UNIVERSITY OF STRATHCLYDE

DEPARTMENT OF HUMAN RESOURCE MANAGEMENT

Work Intensity in the Software Industry: Incidence and Impact on Professional Software Workers

Saira Reid Lee

A Thesis Submitted in Fulfilment of the Requirements for the Degree of Doctor of Philosophy

2012

DECLARATION OF AUTHENTICITY AND AUTHOR'S RIGHTS

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed:

Date:

ACKNOWLEDGEMENTS

Several people deserve mention by way of helping me through the journey of this PhD.

Thank you to Professor Phil Taylor for being such a wonderful, exemplary supervisor to me throughout my PhD. Your guidance, support and mentoring have been invaluable and I am extremely grateful to you for helping me realise my full potential. Thank you for always being available for whenever I have needed advice or feedback on chapters and for your constant enthusiasm. I would also like to acknowledge my late supervisor Peter Bain, who supported me in the very first year of my PhD.

Thank you to the academics, secretaries and other PhD students from the HRM department who have helped make this journey such an enjoyable one. Thank you to my past and present office mates, Pauline and Pearl, respectively, for your friendship and your support.

I would like to express my gratitude to my husband Simon, who was my source of inspiration for this PhD topic. You have been my ‘rock’ throughout this journey. Thank you for your unfailing support, thoughtfulness and belief in me.

Thank you to my parents for encouraging me to follow my dreams, always believing in me and for being there during the good and the hard times of this journey. Thank you to my brothers for your support, always being here for me and for the years of fun and laughter we have shared.

ABSTRACT

Various trends can be seen to characterise the ‘new’ economy, including globalisation, networking, competition, ICTs and transformations in forms of work organisation. Notably, it is claimed that these trends may encourage self-management, peer pressure, the expansion of workloads and greater expenditure of work effort, with implications for experiences of work intensity (Bittman et al, 2009; O’Riain, 2006; Green, 2006, 2004, 2001; Thompson and McHugh, 2002; Findlay et al, 2000; Springer, 1999; Tomaney, 1990).

This thesis aims to explore whether software professionals are necessarily immune from experiences of work intensity, due to their positioning as archetypal knowledge workers (Baldry et al, 2007, 2005; Newell et al, 2002; Alvesson, 1995; Kunda, 1992) and their role in creating a technology which has applicability to virtually all work, business and social situations (Scotland IS: Innovation and Skills in Scotland, 2008; Freeman and Perez, 1998; Quintas, 1994). This thesis has utilised a contextually-based, qualitative, in-depth comparative case study approach, in order to identify and explain the linkages, mechanisms and relationships which influence and shape software professionals’ experiences of work intensity.

This thesis makes an original contribution through identifying that professional software workers are subject to work intensity from diverse sources. Notably, advances in ICTs, globalisation, flexibility and developments in the software industry have had implications for software professionals’ experiences of work intensity. Contextual elements, such as market dynamics, firm characteristics and internal organisational factors influence experiences of work intensity. In addition, aspects of the software labour process, including deadlines, project team structures, specialist knowledge, interruptions, normative control, breaks and worker agency, have implications for software professionals’ experiences of work intensity. Finally, the development of a taxonomy of professional software job roles and the utilisation of the work diary study within this PhD have provided comprehensive insight into the tasks and activities performed by individuals.

CONTENTS

LIST OF DIAGRAMS

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 OVERVIEW

My interest in professional software work first emerged through my husband recounting his day-to-day experiences in his job as a software engineer. I was struck by the apparent intensity of his work. Concurrently, during the second year of my Masters degree in 2005, my lecturers indicated that creative, knowledge-intensive occupations such as professional software work were still relatively unexplored. My preliminary examinations of the literature demonstrated that while existing research on software professionals provided valuable insight into dimensions such as teamworking, identity, skills and knowledge development, careers, union involvement and control, there was a lacuna in the area of work intensity. In addition, existing studies on work intensity appeared to largely focus on the manufacturing sector (see Elger, 1990; Tomaney, 1990), service-related sectors (Taylor and Bain, 2007) or work intensity over time in the economy as a whole (Green, 2006, 2004, 2001; Gallie et al, 1998; Penn et al, 1994), as opposed to emerging sectors such as professional software work. These observations therefore encouraged me to embark on the ‘logic of discovery’ that is a PhD, with the central aim of studying work intensity in the software industry, its incidence and its impact on professional software workers.

The following chapter will provide the rationale for this research and present the five research objectives which have guided this study. This chapter will also outline the research methodology deemed appropriate to achieving these objectives, in terms of philosophical approach, research design and research methods. The final section will provide a brief summary of the chapters which follow.

1.2 RATIONALE FOR RESEARCH

Defining and Evaluating Experiences of Work Intensity

Before explaining the rationale for research, it is necessary to define the terms 'work intensity' and 'work intensification', in order to provide clarity of focus. Common to both terms is the concept of effort that individuals put into their job when they are working (Burchell, 2002). This effort has two components: an extensive component, with regards to the time spent at work and an intensive component, which relates to the intensity of physical or mental input into work (Fiksenbaum et al, 2010; Green, 2006). This effort can encompass two aspects of an individual's workload, namely qualitative – the difficulty and complexity of work – and quantitative – the amount of work an individual has to perform (Wichert, 2002). However, temporal factors underpin the essential differences between the two terms, in that 'work intensity' considers the experience or condition of work at a moment or stage in time, whilst 'work intensification' takes into account the evolutionary nature of work over time. Whilst recognising that developments and changes in the software occupation are important considerations, this research is primarily concerned with understanding and explaining software professionals' current experiences. This research is therefore a study of work intensity, rather than work intensification, which may be regarded as longitudinal in its timeframe.

In addition, it is important to note that whilst work intensity can be viewed as an objective concept, its outcomes may depend on the interpretations, perceptions and responses from individuals. For example, work intensity can result in positive outcomes, with individuals being engaged in their work and experiencing satisfaction or negative outcomes, such as work overload, increased cognitive overload, hypertension and poor well-being (Boisard et al, 2008; Wichert, 2002; Green and McIntosh, 2001). Furthermore, objective measurement and evaluation of work effort in knowledge-intensive occupations such as professional software work can be extremely difficult, due to the demands arising from the mental, intellectual and tacit nature of work. For instance, whilst software professionals may appear to spend a significant amount of time on one task, this may require a significant amount of mental effort that is not readily observable. In this sense, it is necessary to look beyond superficial appearances of work effort. This research therefore aims to capture the experiences and perceptions of software professionals, in order to provide insight into the phenomenon of work intensity. Green (2006) notably proposes a variety of steps which can aid the examination of mental work effort in knowledge-intensive occupations, such as establishing social norms based on people's perceptions of work effort, validating assessments through insight from peers or managers and utilising self-reporting measures by asking individuals to consider present and past work experiences.

The following sections will engage with the literature in greater depth, in order to provide the rationale for conducting research into software professionals and their experiences of work intensity.

The Knowledge Economy

The increased importance and centrality of knowledge, emphasis on creativity and focus on intellectual and social skills are perceived to have encouraged the emergence of the 'new' economy (Castells, 2000; Frenkel et al, 1995; Drucker, 1993; Bell, 1974). Notably, focus on the generation, distribution and production of knowledge within the new economy is claimed to have encouraged the development of knowledge work occupations, such as professional software work, consultancy and research and development. Knowledge work occupations are deemed to place emphasis on symbolic and analytical skills, intangibility, theoretical knowledge, problem-solving, intellectual judgement, self-determination and creativity (Pyöriä, 2005; Alvesson, 2004, 2001; Newell et al, 2002; Thompson et al, 2001; Frenkel et al, 1995). In addition, the distinctive nature of knowledge work means that individuals may be subject to more favourable working conditions, flexible work structures, autonomy, project team structures, lower levels of bureaucracy and minimal managerial supervision (Baldry et al, 2007; Mathews, as quoted in Thompson and McHugh, 2002; Cappelli, 2000; Spender, 1998; Kunda, 1992; Quinn, 1992). Moreover, trends such as globalisation, competition, technology, down-sizing and de-layering are considered to have stimulated the change from hierarchy and stable employment to emphasis on transactional contracts, employability and the increasing self-management of careers (Ituma and Simpson, 2006; Arnold, 2005; Baruch, 2004; Cohen and Mallon, 1999; Cappelli, 1999; Rousseau, 1995).

Crucially, software professionals have been positioned as archetypal knowledge workers, due to the emphasis on intellectual, creative and intangible work, autonomous working conditions, low bureaucracy, project team forms of work organisation and an individualistic view of careers (Baldry et al, 2007, 2005; Newell et al, 2002; Alvesson, 1995; Kunda, 1992). Furthermore, the characterisation of professional software work as an occupation which places emphasis on employability, the self-management of careers and labour market power, due to the possession of valuable skills, knowledge and expertise (Thompson and McHugh, 2002; Birchall and Lyons, 1995) can enable further exploration into these trends.

In this sense, the status of software professionals as archetypal knowledge workers enables consideration of the implications that transformations in working conditions and work organisation may have for individuals, particularly in terms of work intensity.

The Software Industry

Technological advances and the expansion of ICTs can be seen to have been major drivers of growth and change in the computer industry, encouraging the move from hardware to software (OECD Information Technology Outlook, 2006). Crucially, professional software workers can be identified as a core occupation within the new economy, due to the outcomes of their labour – software – having applicability to virtually all work, business, human, social and relational activities (Scotland IS: Innovation and Skills in Scotland, 2008; Freeman and Perez, 1998; Quintas, 1994). For example, software can help companies respond competitively, address consumer demand and improve effectiveness (Intellect Software and IT Services Report, 2009; Key Note Market Report, 2008b; Quintas, 1994). In addition, whilst market and economic conditions may affect IT budgets, priorities and strategies, software has continuing relevance, due to the need for companies to manage costs, address changing circumstances and operate, maintain and support existing systems (Key Note Market Report, 2008b; Baetjer, 1998). Furthermore, society in general can be seen to be reliant on software for social interactions, leisure and entertainment (Scotland IS: Innovation and ICT Skills in Scotland, 2008). In this sense, research into the experiences of software professionals themselves is of significant interest, on the grounds that:

> They are both the creations and the agents of the most spectacular technology yet, which in a generation has launched a transformation of the entire production process (Kraft, 1979: 2).

Developments in ICTs and globalisation are claimed to have increased the international trading of services and encouraged greater flexibility in the choice and location of labour (OECD Information Technology Outlook, 2006; McGrath-Champ, 2005; Sharpe, 1998). Stehr (2004) therefore argues that time, place and distance may be increasingly irrelevant for companies presented with a variety of choices concerning the creation and management of workforces. Notably, a developing trend within the software industry is that organisations may choose to outsource their software services to specialist software organisations in the same or other countries (Sharpe, 1998). Indeed, it is claimed that the outsourcing or offshoring of software activities can help organisations make cost savings, fill in internal skill gaps, receive services required on a short-term basis, take advantage of attractive labour conditions and wages, provide 'round the clock' services and tactically deploy individuals in increasingly competitive, global markets (Key Note Market Report, 2008c; McManus and Floyd, 2005; Arora et al, 2001).

In summary, software professionals' status as a core occupation within the new economy can therefore enable examination of the implications that trends such as globalisation, developments in ICTs and flexibility over choice and location of labour may have for workers themselves, particularly in terms of experiences of work intensity.

The Potential for an Intense Work Process

Existing research has presented conflicting perspectives on the implications that transformations in workplaces and forms of work organisation have had for experiences within the knowledge economy. For example, whilst team-based structures may arguably facilitate problem-solving, creativity and collaboration between knowledge workers, these configurations may equally encourage greater self-management, peer pressure, the expansion of responsibilities and motivate individuals to work harder (Green, 2006, 2005; 2001; Thompson and McHugh, 2002; Findlay et al, 2000; Springer, 1999). Notably, the emphasis within professional software work on heterogeneous, interdependent project team structures provides the perfect setting for exploring these issues in greater detail.

In addition, positive outcomes from the new economy, such as the up-grading of knowledge and skills, empowerment, autonomy and greater satisfaction at work have been juxtaposed with the themes of rationalisation, efficiency, displacement, de-skilling, monitoring and control (Baldry et al, 2007; Kumar, 2005; Alvesson, 2004). Moreover, it is claimed that emphasis in the new economy on flexibility, multi-skilling and empowerment may, in reality, have stimulated the integration of tasks, expanded workloads and required individuals to expend greater levels of work effort, in order to increase productivity (Green, 2006, 2001; Thompson and McHugh, 2002; Tomaney, 1990). Indeed, Thompson and Warhurst (1998) go so far as to suggest that empowerment, flexibility and creativity may merely represent alternative ways of managing and controlling workers. In this sense, it is pertinent to consider whether these trends are in evidence within professional software work and the implications for experiences of work intensity.

Finally, the literature suggests that the restructuring of time and space, flexibility in labour markets, competition and ICTs may have had implications for experiences of intensity. Indeed, ICTs, privatisation and commercialisation may have transformed the nature of competition, stimulated changes in technological processes and products and intensified markets in the new economy (Hornby and Clarke, 2002; Castells, 2000; Webster, 2000; Dyson et al, 1996). Furthermore, these trends may have sped up activities, expanded responsibilities and workloads, increased performance pressures and encouraged greater personal investment in work (Bittman et al, 2009; O'Riain, 2006, 2000; Green, 2004, 2001; Gallie et al, 1998; Green and McIntosh, 1998; Sennett, 1998). In summary, exploration into whether software professionals are necessarily immune from experiences of work intensity is of great significance, due to their role in creating a key technology at the centre of the economy and their status as archetypal knowledge workers.

Labour Process

Consideration of the software labour process can be seen to help address the central aim of this research. Preliminary review of existing literature suggests that objective structures, worker subjectivity, consent and accommodation within the software labour process may have implications for software professionals' experiences of work intensity. For example, objective structures such as the development life cycle, structured methodologies, standards, code reviews and performance metrics may potentially represent attempts to formalise, monitor and control professional software work (Andrews et al, 2005; Sharone, 2004; Baetjer, 1998; Friedman and Cornford, 1989; Beirne et al, 1988; Kraft, 1979, 1977). In addition, the ability of software professionals to exert agency and resist or respond to organisational practices, such as standards or internal ideology, may further serve to influence experiences of work intensity.

Consideration of consent and accommodation can also be deemed relevant to this study on work intensity, in that the dynamics between capital and labour may influence the systems through which tasks are directed and supervised and how individuals adapt to work, produce or reproduce interests (Edwards, 1979; Burawoy, 1978). Furthermore, conventional wisdom suggests that professional software work may be empowered, autonomous and intrinsically motivated by the nature of the work itself (Barrett, 2001; Alvesson, 2000; Tsoukas, 1996), meaning it is important to consider the implications of these aspects for experiences of work intensity. In this sense, labour process analysis can provide valuable insight into the extent to which the structure, design, organisation, management and control of professional software work influences experiences of work intensity.

1.3 RESEARCH OBJECTIVES

The five objectives guiding this thesis and which inform the research questions are:

- To provide a focused examination of the tasks and activities performed by software professionals
- To understand how software professionals respond to and experience the labour process
- To explore contextual and internal organisational factors which may have implications for experiences of work intensity
- To establish whether or not professional software workers experience work intensity
- To the extent that software professionals do experience work intensity, to examine their experiences and perceptions of work intensity and its extent, character, causes and consequences.

1.4 METHODOLOGY AND METHODS

This section will outline the research methodology deemed most appropriate to achieving the central aim and objectives of this research, with regards to philosophical approach, categorising professional software job roles, research design and research methods.

This PhD aims to identify and explain the linkages, mechanisms and relationships which influence and shape software professionals' experiences of work intensity. Critical Realism can be identified as the most relevant philosophical approach, due to its emphasis on establishing causal linkages and explaining mechanisms and wider structures, in order to provide a comprehensive understanding of phenomena. In addition, one of the key objectives of this research is to apply a contextually-based analysis of professional software work, in order to provide a detailed understanding of work intensity. This corresponds with the Critical Realist paradigm, which stipulates the importance of considering contextual circumstances, in order to understand differences in outcomes, identify causal powers or mechanisms which may interact to create particular events and provide a setting for interpreting and understanding individual experiences (Modell, 2009; Sayer, 2008; Easton, 2000; Tsoukas, 1989). Consideration of context within this research can be seen to be especially relevant, in that much of the existing literature on professional software workers appears to be largely de-contextualised, diminishing the relationship between external context, internal organisational factors, the software labour process and how these relate to experiences at the individual level.

In-depth, intensive case study research design can be identified as helping to address the research objectives set out in Section 1.3, through allowing for the tracing of causal relationships, the interpretation of meanings in context and the generation of insight into the complexity of structures, social actions and dynamics within individual settings (Harrison and Easton, 2004; Danermark et al, 2002; Orum et al, 1991). In addition, a comparative approach may potentially enable the identification of commonalities and differences and the consideration of factors which may have an effect on outcomes, helping to develop theoretical and conceptual understandings of work intensity (Danermark et al, 2002; Eisenhardt, 2002; Orum et al, 1991).

Existing literature on software professionals has tended to focus on the development life cycle as a means of explaining professional software work (Andrews et al, 2005; Marks and Lockyer, 2004; Marks et al, 2002; Marks et al, 2001; Barrett, 2001; Beirne et al, 1998), rather than outlining the activities performed by individuals themselves. Variations in job titles and roles, differing levels of involvement in the software life cycle and differing levels of specialisation or generalisation can be seen to have contributed to difficulties in classifying professional software roles and detailing activities (Andrews et al, 2005; Beirne et al, 1998; Sharpe, 1998; Kraft, 1977). However, whilst professional software work may be heterogeneous, complex, intangible and variable in nature, the author argues that it is possible to provide a general understanding of the types of roles and activities that individuals may be engaged in. Notably, the author argues this can be achieved through the examination and collation of information from a variety of internet and paper-based career sources. In addition, providing insight into the nature of activities and tasks within professional software work can help to inform the development of research methods utilised within this research study.

Qualitative research methods can be deemed most appropriate to this study into work intensity, due to the emphasis on exploratory work to identify causal linkages and explain what produces particular stages, changes and situations (Ackroyd and Fleetwood, 2000). Notably, the combination of multiple research methods, such as observation, documentation, work diaries and semi-structured interviews, may help to create a 'chain of evidence', increase the validity of findings and enable consideration of the meaning of differences (Yin, 2003; Fielding and Fielding, 1986).

1.5 THESIS OUTLINE

This final section will provide a brief overview of the chapters which follow, in order to guide the reader.

Chapter Two engages with the key debates on the knowledge economy, the software industry and work intensity, in order to provide the rationale for research and identify aspects which may have implications for software professionals' experiences of work intensity. This chapter argues that research on software professionals, who can be defined as the creators of a key technology and archetypal knowledge workers, can enable the key generalisations surrounding the knowledge economy and knowledge work to broader review. Notably, these debates concern the extent to which transformations in forms of work organisation and working conditions are truly in evidence within knowledge work occupations and the extent to which these represent positive outcomes or negative repercussions for individuals. This chapter also helps to inform our understanding of professional software work through outlining the history and importance of the software industry, recent developments and the main characteristics of professional software work. In addition, this chapter provides definitions of 'work intensity' and 'work intensification' and proceeds to argue that trends such as globalisation, developments in ICTs and transformations in forms of work organisation may have implications for experiences of work intensity.

Chapter Three engages with the four streams of literature which inform our understanding of professional software work and which may be important to this study. Firstly, this chapter focuses on the collaborative aspects of professional software work, arguing that interactions with project team members, project managers and clients can be seen to form a necessary part of the work process. Secondly, this chapter illustrates that software professionals may require a range of technical, interpersonal, business-related and political skills to perform their work effectively. In addition, it is argued that individuals in the new economy may be increasingly responsible for managing their own careers. Thirdly, this chapter suggests that particular organisational and interactional dynamics within professional software work, such as deadlines, working time, interruptions, work location and physical proximity may have implications for software professionals. Fourthly, this chapter demonstrates that a labour process perspective can provide insight into the structure, design, organisation, management and control of professional software work and its implications for software professionals' experiences of work intensity. The chapter concludes by summarising the main themes raised in Chapters Two and Three which may have implications for software professionals' experiences of work intensity and presents the five research questions guiding this study.

Chapter Four outlines the methodological approach employed within this research, with regards to the appropriateness of the Critical Realist paradigm, the qualitative, in-depth and comparative case study research design and the utilisation of multiple qualitative research methods (observation, documentation, work diaries and semi-structured interviews). This chapter also presents the taxonomy of professional software job roles which has been developed by the author, in order to provide insight into the nature of activities and tasks performed by individuals. This taxonomy addresses the failure of the preceding literature to detail the actual activities performed by software professionals and also informs the development of research methods utilised within this research study. In addition, this chapter discusses how work intensity within professional software work has been evaluated and presents an index of possible determinants of work intensity which has been created by the author, in order to enable exploration into software professionals' experiences of work intensity. Finally, this chapter presents information on the two case study companies (SpecSoft and InSoft), the purposive and iterative sampling strategy, response rates, participant characteristics, data recording procedures and strategies for data analysis.

Chapters Five and Six present the sources of intensity identified at SpecSoft and InSoft, respectively, and document the consequent experiences of software professionals. **Chapter Seven** compares and contrasts findings from SpecSoft and

InSoft, in order to identify similarities, patterns and differences in software professionals' experiences of work intensity between the two organisations. **Chapter Eight** positions the findings from the two case studies in relation to the literature, in order to identify the original contributions made by this research and aspects which are consistent with, or challenge, existing accounts on professional software workers. This chapter also examines how the findings sit against the narratives on the knowledge economy. Finally, **Chapter Nine** concludes by summarising the main contributions of this PhD research, limitations and suggestions for future research.

CHAPTER 2 THE KNOWLEDGE ECONOMY AND WORK INTENSITY

INTRODUCTION

Trends such as globalisation, de-regulation, competition, the expansion of ICTs and emphasis on knowledge can be seen to have given rise to a 'new' economy and stimulated the emergence of knowledge work, including consulting and professional software work (Scarborough, 1999). It is claimed that new ways of communicating, facilitative management styles and flexible forms of work organisation may have up-skilled, up-graded and empowered workers in the new economy (Baldry et al, 2007; Kumar, 2005; May, 2002; Castells, 2000; Felstead and Jewson, 1999; Negroponte, 1995). However, surveillance, polarisation and new ways of controlling workers can also be seen as less desirable consequences (Brinkley et al, 2009; Alvesson, 2004; Huws, 2003; Warhurst and Thompson, 1998). In addition, despite expectations in the 1970s that quality of working life would improve and work would become less intense in the future, it is argued that the United Kingdom is experiencing high levels of work intensity, with 1 in 5 employees potentially exposed to high work intensity (Boisard et al, 2008; Green, 2004). Critically, the characterisation of software professionals as archetypal knowledge workers suggests that this occupation is ideal for scrutinising the key debates on the 'new' economy. This chapter therefore engages with the key debates on the knowledge economy, software work and work intensity, in order to demonstrate the rationale for this research.

Section 2.1 of this chapter focuses on the emphasis placed on knowledge and intellectual capital within the 'new' economy and considers the extent to which this represents a new paradigm. This section begins by discussing the main theories on the 'new' economy, such as Bell's (1974) 'post-industrial society'; Reich's (1991) 'symbolic analysts'; Drucker's (1993) 'knowledge economy'; and Castells' 'informationalism'.

This is followed by an account of the main definitions, characteristics and perspectives surrounding knowledge work. It is claimed that the intangible, intellectual and creative nature of knowledge work necessitates different types of work organisation, such as autonomy, flexibility, lower levels of bureaucracy and facilitative management styles (Mathews, as quoted in Thompson and McHugh, 2002; Spender, 1998; Quinn, 1992). However, minimal supervision, teamworking and autonomy are also argued to represent new, insidious ways to manage and control workers in the 'new' economy.

Section 2.2 outlines the composition of the computing industry, identifies the various trajectories in computer systems development and discusses the emergence of professional software work as a separate occupation from routine, maintenance areas. This section also focuses on the adoption of structured methodologies, such as the 'waterfall cycle' and 'Agile' within professional software work, in an attempt to develop industry-standard processes, achieve greater structure and manage increased complexity of systems (Raghaven and Chand, 1989; Kraft, 1979). In addition, it is argued that the revolutionary nature of software and its applicability to virtually all types of work situations and human activity (Scotland IS: Innovation and Skills in Scotland, 2008; Quintas, 1994) makes research on software professionals particularly relevant. Furthermore, the emergence of increased competition and greater flexibility over choice and location of labour within the software industry provides the opportunity to explore the implications that these trends have had for knowledge workers themselves.

Section 2.3 outlines the main characteristics of professional software workers, including their status as knowledge workers, the intangible, creative and challenging nature to work and the presence of occupational, team and organisational identities. It is also argued that professional software work may possess intrinsically satisfying attributes which, coupled with identification at occupational and team levels, may motivate individuals and act as a powerful form of normative control (Baldry et al, 2007; Lockyer et al, 2001; Deetz, 1995). This literature helps to inform our understanding of professional software work and enables the identification of aspects which may be of importance to this study.

Section 2.4 provides the rationale for research into work intensity within professional software work. This section firstly defines the terms ‘work intensity’ and ‘work intensification, in order to provide clarity of focus to the research. Factors such as globalisation, ICTs and forms of work organisation are outlined as having implications for experiences of work intensity in the new economy. For example, the restructuring and intensification of time and space, the speeding up of activities, the facilitation of management control, the expansion of responsibilities and workloads, increased performance pressures and levels of personal investment in work can be identified as potentially contributing to work intensity for individuals. It is argued that exploration into whether software professionals are necessarily immune from experiences of work intensity is of great significance, due to their role in creating a key technology at the centre of the economy and their status as archetypal knowledge workers.

Chapter Two concludes by summarising the implications that these key debates have for this research study on work intensity in the software industry, its incidence and its impact on professional software workers.

2.1 THE KNOWLEDGE ECONOMY

The following section will discuss the main theories on the knowledge economy and consider the extent to which the ‘new’ economy represents a new paradigm. It will then focus on the main definitions, characteristics and perspectives surrounding knowledge work.

The New Economy

It is claimed that the increased importance and centrality of knowledge in society, emphasis on creativity and innovation at work and focus on intellectual and social skills have led to what can be termed the 'new' economy (Castells, 2000; Frenkel et al, 1995; Drucker, 1993; Bell, 1974). The 'new' economy has been summarised as one which has moved from the production and distribution of tangible and physical goods to one concerned with the intangibility of goods and services and the generation, distribution and application of knowledge (Caloghirou et al, 2006; Brown and Hesketh, 2004; Newell et al, 2002; Jacques, 2000). Various trends can be seen to characterise this 'new' economy, such as: the increasing emphasis on knowledge; the use of information and communication technologies (ICTs); the expansion and growth of new sectors, such as the knowledge and service sectors; globalisation; ideas of 'weightlessness'; more flexible ways of working, such as teleworking, homeworking and virtual working; flatter, more fluid working structures; and the expansion of markets (Castells, 2000; Rowe and Thomson, 1996; Frenkel et al, 1995). It is also claimed that in the 'new' economy, the increased demand for symbolic and analytical services has led to the emergence and development of knowledge-based work and knowledge processes. Indeed, Knights et al (1993) point to the increasing use of terms such as 'knowledge worker', 'knowledge-intensive firm' and the focus on particular types of activities in the new economy.

Many theorists (such as Bell, 1974; Reich, 1991; Drucker, 1993; Castells, 2000) have attempted to define and outline the main elements of the new economy, emphasising the centrality of knowledge and its implications for society. At the centre of Bell's (1974) approach is the emphasis on theoretical knowledge, largely deriving from developments in technology and science and giving rise to what is termed 'post-industrial society'. Whilst 'pre-industrial' and 'industrial' societies may have been defined by agricultural, mining, fishing and energy and machine technologies respectively, 'post-industrial society' has been heralded as the age of knowledge, information processing, telecommunications and computers (Bell, 1974). Post-industrial society is considered to have undergone a variety of transformations in the technological, economical, political, social and cultural spheres, such as a move from manufacturing to services, changes in occupations, infrastructure changes and the increased participation of women in work (Bell, 1974). It is also claimed that these technological developments have transformed society through encouraging the creation of new occupational classes, new production methods and changing the nature of communications and interactions. Trends such as the expansion of transportation and public utilities, mass consumption and re-considerations of time and space, as a result of technological innovations are also considered to have had implications for distribution, the location of work and the nature of work performed by individuals (Bell, 1974).

Many writers (Schaff, 1982; Reich, 1991; Castells, 2000; May, 2002; Huws, 2003) have attempted to classify work in the 'new' economy. For example, Schaff (1982) suggests that the 'new' economy can be composed of five main service areas, namely: creatives (knowledge workers engaged in research, development and design activities); organisational workers (engaged in management, organisation, facilitative work or tasks); social workers (individuals working as advisers, counsellors or carers); maintenance and technical staff (individuals working in areas such as plumbing or information technology help desks); and leisure activities (individuals engaged in cultural and sport activities). In the 'new' economy, skills such as problem-solving, customer interactions and interpersonal skills are identified as being important (Brown and Hesketh, 2004). Reich (1991) also suggests that three main job categories have emerged in the 'new' economy, namely routine production services, in-person services and symbolic-analytic services. Symbolic-analytic work encompasses problem-solving, problem-identification, abstract activities and the manipulation of symbols and includes workers such as software engineers, public relations executives, lawyers and consultants (Reich, 1991). In addition, routine production and in-person services relate to tasks which are routine in nature, the key distinction being that in-person tasks are provided on a person-to-person basis (such as those working as hairdressers, cleaners, secretaries and in-flight attendants) and routine production services are repetitive in nature, constituting a large section of blue collar and information-based occupations (Reich, 1991). Despite this diversity in occupational areas, Drucker (1993) suggests that two groups dominate in the 'new' economy: knowledge workers at the core, who are engaged in intellectual, knowledge work and peripheral workers, who support the 'new' economy.

Drucker (1993) also draws attention to knowledge, in terms of its central role and its transformative effects on the structure, economics and politics of modern society. For example, Drucker (1993: 35-36) outlines the impact that knowledge and its application have had on productivity:

> …too few people realise that it is the application of knowledge to work which created developed economies by setting off the productivity explosion of the last hundred years. Technologists give the credit to machines, economists to capital investment. But both were copious in the first hundred years of the capital age, that is, before 1880, as they have been since. In respect to technology or capital, the second hundred years differed little from the first hundred. But there was absolutely no increase in worker productivity in the first hundred years – and consequently also little increase in workers' real incomes or any decrease in their working hours. What made the second hundred years so critically different can only be explained as the result of applying knowledge to work.

Knowledge is claimed to have contributed to the new economy in three ways, through encouraging continuous improvements, constant innovation and allowing for the exploitation of knowledge to create new products (Drucker 1993). The productivity of knowledge can therefore be seen to be increasingly important in a capitalist economy:

> The productivity of knowledge is increasingly going to be the determining factor in the competitive position of a country, an industry, a company. In respect to knowledge, no country, no industry, no company has any 'natural' advantage or disadvantage. The only advantage it can possess is in respect to how much it obtains from universally available knowledge. The only thing that increasingly will matter in national as well as in international economics is managerial performance in making knowledge productive (Drucker, 1993: 176).

Knowledge is also at the centre of Castells (2000) 'informationalism' approach, whereby knowledge and its applications are believed to revolutionise knowledge, information and communications. In this sense, the 'new' economy can be seen as informational, in that knowledge-based productivity is organised around and maximised through the development and diffusion of ICTs (Castells, 2000). Indeed, ICTs are claimed to have stimulated the development of new sectors (such as multimedia, software and e-commerce), encouraged new ways of working and managing and provided new ways of creating, distributing and disseminating knowledge and information (Caloghirou et al, 2006; Foray, 2006; Stanworth, 1998). The importance of ICTs in what has been termed the 'new' economy is emphasised by Gibbons et al (2002: 122):

> The success of the knowledge industry depends on the extent to which it is supported by an information technology infrastructure. This new infrastructure depends upon innovations in the telecommunications and computer industries that will make possible the ever closer interaction of an increasing number of knowledge centres. This new infrastructure is being put in place. Its effects will be pervasive and may in time lead to a new techno-economic paradigm.

ICTs can therefore be considered to have had implications for the nature of knowledge in that:

> …the ICT system gives the knowledge economy a new and different technological base which radically changes the conditions for the production and distribution of knowledge, as well as its coupling to the production system (Lundvall and Foray, 1996: 4, as quoted in Caloghirou et al, 2006: 7).

These ICTs are also claimed to have implications for the nature of work processes, employment structures and occupational structures. Indeed, Castells (2000) argues that employment and occupational structures have transformed, as a result of the move from goods to services, the increase in professional and managerial jobs and increasing emphasis on the information content at work. For example, ICTs may upgrade, downgrade or have an effect on how and where people work (Castells, 2000). Furthermore, technological developments, along with organisational trends such as downsizing and de-layering, may have also encouraged the development of new employment practices such as contracting, outsourcing, portfolio working, self-employment, teleworking and homeworking (May, 2002; Felstead and Jewson, 1999; Negroponte, 1995; Handy, 1994). In this sense, ICTs may have had implications for work and society through speeding up processes and communications, allowing for greater flexibility in work organisation and providing more choice over work location. In particular, occupations with an information and knowledge content, especially those which are professional, technical and managerial, have been argued to play an increasingly important role at the centre of the economy (Castells, 2000). The move towards an economy emphasising the production, distribution, diffusion and application of knowledge is therefore suggested to have led to a new ethos and re-definition of capitalism whereby:

> The language is now of 'high added value' and 'high performance work organisations' who attempt to provide workers with the information, skills, incentives and responsibility to make decisions essential for innovation, quality improvement and rapid responses to

> change. In the knowledge economy, we have moved from high volume to high value work (Brown and Hesketh, 2004: 44)

and

> We're all in the thin air business these days…the real assets of the modern economy come out of our heads, not out of the ground; ideas, knowledge, skills, talent and creativity (Leadbeater, as quoted in Baldry et al, 2007: 28).

In the 'new' economy, it is therefore claimed that competitive advantage and wealth creation may depend more on intangible, specialist knowledge, flexible, adaptable decentralised units and less on manual areas or traditional command and control structures (Drucker, 1993; Hague, 1991; Reich, 1991; Prahalad and Hamel, 1990). The new economy literature also presents arguments centring around the idea that intangible services are replacing tangible goods and that ICTs are leading to an increasingly 'weightless', 'dematerialised' society (Huws, 2003). For instance, Castells (2000) suggests that under 'informationalism', the 'new' economy can be considered to be global, with production, distribution and consumption operating on a global scale and networked, whereby work, production and competition occur through networked interactions. Similarly, Stehr (2004) points to the increasing irrelevance of time and space at work as a result of globalisation and the application of ICTs. Trends such as de-localisation of work and outsourcing can be seen to have emerged, largely aided by the increased digitisation of information allowing for remote access; the standardisation of tasks, enabling activities to be monitored remotely or outsourced; the convergence of skills requirements in occupations and industries; and greater diffusion of ICTs (Huws, 2003). In this sense, companies and regions may have to compete more effectively for services, due to the increased choice in locations, labour markets and skill pools (Huws, 2003).

Baldry et al (2007) also draw attention to the changing meaning of work to individuals in modern society, with a move from traditional Taylorist approaches, viewing work as largely serving economic needs, to human relations schools of thought, with considerations of the importance of motivation, commitment and participation at work. Thus, organisations may have had to consider new forms of work organisation in the 'new' economy to remain competitive, such as lower levels of bureaucracy, flatter, more fluid structures, greater flexibility, networking and more horizontal communications and interactions (Quinn, 1992). In this sense, greater flexibility, multi-skilling and more organic work structures may have enabled organisations to respond more effectively to uncertainty, unpredictability, changing market conditions and consumer demand (Baldry et al, 2007; Elger, 1991; Piore, 1986). In addition, Baldry et al (2007) suggest that as a result of more flexible organisational structures, work roles may have become more interdependent, meaning that individuals could have to undertake more responsibility or carry out a wider variety of roles. Furthermore, developments in the economy and forms of work organisation may also have given rise to changes in the psychological contract, with a move from contracts emphasising hierarchies, stable employment and career ladders, to those emphasising self-managed careers and individuals managing their own work (Thompson and McHugh, 2002; Birchall and Lyons, 1995). In this sense, individuals may be more responsible for managing and improving their employability and careers, through working on skills, knowledge and experience (Baldry et al, 2007; Herriot and Pemberton, 1996; Heckscher, 1995).

In summary, knowledge is claimed to be beneficial for society, in terms of upgrading knowledge and skill levels in existing work, creating and expanding new occupational areas and encouraging empowerment, autonomy and increased satisfaction at work (Baldry et al, 2007; Kumar, 2005). As concluded by Brown and Hesketh (2004: 43):

> Unlike financial capital, human capital appears to be impervious to such fluctuations. Or at least that is what we have been led to believe by those advocating the relentless pursuit of increased

> knowledge and skills. Knowledge reaches the parts other forms of capital cannot reach. It can enhance the economic returns of individuals, regions and nations states; it can grant a healthier life to those fortunate enough to acquire knowledge and even, so we are told, promote greater spiritual fulfilment. Crucially, its elixir transcends the peaks and troughs of the performance of the global economy. Human capital is worth having in the bad times as well as during the good.

Contesting the 'New' Economy

However, despite claims that there is a 'new' economy, Huws (2003) argues it is necessary to question whether the 'new' economy actually represents a completely new paradigm. For example, whilst proponents of the 'new' economy thesis claim that the service and knowledge sectors are of increasing importance, traditional areas such as agriculture and manufacturing may, in fact, still exist but instead be re-located in the global economy, as a result of increased choice eliciting from networking and globalisation (Huws, 2003; Castells, 2000; Cohen and Zysman, 1987). Indeed, Huws (2003) suggests that claims pointing to the demise of manufacturing may have resulted from the difficulty in studying the changing nature of the division of labour and accurately distinguishing between intermediate, final outputs and different types of contractual arrangements. As illustrated by Huws (2003: 146):

> Material goods must be transported in a physical form across national boundaries and are therefore generally recorded in import and export statistics but information sent over the internet leaves no such trace and there is no easy way to access the value of such traffic.

Stehr (2004) also points to the interdependent nature of the manufacturing and service sectors, suggesting that service-related inputs may be required in the manufacturing sector and vice-versa, signifying a connection between goods and services. This point is further exemplified by Huws (2003), who argues that the mechanisation of farming and food production, such as the manufacturing of farm machinery, fertilisers and the preparation and distribution of foods to supermarkets, is still evident. Despite the claimed increase in emphasis on intangible goods and services, Huws (2003: 147-148) also suggests that society may continue to have a need for material goods, especially in the case of new ICTs:

> There is also a need, continuously renewed because of its rapid obsolescence, for hardware: personal computers, mobile telephones, modems, scanners, printers, switches and the many components and accessories involved in their manufacture and use.

It is also suggested that manufacturing may have simply changed its focus from material-intensive commodities to information and knowledge-based commodities, based around knowledge industries (Stehr, 2004). These knowledge-based commodities could be monetary based, relating to the movement of capital, exchange rates and interest rates or non-monetary based, focusing on data, programs, organisational knowledge and information (Stehr, 2004).

In addition, whilst the search for knowledge and innovation may be seen as one of the defining features of the 'new' economy, Callinicos (2001) suggests that this drive may be stimulated by the continued search for profits and profits maximisation. The 'new' economy can therefore be considered to merely represent a re-definition of capitalism to generate profit through new means. This may include attempts to produce tangible or intangible-based goods through methods such as developing human capital via education and training and the commodification of information through CD-Rom based instructions (Huws, 2003). As emphasised by Baldry et al (2007: 29):

> As Callinicos has observed, the ceaseless pressure to upgrade computing systems is not driven by any autonomous technological imperative but by the interest in profit-maximisation shared by Microsoft, Intel and the PC manufacturers.

Whilst supporters of the 'new' economy approach cite the growth in intellectual and knowledge-based areas of work as one of the features of the 'new' economy, Alvesson (2004) suggests there may also be continued expansion in lower levels, with themes such as rationalisation, efficiency and managing costs emerging alongside discussions of flexibility, empowerment and up-skilling. Work in the knowledge economy can therefore be seen to potentially displace and de-skill existing workers, as well as up-skilling and up-grading some areas (Warhurst and Thompson, 1998). Indeed, Castells (2000) points to the possibility of a 'new' economy characterised by an increasingly polarised structure spanning four social groups such as: upper middle class, comprising managers and professionals; middle class, including technicians and craft workers; lower middle class, covering sales, clerical and operator workers; and lower class, those individuals working with services and agricultural areas. Furthermore, Brinkley et al (2009) and Huws (2003) highlight the potential for a polarised structure in the knowledge economy with, on the one hand, core knowledge workers engaged in creation and production of intangibles and, on the other, those engaged in few knowledge tasks and routine work, with responsibility for supporting the system and functions. As illustrated by Reich (1991: 175):

> The foot soldiers of the information economy are hordes of data processors stationed in 'back offices' at computer terminals linked into world-wide information banks. They routinely enter data into computers or take it out again – records of credit card purchases and payments, credit reports, checks that have cleared…subscriber lists, personnel, library catalogues and so forth. This 'information revolution' may have rendered us more productive but it has also

> produced huge piles of raw data which must be processed in much the same monotonous way that assembly-line workers and before them, textile workers processed piles of other raw materials.

In this sense, as highlighted by Huws (2003), when analysing the knowledge economy and its consequences for the nature of work, it may therefore be important to examine the labour process and materiality of the worker, in order to understand contributions to the final process or commodities.

Many of the arguments presented on the centrality and importance of knowledge as an original feature of the 'new' economy may, at first glance, be persuasive. However, the production and diffusion of knowledge, as well as the management and processing of information, could be argued to have always been important for the functioning of society and the economy (McNicoll et al, 2002). For example, Caloghirou et al (2006: 3) suggests that knowledge may have always played a significant role in the economy and society in general:

> The contribution of knowledge to the process of economic, technological and social change is not new: it has always played a part and often a crucial part in such change. Every economic activity is based on the use of knowledge in some way both in modern societies where the central phenomenon is…that as an aggregate we know more…and that every aspect of our material existence has been altered by our knowledge' (Mokyr, 2002: 2), and in pre-historic (Paleolithic and Neolithic) societies (Smith, 2002: 9). Storytelling is one of the oldest forms of knowledge sharing used throughout human society. Thus, in this sense, there has always been a knowledge dependent economy.

In addition, the enduring nature of knowledge through the centuries is emphasised by Brown and Hesketh (2004: 44):

> Its role in economic competition between nations also has a long pedigree as David Landes has observed, in the early eighteenth century France sent out ‘explorers’ to acquire the secrets of new British technologies and in 1718 it ‘launched’ a systematic pursuit of British technicians: clock and watch makers; woollen workers; metallurgists; glassmakers; shipbuilders.

Mokyr (2002) also suggests that during the Industrial Enlightenment era, cultural and intellectual development, as well as technological changes, enabled interactions and accessibility to different types of knowledge. In addition, the generation, application and management of knowledge has played an important role throughout the industrial era, through contributing to technological change, growth and management of knowledge. For example, it can be seen that the principles and applications of Taylorism in the twentieth century were fundamentally concerned with extracting and capturing the knowledge of workers, in a systematic attempt to access and apply the knowledge that individuals possessed (Chumer et al, 2000; Warhurst and Thompson, 1998). Thus, in this sense, managers may have always been aware of the knowledge workers possess and concerned with how to access this knowledge (Warhurst and Thompson, 1998). It could be argued that:

> And yet there is something terribly obvious and familiar about this. Isn’t the management – identification, codification and application – of knowledge a defining feature of Taylorism? The quest to harness, monopolise and systematise knowledge is hardly a novel application and pre-occupation of management (Chumer et al, 2000: xvi).

In summary, the 'new' economy can be seen to be subject to many contradictions and indeed, emerges as a highly contested and debatable paradigm. The next section will focus on an emerging aspect from the 'new' economy literature, namely the increasing emphasis on theoretical and intangible knowledge and the move towards what has been termed 'knowledge work'.

Defining Knowledge Work

As highlighted earlier, the increased demand for symbolic and analytical skills has led to the emergence and development of knowledge work and knowledge processes (Pyöriä, 2005). It is useful to distinguish 'knowledge work' from 'working knowledge', and between 'manual work' and 'knowledge work', as there can be ambiguities with regards to the concept of knowledge itself (Pyöriä, 2005; Frenkel et al, 1995). Indeed, knowledge as a concept can have a variety of meanings (Alvesson, 2004; Brown and Hesketh, 2004; McNicoll et al, 2002; Blackler, 1995; Nonaka, 1994). For example, Blackler (1995: 1032-1033) states that:

> Knowledge is multi-faceted and complex, being both situated and abstract, implicit and explicit, distributed and individual, physical and mental, developing and static, verbal and encoded.

The area of knowledge at work can therefore potentially include those with 'working knowledge' - individuals who have knowledge of their work and utilise knowledge - or, alternatively, refer specifically to 'knowledge workers' - individuals who use abstract and theoretical forms of knowledge in their work and are engaged in the generation, diffusion, distribution and application of knowledge itself (Brinkley, 2009; Baldry et al, 2007). In addition, whilst manual work may be more clearly defined, physical in nature and rely on contextual knowledge for understandings, knowledge work can be seen to have mental, intellectual and theoretical elements dealing with more abstract, symbolic notions and ideas (Pyöriä, 2005). In addition, as suggested by Newell et al (2002) and

Frenkel et al (1995), knowledge can be considered to be central to knowledge work and work processes, forming the input, medium and outputs of work. In this sense:

> …a 'knowledge occupation' can be defined as an occupation in which most activities are knowledge – (or information) – based, knowledge-intensive and knowledge-generating (Kochen, 1984: 150). This definition thus restricts knowledge workers to roles where knowledge is central to what is worked on (medium of work), what is produced (output of work) and how work is undertaken (the act of work) (Frenkel et al, 1995: 778).

Within the area of those engaged in work with mental, intellectual and theoretical elements, Reed (1992) suggests that there can be three categories of professionals. These include: traditional professionals, those workers utilising codified, theoretical and rational knowledge, such as lawyers and doctors; organisational professionals, workers whose knowledge may be tacit, technical and contextually-based, such as managers and administrators; and knowledge workers, those individuals who deal with an esoteric, intangible knowledge base, working with, generating and manipulating symbolic forms, such as software engineers and consultants (Reed, 1992). Traditional professions may typically be characterised by codified and rational knowledge bases, standardised educational paths, strict entry routes and common professional norms and affiliations (Newell et al, 2002). However, knowledge professions may be characterised by diverse educational backgrounds, less in the way of controls from occupational institutions or professional associations (Newell et al, 2002) and intangible knowledge bases which cannot easily be formalised, meaning that workers may have skills and knowledge which are valued in the market place. As claimed by Hull (2000), new professions such as software engineering and consultancy can be seen to have emerged as a result of changes in the economy, technology and managerial ideologies. Thus, it can be suggested that:

> The argument is that the established form of professionalism, professional organisation and professional power have recently been altered by the combined impact of economic, technological and ideological changes. Among other factors, the growth of 'organisational professions', such as managers and administrators, together with the financial crisis of the welfare state, are forcing challenges to the traditional professions associated with welfare and other professions; they are heralding and enhancing the emergence of new forms of expertise and new 'expert divisions of labour' and they are breaking the links between the state and professional expertise which once legitimised and guaranteed the neutrality of those professions (Hull, 2000: 51).

Characteristics of Knowledge Work

A variety of terms have been used to describe those engaged in knowledge work, such as 'knowledge workers', 'symbolic-analysts' (Reich, 1991) and 'gold collar workers' (Kelley, 1990). Knowledge work is considered to be characterised by a variety of traits, including: high educational attainment; an emphasis on symbolic and analytical skills; intangibility of work; theoretical knowledge; problem-solving; intellectual judgement; self-determination; and creativity (Pyöriä, 2005; Alvesson, 2004, 2001; Newell et al, 2002; Thompson et al, 2001; Frenkel et al, 1995). In addition, due to the nature of work, knowledge work organisations are perceived to include favourable working conditions and terms of employment, non-bureaucratic working environments, flatter, more fluid working structures, autonomous working conditions and opportunities for development (Baldry et al, 2007; Cappelli, 2000; Alvesson, 1995; Kunda, 1992). Knowledge work can include those working within consultancies, advertising, software engineering and research and development areas. Indeed, it has been suggested that software professionals may be the archetypal knowledge workers, in that this occupational group is considered to place a strong emphasis on intellectual, creative, challenging work, commitment to the occupation and an often individualistic view of careers (Baldry et al, 2007; Baldry et al, 2005; Newell et al, 2002). For example, stated by Baldry et al (2007: 132):

> Software workers especially are often regarded as the consummate knowledge workers…these more highly qualified workers enjoyed greater labour market leverage, status and flexibility in determining the course of a career within a chosen field of expertise and indeed a high proportion of these employees tend to regard their software careers as 'boundaryless', taking them to other organisations.

As discussed earlier, knowledge work is also considered to be ambiguous, complex and intangible in nature, due to the difficulty in identifying, articulating and formalising what individuals actually do (Alvesson, 2004; Thompson and McHugh, 2002). Through a framework of knowledge conversions, Nonaka (1994) outlines the variety of ways in which tacit and explicit knowledge (knowledge which can be codified, shared and transmitted) can be generated and diffused, such as through education, experience, socialisation, language and interactions. This fits with definitions of tacit knowledge, whereby Polanyi (as quoted in Thompson et al, 2001: 926) states that:

> Tacit knowledge is based upon the 'indwelling' of awareness and understanding by individuals – 'we know more than we can tell'.

The tacit nature of knowledge work can be further illustrated through the lens of professional software work, in that these workers may rely on tacit knowledge, experience and interactions for the generation and application of knowledge:

> …software developers are in essence using their knowledge and experience of extant technology to innovate and produce new applications. They are also expanding their knowledge base when

> applying existing knowledge to the new application in order to determine how to the new system performs. However, neither the initial development or the interpretation of the new technology is codified. The developers are using *tacit* knowledge, which cannot easily be replicated because it has not been stated in explicit form (Thompson and McHugh, 2002: 13).

Indeed, Reich (as quoted in Webster, 2004: 210) draws attention to the difficulty in articulating and expressing the process of knowledge work, due to its intangibility and complexity:

> How, then, do symbolic analysts describe what they do. With difficulty. Because a symbolic analysts' status, influence and income have little to do with formal rank or title, the job may seem mysterious to people working outside the enterprise web, who are unfamiliar with the symbolic analysts' actual function within it. And because symbolic analysis involves processes of thought and communication, rather than tangible production, the content of the job may be difficult to convey simply…it is not always instructive or particularly edifying to say that one spent three hours on the telephone, four hours in meetings and the remaining time spent gazing at a computer screen trying to work out a puzzle.

Baldry et al (2007) suggest that during the 1990s, organisations became increasingly concerned with knowledge management. This interest can be seen to have emerged from the desire of organisations to understand, manage, share and acquire knowledge, with the recognition that they may be reliant and dependent on workers for knowledge (Baldry et al, 2007). However, as demonstrated by Blackler (1995), the nature of knowledge can be complex and fall into a number of categories. For example, knowledge can be 'embrained', implicit and tacit in nature; 'encultured', part of processes and informal workings; ‘embodied’, part of what workers naturally do and based around contexts; ‘embedded’, based in routines, roles and procedures; and ‘encoded’, that which is more tangible in nature (Blackler, 1995). ICTs can be argued to have helped the capturing, storing and sharing of information, through methods such as databases, groupware software, e-mails and intranets (Newell et al, 2002). In addition, knowledge can also potentially be shared and accessed through methods such as teamworking and informal interactions (Hansen et al, 1999). However, as Newell et al (2002) argue, it may be more challenging and complicated to capture and identify knowledge and processes for those workers engaged in knowledge work. Indeed, following Blackler’s (1995) categories of knowledge, knowledge work can be seen to be ‘embrained’, through its tacit and implicit nature, ‘encultured’, with knowledge being part of processes and informal working, and ‘embodied’, being based around what workers naturally do. As a result, Newell et al (2002) suggest knowledge work may be difficult to express and codify, due to its tacit, intangible nature, its basis in personal experiences or intuition, dynamism and change and relation to context. Thus, as summarised by Alvesson (2004: 183), it could be argued that:

> Knowledge is a concept far too loose, ambiguous and rich and pointing in far too many directions simultaneously, to be neatly organised, coordinated and controlled. Given the complexities, tacitness and ‘dispersed presence’ of the knowledge phenomenon, there is a tension between knowledge and management. Given the problems of the objectivity and functionality pointed out above, there is, for example, the need for constant discussion, reflection, questioning and debating on what is ‘valid’ and how knowledge, as a resource, can be transformed into knowing in specific non-standard situations. All this goes beyond what management as structural, behaviour or normative control may deal with.

Knowledge workers may therefore be in the enviable position of being able to choose how and when to work, control and capitalise on their labour, as a result of having knowledge, skills and abilities which may be intangible and difficult to formalise (Spender, 1998). Individuals may, as a result, represent a source of competitive advantage to the organisation, with regards to their skills, abilities and expertise (Alvesson, 2004). In this sense, the skills, knowledge and expertise, as well as networks held by knowledge workers may be highly valued in the market place, meaning that knowledge-intensive firms may be more dependent on knowledge workers to retain competitive advantage (Donnelly, 2006). Indeed, Knell (2000) suggests that knowledge workers can be seen as 'free workers', in that they may have highly valued expertise and skills and more labour market power, making them less dependent on the organisation. Indeed, as Löwendahl (as quoted in Alvesson, 2004: 141) emphasises:

> These human resources – people – contribute much more than their labour. The significance of their networks and relationships is often great. The professionals bring to the firm their expertise, their experience, their skills in relationship building and maintenance, their professional reputation, their network of professional peer contacts and their established relationship with past, present and future clients.

Knowledge workers may therefore have greater opportunities for mobility as a result of their valuable knowledge and skill bases, making the retention of knowledge workers and how to make work conditions attractive to these workers important priorities (Alvesson, 2000).

Perspectives on Knowledge Work

As illustrated in earlier sections, it can be argued that knowledge workers may experience more favourable terms and conditions, more flexible work structures and looser forms of management, as a result of the distinctive features of knowledge work.

Indeed, Newell et al (2002: 20) summarises some of the key arguments concerning forms of work organisation and structures in knowledge work, claiming that:

> The traditional bureaucratic command and control structure will have to be jettisoned in favour of more flexible team – and project – based management approaches. Management will need to abandon traditional styles and structures and act more like conductors who coordinate the activities of knowledge workers. Managing knowledge within the knowledge-based occupation is, therefore, more about the management of the people employed in these firms, typically organised in teams, than about the development of information and communication technologies to extract and capture this knowledge.

Knowledge workers may require and experience more distinct, favourable working conditions, such as autonomy at work and more facilitative styles of management to encourage commitment and motivation, as well as enabling judgement, problem-solving and creativity processes (Thompson and McHugh, 2002; Spender, 1998). As claimed by Donnelly (2006: 78):

> The emergent category of 'knowledge worker' is representative of such changes: knowledge workers are being viewed contemporaneously as the vanguards of new organisational arrangements and the precursors of a new employment relationship, as they are expected to be able to extract deeper concessions from their employer(s) than traditional employees due to the level of employer-employee interdependency that is said to be in operation between the knowledge-intensive firm (KIF) and the knowledge worker.

The nature of education and qualifications that knowledge workers attain may also lead them to naturally emphasise particular conditions and types of work as being important (Newell et al, 2002). Thus, in this sense:

> A person, who, through education, training and on-going social relations, experiences herself strongly as a professional, may as a result give priority to certain values and motivators at work e.g. autonomy, knowledge development, specialisation as an outcome of the identity rather than an effect of the pure motivating power of instrumental sources of gratification (Newell et al, 2002: 208).

In this light, the distinct characteristics of knowledge work may require more facilitative styles of management, opposed to more traditional command and control structures such as those in traditional, manual and industrial areas of work (Mathews, as quoted in Thompson and McHugh, 2002). For example, Flecker and Hofbauer (1998) highlight the importance of achieving a balance between providing terms and conditions which may be valued by knowledge workers, as well as considering how work processes can be coordinated, facilitated and managed effectively. Due to the intangible and ambiguous nature of knowledge work, it may be more difficult for managers to access and retrieve knowledge from these workers. Autonomous working conditions may therefore be beneficial, through enabling knowledge workers to manage timing and techniques needed for work which is more ambiguous in nature (Gibbons et al, 2002; Warhurst and Thompson, 1998). Indeed, knowledge workers may have more insight into work areas, compared to managers who, whilst having a general overview of issues, may have less understanding of specific areas (Alvesson, 2004). As stated by Alvesson (2004: 22):

> Knowledge work – analysis, the exercise of judgement and problem-solving – is thus carried out by the majority of the personnel and not centralised on a managerial or technocratic elite, designing systems and procedures for others to follow. In KIFs the division between

> conceptualisation and execution is limited. There is not much space for management to establish a monopoly of intellectual work – in opposition to classic organisational forms, which build upon an extensive hierarchical division of labour.

Teamworking can also be considered to be important for knowledge work areas, through enabling individuals to interact, share and form new knowledge (Kofman and Senge, 1993). Indeed, Thompson and McHugh (2002) suggest that even though teamworking has been evident in the past, its current application can be seen as longer lasting and beneficial to knowledge work organisations. For example, Vygotsky (as quoted in Blackler, 1995) and Nonaka (1994) argue that team interactions may help to shape individuals, allowing for the circulation of knowledge and information and for the development of skills and knowledge. In addition, methods such as teamworking can be beneficial in that some types of knowledge possessed by individuals may interact with other types of individual knowledge, creating new knowledge at the individual, collective and organisational levels (Spender, 1998, 1996). This collective, informal knowledge can be seen as being the most beneficial and useful for organisations, as it may be more difficult for other organisations to understand and imitate (Spender, 1998, 1996). The area of professional software work can serve as an illustration of the importance of teamworking, in that individuals may need to collaborate in order to bring together different skills and knowledge, as well as to supplement levels of expertise and solve complex work problems (Baldry et al, 2007; Tam et al, 2002).

Alvesson (2000) and Tsoukas (1996) also draw attention to the intrinsic and normative nature of knowledge, stating that knowledge workers may enjoy the challenging and intellectually stimulating nature of the work itself. Knowledge workers may also emphasise intangible rewards, such as reputation, status, recognition, challenging and interesting work, career aspirations and development, as well as more tangible rewards, such as pay and bonuses. In addition, Frenkel et al (1995) and Tampoe (1993) suggest that knowledge workers may be motivated by being able to realise their potential, having responsibility over assigned tasks, accomplishing tasks and receiving monetary rewards to recognise contributions. Thus, in summary, it may therefore be useful for organisations to have an understanding of how norms are established, as well as areas which knowledge workers may consider desirable, in order to identify the most favourable working conditions and organisational forms (Alvesson, 2000).

However, whilst knowledge work may be held up as the vanguard of transformations in work structures and organisation, themes such as empowerment, flexibility, creativity and innovation could also be viewed as methods to help extract potential from knowledge workers, akin to Taylorist methods which attempted to identify, capture and extract knowledge from workers (Warhurst and Thompson, 1998). In this sense, practices in the new economy could be considered to merely represent new ways of managing and controlling workers. Indeed, Deetz (1995) suggests that knowledge workers, such as software professionals, may require minimal managerial supervision as they may derive their identity from their occupation and enjoy the work, which in turn can intrinsically motivate and act as a form of normative control. In addition, Mathews (as quoted in Thompson and McHugh, 2002) claims that new ways of working, such as teamworking, may encourage workers to self-manage themselves. For example, Beirne et al (1998) suggest that within professional software work, the devolution of responsibility to employees in team-based structures to deliver good quality products may also require monitoring of peer team members' work efforts. Thus, team-based structures may require individuals to review from within, co-operate more extensively and be self-disciplined (Thompson and McHugh, 2002). Furthermore, Causer and Jones (1996) suggest that whilst knowledge professionals may typically be managed through high trust strategies, hidden control measures, such as performance appraisals and performance-related pay can act as mechanisms to allow monitoring and maintenance to be carried out and encourage greater levels of self-management. Therefore, it can be suggested that:

> Contemporary initiatives fundamentally re-organise the process of work: 'In place of command and control structures designed to enforce rigidity and compliance, the new production systems call for management that offers facilitation and guidance and co-ordination between self-managing groups of employees who are capable of looking after the details themselves (Mathews, as quoted in Thompson and McHugh, 2002: 165).

Furthermore, a variety of occupations, including those which may be considered to be professional, may be subject to elements of specialisation and de-skilling as a result of new technologies and processes, affecting work organisation, work relationships and the overall quality and structure of jobs (Lyon, 1996). This may therefore present a less optimistic picture of the knowledge sector. For example, Rueschemeyer (1986) suggests that in knowledge-based occupations, division of labour can come from capital's need to produce more efficiently through specialisation. Indeed, within software work, the adoption of electronic engineering principles in the 1960s can be seen to have played a major role in the development of professional software engineering as a separate occupation from less skilled areas of software work, such as routine information technology work (for example, data entry) and problem-solving and maintenance work (for example, help desk and support functions). The segregation of occupations within the information technology field can therefore be seen to have potentially stratified skill, experience and educational requirements for these functions.

The following section will continue this discussion by focusing on the development of the software industry and the emergence of professional software work as a separate occupation from routine and maintenance areas.

2.2 SOFTWARE: HISTORY, IMPORTANCE AND DEVELOPMENTS IN THE INDUSTRY

Advances in technology and engineering can be seen to have accelerated the development of computers, allowing for greater hardware miniaturisation, high speed buffers and greater memory to receive and store data (Friedman and Cornford, 1989; Kraft, 1977). Furthermore, developments in hardware are claimed to have created the conditions for developments in software and systems (Friedman and Cornford, 1989; Couger, 1982). Many writers (Baetjer, 1998; Sichel, 1997; Quintas, 1994; Kraft, 1979) suggest the development of the software industry can be explained through examining the history of the computing industry and associated technologies. For example, Quintas (1994) suggests the software industry has emerged through technical change in terms of the evolution of computing languages, tools, methods and techniques; organisational change, through new ways of managing people and processes; and the development of software into packaged, custom and services areas. Economic, political and organisational factors, in addition to changes in the way computers are used in businesses and society, are also considered to have influenced computer systems development, purchasing habits and the importance of software in general (Key Note Market Review, 2004; Friedman and Cornford, 1989).

This section will provide an overview of the hardware and software industries, the history and development of the software industry and the evolution of software engineering. It will then demonstrate the importance of software to work, business and society and discuss recent developments within the software industry.

The Hardware and Software Industries

The computer industry can be segmented into hardware, packaged software and software services, with each exhibiting different structures and levels of concentration. The hardware industry is composed of two main areas: data processing equipment (such as PCs, portables and servers) and computer peripherals (such as printers, laptops, keyboards and monitors (Key Note Market Review, 2004). Datacoms and PC hardware are the largest sectors of the market, with increasing emphasis on storage devices and portables, such as laptops and internet capable mobile phones (Key Note Market Report, 2008a). The hardware sector is typically structured around economies of scale and dominated by large global organisations, such as Hewlett Packard and Compaq. Mergers are an increasing trend in the hardware sector, in an attempt to expand and achieve greater geographical reach and to help cover growing development costs for new products and technology (Key Note Market Review, 2004). The hardware sector is characterised by high volume trading and competitive pricing, with organisations claiming technological superiority and attempting to reduce prices, in order to achieve competitive advantage (Key Note Market Report, 2008a).

Improvements in processing power, memory and decreasing costs in computational devices are suggested to have facilitated the continued development of computers and programming (Baetjer, 1998; Lavoire et al, 1993, 1991). Furthermore, developments in hardware are claimed to have created the conditions for developments in software and systems (Friedman and Cornford, 1989; Cougar, 1982). Research and development are also considered to be a major driver of growth and change in the information technology sphere, encouraging the move from hardware to packaged software and software services (OECD Information Technology Outlook, 2006).

The software industry includes three discrete areas: customised (secondary) software, which is made specifically for an individual organisation; services, provided by consultants and specialist organisations to a range of firms; and packaged (primary) software, which can be purchased by a wide variety of firms and consumers and adapted to suit specific organisational needs (Barrett, 2004; Marks et al, 2003; Marks et al, 2001; Sharpe, 1998). The packaged software market is populated by a range of large, global multi sector organisations producing both software and hardware (such as Hewlett

Packard, IBM and Sun Microsystems), small organisations and independent software organisations, operating in a variety of markets and supplying various types of systems and software (Key Note Market Review, 2004). The packaged software market includes business software (dealing with operating systems and application systems) and leisure and entertainment software (relating to games software for consoles, handheld systems, PCs or laptops). Applications systems software presents development opportunities to software organisations, as a result of increasing interest in e-business, tools, online sales and marketing (Key Note Market Report, 2008b).

It is claimed that whilst high levels of expertise and skill may be required to create packaged software, once in use, packaged software can be used and modified by individuals with less skill (Kraft, 1979). Some organisations may therefore choose to purchase packaged software instead of having systems custom built, allowing lower level workers to customise and integrate standard packages with other systems (Quintas, 1994). Sharpe (1998) argues that as a result of replacing bespoke with packaged software, levels of expertise and skill in non-specialist organisations may gradually begin to erode. However, Quintas (1994) emphasises that, in practice, it may be difficult to predict how packaged products may interact and behave with existing systems. As a result, expertise and skill from software professionals may be important to make modifications and customise software to fit with user requirements.

The customised and services software sectors encompass analysis, design, development, operation, installation and maintenance of software and systems (Key Note Market Report, 2008c). The services sector is extremely diverse, featuring small partnerships, consultancy services and outsourcing, which focus on personal contracts for selling services, through to multinational companies (such as IBM and Fujitsu) and large-scale freelancing, requiring substantial resources and budgets in order to cover investments (Key Note Market Review, 2004). The services sector includes three main areas: professional services, encompassing the development, configuration, testing, installation and management of IT systems; support services, involving maintenance, third-party maintenance and telephone support; and operational services, relating to the outsourcing of IT-related activities, such as operations, infrastructure, applications or security, by firms trying to either exploit the current system or focus internal resources on core operations (Key Note Market Review, 2004). Planning and design is an additional, growing sector which is largely project based, ad hoc in nature and offering consultancy-style services, in terms of helping clients decide what to buy and how to organise information technology strategies.

The History and Development of the Software Industry

The generation and development of hardware and software can be witnessed through technological developments in electrical components, design, materials and the capacity, speed and reliability of machines (Friedman and Cornford, 1989). Hardware is considered to have developed from early valve technology, progressing through to the development of transistors, integrated circuits and large-scale integrated circuits (Friedman and Cornford, 1989). Early computers used electronic mechanical devices as the major input and output devices, such as punch cards, magnetic tapes and paper readers. The first computer, ENIAC, was built during World War Two and required a complete program to be written and installed in order to function. Programming was initially considered to be a largely routine, clerical function, exhibiting a high concentration of female employment as a result. However, as programming developed in practice, it was seen to be a more complex occupation than originally believed, requiring an understanding of the logics, physical structure and mechanical operation of the computer (Quintas, 1994; Friedman and Cornford, 1989). Software therefore emerged as the necessary complement to hardware, providing the instructions for computers to function:

> Along with machine operators who oversee the physical operation of the “hardware”, programmers are responsible for a computer’s day-to-day operation. In particular, they provide the computer with a detailed

> sequence of operating instructions, which can be in any one of a large and growing number of program "languages". Without these human-entered instructions – the "programs" – the most advanced computers would sit dumb and idle and useless (Kraft, 1979: 1).

The recognition that software was indispensable to hardware resulted in the definitional change in programming from a relatively unskilled, unimportant function to one considered intellectual in nature, rapidly increasing the attractiveness of the occupation to males. Thus, gender, en-skilling and organisational change were intertwined in the early development of the software industry.

Early programs were written in machine binary code, where numeric codes defined operations and the locations and codes corresponded with binary functions or hardware (Quintas, 1994). Programmers required a detailed understanding of machine functions, in order to anticipate what circuits should be used, to enter instructions in the correct sequence and to make any changes or alterations to programs (Friedman and Cornford, 1989; Quintas, 1994). However, binary coding was considered to be a time-consuming and error prone process. In the 1940s, the stored program computer was created - coinciding with the invention of the transistor - which could be programmed and was less time-consuming as a result (Kraft, 1977). Instructions could be stored in the computer memory, where the hardware unit within the computer body decoded stored instructions, enabling the computer's arithmetic logic to process data (Kraft, 1977). Assemblers were introduced in the early 1950s, following the introduction of decimal/alpha-numeric codes, representing a move towards natural language, even though they were still machine specific in nature (Quintas, 1994). Assemblers involved translating the mnemonic (memory-aiding) representations of instructions into binary equivalents to keep track of locations, making it easier to check programs, make alterations and reduce errors (Friedman and Cornford, 1989; Lavington, 1980; Collin, 1978). However, the machine oriented nature of assembler language meant that syntax and vocabulary continued to be restricted to basic machine operations and instruction sets which corresponded to hardware functions, rather than real world tasks required by applications. The machine specific nature of assemblers also meant that programs could not be run on other machines and had to be re-coded, in order to be run on other computers (Quintas, 1994; Friedman and Cornford, 1989). Furthermore, programming continued to be a time-consuming and elaborate process with constraints in the mechanical devices, due to the presence of moving parts and material strength (Friedman and Cornford, 1989; Kraft 1977).

In the 1950s, the emergence of high level languages meant that individuals required a significantly less detailed understanding of the circuitry or physical operations of computers, leading to more of a division between hardware and software (Friedman and Cornford, 1989; Greenbaum, 1976). High level languages also represented the increased recognition and importance given to software, software cost and programmer productivity (Friedman and Cornford, 1989). Third generational languages (3GLs) such as FORTRAN (Formula Translation), which was used for scientific and engineering applications and COBOL (Common Business Oriented Language), which was used for commercial applications, used vocabulary and syntax reflecting natural language, meaning that programmers could write complex or extensive programs without having detailed knowledge of machine functions or machine languages (Quintas, 1994; Kraft, 1979). Fourth generational languages (4GLs) evolved during the late 1970s and early 1980s, with vocabulary and syntax even more like natural language, opening up the spectrum to a wider group of individuals (Baetjer, 1998; Quintas, 1994). Object-Oriented programming (OO) is considered to be one of the most recent technological innovations in programming language, allowing for the identification of discrete objects within the system being developed and delivered (Quintas 1994). Whilst older languages required whole programs to be completed before running, OO languages enable a more incremental and detailed understanding of systems, as well as allowing for a clear division of knowledge and ensuring objects do not interfere with data (Baetjer, 1998).

From the 1960s, various tools have been developed to assist and support programmers in areas such as planning and estimation, systems analysis, design, testing, debugging and maintenance (Voelcker, 1988). These tools include debuggers, which can help programmers to find or fix mistakes on screen, as well as viewing what has been created and how it works; code generators, which help to translate higher-level specifications into code; browsers, which allow programmers to examine different aspects of programs and systems from various viewpoints; and version control tools, which help to coordinate work, help to integrate changes and ensure workers can keep track of different versions of different models (Baetjer, 1998). Computer Aided Software Engineering (CASE) is another tool which can facilitate software development, as well as automate certain processes. CASE was originally considered to help with the automation of software development, automatically transforming knowledge of what the system was to do into code (Baetjer, 1998). However, due to the evolutionary, iterative and creative process of software engineering, CASE tools have been utilised more as translation tools, in order to help software engineers to better understand what they are doing. It can therefore be argued that:

> In this respect, software development is like any other design process. A bridge is fully designed only when the drawings, materials to be used and other specifications are fully worked out. What the engineers are doing in designing a bridge is thus coming to understand it fully so that they can articulate their design in complete detail in the appropriate code – in their case detailed drawings, material specifications and so on…In short, software development can be understood as a process of coming to understand what a software system should do. Any software application starts as a rather sketchy, abstract conception; it finishes when this vague conception has been satisfactorily articulated in executable code. Thus the process of building a piece of software is a process of coming to understand it fully in the sense of being able to

> articulate it in detail. Developing software is a matter of understanding fully what we are trying to achieve (Baetjer, 1998: 67).

In this context, it is suggested CASE can be used to help provide diagrammatic support (allowing for visualisation of the system, as well as assisting with the drawing and checking of diagrams), data management (helping to maintain coordination between work giving information on the evolution of systems), speed up changes and allow perspectives in programs to be seen which may be different from the code (Baetjer, 1998).

At first appearance, the evolution of programming languages and tools may appear to have enabled software work to be carried out by non-specialists. However, computing languages based on natural language and syntax may not necessarily have opened up all areas to novices (Marks et al, 2001). For example, whilst pure 4GLs may require little programming expertise and allow less trained and skilled individuals to modify some standard programs, hybrid 4GLs require considerable programming expertise, advanced skills and knowledge (Grindley, 1986). Similarly, tools such as CASE, whilst intended to support and improve the development process and provide some standardisation of processes may not always be used by software professionals (Marks et al, 2001).

The Evolution of Software Engineering

Initially, programming was considered to be an all-encompassing task, where design, development, testing and maintenance activities were all carried out by individuals and groups, with no clear-cut divisions of labour (Kraft, 1979). Computers were less complex, leading to more unstructured programming approaches, such as ‘hacking’. ‘Hacking’ involved programmers writing code without any planning and then ‘hacking’ at it to get results and remove bugs (Baetjer, 1998; Quintas, 1994). However, in the 1950s, the electronic engineering discipline was introduced to provide structure to programming and eradicate the capacity for human error. Engineering is considered to be one of the oldest, most organisationally developed of modern technical occupations, invented by science-based industries to apply theoretical and practical scientific principles to the production process (Kraft, 1979, 1977). Programming therefore became re-defined as 'software engineering' due to the adoption of scientific, mathematical and practical engineering principles (Naur and Randall 1969). Engineers could also be seen to perform two major roles: firstly, to re-arrange and re-design work tasks to increase managerial control over production processes and secondly, to simplify work tasks so the same work could be carried out by less skilled workers (Kraft, 1979).

By the end of the 1960s, programming had broken down into three sub-divisions: systems analysis, which dealt with the specifications and designs of a system; programming, relating to the development and coding of systems; and coding, for more routine coding functions (Kraft, 1979). This led to the division of labour and separation of what was considered to be more the more intellectual task of design from the more routine task of code writing. The adoption of electronic engineering principles also played a major role in the development of professional software engineering as a separate occupation from the less skilled areas of software work, such as routine information technology work (for example, data entry) and problem-solving and maintenance work (such as help desk and support functions). Indeed, it is claimed that the segregation of occupations within the software industry has stratified skill, experience and educational requirements, from software professionals requiring degrees, expertise and experience, through to data processors, requiring generic information technology skills.

More formal structured approaches to software engineering also began to emerge in the 1960s and 1970s, representing further attempts to develop industry-standard processes and achieve greater orderliness, simplicity and structure in software engineering (Raghaven and Chand, 1989; Kraft, 1979). Structured programming methodologies represented an attempt to deal with increased complexity in software engineering, as well as potentially freeing managers from dependencies on higher skilled workers in designing or writing programs and making fragmentation of labour more possible (Kraft 1977). A methodology in software engineering can be defined as:

> …a set of procedures that a software development organisation follows (or tries to follow) in producing new software (Baetjer, 1998: 36).

Under structured methodologies, large programs are broken into sub-systems and different modules, allowing different individuals to work and focus on different modular parts (Baetjer, 1998). For example, the 'waterfall cycle' methodology involves the cascading of set stages in the software development process (Requirements and Analysis, Design, Development, Testing, Installation), allowing work to be split into modular sections. The 'Agile' methodology also involves the cascading of software development stages but is more iterative in nature, with the running of parallel, overlapping releases. Modularity can also aid the maintenance and enhancement of systems, through making it easier to understand individual system elements and the relationships between these, as well as making it easier to identify and make changes without affecting other parts of the system (Baetjer, 1998). These methodologies have also emerged to assist software engineers in dealing with and managing unclear requirements and consistency between stages, as well as enabling greater client involvement. Indeed, as discussed in greater detail in Chapter Section 3.1, user interactions are increasingly important through helping software professionals understand more specifically what the user is looking for and helping to improve the effectiveness of systems (Baetjer, 1998).

The division of labour into modular sections under the 'waterfall cycle' and 'Agile' methodologies also enables work to be distributed between individuals and teams, depending on the knowledge, skills and expertise possessed by individual software engineers (Baetjer, 1998). In this sense, structured programming methodologies such as the 'waterfall cycle' and 'Agile' may give rise to an authority hierarchy, with the fragmentation of work and rank based on levels of skill and expertise (Kraft, 1979).

Developments in structured programming and methodologies could therefore be considered to have polarised the activities and work areas performed by software professionals (Kraft, 1979). Chapter Section 4.3 will look further at the nature of work performed by software professionals by categorising job roles and providing insight into tasks and activities.

The Importance of Software

Software is considered to have applicability in virtually all work situations and types of human activity, transforming business, work and society. For example:

> …software may control a heart pace maker or the braking system in a car, or process credit transactions or images from the hubble telescope i.e. software is code that creates virtual machines that do things and it gets just about everywhere (Quintas, 1994: 29).

Software can also be seen to be extremely important in driving innovation and aiding companies in the search for competitive advantage:

> Today it is clear that software is the key technology behind much of what is driving the innovative aspects of the economy and is at the heart of the revolution taking place in the use of electronics and telecommunications services by the consumer. In economic terms, it offers the potential opportunity to increase UK productivity and enable the UK to compete in an ever more competitive global economy (Intellect Software and IT Services Report, 2009: 25).

Whilst businesses are increasingly reliant on software for functions, interactions and to remain competitive, society is also considered to have become increasingly reliant on technology for social interaction, leisure and entertainment (Scotland IS: Innovation and ICT Skills in Scotland, 2008). As a result:

> This means that ICT skills are increasingly in demand, both within ICT-based businesses who design and create solutions for business and leisure and for those within non ICT-based businesses to have sufficient knowledge of IT to be able to identify and implement new ways of using ICT. This trend is likely to increase further with the next generation of consumers and ICT workers having been raised in an ICT-heavy society ('digital natives') (Scotland IS: Innovation and ICT Skills in Scotland, 2008: 36).

Private sector organisations are considered to be amongst the main purchasers of software and services, especially in areas such as finance, banking, telecommunications, retail and computer services (Key Note Market Review, 2004). For example, the financial sector is one of the largest sectors for outsourced computer services, with software used to maintain and update current and planned systems (Key Note Market Report, 2008c). Software is applied within investment banking to enable 24/7 operations, the execution of large transactions and also facilitate trading activities. In addition, retail banking employs software to manage high volume customer transactions, online banking services, cash machines and over the counter services (Target IT, 2007). Software is also utilised in the retail sector, through helping to meet changing customer demand, customer service expectations and supporting business functions and company infrastructure (Key Note Market Report, 2008c; Target IT, 2007). The telecommunications sector makes use of software to provide the infrastructure for telecommunications technology development, as well as to enable the diversification and broadening of products and services (Target IT, 2007). Telecommunications firms may also make greater use of those firms providing specialist software services by outsourcing non-core computer activities and functions, such as the maintenance of call centre infrastructure (Target IT, 2007). The public sector has also experienced increased demand for software services, due to technology investments in education, health, central and local government (Key Note Market Review, 2004).

Various political, economic and organisational factors, such as economic and market conditions, regulatory changes, competition, globalisation, downsizing, networking and outsourcing can have implications for information technology budgets, priorities and strategies (Intellect Software and IT Services Report, 2009; Quintas, 1994). For example, Key Note Market Review (2004: 93) highlights the impact of economic conditions on information technology decisions:

> In times of stable economic growth, IT budget decisions are normally taken over in the long-term (buying new systems and products involves significant costs in staff time) but the major budget holders have renewed their spending plans over shorter timescales since 2001…When IT budgets are squeezed, companies do not necessarily stop spending but they give higher priority to 'must have' new products and services (such as security) and a lower priority to the upgrading of older systems that still function. Companies also switch to products and services that have lower on-going running costs (e.g. Linux instead of Microsoft Windows or Unix).

Despite economic and organisational changes, software therefore remains an area of importance, due to organisations' continuing need for software to be operated, maintained and updated, as well as to reduce costs and increase efficiency (Key Note Market Report, 2008b). Indeed, Baetjer (1998) highlights that the maintenance and adaptability of systems is important, in order to manage existing applications effectively and allow for adaptations or updates to address changing scenarios or circumstances. For example, whilst economic factors have led to some decline in software business within the financial sector in Scotland, financial organisations are still one of the largest users of business software, due to the need to reduce spending and improve efficiency of systems (Scottish Technology Industry Survey, 2009; Key Note Market Report, 2008b). In addition, Scottish markets are increasingly looking at software to help supply business critical systems, such as infrastructure outsourcing and applications management (Scottish Technology Industry Survey, 2009). As a result, software professionals are responsible for building systems and software that can evolve and support changing organisational requirements, as well as helping to increase productivity and improve effectiveness (Quintas, 1994). Changing economic and organisational conditions can therefore enable software workers and companies to play a continuing important role in the economy (Scottish Technology Industry Survey, 2009). In this sense:

> The strength of the business software market is that modern businesses and government organisations cannot function without state-of-the-art software, which must be maintained and updated regularly. Business software is said to be 'recession proof', because businesses need it to help reduce costs when budgets are cut and to maximise their reach when marketing to customers (Key Note Market Report, 2008b: 28).

In summary, research on software professionals can therefore be argued to be of particular interest, due to their role in creating a technology which can be seen at the heart of the economy and society in general.

Recent Developments in the Software Industry

New technologies and trends have emerged in the software industry, with emphasis on server based application services and internet based devices, emerging from developments in digital technology, convergence and demand for mobility from computer services providers and consumers (Key Note Market Report, 2008b, 2008c; Key Note Market Review, 2004). 'Open source software' is a developing area, where individual software engineers can make changes to software, add features or build on applications and distribute changes, whilst keeping source code open for others to access

(Key Note Market Report, 2008b; Key Note Market Review, 2004). In addition, 'Software as a Service' (SaaS) allows applications to be provided to customers over the internet opposed to on the computer, as a result of changes in the marketplace, with software workers supporting applications for clients, assuming more operational and maintenance tasks to support client software (Key Note Market Report, 2008b; Key Note Market Review, 2004). Indeed, online companies such as Amazon and Google which offer SaaS services may represent a competitive alternative to traditional suppliers of computer services. Online retail is a further expanding area, aided by developments in software, telecommunications, technologies and security systems (Key Note Market Report, 2008b, 2008c). Concerns in modern society over terrorism and security have also led organisations and governments to consider the security of systems more closely, placing more importance on security software systems, data management and systems administration (Key Note Market Review, 2004).

As discussed in Section 2.1, developments in ICTs and globalisation are also claimed to have increased international trading of services and encouraged greater flexibility in choice and location of labour (OECD Information Technology Outlook, 2006; McGrath-Champ, 2005). In the Scottish context, these factors are considered to have resulted in greater competition for business and talent, in that:

> Globalisation is having a major impact on the ICT sector, driving a complex restructuring and skills shift. Improved communications and the globalisation of supply chains means that Scottish companies are now often in direct competition with companies from around the world, including those from economies with a strong track record in high level ICT (particularly the USA), leading to increased pressure for constant technological innovation. As well as competing for business, there is also global competition amongst firms in attracting the brightest talent (Scotland IS, 2008: 5).

Globalisation and developments in ICTs are also claimed to have stimulated an increase in mergers and acquisitions and re-location of work to developing countries, in order to make cost savings and attain greater access to production capabilities and markets (OECD Information Technology Outlook, 2006). In this sense, it could be argued that:

> There is a global restructuring of ICT production activities, with the emergence of Eastern Europe and non-OECD developing countries as both producers and growth markets and the global rationalisation of ICT and ICT-enabled services production. This new wave of globalisation has largely been driven by efficiency-seeking competition, with firms taking advantage of cost differences and the rapid development of goods and services production capabilities in developing economies to service global markets. With the growth of developing countries there is also an increasing need for the presence of ICT firms in these new growth markets. Once again, the ICT sector is at the forefront of globalisation and is enabling the continued international rationalisation of production within the sector and beyond (OECD Information Technology Outlook, 2006: 86).

A developing trend within the software industry is that organisations may choose to outsource their software services to specialist software organisations in the same or other countries (Sharpe, 1998). Outsourcing is argued to be beneficial to organisations through enabling them to focus on core functions, take advantage of innovative services, fill in internal skill gaps, receive services required on a short-term basis and improve business processes (Key Note Market Report, 2008c; McManus and Floyd, 2005; Arora et al, 2001). Many organisations may also attempt to make savings on software services through outsourcing or offshoring work to other regions, such as Eastern Europe, the Pacific basin, South America and Asia (Intellect Software and IT Services Report, 2009; Key Note Market Report, 2008b). Outsourcing or offshoring software to other countries is heralded as being beneficial to organisations, through providing cost savings, attractive labour conditions and wages, the potential for 'round the clock' services and the tactical geographical deployment of individuals in increasingly competitive, global markets (McManus and Floyd, 2005).

India has emerged as a strong contender for firms wishing to outsource and offshore their software services, as a result of the advantageous conditions and opportunities it presents. In addition to the above cited benefits, outsourcing services to India is argued to have become attractive due to lower wages, English speaking and well-educated software engineers and favourable government policies, such as economic liberalisation and tax breaks (McManus et al, 2007; Athreye, 2005). Whilst Indian software firms deal with a range of professional services, products, consultancy, training and data processing, many Indian software firms may focus on services, as opposed to packaged software (Balasubramanyam and Balasubramanyam, 2000). Services are highlighted as being more appealing to Indian software organisations, due to favourable government policy, incentives and investments for services, as well as a lack of intellectual property rights and piracy issues for packaged software (McManus et al, 2007; Arora et al, 2001). Initially, the Indian information technology sector was not well developed domestically, providing fewer opportunities for developing products for domestic markets, making services the preferable option (McManus et al, 2007; Arora et al, 2001). Indian software firms may conduct activities offshore, on-site on client sites, or with a mix of on-site and offshore work. Whilst many activities offshored may be considered to be lower cost and lower value to the client organisation, some Indian software companies are attempting to advance up the value chain to compete through quality and complexity of services, rather than merely cost advantages (Balasubramanyam and Balasubramanyam, 2000). As a result, it is suggested that Indian software companies have begun to develop more mature software development processes, developing products from earlier services provided and moving into emerging areas such as e-commerce (Arora et al, 2001).

China is also claimed to be another country developing its information and communication technology industries in an attempt to attract companies seeking to outsource or offshore software activities (Key Note Market Report, 2008c). Many hardware firms have developed software sections and specialist software firms have emerged, leading to many software firms and sectors in Chinese states (Tschang and Xue, 2005). Around 80% of domestic software demand is suggested to be generated internally, where many user organisations use domestic software firms as they understand domestic markets which are fragmented by provincial boundaries (McManus et al, 2007; McManus and Floyd, 2005; Tschang and Xue, 2005). China also has a fairly fragmented and competitive market, with domestic firms competing on the basis of low cost and low wage strategy, coupled with multinational and foreign firms which may squeeze middle firms (McManus et al, 2007; OECD Information Technology Outlook, 2006; Tschang and Xue, 2005). Tschang and Xue (2005) suggest that many Chinese software firms are small and spread over a large number of sectors, making it difficult for organisations to compete for large-scale work or specialise and to achieve greater expertise, in order to compete in markets for more complex or advanced work. In addition, China is also suggested to have weak intellectual property rights protection and a weak industrial base, presenting a challenge for movement into higher level areas (OECD Information Technology Outlook, 2006).

Government policy has influenced the development of the Chinese software industry, through the liberalisation of restrictions on foreign direct investment and the encouragement of software research, development and firm formation (OECD Information Technology Outlook, 2006; Tschang and Xue, 2005). These policies also attempt to enable Chinese organisations to invest overseas through mergers and acquisitions, in order to obtain access to technology and distribution channels (OECD Information Technology Outlook, 2006). It is suggested that China may also be able to compete more effectively through building market alliances across provincial boundaries, attempting to specialise in areas to differentiate from competitors, reducing reliance on domestic markets, improving English skills and obtaining more access to external high technology markets (McManus et al, 2007; Tschang and Xue, 2005).

2.3 CHARACTERISTICS OF PROFESSIONAL SOFTWARE WORK

Sections 2.1 and 2.2 of this chapter have provided valuable insight into the emergence of knowledge work within the new economy and the development of new industries, such as software work. This section will explore professional software work in greater depth, in order to develop an understanding of this occupation and identify areas of interest for this research. Notably, Section 2.3 focuses on the main characteristics of professional software work in terms of its archetypal status as a knowledge work occupation, issues of identity and its intrinsically interesting and satisfying attributes.

Software Workers as Knowledge Workers

Conventional wisdom and stereotypes of professional software work suggest that workers tend to be typically in their twenties or thirties and predominantly male, with a minority of females in the industry (Key Note Market Review, 2004). Professional software work is perceived to place a strong emphasis on intellectual, creative, intangible, challenging work, with autonomous working conditions, low bureaucracy, flatter, more fluid working structures and project team work organisation, resulting in many researchers positioning it as the archetype of knowledge work (Baldry et al, 2007; Baldry et al, 2005; Newell et al, 2002; Alvesson, 1995; Kunda, 1992).

It is also claimed that the emergence of knowledge work in areas such as professional software work have shifted traditional perceptions of professionalism based solely on institutional foundations, professional affiliations and ethical codes to new ideas of professionalism, defined by the intangibility of knowledge and the occupational and organisational context (Fincham et al, 1994; Gibbons et al, 1994). Professional software workers are deemed different from traditional professionals, such as lawyers or doctors, as they may not always have professional affiliations or subscribe to formal ethical codes. Arguably, these new, contemporary professionals are concerned with business and organisational needs, with an emphasis on the creation of new skills, knowledge and their application, as opposed to the monopolisation of a field of specialist knowledge like traditional professions (Flood et al, 2001).

Software professionals are largely perceived to have an intangible knowledge base which cannot easily be formalised (Andrews et al, 2005). This tacit, intangible knowledge and understanding may be difficult to express, as it resides in experience, intuition, speculation and context (Newell et al, 2002; Bird, 1995). Indeed, professional software workers can be seen to fit Reich's (1991) 'symbolic analyst' category, due to the need for problem-solving, problem-identification and abstract thinking, in order to understand the linkages, causes and consequences between concepts and processes. Professional software work is therefore held to be a creative process, where individuals need to make sense of concepts, relationships and processes and learn by 'doing' (Baetjer, 1998). For example, as illustrated by Baetjer (1998: 64):

> A creative process such as software design is not deterministic, with output dictated by input through some sort of black-box optimisation. This would require the designer to grasp the problem in its entirety at a glance and on that basis to grasp its "correct" solution. On the contrary, software design is an evolutionary process in which the designer "makes sense" of the problem over time and gradually puts the design together. In this respect software design would seem akin to writing. Composition is not a matter of copying out a book that has somehow popped into the writer's head. Rather the writer works gradually from a vague idea to a fully-conceived book, through a process of fleshing out, defining and refining, finding out what works by trial and error. Similarly, the software designer uses feedback from the design itself, seeing what works, what has promise, what relationships are revealed that were unclear before.

Identity

It has been suggested that professional software workers have multiple identities, due to the influence of education, status, career, work, conditions and pay (Alvesson, 2004). For example, features such as work autonomy, self-management, working structures and arrangements are argued to potentially impact on professional identification and encourage occupational identity (Barrett, 2001; Alvesson, 2000). Furthermore, the intrinsic orientation to work, as well as the utilisation and development of skills and knowledge to provide services to the organisation and clients may encourage the identification with the occupation:

> …values as software professionals are very much tied up with their self-image as providers of a professional service and the necessity to perform well on behalf of both the customer and the company (Baldry et al, 2007: 125).

Software professionals may also identify with organisational sub-units, such as departments or teams, as these may have more relevance to the individual than wider organisational identity (Lockyer et al, 2001; Alvesson, 2000). For example, professional software workers carrying out on-site work may experience an increased sense of team identity, due to the physical and suggested psychological distance from the employing organisation (Lockyer et al, 2001).

Organisations may therefore face the challenge of encouraging professional software workers to identify with the organisation as a whole, due to the presence of competing identities (Alvesson, 2004). Identification with the organisation can be considered important in terms of encouraging individuals to support and be committed to the organisation, as well as strengthening retention. It is claimed that if organisations attempt to meet the needs and goals of professional software workers, individuals may give their commitment and loyalty, potentially reducing conflicts between professional and organisational identity (Alvesson, 2004; Herriot and Pemberton, 1995; Wallace, 1995; Lachman and Aranya, 1986). The presence of occupational, team and organisational identities can therefore have implications and consequences for employment policies and working conditions (Lockyer et al, 2001).

Occupational, team and organisational identity are also suggested to encourage and motivate software professionals to adapt themselves to the cyclical nature of software work, such as where overtime and an increase in work effort may be necessary to meet approaching deadlines. Indeed, the propensity to adapt flexibly to changing work demands can be seen to reflect the intrinsic nature of professional software work:

> Here, as in the case of most software developers, concessions to organisational demands for flexibility and extra work are accepted as part of the job and point to different attitudes towards the elastic demands of the contemporary contract (Baldry et al, 2007: 140-141).

Attributes of Professional Software Work

Professional software work is argued to have attributes which make it intrinsically interesting and satisfying for those who perform it (Baldry et al, 2007). For example, Tampoe (1993) suggests that knowledge workers such as software professionals may experience satisfaction through personal growth (realising potential), operational autonomy (being able to achieve assigned tasks) and task achievement (feeling a sense of accomplishment). The areas which software professionals emphasise as being appealing and motivating may also be related to expectations and the educational and social context surrounding work (Newell et al, 2002). As emphasised by Alvesson (2004: 208):

> A person who, through education, training and on-going social relations, experiences herself strongly as a professional may as a result

give priority to certain values and motivators at work e.g. autonomy, knowledge development, specialisation as an outcome of the identity rather than an effect of the pure motivating power of instrumental and hedonistic sources of gratification.

In addition, attention to intrinsic and normative areas of professional software work, along with the provision of interesting and challenging work may potentially influence decisions to apply to and remain with an organisation (Alvesson, 2004).

From a managerial and organisational perspective, whilst some occupations may be subject to rules, procedures and performance indicators in order to manage work, the intrinsic and normative nature associated with professional software work can therefore be seen to potentially influence the "ideas, expectations and subjectivities of people" (Alvesson and Kärreman, cited in Alvesson, 2004: 137). For example, professional software workers may therefore potentially require minimal managerial supervision as they may derive their identity from their occupation, which, in turn, can motivate them and act as a form of normative control (Lockyer et al, 2001; Deetz, 1995). Chapter Section 3.3 further explores this area, emphasising that the literature tends to place considerable emphasis upon the intrinsically satisfying aspects of professional software work, at the expense of critical evaluation that might indicate alternative realities, such as normative control, performance indicators and the pressure of deadlines.

2.4 WORK INTENSITY

This chapter has argued that the status of professional software work as an archetypal knowledge work occupation makes it ideal for scrutinising the key debates on the new economy. In addition, this chapter has discussed the emergence of the software industry and outlined the main characteristics of professional software work, in order to identify potential areas of interest for this research. The following section will provide the rationale for research into work intensity, define the terms ‘work intensity’ and ‘work intensification’ and identify factors which may have implications for experiences of work intensity in the new economy. Notably, this section argues that it is essential to explore whether software professionals are necessarily immune from work intensity, due to their role in creating a key technology at the centre of the economy and their status as archetypal knowledge workers.

Defining Work Intensity and Work Intensification

Definition and terminological imprecision frequently conflate terms such as ‘work intensification’, ‘work intensity’, ‘work overload’, ‘work strain’, ‘high job demands’ and ‘role overload’. In order to provide clarity of focus to this research study on work intensity, it is therefore essential to define ‘work effort’ ‘work intensity’ and ‘work intensification’. Common to both ‘work intensity’ and ‘work intensification’ is the emphasis on:

> …the effort that employees put into their jobs during the time that they are working (Burchell, 2002: 72).

This effort encapsulates two areas, namely an extensive component, with regards to the time spent at work and an intensive component, with regards to the intensity of physical or mental input of effort into work (Fiksenbaum et al, 2010; Green, 2006). With regards to effort, ‘work intensity’ and ‘work intensification’ can also be seen to relate to two aspects of an individual’s workload: qualitative, being the difficulty and complexity of work and quantitative, referring to the amount of work an individual has to perform (Wichert, 2002). The focus on ‘work intensity’ as opposed to ‘work intensification’ in this research emphasises the acute but nevertheless important distinction between these terms and the resulting approach deriving from this research. Closer examination of existing literature (Green, 2006, 2004, 2001; Burchell, 2002; Gallie et al, 1998) suggests that temporal factors underpin the essential differences. Crucially, work intensity considers the experience and condition of work at a moment or stage in time, whilst work intensification takes into account the evolutionary nature of work over time. This conceptual dichotomisation prefigures and informs this study of professional software work.

Work intensity can also be viewed as an objective concept in that its implications and outcomes may be positive or negative, depending on interpretations, perceptions and responses. For example, work intensity can be experienced positively, through individuals experiencing greater levels of engagement and enjoyment with work (Burke et al, 2009). Work intensity can also be seen to positively encourage productivity growth in the economy:

> …it is frequently asserted that intensification, rather than increased efficiency, must account for the 1980s productivity growth in the manufacturing sector (Green, 2001: 64).

There is a tendency for outcomes for individuals to be perceived negatively in the work intensity literature, with terms such as ‘increased cognitive load’, ‘work overload’, ‘long working hours’, ‘repetitive working’ and psychological and physical effects, such as hypertension, stress, internal aggression, sustained alertness and poor well-being often accompanying the term (Boisard et al, 2008; Wichert, 2002; Green and McIntosh, 2001). Indeed, as summarised by Boisard et al (2008: 46):

> …the intensification of work makes it difficult, if not impossible, to institute practices that are essential to the preservation of health: changing position, taking a ‘breather’, organising one’s workplace, selecting the right tool or document, obtaining correct information, securing useful assistance at the right time, anticipating problems in order to avoid them and preparing for emergency situations.

Green (2004) also claims that human physical and mental capabilities may not allow for an infinite extension of effort. In this sense, it can be argued that the continuing intensification of work may be unsustainable, meaning that once there is an efficient match between work flows and individual readiness, efficiency may have to be found elsewhere (Green, 2006). For example, unmanageable levels of work intensity may potentially lead to individuals adopting less preferable ways of working to be productive, potentially endangering their health and safety:

> For a worker fitting a car seat, the postures that enable him to work quickly are not always those that avoid excessive pain in the back or elsewhere. In hospital, the intensification of work often leads assistant nurses to abandon the use of patient lifts, which take too long to install; in addition, their tight schedules focus their activities on purely technical actions, whereas handling the patient enables them to have contact with him (Villatte et al, 1993) (Boisard et al, 2008: 45).

Experiences of work intensity may also be influenced and mediated by contextual, institutional and personal factors. At the contextual level, work intensity may be affected by type of industry, whether an organisation is public or private, company size, business cycle stage, or the global business situation (Burchell and Fagan, 2004; Green, 2001). For example, Boisard et al (2008) suggest that the hospitality and transport sectors are industries which can be seen to experience high levels of work intensity, due to high working speeds and tight deadlines, respectively. At the level of the institution, experiences may be influenced by employment status, work organisation, management style, rewards, job content and employee disposition (Baldry et al, 2007). It is claimed that individuals with temporary or contracting status may experience greater levels of work intensity, due to increased job insecurity and time pressure (Boreham et al, 2008; Green, 2004). In addition, personal, social and environmental factors, such as occupational group, level, age, gender and personality may influence the resilience and vulnerability of individuals to work intensity (Wichert, 2002). Indeed, Boisard et al

(2008) suggest that work intensity may have an age-related dimension, in that younger, less experienced workers may perceive greater levels of work intensity, due to lack of familiarity in dealing with intense situations, compared to older, more experienced colleagues.

In this sense, consideration of firm characteristics, internal organisational factors and job-related characteristics (such as level, age, gender, permanent or contractor status) may assist the examination of software professionals' experiences of work intensity.

Importance of Research on Work Intensity

Existing research has attempted to study work intensity and work intensification through the utilisation of case study methods, productivity analyses, time series analyses, evidence of industrial accidents and comparisons of representative surveys (Burchell and Fagan, 2004; Green, 2001). For example, the 'Percentage Utilisation of Labour Index' aims to measure the changing intensity of human exertion per hour by employees, in order to provide an indication of whether employees are working harder (Bennett and Smith-Gavine, 1988). The SCELI survey of the 1980s and more recent surveys by Gallie et al (1998) have attempted to measure work effort over time to determine whether levels of work intensity have changed. Similarly, the European Working Conditions Surveys compared and contrasted the nature of working conditions in the European Union in 1991, 1996 and 2000, in order to ascertain whether work intensity has changed over the years. Taylor and Bain (2007) have also drawn attention to work intensity in the context of call centre work, with the infrequency of breaks, impact of targets and levels of monitoring, supervision and under-staffing identified as contributing factors. More recently, Fiksenbaum et al (2010) have focused on the Chinese hotel industry, suggesting that work intensity can be a consistent and significant predictor of work outcomes, such as work engagement and psychological well-being.

In this sense, the majority of studies on work intensity can be seen to have largely focused on the manufacturing sector (see Elger, 1990; Tomaney, 1990), service-related sectors (Fiksenbaum et al, 2010; Taylor and Bain, 2007) or work intensity over time in the economy as a whole (Green, 2006, 2004, 2001; Gallie et al, 1998; Penn et al, 1994), as opposed to emerging sectors such as professional software work. Existing research on professional software work has attempted to provide insight on these workers through areas such as teamworking (see Marks and Lockyer, 2004; Scarborough, 1999; Carmel and Bird, 1997); identity (Marks and Lockyer, 2005; Marks et al, 2002; Lockyer et al, 2001); skills and knowledge development (Lerouge et al, 2005; Marks and Lockyer, 2004; Scarborough, 1999; Sonnetag, 1995; Cheney et al, 1990); careers (Ituma and Simpson, 2006; Lee et al, 2002; Couger et al, 1979); class (Marks et al, 2003); union involvement (Hyman et al, 2004); and control (Barrett, 2001; Beirne et al, 1998; Causer and Jones, 1996; Barker, 1993; Kraft and Dubnoff, 1986). However, despite these contributions, there continues to be a lacuna in research on work intensity in professional software work, which this research has attempted to address.

Crucially, research on software professionals can provide valuable insight into the implications that trends such as globalisation, ICTs and changes in work organisation viewed in the knowledge economy may have for workers themselves. In addition, software professionals perceived status as archetypal knowledge workers and their role in creating a key technology at the centre of the economy means it is important to examine whether these individuals are necessarily immune from experiences of work intensity. Furthermore, research on work intensity can be seen to have more general relevance, through helping to explore contributing factors, changes in intensity and implications for individuals, labour markets, economic growth and wider society (see Burke, 2009, 2008; Green, 2008, 2004, 2001; Burchell and Fagan, 2004).

Section 2.2 illustrated that professional software work has radically altered since its inception, from early programming, which was an all-encompassing task, through to its current form, with the separation of intellectual work from less skilled areas, as a result of the adoption of electronic engineering principles in the 1950s. Whilst the development and changes in the software occupation are important considerations, a study into work intensification would entail a longitudinal research study, with the emphasis on how work organisation has changed and thus affected work experiences over the years. This research is primarily concerned with understanding and explaining how present firm characteristics, internal organisational factors and current forms of work organisation impact on workers, making it a study of work intensity. For the purposes of this study, respondent reflections on past and present experiences have enabled consideration of how work has changed and impacted on individuals. This research can therefore be seen to make an original contribution, through providing insight into work intensity in the software industry, its incidence and its impact on professional software workers.

The Potential for an Intense Work Process

The following section will consider trends which have emerged in the new economy and the potential for an intense work process. Most notably, globalisation, ICTs, teamworking, multi-skilling, flexibility and personal investment in work can be identified as having potential implications for software professionals' experiences of work intensity.

Sections 2.1 and 2.2 drew attention to the implications that globalisation and ICTs may have had for labour markets, work organisation and skills. Globalisation is claimed to have restructured ideas of time and space, through weakening institutional buffers between national economies, enabling greater flexibility in labour markets and changes in work organisation (O'Riain, 2006). ICTs can be seen to have enabled organisations to globalise and re-organise, allowing for fewer constraints with regards to space, organisation and local institutional arrangements (Reich, 1991). These trends are also argued to have contributed to the intensification of work:

> These interactions between exploitation and embedding *intensify* time with more rapid rates of change in technologies, product markets and organisational structures; increased turbulence, turnover and mobility in the labour market; the importance of deadlines in work organisation; and a revolution in experience of time and shortening of horizons (O'Riain, 2006: 512).

For example, the application and utilisation of ICTs in workplaces are claimed to have stimulated an increase in work effort (Bittman et al, 2009; Green, 2004, 2001; Green and McIntosh, 1998). New technologies, along with trends such as downsizing and de-layering, are perceived to have encouraged greater presence at work and intensity of work, due to concerns over potential job security (Boreham et al, 2008; Green, 2006, 2004; Bunting, 2004; Burchell, 2002; Beirne et al, 1998). Indeed, an increasing number of studies have noted an increase in work pressure in the United Kingdom, in terms of longer working hours and shorter lunch breaks (Hyman et al, 2003; Green, 2001). It is also claimed that new technologies may increase workloads and speed up activities (see Green, 2001; Edwards et al, 1998; Gallie et al, 1998; Sennett, 1998). For instance, scanners are perceived to have intensified the pace and repetitive nature of retail check-out work, through increasing the coordination of movements and creating the need for sustained alertness, in order to reduce check-out waiting times and meet production targets (Boisard et al, 2008). Computers and mobile technology are also claimed to have enabled faster delivery and greater flexibility amongst those working in jobs with high levels of discretion and autonomy (Green, 2006).

ICTs are also considered to have facilitated greater management control and monitoring of work, potentially impacting on the intensity of work (Boreham et al, 2008). For instance, ICTs may increase surveillance and monitoring, through enabling supervisors to monitor work visually or interpret data based on individual technological output (Baldry et al, 1998). Mobile phone technology, whilst arguably allowing individuals to coordinate and control schedules more flexibly (Haddon, 2004; Ling, 2004; Katz and

Aakhus, 2002), can also be seen to potentially increase intensity of work, through enabling continuous, mediated interactions with less 'dead time' (Bittman et al, 2009; Golden and Geisler, 2007; Felstead et al, 2005). For example:

> Perhaps the mobile phone, along with the computer, has increased management's ability to monitor the flow of work. After all, being in perpetual contact increases the ability of management to stay in touch with employees who are away from their desks for work-related reasons. Also, the ability of employees to stay in touch with office staff from remote locations and the increasing ability to make discretionary decisions may increase the engagement of employees, resulting in an increase in effort per hour of work. This kind of change in work organisation will be viewed by managers as increased productivity…workers are likely to experience more work, at an intense pace, under greater time pressure with more stress and heavier use of the mobile phone, as a single package (Bittman et al, 2009: 687).

In this sense, it is important to consider whether software professionals are necessarily immune from the effects of globalisation and ICTs and experiences of work intensity, due to their role in creating a key technology which has revolutionised and transformed work, business and society.

Globalisation and ICTs are also considered to have encouraged transformations in work organisation, such as increasing emphasis on de-layering, teamworking, multi-skilling and flexibility. Whilst these changes may be positioned as representing an efficient, 'smarter' way of structuring and organising work, these transformations may also encourage the intensification of work (Thompson and McHugh, 2002). Indeed:

> Under all this rhetoric about new-wave management, the most important trend appears to be people working harder. Pressures on the

> effort-bargain are, of course, a constant pressure of market relations. But the combination of increased competitive pressures for cost reduction on private and public sector organisations, with expanded means for reducing or recording 'idle time', are leading to substantial work intensification, whether through reductions in manning levels and job demarcation or other means (Elger, 1991; Nichols, 1991) (Warhurst and Thompson, 1998: 9).

For example, whilst teamworking is held to facilitate problem-solving, creativity, and co-operation amongst workers, it may also enable management to encourage the self-management of teams, with implications for the intensity of work (Thompson and McHugh, 2002). As emphasised by Sharpe (1998: 37):

> The ideology of 'empowerment' reflects the new management conviction that 'de-layered' or flattened organisations – organisations in which work teams monitor and intensify their own labour – can extract value more quickly than traditional systems of fragment and flog.

Critically, teamworking can be seen to intensify work, through expanding responsibilities, increasing performance pressures and encouraging individuals to work harder (Green, 2006, 2004, 2001; Findlay et al, 2000; Springer, 1999). Green (2001) demonstrates that teamworking structures may impact on work intensity, illustrating that whilst only 29% of employees in 1986 stated colleagues affected how hard they worked, this increased to 36% in 1992 and 57% in 1997. In this sense, whilst teamworking may appear to be beneficial, through encouraging collective effort, skills and knowledge development and self-management of tasks, increased surveillance, peer pressure and work overload emerge as negative work outcomes (Baldry et al, 1998). Crucially, the posited presence of heterogeneous, interdependent project team structures within professional software work enables exploration into whether teamworking has implications for software professionals' experiences of work intensity.

It is also claimed that emphasis on flexibility, multi-skilling and empowerment at work has stimulated the integration of tasks and expansion of workloads, marking a reduction in 'idle time' and an increase in work effort in order to increase productivity (Green, 2006, 2001; Thompson and McHugh, 2002; Tomaney, 1990). For example, Warhurst and Thompson (1998) suggest that flexibility and multi-tasking may intensify work for software professionals, in that fewer workers may be expected to perform the same quantity of work.

Individuals may also be encouraged to place greater levels of personal investment in work as a result of teamworking, flexibility and empowerment approaches, influencing levels and experiences of work intensity (Green, 2004; Thompson, 2003; Gallie et al, 1998). Ichniowski et al (1996) suggest that greater personal investment in work may have positive work outcomes, through encouraging worker engagement and intrinsic satisfaction with work. Indeed, in the case of software professionals, Baldry et al (2007: 140-141) suggest that flexibility and the acceptance of additional responsibilities at work may be viewed as part of the nature of professional software work by workers themselves:

> Here, as in the case of most software developers, concessions to organisational demands for flexibility and extra work are accepted as part of the job and point to different attitudes towards the elastic demands of the contemporary employment contract.

Professional software work can therefore be argued to have attributes which make it engaging and intrinsically satisfying for those who perform it, potentially influencing experiences of work (Baldry et al, 2007). In this sense, research on professional software workers can enable arguments centred around flexibility, multi-skilling and personal investment in work to be subjected to further scrutiny. Chapter Section 3.4 will expand on issues such as internal motivation to work and normative control to more fully explore the software labour process and its implications for work intensity.

CONCLUSION

This chapter has engaged with the key debates concerning the knowledge economy, software work and work intensity, in order to identify factors which may have the potential to influence software professionals' experiences of work intensity and help towards the generation of research questions. On the basis of observations within this chapter, professional software workers can be viewed as an archetypal group of knowledge workers. In addition, conventional wisdom on the knowledge economy suggests that software professionals are presumed not just to be knowledgeable, but also to be empowered, self-reliant, autonomous and intrinsically motivated by the nature of the work itself (Barrett, 2001; Alvesson, 2000; Tsoukas, 1996). Professional software work may also be subject to favourable forms of work organisation which characterise knowledge work occupations, such as facilitative management styles, supportive and collaborative project team structures, lower bureaucracy and autonomous working conditions (Donnelly, 2006; Newell et al, 2002; Thompson and McHugh, 2002; Spender, 1998).

In addition, this chapter has examined more critical literature which questions the optimistic pictures presented on the 'new' economy and knowledge work. Most notably, it has been argued that rationalisation, efficiency, managing costs, de-skilling and displacement of work in the 'new' economy may exist alongside flexibility, empowerment, up-skilling and the up-grading of work (Baldry et al, 2007; Kumar, 2005; Warhurst and Thompson, 1998). This chapter has also argued that whilst knowledge workers may experience more favourable terms and conditions and forms of work organisation, the notion of 'free' workers may be overstated. For example, teamworking structures may encourage greater self-management, review from within and peer control, potentially representing a new way of controlling workers (Beirne et al, 1998). In addition, the intrinsically motivating nature of professional software work may encourage individuals to adapt themselves to the cyclical nature of work, acting as a form of normative control (Baldry et al, 2007; Beirne et al, 1998). Crucially, the status of software professionals as archetypal knowledge workers enables consideration of the implications that transformations in working conditions and work organisations may have had for individuals, particularly in terms of work intensity.

Moreover, this chapter has illustrated the importance of considering the implications of developments in the software industry for the structure, design, organisation and management of professional software work. Crucially, the evolution in computing languages, the application of structured methodologies, emphasis on interactions with clients, increased flexibility in choice and location of labour and trends towards the outsourcing and offshoring of work (OECD Information Technology Outlook, 2006; Arnold, 2005; McGrath-Champ, 2005; Alvesson, 2004; Marks et al, 2001; Baetjer, 1998) can be recognised as factors which may have implications for software professionals' experiences of work intensity. Furthermore, the role of software professionals as the creators of a key technology within the economy suggests that research on these individuals is of great significance.

Software has been identified as being significant and increasingly important due to its widespread utilisation throughout the economy, its applications to virtually all work situations and to many types of human, social and relational activities (Freeman and Perez, 1998). Professional software workers engaged in the design, development, testing and installation of software can therefore be considered to be central to debates relating to the utilisation and implications of technology in the new economy, in that:

> They are both the creations and the agents of the most spectacular technology yet, which in a generation has launched a transformation of the entire production process (Kraft, 1979: 2).

In addition, the continued growth of the software industry, its employment and the implications for the future of work more generally suggests that more detailed focus on this occupation is of considerable importance (Barrett, 2005; Marks et al, 2003; Beirne et al, 1998).

This chapter has also presented the case for research on work intensity and demonstrated the importance of examining the experiences of professional software workers. Trends identified in the knowledge economy, such as globalisation, ICTs and new forms of work organisation can be seen to have had implications for experiences of work intensity, through speeding up activities, expanding workloads and creating performance pressures for individuals. Research on software professionals, who can be categorised as an archetypal group of knowledge workers, can therefore provide valuable insight into the implications that these trends may have had experiences of work intensity in the new economy. Furthermore, it has been emphasised that addressing the research lacuna of work intensity within professional software work is paramount, in order to establish whether software professionals themselves are immune from work intensity, due to their role in creating a key technology within the economy. This chapter has also demonstrated the importance of considering firm characteristics, internal organisational factors and job-related characteristics within this research, in order to fully examine factors which may influence software professionals’ experiences of work intensity.

In summary, this study will therefore illuminate the experience of a key group of prolific knowledge workers and subject to broader review the conventional wisdom surrounding professional software workers, developments within the industry and some of the key generalisations surrounding the knowledge economy.

CHAPTER 3 SOFTWARE WORK

INTRODUCTION

Chapter Two provided an in-depth review of the key aspects surrounding the knowledge economy, the software industry and work intensity. Chapter Three will continue the discussion by engaging with the four streams of literature which inform our understanding of professional software work and help to identify aspects which may be of importance to this study.

Section 3.1 concentrates on the collaborative aspects of professional software work, maintaining that interactions with project team members, project managers and clients are necessary to improve levels of understanding and bring various types of knowledge together (Alvesson, 2004; Newell et al, 2002; Baetjer, 1998). Project team structures are recognised as being an important feature of professional software work, through aiding the effective integration of work activities and providing the opportunity for knowledge sharing and acquisition, due to the presence of heterogeneous skills and knowledge (Marks and Lockyer, 2004; Tam et al, 2002; Baetjer, 1998; Walz et al, 1993). In addition, user interaction is identified as an emerging feature of professional software work, which arguably enables software professionals to achieve a greater understanding of user requirements and improve the effectiveness of systems.

Section 3.2 illustrates that software professionals may require a range of technical, interpersonal, business-related and political skills to perform their work effectively. It is also claimed that trends such as globalisation, downsizing and de-layering in the 'new' economy may have placed greater emphasis on individuals being responsible for their own careers and general employability (Ituma and Simpson, 2006; Arnold, 2005; Baruch, 2004; Cappelli, 1999; Cohen and Mallon, 1999; Rousseau, 1995).

Section 3.3 suggests that particular organisational and interactional dynamics within professional software work may have implications for individuals. It is claimed that deadlines may enable organisations to meet market requirements, release products quickly and regulate individual work effort (Andrews et al, 2006; O'Riain, 2006; Kunda, 1992). In addition, this section suggests that the interplay between markets, organisations, employers, clients and internal organisational processes may have implications for software professionals' management of working time (O'Riain, 2010; O'Carroll, 2008; Baldry et al, 2007; Sharone, 2004; Perlow, 1997). Interruptions resulting from office layout, technological tools, client unpredictability and interactions are argued to potentially have further consequences for work rhythms, schedules and the management of working time (O'Carroll, 2008; Boisard et al, 2008; Voss-Dahm, 2005; Perlow, 1997; Kunda, 1992). Moreover, it is argued that changes in work location and the physical proximity of project team members may affect communication and the quality of interactions within software project teams.

Section 3.4 focuses on the software labour process and implications for software professionals' experiences of work intensity. It is argued that labour process theory can provide insight into dimensions of the professional software labour process, such as objectivity of work organisation (Thompson, 1989; Burawoy, 1978; Braverman, 1974), subjectivity, worker agency and consideration of consent and accommodation (Storey, 1985; Edwards, 1979; Burawoy, 1978; Friedman, 1977) and the implications that these aspects may have for work intensity. Furthermore, the utilisation of a labour process approach within this study can be seen to facilitate understanding of the contradictions and complexities of control, power and resistance claimed to be present in knowledge-intensive occupations (Sewell, 2005). It is argued that labour process elements, such as the development life cycle, project team structures, autonomy, normative control and cultural control, potentially represent attempts to formalise, monitor and control professional software work. However, it is also posited that software professionals can exert power over the labour process, due to the intangible and intellectual nature of work. In this sense, it is argued that labour process analysis can provide valuable insight into the extent to which the structure, design, organisation, management and control of professional software work influences experiences of work intensity.

Finally, this chapter concludes by summarising the main themes raised in Chapters Two and Three which may have implications for software professionals' experiences of work intensity and presents the five research questions guiding this study.

3.1 COLLABORATIVE ASPECTS

Interactions between team members, managers and users are considered to be important in knowledge-intensive areas such as professional software work, as these workers are often dealing with complex ideas and tasks which need to be communicated for improving knowledge bases, understanding and levels of information (Alvesson, 2004; Newell et al, 2002; Baetjer, 1998). As illustrated by Baetjer (1998: 147-148):

> The human knowledge that must be brought together and embodied in a new capital good is dispersed, incomplete, changing and largely tacit. It is *dispersed* in that many different people possess different bits of knowledge that the new capital goods must incorporate. It is *incomplete* in that often we have not figured out all the knowledge we will need to build a product whose general nature and purpose we have in mind. It is *changing* constantly – that we are always learning; those who need the product are learning better what they need, what is possible and how the conditions in which they will use it are changing; those who are trying to build it are coming to understand better what their customers (think they) need, what design elements will be most suitable, and what tools they themselves might use in crafting the design.

The following sections will therefore consider the nature of project teams and user interactions as part of the professional software work process.

Project Teams

Whilst methods, such as project websites, e-mails, diary entries and meetings can facilitate the learning process, it is argued that bringing together individuals in project team structures is one of the most important ways of encouraging the acquisition and development of new skills and knowledge (Marks and Lockyer, 2004; Swart and Kinnie, 2003). Indeed, it is claimed that the dynamic nature of professional software work, in terms of changing project requirements, specifications and need for information sharing between team members, necessitates a collaborative team-based approach to work.

Whilst the traditional idea of teams typically encompasses the bringing together of complementary skills, Marks and Lockyer (2004) point to the distinctiveness of software project teams, typified by heterogeneous skills and knowledge and differing individual responsibilities, degrees of specialisation and aspirations. Project teams are considered to provide the opportunity for knowledge sharing and acquisition, as well as allowing for the effective integration of work activities (Tam et al, 2002; Baetjer, 1998; Walz et al, 1993). In this sense, project team structures are suggested to give rise to interdependencies between team members in order to supplement expertise and solve complex problems. Project team structures are also suggested to operate within flexible organisational arrangements which support the interdependency of roles and require individuals to manage a variety of responsibilities (Baldry et al, 2007). For example, Kraft (1977: 72) points to the interdependent and mentoring nature of software project team work:

> Programmers as a matter of course act co-operatively. That is, when one confronts a problem or a difficult task, he or she will walk over to another programmer for help. The help, in turn, is routinely given.

> Thus, a sort of master/apprentice heritage of the programmer as an artisan persists even among the least skilled of the occupation.

The collaborative nature of project team interactions is also argued to potentially reduce the reliance of individuals on organisations for skills and knowledge development. As emphasised by Baldry et al (2007: 118):

> The high degree of collaboration and inter-disciplinarity suggests a more collective construction of skills development. Given that software workers cannot, and do not, rely exclusively on the employing organisation to attend to their development needs (Foote, 2000), the role of teamworking assumes particular significance. Not only would software employees tend to be integrated into inter-disciplinary teams but, in the interests of career development, they might seek to move between teams and projects.

The continuous updating of skills and knowledge is therefore also perceived to be important to achieve access to other projects in the future (Sonnetag, 1995). Indeed, the ability to rotate between project teams is suggested to allow software professionals to continuously experience challenging, intrinsically motivating work and broaden their technical expertise. For example, Sonnetag (1995) states that software professionals considered excellent by others in their field often have greater variability in terms of project involvement, social networks and contacts with other teams.

However, whilst project teams may play an important role in the development of skills and knowledge, there may be limited opportunities for movement into new project areas. For example, Baldry et al (2007: 73) suggests that project team leaders may possess power in terms of determining the nature of moves to other projects or areas:

> Team leaders, particularly those in charge of more prestigious projects, were powerfully placed to determine who should be included in these programmes. For many employees these relationships impacted on their work opportunities, with the outcome of team leaders' decisions paralleling and reinforcing the major division in software work; between those who worked on 'cutting edge' projects and those involved in more routinised lower level work.

In this sense, there may be potential for conflict between an individual's desire to acquire valuable skills and knowledge and move to different positions and the organisation's concern for them to remain in a specific area of work, with a view to developing a particular expertise or specialism (Marks and Lockyer, 2004). For example, Marks and Lockyer (2004) suggest that older software professionals may feel their careers are focused in a particular specialist area, compared to new entrants who may slot into new niches utilising new technology. Thus, whilst it may be necessary for individuals to gain new skills in order to gain access to new projects, Fincham et al (1994) suggest an emphasis on expertise and specialist niches can commit software professionals to a particular area, creating barriers to accessing new areas.

Furthermore, whilst the development of individual expertise and transferability of knowledge within project teams can potentially improve the value of software companies, companies may be at risk of enhancing an individual's reputation in the wider software community and increasing their potential labour market mobility (Glaser, 1976). Indeed, it is argued that employers may be highly dependent on employees for their intangible knowledge, ideas, creativity, professional expertise, skills and networks as sources of competitive advantage, meaning that software professionals may possess considerable labour market power (Andrews et al, 2005; Scarborough, 1999). For example, more experienced software professionals may have greater bargaining power, as they may possess greater external market value and have established valuable networks and relations with clients (Donnelly, 2006). As emphasised by Donnelly (2006: 81):

> …the development of new areas of esoteric expertise – which are often highly valued in the external marketplace – shifts power to the knowledge worker – as they become less dependent upon their immediate employer for employment, as demand for their services also arises from their clients; reflected by the rising bargaining power and pay levels of knowledge workers.

It is also claimed there may be clashes between collective work processes and the need for individuals to balance their team and project requirements with their individual development aspirations (Marks and Lockyer, 2004; Marks et al, 2002). In this sense, whilst the utilisation of team collaboration and rotation can enable software professionals to broaden their skills and knowledge base (Marks and Lockyer, 2004), there may be the potential for conflict between individual and employer aspirations, resulting in limited opportunities. As a consequence, an organisation's inability to address individual needs may result in a higher turnover of staff leaving to satisfy their aspirations (Amakawe et al, 2000). The effects of team members leaving have been documented by Appelbaum et al (2004), highlighting that specialist skills and knowledge may be lost, creating major skill gaps which may affect remaining team members in terms of increasing workloads, limiting resources and meeting impossible deadlines.

Users

An emerging feature of professional software work is that user contact is no longer solely the private domain of managers, with non-managerial software professionals' job requirements expanding to include user interactions. During the 1960s, interaction between users and computer specialists was sporadic, with systems often being sold based on technical opposed to operational specifications, making it difficult to understand user needs and provide systems fitting user requirements (Friedman and Cornford, 1989). Software reliability, quality, costs and deadlines were also concerns relating to the effectiveness of systems. As a result of these issues, user participation and involvement began to change in the 1970s, from a one-way developer to user approach to a two-way process, placing more emphasis on user-oriented systems and interactions (Friedman and Cornford, 1989; Boland, 1978). However, whilst software professionals may become more involved with users, it is important to note that this does not always lead to a move into managerial or project management grades, as this may not always be preferential for those emphasising technically-oriented careers (Marks et al, 2002).

User interactions are therefore perceived to be important, through helping software professionals to define problems, make clarifications, discuss solutions and develop an understanding of what the client is looking for (Alvesson, 2004). Indeed, achieving understanding of the user's requirements and overall vision throughout the process is argued to be especially important for the effectiveness and success of software systems, in that:

> The tacitness and incompleteness of the user's knowledge of what they need are the main reasons for the failure of traditional methodologies in modern software development. Software requirements cannot be articulated completely in the first stages of development because the necessary knowledge is incomplete and because much of it is inarticulate. Only through interacting with the developing tool do users discover and communicate to designers what they need (Baetjer, 1998: 62).

Furthermore, the tacit and intangible nature of professional software work may mean that face-to-face interactions between clients and software professionals are necessary to define reference points and understandings:

> For example, in a situation where a group of IT specialists is designing a software program for a group of business analysts, there may be little common understanding of contexts between the two groups. Moreover, much of the knowledge on both sides is likely to be tacit and difficult to articulate. In this situation, considerable face-to-face contact will be necessary before each side understands the issues confronting the other and so can develop a situation that is feasible and useful. This will involve what has been termed 'perspective taking', such that one group can begin to appreciate the world-view and context that underpins the knowledge and experience of the other (Boland and Tenkasi, 1995) (Newell et al, 2002: 177).

Users can be categorised into four main groups, including patrons or clients, who are the overseers and initiators of the system; end-users, those who use the system when it is in operation; secondary users, whose work may be displaced, automated, de-skilled, up-skilled or affected by the system; and other interactors in the systems development process (Friedman and Cornford, 1989). In addition, as will be discussed in later chapters, the various stages encompassed in software development (Analysis and Requirements, Design, Development, Testing, Installation) may involve varying levels of client interaction, depending on resources, technical knowledge and the nature of project initiation. For example, users may have little involvement due to lack of technical knowledge or preference, regulated involvement according to knowledge, power and resources or be involved throughout the process, with full co-operation and involvement between users and software professionals (Beirne and Ramsay, 1988).

3.2 THE NATURE OF SKILLS AND CAREERS

Skills

Professional software work is considered to be evolutionary in nature, as a result of changing technology and circumstances, which impact on skill requirements and create the potential for gaps between job requirements and existing skill sets (Lerouge et al, 2005). For example, transformations in the era of dot.com companies and changes in organisational structures are claimed to have led to a change in emphasis in leadership styles and skill sets (Edward and Wacjman, 2005; Woodfield, 2000).

Analysis of existing research suggests that software professionals require a range of technical, interpersonal and political skills to perform work effectively, opposed to focusing solely on one skill area (Sonnetag, 1995; Curtis et al, 1988). It is claimed that the skill sets utilised by software professionals cover five main areas: systems development task skills, relating to the ability to analyse, design, develop, implement and maintain systems; technological skills; interpersonal skills, especially those required for team and client interactions; political skills, with regards to the ability to mediate and negotiate; and business knowledge, which can help facilitate understanding of business problems (Lerouge et al, 2005). Curtis et al (1988) suggest that four main characteristics can be identified in excellent software professionals, including technical knowledge; project identification; familiarity with application domains and the ability to integrate this with general computer knowledge; and interpersonal and political skills, in order to communicate and manage demands effectively with team members and clients.

Within existing literature, there is some disagreement between researchers emphasising the greater importance of interpersonal and political skills (see Trauth et al, 1993; Khan and Kukalis, 1990) and those identifying technical skills as being of crucial importance (Cappel, 2001; Cheney and Lyons, 1980). Indeed, the emphasis placed on technical or interpersonal skills may differ according to the group surveyed. For example, Sonnetag

(1995) suggests that clients may emphasise business knowledge, communication, and user orientation more than technical skills, as they may rely on clear explanations to understand concepts or may lack the technical background to evaluate software professionals' technical skills. Beirne et al (1998) also suggest that organisational and commercial knowledge and the ability to network with users, coupled with technical expertise, may be necessary for software professionals to be considered effective. At the same time, Sonnetag (1995) states that those software professionals deemed excellent by co-workers may place more importance on technical knowledge, good working style in terms of structuring problems, the adoption of an individual approach whilst operating to team standards, as well as interpersonal skills. In this sense, whilst software professionals may be required to continue developing and maintaining technical knowledge, they may also be increasingly responsible for enhancing organisational and business development skills, in order to meet the requirements of both employers and clients (Curtis et al, 1988).

Careers

Increasingly, factors such as globalisation, increasing competition, new technology, downsizing and de-layering are argued to be impacting on the nature of careers and contracts. It is claimed that in the new economy, traditional careers and relational contracts emphasising hierarchy, status, long-term employment and promotion from within have given way to self-managed, 'boundaryless' careers and transactional contracts. Under the new arrangements, careers are perceived to transcend organisational boundaries and be flexible in nature, whilst the employment relationship involves employees accepting more responsibility and potentially working longer hours, in return for pay, rewards and access to training and development opportunities to secure employability (Ituma and Simpson, 2006; Arnold, 2005; Baruch, 2004; Cappelli, 1999; Cohen and Mallon, 1999; Rousseau, 1995). In this sense, the impetus for managing and developing careers may rest with the individual software professional (Arnold, 2005).

Software professionals are argued to have a unique career situation, in that they need to develop skills and knowledge continuously to ensure these remain up-to-date, due to the expansion and rapidly changing nature of technology in the software industry (Couger et al, 1992). In order to manage careers and employability, it is claimed that individuals may require various career management skills, such as understanding what skills, competencies and knowledge should be developed, how to do this effectively, where to go for this development and how to coordinate training and development with other activities (Baruch, 2004; Jones and DeFillippi, 1996; Arthur et al, 1995). As emphasised earlier, contextual factors, such as economic conditions, globalisation, societal context, labour market conditions and new technology can be seen to play an important role in dictating individual career patterns (Ituma and Simpson, 2006). For example, Ituma and Simpson (2006) suggest that software professionals may utilise different career patterns depending on contextual factors, either emphasising traditional careers due to the need for security and stability, mobility between projects and organisations to increase wages, skills and experience, self-employment, or utilise elements of these, in order to accommodate personal interests or family obligations. Concurrently, the career paths, orientation and choices made by software professionals may also be influenced by workplace conditions, informal structures and organisational approaches, as well as personal preferences and levels of self-awareness (Baldry et al, 2007).

In terms of the nature of careers pursued, professional software workers can potentially follow two main career paths, either focusing on technical areas or moving into managerial roles, with some elements of overlap between the two areas. For example, software professionals pursuing technical paths may have purely technical roles or also assume some supervisory and management tasks in addition to their technical responsibilities (Kraft, 1977). On the other hand, software professionals may choose to move into project management roles, concentrating mainly on managerial duties or also including some elements of technical expertise. Indeed, those working as project managers may have had initial backgrounds and training in professional software work before moving into more managerial areas (Kraft, 1977). Chapter Four will explore categories of professional software work in greater depth, in order to give a more detailed understanding of software job roles and tasks and activities performed by individuals.

It is also claimed that women may be more concentrated in particular types of roles, such as systems analysis, design, project management or support, with men dominating higher end functions (Adam, 2002; Marks et al, 2001; DiDio, 1997; Baroudi and Igbaria, 1995). Guerrier et al (2008) suggests it is unclear whether women are concentrated in particular roles due to personal choice or if they are channelled into these by organisations due to existing structures. On the one hand, it is suggested that software firms may exhibit masculine-oriented traits, such as long working hours and intense working conditions which are embedded in organisational structures and daily practices, which may affect the participation of females in particular roles (Kerfoot and Knights, 1997; Witz and Savage, 1992). However, at the same time, Hakim (2000) suggests that the preferences and choices made by females could also be seen as the main source of inequalities with regards to employment and opportunities. Indeed, factors such as career breaks, the combination of child rearing or caring responsibilities with careers and the uptake of teleworking or homeworking are considered to have potentially impacted on the development of female technical skills, role choices and levels of involvement in the profession (Guerrier et al, 2008; Marks et al, 2001; Baroudi and Igbaria, 1995).

3.3 ORGANISATIONAL AND INTERACTIONAL DYNAMICS

The following section will discuss organisational and interactional aspects of professional software work, such as deadlines, working time, interruptions and work location, which may have implications for software professionals' experiences of work intensity.

Deadlines

Pressures such as the speed of technological change, globalisation and competition, coupled with the availability of capital and financial concerns, may compel companies to develop and release software more quickly to the marketplace (Andrews et al, 2006; O'Riain, 2006). The need to meet market requirements and release products quickly therefore means that aggressive scheduling and time constraints can be seen to be important aspects of knowledge work (Kunda, 1992). Indeed, Perlow (1997) contends that pressure to release products quickly, coupled with insufficient planning, may encourage a 'vicious time cycle' which has adverse consequences for software professionals. Notably, knowledge workers may have to contend with heavy workloads, scheduling pressures and tight deadlines, with implications for the management of working time (Kunda, 1992). In this sense, deadlines can be recognised as a key organisational mechanism within professional software work which forms the focus for management and team effort:

> The deadline is the mechanism by which management brings the time pressures of product markets into the heart of the team (O'Riain, 2010: 341).

Crucially, deadlines can be seen to enable the regulation of work effort within a work process which is otherwise intangible, ambiguous, autonomous and creative in nature (O'Riain, 2006, 2000; Rasmussen and Johansen, 2005; Barrett, 2004). For example, Sharone (2004) suggests that software professionals may naturally increase their expenditure of work effort and working hours during the final stages of the development life cycle, in order to meet impending deadlines. In addition, deadlines may facilitate the application of punitive and incentivising systems within professional software work, such as those which deny individuals' access to prestigious projects when targets are not met and those which provide material rewards and new opportunities for those who succeed (Baldry et al, 2007).

The immediacy of deadlines may also have implications for the internal dynamics within software project teams. For instance, O'Riain (2010) suggests that project teams may become more introverted when approaching deadlines, in order to assist communication, co-operation and the overall work process. However, individuals may become more extroverted after fulfilling deadlines as they consider their next direction and future opportunities. Notably, changes in the internal dynamics of project teams may have powerful implications for organisations, in that introversion may affect cross-team co-operation and extroversion may contribute to turnover and loss of firm-specific knowledge (O'Riain, 2000).

However, whilst deadlines may be immovable, software professionals may be able to utilise various strategies to ensure they meet deadlines and protect themselves from adverse outcomes. For example, software professionals may choose to screen information from project managers located elsewhere, enabling them to manage their work time more effectively:

> Space became an asset in handling the politics of time. Having a manager on the other side of the world allows the team, including the team leader, to screen information from Ramesh [project manager] in order to let the team balance the technical and time demands to their own satisfaction (O'Riain, 2006: 520).

In addition, it could be argued that the approach that software professionals take to planning work may influence the extent to which deadlines impact on experiences of work intensity. For instance, Perlow (1997: 90) argues that working reactively, rather than pro-actively, may have a detrimental effect on the ability of software professionals to complete work to deadlines:

> If each task did not become a crisis, the engineers would have more time to solve problems as they arose. Instead, because everything is

> left until the last minute, routine problems often become crises, and tasks take longer than they otherwise would.

Furthermore, it is useful to consider whether organisational and personal characteristics may influence how individuals respond to deadlines. Certainly, Boisard et al (2008) suggest that whilst software professionals may be subject to tight deadlines or have insufficient time in which to complete work, experiences may vary depending on sector, company size, occupation, status, gender and age.

Working Time

Working time is arguably an important aspect within professional software work and may be influenced by the interplay between the market, organisations, employers, clients and individual self-management strategies (O'Riain, 2010). Crucially, O'Carroll (2008) argues that the compression of working time and the need to fit tasks within delineated time frames may intensify work for individuals based within knowledge-intensive occupations. In particular, 'fuzzy holes' (spaces in the working day which occur through movement between tasks or breaks from work) may become compressed when time is scarce, whilst 'intangible time' (activities such as information gathering, thinking, processing ideas and concepts) may increase the length of the working day (O'Carroll, 2008). Indeed, whilst software professionals may have discretion over the scheduling of breaks, breaks may become shorter or be taken less frequently when deadlines are looming (Baldry et al, 2007; Perlow, 1997). In this sense, individual strategies for organising personal work time may be influenced by corporate time frames and set deadlines:

> …these strategies are applied within the shadow of the clock; the time-frame within which these tasks are to be completed is determined by deadlines set by others. Corporate time frames, which equate speed

with efficiency, are very different from the individual time-frames of the work process (O'Carroll, 2008: 188).

Internal organisational processes, such as performance review schemes, may further influence how individuals manage their working time and expenditure of work effort. For example, Sharone (2004) stresses that competitive, self-managed performance grading structures may engender intense anxiety amongst software professionals over professional competence, encouraging individuals to self-impose long working hours or increase expenditure of work effort. Furthermore, clients may have implications for working time within the software work process. For instance, client indecision, unanticipated crises and customer use of systems may introduce elements of unpredictability to the software labour process (O'Riain, 2010; O'Carroll, 2008). Moreover, software professionals based on client sites may be subject to greater monitoring and attention to working time by clients (O'Riain, 2010).

Barley and Kunda (2004) suggest that software professionals possessing contractor status may potentially be able to avoid temporal constraints if they are able to exert greater control over work rhythms and choose which hours or days they wish to work with clients. However, personal responsibility for deciding how and when to work may mean that contractors have to be particularly conscious of how they manage their working time (Barley and Kunda, 2004). For example, contractors may need to invest time working on various contracts, secure future work, develop new skills, carry out administrative tasks and maintain networks (Osnowitz, 2010; Barley and Kunda, 2004).

Interruptions

Interruptions are considered to be a normal part of the work process within knowledge work and may be caused by office layout, technological tools, client unpredictability and interactions (Boisard et al, 2008; O'Carroll, 2008; Voss-Dahm, 2005; Perlow, 1997;

Kunda, 1992). For instance, open plan offices may subject knowledge workers to noise and distractions, creating interruptions during the working day:

> On a typical day, engineers may be seen in their cubicle, attached to the ubiquitous terminal, often with ear plugs to keep out the unending background noise and prevent interruptions that are inevitable in a space designed for openness and communication (Kunda, 1992: 47).

Individuals utilising technological tools such as e-mails may equally experience interruptions to work rhythms due to ambiguity over e-mail content and difficulty in accurately predicting the time it may take to read and complete messages (O'Carroll, 2008). In addition, the interactive, collaborative and interdependent nature of professional software work may subject software professionals to spontaneous interruptions from team members who need to discuss issues, solve problems or request help, in order to complete work within specific time periods (Perlow, 1997). Clients may also interrupt or delay the software work process by requesting last-minute changes in requirements or omitting valuable information (Voss-Dahm, 2005). Indeed, the likelihood of projects adhering to schedules may decrease if clients frequently disrupt the software work process (Voss-Dahm, 2005).

Crucially, it is important to consider the implications that interruptions created by technological mediums, clients and colleagues may have for work rhythms, schedules and the management of working time within professional software work. Perlow (1997) has documented the consequences that interruptions may have for the software work process and outlined measures which could help interactions to be managed more effectively. Notably, whilst software professionals arguably require extended periods of uninterrupted time to focus on work tasks, the unpredictable and spontaneous nature of interruptions may affect thought processes, disrupt work rhythms and fragment schedules. Consequently, Perlow (1997: 116) suggests that managing interactions according to priority or scheduling time blocks for ‘quiet time’ may help to reduce the disruptive and unpredictable nature of interruptions:

> When interruptions are scattered throughout the day they are perceived as interruptions, but if they could be given a set time of their own, their value might be recognised. In other words, the attempt to provide individuals with quiet time to get their own work done was an interest in creating alternative times for those interactions essential to the work process.

However, it is important to note that individuals who are willing and supportive may experience interruptions to a greater extent, in comparison to those who control their availability and thus may be able to prioritise their working time more effectively (Perlow, 1997). In addition, personal approach to managing working time may influence the extent to which software professionals interrupt colleagues spontaneously or engage in discussions during agreed time slots. For example, individuals may be more inclined to interrupt others spontaneously if they are crisis-driven and reactive, rather than pro-active, when planning their own working time (Perlow, 1997). It may also be useful to consider whether organisational practices and management styles influence the approach individuals take to managing interactions with team members. For example, organisational systems which place emphasis on the individual over the collective, coupled with reactive management strategies, may encourage software professionals to be self-invested and competitive, to the detriment of their colleagues:

> Individuals trapped in this cycle do “whatever it takes” to get the job done because that is the appropriate approach that promotes their individual success. Yet their very attempts to succeed at work perpetuate a way of working which is disruptive and reinforces the crisis mentality, requiring individuals to put extraordinary amounts of time into their work. In the end, because each individual is concerned

> with his or her own success, everyone must work harder than necessary and neither the work nor individuals' lives outside of work benefit (Perlow, 1997: 96-97).

In this sense, it could be concluded that organisational emphasis on collective goals, teamwork and co-operation may encourage software professionals to manage their working time and interactions more effectively, helping to minimise disruption to team members (Perlow, 1997).

Location and Physical Proximity

Globalisation and ICTs have arguably enabled workplaces in the new economy to overcome the constraints of time, space and local institutional arrangements (O'Riain, 2006, 2000). For example, companies may have more opportunities to sub-contract work, take advantage of cheaper labour in different time zones and externalise activities which are deemed less important to central operations (Baldry et al, 2007; Aneesh, 2006; O'Riain, 2006; Castree et al, 2004). Developments in ICTs can also be seen to have increased interconnectedness and enabled organisations to transcend horizontal, vertical and organisational boundaries (Boreham et al, 2008; Barley and Kunda, 2004). Crucially, globalisation and developments in ICTs have arguably encouraged changes in workplace organisation, through enabling the geographical dispersion of project teams, the extension of working hours and the re-definition of employment practices, in order to satisfy project needs (Osnowitz, 2010). In addition, the ability to secure workers across a variety of different time zones can be seen to have provided the opportunity for 24/7 working:

> The global twenty-four hour office was always the hidden possibility and agenda of all programs of globalisation. Now it is a reality with which practices of business and labour management must contend. From the perspectives of corporate governance, the new arrangement

> allows organisations in two time zones to be sequentially patterned for competitive advantage – this is called the follow-the-sun approach (Aneesh, 2006: 84).

Software professionals may therefore be subject to a variety of locational configurations, with conventional on-site locations, project team members being distributed across different sites or individuals working from home. Consequently, software professionals may interact face-to-face in the workplace through weekly team meetings, at workstations, in kitchens, in restrooms or through the use of technological mediums, such as phone calls, e-mails and video conferencing (Osnowitz, 2010; Baldry et al, 2007). In this sense, it is important to consider the implications that changes in work location and physical proximity may have for interactions and communication within software project teams (O'Riain, 2006, 2000; Benner, 2002).

Notably, the re-configuration of time and space may have increased the ability of capitalists to dominate workers at local, regional and transnational scales, raised competitiveness, shortened work times and intensified working arrangements, with potential implications for software professionals' experiences of work intensity:

> …this dynamism comes at a cost – organisations seesaw between isolation from the world around them and internal fragmentation while employees experience pressure and burnout at work and increased volatility in the labour market, even those who benefit financially from the global workplace (O'Riain, 2000: 175).

In addition, despite suggestions that the ability of individuals to communicate via technological mediums may have decreased the relevance of physical space, locality may continue to be of importance in an increasingly global economy (Aneesh, 2006; Castree et al, 2004; O'Riain, 2000). In particular, the ability to communicate via technological means may not compensate for the benefits to be secured from close face-to-face interactions within professional software work. For instance, the use of flexible working practices (such as homeworking and teleworking) within professional software work may remove individuals from the pace and rhythm of the workplace and prevent informal interactions which may be important to the work process (Boreham et al, 2008; Baldry et al, 2007). Indeed, close physical proximity may enable software professionals to engage in informal discussions, promptly handle issues, manage interdependencies and connect with team members at a level which may not be possible through virtual means:

> Being interactionally present does not mean that one can discuss things over a cup of coffee with colleagues across the globe or share a hearty laugh with the team on the other side of the video screen; audio visual links fail to carry jokes to the other side (Aneesh, 2006: 81).

Spatial embeddedness may also encourage solidarity, cohesion, social support and shared meanings, which are important for the effective functioning of software project teams (Lockyer et al, 2001; O'Riain, 2000). Certainly, parallels can potentially be drawn between geographically dispersed project teams and 'occupational communities', which may possess particular sub-cultural knowledge and internal norms, creating difficulties in creating common understandings (Bechky, 2003).

Moreover, close physical proximity and locality may enable individuals to share information, interact and communicate more promptly, enabling them to respond to changing conditions in the economy more effectively (O'Riain, 2006, 2000). Indeed, local spaces may possess specific territorial and cultural capacities which help to buffer individuals from the impacts of globalisation (Castree et al, 2004; O'Riain, 2000). In contrast, reliance on technological mediums for communicating with team members located elsewhere may contribute to misunderstandings, political issues and potentially affect the development of team cultures. Geographical dispersion may additionally affect the ability of individuals to embed informal working practices at off-site locations or effectively share knowledge which is tacit, informal and specialised in nature:

> …practices that depend to a great extent on co-location, such as tacit understandings about how work is embedded in clearly specified corporate goals, shared organisational vision and relevant normative orientations, pose a substantial challenge to the ‘organisation’ of distributed work (Boreham et al, 2008: 121).

Furthermore, O’Riain (2000) argues that geographical dispersion may enable software professionals to ignore or screen problems from project managers or other team members located elsewhere. Crucially, whilst these actions may be deemed necessary to minimise unnecessary interference and the allocation of additional work tasks when nearing deadlines, problems may be overlooked, potentially creating difficulties at a later date:

> Problems which would require a solution in a face-to-face context can be swept under the carpet or become a figure of fun in a context where communication is by phone and the Internet (O’Riain, 2000: 179).

It is also important to note that whilst face-to-face interactions do not necessarily guarantee that team members will communicate effectively, this process may be more complicated when individuals are distributed across various time zones (O’Riain, 2000). Notably, differences in time zones may hinder communications, the ability to promptly resolve problems and individual’ capacity to effectively organise day-to-day work activities:

> The natural order of things comes into conflict with the emerging real-time regime of transnational integration. The temporal dissonance is a contentious issue when setting deadlines or phone or video meetings (Aneesh, 2006: 91).

In this sense, it is important to consider how project teams which are distributed globally can be integrated and coordinated effectively (Aneesh, 2006). Huws and Flecker (2004) suggest that in order for remote working structures to work effectively, it is important to establish clear communication patterns, ensure mutual cultural understandings, instil quality control standards and explicitly define working procedures. In addition, it may be important to assess the quality of technological links to ensure that temporal delays do not interfere with the integration of work activities on a global scale (Aneesh, 2006). Osnowitz (2010) has also drawn attention to the importance of telecommuters devising effective lines of communication with colleagues and managers to prevent their exclusion from internal processes and decisions affecting their performance. Indeed, combining processes, such as phone calls, e-mails and periodically travelling to other sites may be necessary to prevent unnecessary misunderstandings (Huws and Flecker, 2004).

3.4 THE SOFTWARE LABOUR PROCESS

The following section will explore the labour process of professional software work, in order to identify further areas of consideration for this study. Crucially, it is argued that labour process analysis can provide valuable insight into the extent to which the structure, design, organisation, management and control of professional software work influences software professionals' experiences of work intensity.

Labour Process Theory and Knowledge Work

Labour process theory can be applied to provide insight into contemporary transformations in labour, work organisation and employment, such as the transcending of regional and national boundaries for global production and distribution, and the emergence of the knowledge economy (Thompson and Smith, 2000). For example, as emphasised by Thompson and Newsome (2004: 156):

> Bolstered by an expanded scope of analysis and methodologies that add to the traditional case study orientation, LPT can make a strong

> contribution both to the understanding of work and employment trends and to new forms of labour politics that are generated from them.

Indeed, it is argued that reflecting on the global political economy and perceived best practice can be beneficial to labour process analysis, due to potential global influences on employment relations and work at a regional and national scale (Thompson and Newsome, 2004). The emergence of multinational enterprises and international divisions of labour may also offer the potential for comparative analyses of the labour process, due to differences in qualities of capitalism, variability in structural features between countries and diversity in stages, firms and sectors (Thompson and Smith, 2000; Rowlinson and Hassard, 1994). Furthermore, with the emergence of the knowledge economy, labour process theory can be seen to have experienced a transition from the predominant focus on physical labour effort to considerations of cognitive, mental, knowledge-based labour. In this sense, labour process theory is an ideal approach for studying the labour process of professional software work, due to its status as an intellectual, knowledge-intensive occupation.

The Indeterminacy of Labour

Indeterminacy of labour – "the gap between an employee's notional capacity to labour (i.e. their 'labour power') and what the employee actually ends up doing" (Sewell, 2005: 86) - is cited as being at the centre of labour process theory and the source of workplace antagonism between employers and employees (Smith, 2006; Thompson and Smith, 2000). It is also claimed that under the capitalist labour process, there is a tension between capitalist attempts to maximise surplus value and employees' wish to minimise exploitation (Sewell, 2005*)*. Indeed, historically, the struggle between labour and management over the right to determine and control the organisation of work and the use of skills can be seen to have shaped the social relations of production (Giordano, 1988). The indeterminacy of labour can therefore present employers with the challenge of how to ensure potential purchased labour power transforms into actual labour, ensuring profitable outcomes for capital.

However, it is claimed that in contrast to traditional industries, knowledge work occupations such as professional software work may not exhibit conventional structures of control between employers and employees, due to the intangible and tacit nature of work. Chapter Two illustrated that the labour process of knowledge work occupations such as professional software work can be seen to be different from traditional industries, such as manufacturing. Thus, whilst direct control methods could be seen to be appropriate in the early twenty-first century in order to establish order, control and efficiency, Chapter Two argued that responsible autonomy strategies could be deemed more appropriate for those engaged in intangible, creative, challenging and intellectual work (Alvesson, 2004). Management may therefore choose to utilise different control strategies to organise and control labour, depending on technology, competitive conditions in product and labour markets and nature of work performed (Friedman, 1977). The discourse of labour process theory within knowledge work in terms of managing the indeterminacy of labour may therefore be seen to involve the following transformation:

> 'How do we ensure that employees do as managers say?' to 'How do we ensure that employees realise the full fruits of their own expertise and ingenuity for the purposes of the organisation?' (Sewell, 2005: 688).

The indeterminacy of labour within professional software work can be further complicated by the ability of individuals to terminate employment contracts and move to other firms, due to the possession of desirable skills, knowledge and expertise (Smith, 2006; Littler and Salaman, 1982). Indeed, Chapter Two argued that the move from employer ownership of physical capital to worker ownership of intellectual capital in the knowledge economy may have enabled knowledge workers to exert greater levels of influence and negotiation within the employment relationship (Robertson and Swan, 2004). The assessment of worker mobility can therefore be seen to be an important consideration, in that whilst certain categories of workers may have greater levels of organisational dependency, others, such as software professionals, may have greater freedom of movement (Smith, 2006).

However, under the capitalist system, managers may be faced with two potentially contradictory requirements: firstly, to successfully make a profit from the workforce, whilst secondly, simultaneously creating the conditions under which this is possible. Barrett (2005, 2001) argues that knowledge workers, such as software professionals, can therefore be subject to the same conditions in a capitalist society as any other workers: their capacity to work has to be bought and sold like any other commodity and their potential labour has to be converted into actual productive labour to ensure a viable and profitable enterprise. Indeed, Chapter Two emphasised that optimistic accounts of workers being liberated through empowerment, flexibility and team-based structures can be challenged through labour process analysis which emphasises that these methods may merely extend and re-conceptualise control and exploitation (Thompson and Newsome, 2004). Thus, as illustrated by Baldry et al (2007: 30):

> …far from liberating workers' creativity, new forms of work embody the same codification of workers knowledge and inherent opportunity for exploitation typified in the scientific rationality of 'old' low-skilled jobs.

In this sense, a labour process perspective can provide valuable insight into whether software professionals are immune from experiences of work intensity due to their status as knowledge workers, or, if in fact, they are subject to control and exploitation like any other occupation (Andrews et al, 2005). The following section will engage with the main trajectories in labour process theory to consider the extent to which the software labour process may subject software professionals to experiences of work intensity.

Objectivity Within the Software Labour Process

Under early de-skilling and degradation theories on work, the work context can be viewed objectively, with economic, political and ideological elements existing independently of individuals engaged in production (Burawoy, 1978). The application of science and technology to the labour process is deemed to have encouraged the development of ideologies on technical matter and the opportunity for real subordination, through de-skilling, fragmentation of work, hierarchical organisation and division of labour. Consequently, it is claimed that management may potentially be able to secure control over the labour process and address the indeterminacy of labour (Thompson, 1989; Burawoy, 1978; Braverman, 1974). Furthermore, it is argued that the expansion and creation of industries and development of new skills under the capitalist mode of production can potentially reproduce the processes of subordination, fragmentation and degradation of work:

> Capital first destroys old occupations, creates new occupations, then subjects these to the separation of conception and execution (Burawoy, 1978: 299).

Crucially, objectivity within labour process theory can provide valuable insight into the structure, organisation, management and control of professional software work and implications for work intensity. Most notably, the development life cycle, the application of deadlines, the emergence of structured methodologies, adoption of project team structures, standards and performance metrics can be identified as potentially having implications for work intensity. It is claimed that the concern to make software development easier to understand and monitor, as well as considerations surrounding the cost, delivery, performance and reliability of software, encouraged the transformation of professional software work from an unregulated and individualised 'art' to a formal, structured discipline (Friedman and Cornford, 1989; Kraft, 1977; Donaldson, 1973). Professional software work can be seen to be structured around five development life cycle stages, which include: Analysis and Requirements - initial discussions and negotiations with clients concerning specifications, needs, budgets and deadlines; Design - what the software should look like and the most effective way to bring together time, resources and finances; Development - work to make the software to client specifications; Testing - the process of checking the software for errors or bugs, as well as adding value to the finished product; and Installation and Maintenance - installing and maintaining the software (Andrews et al, 2005). As emphasised in section 3.3, deadlines can be seen to form the main mechanism around which all professional software work activities are structured, organised, managed and controlled (Baldry et al, 2007; O'Riain, 2006; Andrews et al, 2005; Barrett, 2004). In addition, the emergence of structured programming methodologies, such as the 'waterfall cycle' and 'Agile', could be seen to allow for greater production efficiency and coordination of workers, delivering a degree of internal control to the labour process. Furthermore, structured methodologies can be seen to have aided the fragmentation of work activities, potentially giving rise to an authority hierarchy, with task distribution being based on levels of expertise, skills and knowledge (Baetjer, 1998; Kraft, 1979).

Project team structures, which were initially introduced as a complementary organisational form to structured programming methodologies, can also be seen to help with the management of fragments of work, assisting coordination and control (Kraft, 1977). In addition, teamworking structures may enable the re-configuration of control, replacing management supervision with the formalisation of peer control and self-management:

> Ironically, perhaps, whilst the redistribution of power is central to the teamworking philosophy, in reality power quickly becomes formalised within teamworking situations and leads to high levels of team control and coercion over individual team members. Power is passed from the hierarchy to the team members themselves so that they become self-managing teams. This power is used by the team

members to police and control each other's behaviour (Kraft, 1979: 54).

The empowerment of individuals through project team structures can therefore potentially be seen as being more effective at securing and extracting value than traditional control methods (Sharpe, 1998). In this sense, it is useful to consider the implications of project team structures for software professionals' experiences of work intensity.

Software professionals may also be expected to follow certain standards, with regards to documentation, coordination and quality. Documentation, through providing specifics of programs, systems, decisions and coordination standards, such as style, layout and format conventions may increase transparency and encourage consistency of approach (Friedman and Cornford, 1989). Peer code reviews may also be utilised within the software work process to check code reliability and quality. In addition, increasing competition, pressures from purchasers and concern to improve system quality, performance and reliability are perceived to have encouraged the application of quality standards (Quintas, 1994).

Methods such as performance metrics, worksheets to record time spent on client projects and tasks, targets for task completion and users signing off work are also asserted to represent additional ways in which management can potentially exert control over the software labour process (Friedman and Cornford, 1989). For example, performance metrics, such as log sheets and status reporting procedures may enable managers to schedule workloads and signal completion dates to clients more effectively (Beirne et al, 1998). Worksheets to log time spent on client projects and tasks can also be seen to assist the monitoring of individual output and cost efficiency. Indeed, cost efficiency is considered to be an important criterion for project teams and their survival. For example, teams may have responsibility for budgets and resources may be determined by revenues generated from software sales to clients and the wider market (Voss-Dahm, 2005). At the individual level, performance appraisals and performance-related pay may also place intense pressure on software professionals to provide information on progress and increase effort levels in order to meet targets, personal development goals and avoid sanctions (Beirne et al, 1998). For instance, Green (2006) states that between 1993 and 1998, work intensification was observed in 44% of organisations that increased their use of performance-related pay.

The recognition that objective structures, such as the development life cycle, deadlines, methodologies, standards, performance metrics and documentation, exist within the software labour process suggests it is important to explore the extent to which these aspects influence software professionals' experiences of work intensity.

Subjectivity and Worker Agency Within the Software Labour Process

It is important to note, however, that focusing solely on objective aspects of the software labour process may neglect the presence of subjective aspects of work and variability in worker resistance (Thompson, 1989). Indeed, it can be argued that a purely objective approach may underestimate the knowledgeability and capability of workers faced with a range of management imperatives and the ability of management to appropriate and monopolise elements of knowledge (Wilson, 1988; Burawoy, 1978). For example, whilst standards may arguably help management to obtain greater visibility over the software work process, these methods may be equally valuable to workers themselves through enabling work to be shared and understood more effectively. In addition, whilst peer code reviews can help to detect errors and improve the quality and productivity of software, they can also be seen as an important part of the learning and mentoring process for professional software workers:

> The most popular and perhaps the most effective way to learn and to improve programming techniques is to have an experienced

> practitioner go over your work with you (Friedman and Cornford, 1989: 164).

Similarly, documentation can allow software professionals to make their work more comprehensive to other team members, due to the discretionary and complex nature of work. Workers themselves may also introduce methods to enhance and maintain the software development process; indeed, structured methods are argued to have been used implicitly before the implementation of formal techniques. In this sense, consideration of subjectivity can enrich understanding of the nature of control within the software labour process:

> In other words, Braverman's restriction of attention to the "objective" elements of work is illegitimate if he is to understand the nature of control since, by definition, control involves what we would refer to as "subjective" aspects of work and what I will refer to as political and ideological processes. Only when these processes are understood can we proceed to examine the variety of forms of the capitalist labour process, the transition from one to another and the relationship between the separation of conception and execution and the obscuring and securing of surplus (Burawoy, 1978: 266).

Worker subjectivity may also be influenced by attempts to encourage the voluntary production and reproduction of power and domination within the work context through organisational culture, discourses and disciplines (Thompson and Smith, 2000; Spencer, 2000; Lash and Urry, 1993). Indeed, individuals may potentially reproduce capitalist relations of exploitation and domination through the search for validation of actions:

> That individual workers locate opportunities for self-confirmation under existing capital structures has the 'unintended consequences of concealing from labour the extent to which its pursuit of such

> opportunities has the contradictory effect of reinforcing its dependence on capital (Willmott, quoted in Spencer, 2000: 236).

It is claimed that organisations may place emphasis on internal ideology in an attempt to control individual attitudes and behaviour, internalise performance norms, and encourage greater intensity of work effort from workers (Green, 2006; Kraft, 1977). For example, organisations employing knowledge workers may utilise artefacts and symbolism, such as company history, artefacts, stories and practices in order to encourage cultural orientations:

> Cultural control overlaps and informs identity focused control. Cultural material – symbolic management behaviour, rites, rituals, stories, jargon, material artefacts – not only provides guidelines for orientation in a social landscape but gives clues for how individuals working there should see themselves (Alvesson, 2004: 213).

However, in the context of professional software work, it may potentially be difficult for organisations to encourage sole identity with corporate culture, due to the prevalence of identities at the team, departmental and occupational levels (Newell et al, 2002). In this sense, it may be more appropriate for organisations to instil a sense of community and shared ethos through placing emphasis on the interpersonal bonds between individuals:

> The meanings, beliefs and values around knowledge, support, collaboration and expectations of reciprocity and a shared feeling of togetherness and a common identity associated with corporate belongingness are crucial for the active sharing and offering of experience and insights. The inclination to take time and make an effort to respond positively to requests for assistance or invitations to collaborate outside one's closet set of relations is an outcome of cultural control (Alvesson, 2004: 176).

Consideration of subjectivity can therefore enrich understanding of the ability of software professionals to exert power within the software labour process and resist or respond to strategies implemented by management (Knights and Willmott, 1989; Friedman, 1977). However, it should be recognised that a purely subjective approach may be equally limiting, through the predominant focus on the individual instead of the collective, identities opposed to interests and the neglect of institutional underpinnings and context, in terms of capitalist forces, production relations, market competition and economic concerns (Thompson and Newsome, 2004; Spencer, 2000). Thus, consideration of both objective and subjective aspects of labour process theory can arguably help to provide a more comprehensive understanding of the software labour process:

> …the *raison d'etre* for LPT is that it links the subjective and objective sides of labour's position as paid labour to a capitalist political economy which operates within temporal and historical circumstances and constraints. Analysis of the labour process within the workplace has to be socially and economically *embedded* and not cast adrift through technological ideal types of existential traumas of the human condition (Thompson and Newsome, 2004: 62).

Consent and Accommodation Within the Software Labour Process

Labour process theory also attempts to explain the contradictory relations of control, consent and co-operation between capital and labour at the point of production. For example, it can be argued that the internal dynamics of struggles between capital and labour may influence the systems through which employers direct work tasks, supervise, evaluate performance, discipline and reward workers (Edwards, 1979). Workers may also play a role in creating conditions of consent through their means of adapting to work, producing and reproducing interests in particular ways (Burawoy, 1978). Indeed, whilst systems of control may be necessary for the extraction of surplus value, workers can resist this control passively or actively on an individual or collective basis (Storey, 1985). In response, capital may potentially re-organise the labour process with a shift towards consensual methods for organising workers, in order to minimise worker opportunities and desire for resistance (Edwards, 1979). Control can therefore be viewed as a dialectical process, generated through the struggle, degrees of rationality and negotiation between different management groups and workers (Littler and Salaman, 1982; Zimbalist, 1979).

Consideration of consent and accommodation is of significant value to this research, in that the intellectual and intrinsically interesting nature of professional software work may potentially motivate individuals and enable employers to harness the creative and reproductive powers of labour (Boreham et al, 2008; Edwards and Scullion, 1982; Cressey and MacInnes, 1980). Furthermore, observations within Chapter Two that professional software work may be subject to minimal supervision, facilitative and co-operative styles of management, autonomy, internal motivation to work and normative aspects suggests it is important to consider whether these dimensions influence experiences of work intensity.

Increasing user involvement can be seen to have necessitated more flexible and autonomous approaches to managing professional software work, due to the unpredictable impact of clients on work (Friedman and Cornford, 1989). Software professionals may therefore utilise operational autonomy, with freedom to deal with a set problem through self-determined means, technical autonomy exercised over tasks and time autonomy, in terms of the ability to influence the duration, scheduling and distribution of work time (Barrett, 2005). Indeed, time sovereignty is claimed to be crucial to professional software work, due to the interdependent and co-operative nature of work, the reliance on clients to provide information on time and the need to depart from planned schedules when necessary (Voss-Dahm, 2005; Newell et al, 2002). However, autonomous working conditions may create conditions of consent and accommodation, with employees controlling themselves in the economic interests of the firm by working harder and longer to complete tasks to deadlines (Rasmussen and Johansen, 2005). Thus, as a result of autonomous time management, working time may paradoxically increase for software professionals (Voss-Dahm, 2005). Furthermore, the presence of autonomous working conditions, coupled with intrinsically satisfying work attributes and perceived identity with the occupation, may encourage normative and cultural dimensions of control within professional software work (Alvesson, 1995; Kunda, 1992). For example, intrinsically satisfying work may encourage individual self-supervision, employee loyalty to organisational objectives and assist in the management of individual behaviour and attitudes (Alvesson, 1995; Kunda, 1992; Boreham, 1983). In this sense, it is useful to consider employee responses to the software labour process and implications for experiences of work intensity.

CONCLUSION

This chapter has discussed the key streams of literature surrounding professional software work, in order to identify aspects which may be of importance to this study. Notably, whilst knowledge workers may experience more favourable terms and conditions and forms of work organisation, this chapter has argued that the notion of 'free' workers may be overstated. For example, teamworking structures may encourage greater self-management, review from within and peer control, potentially representing a new way of controlling workers (Beirne et al, 1998). In addition, the intrinsically motivating nature of professional software work may encourage individuals to adapt themselves to the cyclical nature of work, acting as a form of normative control (Baldry et al, 2007; Beirne et al, 1998). In this sense, the status of software professionals as archetypal knowledge workers enables consideration of the implications that transformations in working conditions and work organisations may have had for individuals, particularly in terms of work intensity.

This chapter has also identified that particular organisational and interactional aspects of professional software work may have implications for software professionals' experiences of work intensity. Crucially, software professionals may be subjected to heavy workloads and scheduling pressures, in order to meet aggressive deadlines. Deadlines may regulate individual expenditure of work effort, influence the internal dynamics within project teams and facilitate the application of strategies which reward or penalise workers. Deadlines, clients, internal organisational processes, status and individual strategies have also been shown to have implications for the management of working time within professional software work. In addition, interruptions, whilst being considered to be a normal part of the work process within knowledge-intensive occupations, may have adverse consequences for software professionals, through affecting thought processes, disrupting work rhythms and fragmenting work schedules (Perlow, 1997). Furthermore, whilst globalisation and ICTs may have enabled interconnectedness, provided the opportunity for 24/7 working and encouraged more flexible ways of working, this chapter has argued that changes in work location and physical proximity may adversely affect the ability of software professionals to interact, communicate and collaborate effectively.

This chapter has argued that labour process theory can be applied to understand the software labour process and assist in the identification of aspects which may have implications for experiences of work intensity. Most notably, forms of work organisation, worker agency, consent and accommodation, cultural aspects and normative elements within professional software work can be identified as potentially influencing software professionals' experiences of work intensity. The development life cycle, deadlines, structured programming methodologies, project team structures, standards and performance metrics can be seen to have introduced elements of rationalisation and structure to the software work process, which may influence software professionals' experiences of intensity. However, this chapter has suggested that these methods may be beneficial to workers themselves, through allowing work to be shared and understood more effectively, as well as aiding personal development. In addition, possession of an intangible and intellectual means of production may potentially enable software professionals to exercise agency within the software labour process and mitigate managerial control:

> Programmers thus persist in being something of an anomaly in the modern workplace: they are employees but they are in a position to control much of how they will go about doing their programs – the final product – and to some extent the form the final product will take. Even the careful use of structured programming methods does not for the present give managers absolute control over all aspects of software production (Kraft, 1977: 62).

Knowledge work characteristics such as autonomy may arguably be necessary in order for software professionals to deal with uncertainty and ambiguity. However, this chapter has illustrated that normative and cultural dimensions may encourage individuals to have greater levels of personal investment in work and control themselves in the economic interests in the firm. In this sense, this poses the question of whether those working in knowledge-intensive occupations, such as professional software work, are necessarily 'free' from attempts to structure, control, organise and manage work:

> …it is unlikely that all knowledge workers may be classed as 'free workers', as many are unlikely to influence their working environment and also may have little say in the organisation of their work due to a variety of factors including the degree of employer dependency, work commitments and client/professional pressures. Consequently, the concept of knowledge workers exercising significant control over their work organisation/environment may merely constitute spurious 'futurology', for as Scarborough (1999: 6) asserts, rather than being 'free workers', knowledge workers are located "within relations of employment and control, where the social and institutional conditions of the work process

are shaped by endemic, multi-level conflicts and contradictions (Donnelly, 2006: 82).

The labour process of professional software work may therefore be subject to tensions, contradictions and complexities, which may have implications for work intensity. It can therefore be concluded that labour process analysis can enable consideration of the extent to which the structure, design, organisation, management and control of professional software work influences experiences of work intensity.

In summary, Chapters Two and Three have helped towards the identification of particular areas which may have implications for software professionals' experiences of work intensity. For example, Chapter Two drew attention to the presence of facilitative working conditions within knowledge work, developments in work organisation, the intrinsic attributes of professional software work and normative aspects. Chapter Two also emphasised that consideration of contextual factors, institutional elements, job-related characteristics and personal factors may assist the examination of software professionals' experiences of work intensity. Moreover, Chapter Three has demonstrated that the software labour process, forms of work organisation and organisational and interactional dynamics within professional software work may have implications for software professionals' experiences of work intensity. The insight generated from Chapters Two and Three has therefore enabled the generation of five research questions which will guide this study:

- How do firm characteristics/internal organisational factors (organisational type, company size, product market, skills and knowledge development opportunities, leadership style) influence experiences of work intensity?
- To what extent are software professionals subject to experiences of work intensity as a result of the way work is structured, designed, organised, managed and controlled?

- In what ways are software professionals affected by associated internal organisational factors?
- How are software professionals affected by the way their work is structured, designed, organised, managed and controlled?
- What are software professionals’ perceptions and experiences regarding intensity/intensification?

CHAPTER 4 METHODOLOGY

INTRODUCTION

This chapter outlines the methodological approach taken in this research. Section 4.1 discusses Critical Realist philosophy and its applicability to this research, through the emphasis on establishing linkages and identifying mechanisms, structures and relationships to provide detailed explanations of work intensity. Section 4.2 outlines the intensive, in-depth qualitative case study approach utilised in order to develop a comprehensive, theoretical and conceptual understanding of work intensity. Section 4.3 presents a taxonomy of professional software job roles which has been developed by the author to provide insight into the nature of activities and tasks performed by individuals, variations in work experiences and relationships to work intensity. Section 4.4 discusses the design and application of qualitative research methods (observation, documentation, work diaries and semi-structured interviews) within this research, arguing that the utilisation of multiple methods can enable phenomenon to be considered over a variety of levels and allow for the development of an in-depth, comprehensive understanding of work intensity. Section 4.5 considers existing frameworks to obtain insight into how perceptions of work intensity can be evaluated and presents an index of possible determinants of work intensity, delineating dimensions which existing frameworks have not acknowledged. Sections 4.6 and 4.7 provide information on the two case study companies, SpecSoft and InSoft, respectively, in terms of company background, project overview and project team structures. Section 4.8 documents the purposive and iterative sampling strategy adopted, outlining response rates and participant characteristics for work diaries and semi-structured interviews. Section 4.9 outlines the main procedures for recording of data such as field notes, work diary pie charts and tables, interview notes, recordings and transcriptions. Section 4.10 presents the strategies utilised to analyse data, including incremental analysis, combination of data methods to identify themes, case study write-ups to compare case study data and identification of general themes, relationships and mechanisms to explain work intensity.

4.1 PHILOSOPHICAL APPROACH

It is claimed that an understanding of research philosophy can help to identify underlying assumptions in terms of the way the world is viewed and the rationale behind research approach and design. Critical Realism is an appropriate paradigm for this research due to its emphasis on establishing causal links and explanation of mechanisms and wider structure to provide a comprehensive understanding of phenomena. Critical Realism emerged in the 1970s as an alternative to Empiricism, Positivism and Interpretivism in social science philosophy. Bhaskar (1989) is generally acknowledged to have developed the most comprehensive and influential version of Critical Realism, with the term originating from the hybridisation of Bhaskar's 'Transcendental Realism' and 'Critical Naturalism' philosophies. 'Critical' refers to the philosophy's critique of structure and agency, the desire to change unsatisfactory or oppressive realities and the suggested limitations in applying universalistic laws and truth in social science. 'Realism' refers to the 'resigned acceptance' that a world can be seen to exist independently of our knowledge, whilst also emphasising that an understanding of this world can be accessed through social science research (Danermark et al, 2002; Benton and Craib, 2001; Bhaskar, 1989).

4.1.1 Principles of Critical Realism

Critical Realism postulates that the world can be viewed objectively and exists largely independently of our knowledge. This objective reality means that structures and institutions are, to a considerable extent, beyond the direct control of individuals and groups and have the power to lead them into particular patterns of relationships

(Ackroyd, 2002). Structure can be seen as a necessary condition for human activity, in terms of providing the rules, means, resources and underlying mechanisms through which institutions, organisational forms and collectivities are reproduced (Reed, 2000; Bhaskar, 1989). For example, Tsoukas (1989) suggests that the superior-subordinate relationship can be viewed as part of a wider objective structure which includes labour power, the division of labour and capital owners and the production of services being made possible and constrained through that structure. Indeed, it is claimed that:

> …the prior emergence of the relational properties that inhere in social structures necessarily impinge on current actors and their situations as unavoidably they find themselves operating in pre-structured contexts and interests that shape the social struggles in which they are implicated (Reed, 2000: 55).

Critical Realism also draws attention to the interplay of structure and agency with regards to the capacity for individuals to reproduce or transform social relations, structures and relations through intentional or unintentional agency (Alvesson and Sköldberg, 2009; Benton and Craib, 2001; Ackroyd and Fleetwood, 2000; Reed, 2000; Bhaskar, 1989). For example, Archer et al (1998: xvi) suggest that:

> …agents are always acting in a world of structural constraints and possibilities that they did not produce. Social structure, then, is both the ever-present condition and the continually reproduced outcome of intentional human agency.

In addition, it is argued that individuals may vary in the extent to which they are complicit in reproducing structures or capable of producing changes, depending on interests, powers, resources, constraints and the nature of relationships (Ackroyd, 2004; Archer, 1998).

Critical Realism shares some similarities with Positivism with regards to the importance of identifying causalities, patterns and the considered objective nature of reality. However, Positivism can be seen to have a flat ontology, viewing the world as consisting solely of that which can be observed and causation relating to regularities amongst events. Positivism also postulates that social research can be treated scientifically and that causal objects under investigation can be treated independently of events and individuals, allowing for the identification of universal laws and truths (Bhaskar, 1989). However, Critical Realists argue that there may be difficulties in sustaining universal and independent laws within social science, due to the absence of natural laws and the constant conjunction of events in social systems (Archer et al, 1998). Indeed, it can be argued that the Positivist position to deduce universal statements and laws from purely empirical observations may neglect the presence of additional entities or processes (Benton and Craib, 2001). For example, the Positivist approach may fail to recognise human attributes, such as reflexivity and creativity and less observable types of knowledge, such as implicit rules and shared understandings, which may be difficult to articulate and cannot easily be reduced to statements or propositions (Benton and Craib, 2001; Archer, 1998). It can therefore be argued that:

> Since positivists cannot have recourse to unobservables they cannot explain in any meaningful sense. They may be able to show that there are complex patterns in the data that they collect based on measuring techniques they have devised but they cannot say why. Their 'explanations' are simply summaries of relationships among a set of variables (Easton, 2000: 213).

Critical Realism also possesses some similarities with Interpretivism, through emphasis on the context-dependent nature of social phenomena and the importance of meaning (Sayer, 2008). However, these philosophies differ in their overall focus, in that Critical Realism emphasises the importance of causal explanation, as well as regarding research as an interpretive and creative activity. This is in opposition to Interpretivism, which is considered to reduce social life to meaning and ignore material changes (Ackroyd, 2004). In this sense, Critical Realism can be considered an attractive philosophical alternative, through emphasising the importance of meanings, whilst also stressing the need for insight into mechanisms, structures and layers of reality (Sayer, 2008). Critical Realism therefore argues that whilst it is important to identify causal links between objects, it is equally necessary to ask questions about their status and identify mechanisms, structures and relationships to provide deeper levels of explanation on phenomena (Mingers, 2004; Sayer, 2004; Benton and Craib, 2001; Ackroyd and Fleetwood, 2000; Bhaskar and Lawson, 1998). Indeed, as emphasised by Sayer (2008: 15):

> There is more to the world, then, than patterns of events. It has ontological depths: events arise from the workings of mechanisms which derive from the structure of objects, and they take place within geo-historical contexts.

Critical Realism suggests that reality is stratified in nature, spanning the real, actual and empirical domains (Ackroyd and Fleetwood, 2000). The 'real' level relates to the mechanisms and structures which lead to events. The 'actual' domain refers to the events which may occur or are created by the interaction of mechanisms, independently of our knowledge. The 'empirical' realm is narrower in focus, relating to the area of experience where events may be observed, either directly or indirectly (Alvesson and Sköldberg, 2009; Danermark et al, 2002; Easton, 2000; Bhaskar, 1989; Outwaithe, 1987). This stratification allows for an in-depth understanding of reality, with regards to the causal powers of objects, identification of mechanisms, the association of patterns and the incorporation of these constituent parts into a wider structure (Archer, 1998; Bhaskar, 1989; Tsoukas, 1989).

Critical Realism also argues that whilst the world may exist largely independently of our knowledge, the intransitive dimension (in terms of the reality or object for analysis which can be studied, such as phenomena, events or mechanisms) can confirm or disconfirm gathered understandings. This means that the transitive dimension (the process of generating and applying theories to make sense of and provide insight into reality) may be provisional or subject to change as a result of further or future research (Alvesson and Sköldberg, 2009; Mingers, 2008; Danermark et al, 2002; Benton and Craib, 2001; Ackroyd and Fleetwood, 2000). Critics of Critical Realism therefore suggest that concepts and constructs may not always be applicable if understandings on the objective reality are subject to revisions (Outwaithe, 1987). However, Critical Realists argue that the on-going, reflexive and iterative nature of Critical Realism and re-conceptualisation is essential, in order to reflect underlying mechanisms and the nature of reality more effectively and with greater accuracy (Modell, 2009; Sayer, 2008).

4.1.2 Methodological Implications of Critical Realism

This research aligns ontologically with the Critical Realist position and has deductive elements, through recognition of an objective, independent reality within professional software work. The pre-conceived understanding of this objective reality has emerged from themes in Chapters Two and Three concerning the characteristics of professional software work and forms of work organisation and management control. For example, Chapter Three illustrated that professional software work may be subject to project team forms of work organisation, client involvement, standardisation, rationalisation and normative elements, which have the power to lead individuals into particular ways of working.

However, a purely deductive approach can be seen to be of limited value, due to the overall research aim of providing a comprehensive understanding of work intensity in professional software work through the identification of causal links and insight into mechanisms, structures and relations. For example, an analysis of the objective reality, in terms of how professional software work is structured, designed, managed and controlled and the implications this has for individual experiences is crucial to obtaining a theoretical and practical understanding of work intensity. For instance, Baldry et al (2007: 16) argue that:

> How work is actually experienced will be influenced by the managerial style of the organisation, the intrinsic and extrinsic reward systems in place, the structure of work organisation (such as team work or assembly line) and the content and design of the actual job, tempered by prior dispositions of the employees themselves.

This research therefore also utilises inductive reasoning, positing the importance of conceptualisation and the application of data to create greater theoretical understanding of work intensity. Whilst it is recognised that an inductive approach may be problematic through difficulties in applying certain descriptions or generalisations to unobserved occurrences or unstable circumstances, these limitations may lessen when applying an inductive approach to an objective reality:

> The possibility of making well-founded empirical generalisations depends on what the reality under investigation is like. To put it simply, when we draw conclusions about a fairly stable reality, the risk is comparatively small of generalisations turning out to be false (Danermark et al, 2002: 86).

Critical Realism also suggests it is important to consider the contextual circumstances under which meanings manifest and give rise to phenomena (Modell, 2009). Indeed, the Critical Realist focus on necessity and contingency, as opposed to regularity and identification of universal principles, can help to understand differences in outcomes according to context and the identification of causal powers and mechanisms which may operate and interact to create particular events (Sayer, 2008; Easton, 2000; Tsoukas, 1989). The contextual element can also be seen to be crucial in terms of allowing for the interpretation and understanding of the experiences of individuals (Sayer, 2008).

However, much of the existing research on the software industry remains de-contextualised, tending to diminish the relationship between the software labour process, institutional factors and how these relate to the experiences of software professionals. This relates to analysing the extent to which there are links between contextual variables (such as organisational type, company size, product market, skills and knowledge development opportunities and leadership style) and experiences of work intensity. A key contribution of this research is therefore the application of a contextually-based analysis of professional software work, in order to provide a detailed understanding of work intensity according to context and the generalisation of particular themes and issues across a wider population of software firms. The importance of this contextual analysis is further illustrated by Green (2001), who suggests the impact of work pressures may be affected and mediated by differing organisational contexts. Similarly, Wichert (2002) argues that personal, social and environmental factors can influence the vulnerability of individuals to pressures from work intensity. This research therefore recognises the importance of a contextual analysis utilising multiple data sources, in order to obtain a greater understanding of causal linkages and phenomena.

Critical Realism also suggests that reality is differentiated, structured and stratified, presenting the potential for conflicting practices and interests, interpretations and variability in experiences of the world (Danermark et al, 2002; Easterby-Smith and Thorpe, 2002). A key assumption within this research is therefore the recognition of conflictual realities, as a result of individual personalities, dispositions and individual working conditions, which may potentially influence the experiences of individual software professionals with regards to work intensity. It is therefore imperative to explore how individuals react to the software labour process and internal organisational factors, in order to determine and understand experiences of work intensity.

4.2 NATURE OF ENQUIRY

Case study research is argued to be valuable in terms of enabling insight into phenomenon and the complexity of social actions, structures and dynamics within individual settings (Danermark et al, 2002; Orum et al, 1991). Case studies can be quantitative in nature, utilising statistical and empirical methods to indicate relationships and links within populations or qualitative, in order to obtain in-depth, comprehensive understandings of phenomena and allow for the development of theory (Eisenhardt, 2002). Case studies following a Positivistic approach would place emphasis on hypothesising, empirical data, the rejection of unobservables and establishing law-like relationships, in order to generalise findings to the wider population (Yin, 2003). However, whilst an empirically driven case study could be beneficial through establishing empirical links, the Critical Realist approach and application of qualitative research methods are deemed more appropriate to this research on work intensity, due to the emphasis on exploratory work to explain what produces particular states, changes and situations, alongside the identification of causal linkages (Ackroyd and Fleetwood, 2000). Thus, as exemplified by Easton (2000: 212):

> Case studies which would wish to lay claim to a realist philosophy should be carried out in a different way: to be inquisitive, to look for the roots of things, to disentangle complexities and to conceptualise and re-conceptualise, test and retest, to be both rigorous and creative and above all to seek for the underlying reality through the thick veil which hides it.

4.2.1 Case Study Research Design

Gerring (2007) suggests that it is useful to identify cases which reproduce relevant features typified by the population, as well as those which provide elements of variation on dimensions of interest, in order to provide more comprehensive understandings of phenomenon. For example, typical cases can be seen to be representative of the wider population, potentially improving applicability and reliability of results. Diversity in cases may also allow for contrasts and comparisons to be drawn between cases (Gerring, 2007; Yin, 2003; Danermark et al, 2002). Through examination of the software industry in Chapter Two, four main organisational types were identified as representing the breadth across which software professionals may work: specialist software firms; in-house IT departments in large firms; computer games firms; and multimedia firms. The taxonomy of professional software job roles (see Section 4.3) also provided valuable insight into the nature of work performed by software professionals, aiding the identification of cases which demonstrated variability in these functions. In an ideal, constraint-free world, a full spectrum of firm types could have formed the subject of study. However, constraints of time prevented the application of in-depth investigation, as employed in SpecSoft and InSoft, to other parts of the software industry, such as the computer games and multimedia industries. In addition, the decision to undertake detailed inquiry into two firms, sacrificing breadth for depth, can be justified on several grounds.

Firstly, the choice to focus on depth over breadth can be seen to be related to the nature of phenomenon being studied, research questions and the overall aims and objectives of research (Yin, 2003). The overall aims and objectives of this research meant that in-depth and exploratory investigation into each case study was necessary in order to fully consider the range of factors which may have had implications for experiences of work intensity. Furthermore, intensive, in-depth case study designs are argued to be more fitting to research objectives which emphasise the importance of interpreting meanings in context, the tracing of causal relationships and the development of comprehensive explanations underlying phenomena. This is in contrast to extensive, breadth-based case study designs which, while allowing the population of primary inference to be covered more fully, places more emphasis on the establishment of regularities, empirical patterns and law-like relationships and is deemed to be less applicable to research requiring comprehensive and detailed explanatory outcomes:

> …in this situation, there is no attempt to get behind the conjunctions to discover what the causal powers and mechanisms might be. Researching greater number of cases, with the same resources, means more breadth but less depth. One may be able to identify other contingent causal powers but at the expense of discovering how they operate 'in reality'. This is clearly an alternative research strategy but one which should build on deeper knowledge to start with (Easton, 2000: 214).

Secondly, it can be argued that intensive research concerned with exploring and understanding structures and mechanisms which give rise to phenomena may make it more difficult to study a larger number of cases (Harrison and Easton, 2004; Danermark et al, 2002). This leads to the conclusion that:

> …there is typically a trade-off to be made between the increased potential for generalisability flowing from studying a large number of sites and the increased depth and breadth of description and understanding made possible by focusing on a small number of sites (Schofield, 2002: 185).

Whilst the selection of two cases could be criticised on the grounds of the potential for case specifics giving rise to certain conclusions, the comparative case study approach can provide valuable insight into variations in causal mechanisms, patterns and relationships according to contextual factors:

> …the contexts of the two cases are likely to differ to some extent. Under these varied circumstances, you can still arrive at common

> conclusions from both cases, they will have immediately expanded the external generalisability of your findings, again compared to those of a single case alone (Yin, 2003: 53).

The two cases selected were deemed to provide valuable insight into work intensity, due to variation in organisational type, market representation, software engineering roles and team locations. Thus, thirdly, diversity between cases is perceived to allow for typological theorising, allowing for consideration of factors which may have an effect on the outcome, enabling the identification of mechanisms and informing conceptualisation and theory development (Sayer, 2008; Gerring, 2007). Indeed, heterogeneity between case study sites is deemed to be potentially useful in terms of general theoretical application to understanding other sites:

> Generally speaking, a finding emerging from the study of several very heterogeneous sites would be more robust and thus more likely to be useful in understanding various other sites than one emerging from the study of several very similar sites (Kennedy, 1979, as referred to in Schofield, 2002: 184).

SpecSoft was identified as being representative of a typical specialist software firm, due to its small-medium size, range of software solutions and varied client base. The representation of niche markets based around industry sectors within the divisional organisational structure also offered the opportunity to examine experiences of work intensity according to two contrasting market types, the longer-term projects typical of travel and transport work and the faster paced telecommunications sector. The traditional software function performed within SpecSoft also represented the opportunity to examine how this influenced experiences of work intensity.

InSoft was typical of a large firm with an in-house IT service and provided the opportunity to examine the evolving roles of software professionals at the organisation, with regards to the impact of industry trends and implications for experiences of work intensity. The offshoring of so-called ‘low value’ software engineering functions within InSoft also captured current trends identified within the software industry (as discussed in Chapter Two) for organisations to outsource software services to specialist software organisations in the same or other countries. There are difficulties in generalising to the Indian case from the experience of three Indian offshore workers. However, their inclusion provided valuable insights into the geographical dispersal of work and the teamworking relationships between Glasgow, other UK-based sites and remote service delivery sites. Furthermore, indicative findings and suggestions can present signposts for potential future research on other cases (Gerring, 2007). The differences in team physical proximity between SpecSoft and InSoft also allowed for comparisons to be drawn with regards to implications for experiences of work intensity. In this sense, the comparative approach adopted allowed contextual factors to be taken into consideration, helping to identify commonalities and differences and, crucially, assisting in the development of theoretical understandings (Yin, 2003; Danermark et al, 2002; Bhaskar and Lawson, 1998; Tsoukas, 1989).

4.2.2 Analytical Generalisation

Two types of universal concepts can be identified within social science methodology with regards to generalisation, namely empirical concepts, which relate to the population sharing a formal property and abstract concepts, in terms of structure, processes or mechanisms of the constituent (Danermark et al, 2002). The nature of approach therefore has implications for the type of generalisation which can be applied:

> Realists believe that identifying a plausible and defensible ‘deep’ explanation in one instance can be a major contribution to theory. The question of whether this theory, or any of its components, applies elsewhere then becomes an empirical one. It may be widely applicable

> or it may be narrowly applicable. By contrast, positivists accept that constant conjunction is the best 'indicator' of causality though it cannot guarantee it. This now leads to the assumption that one theory, in the sense of the same set of empirical regularities, will fit all situations. For this to be the case a model built around a set of variables must be assumed to 'capture' the underlying dynamics of the phenomena in question (Harrison and Easton, 2004: 195).

Statistical generalisation of inferences to be applied to a wider population can therefore be seen to be consistent with a Positivistic approach which has emphasised the importance of identifying causal links and law-like relationships, in order to establish universalities. However, this contrasts with analytical generalisation made possible through the Critical Realist perspective, where the identification of entities, processes and mechanisms can allow for the development of theoretical and conceptual understandings (Danermark et al, 2002; Eisenhardt, 2002; Orum et al, 1991). In addition, whilst it may be difficult to generalise from the particularities of a case to the wider population, generalisation can occur through 'reasoning by analogy' by applying knowledge gained from one case to another perceived to exhibit similar traits (Rossman and Rallis, 2003).

4.2.3 Validity and Reliability of Case Study Research

Criticisms have been levied at case study research concerning perceived lack of rigour, the absence of systematic procedure with regards to data collection, data management and analysis and the potential for bias, assumptions or prejudice to influence findings and conclusions (Gerring, 2007; Silverman, 2000; Orum et al, 1991). Case studies may also be criticised on the grounds of construct validity, whereby researchers may use subjective judgements for data collection, opposed to applying operational measures (Yin, 2003). However, the utilisation of multiple sources of evidence can help to provide multiple measures of the same phenomenon, forming a ‘chain of evidence’ and increasing the validity of findings:

> …the most important advantage presented by using multiple sources of evidence is the development of *converging lines of inquiry*, a process of triangulation…Thus any finding or conclusion in a case study is likely to be much more convincing and accurate if it is based on several different sources of evidence, following a corroboratory mode (Yin, 2003: 99).

In addition, whilst diverging or converging causal explanations may emerge from the utilisation of multiple methods, it is argued that this can allow for further scrutiny into the meaning of differences, allowing for the refinement and extension of theory (Eisenhardt, 2002; Fielding and Fielding, 1986).

It is also claimed that researchers may experience difficulty in detaching themselves from the research context in social science, due to the need to examine how individuals understand themselves and their settings (Gillham, 2000). However, this can be seen to relate to the essence of case study research design, which is to provide more comprehensive insight into the experiences of individuals, groups or organisations within contextual settings:

> The researcher who uses the case study typically seeks to grasp the nature of social action as it has been experienced by people themselves. He or she has chosen the case study to get at the human understandings that underlie the actions he or she portrays. The narrative form is precisely adapted to communicating these meanings and understandings – the “lived” experience, as experienced by people (Orum et al, 1991: 21).

Thus, with the recognition that reflexivity and elements of subjectivity exist within social science research, it is essential that case study research is systematic and rigorous in terms of clarifications of research aims, documentation and procedure. For example, it is argued that case study researchers should be explicit on purpose and meticulously document and record information on research design, data collection methods and data analysis processes. In addition, participant validation of information or review by peers can also be useful in verifying the validity of information (Gerring, 2007; Rossman and Rallis, 2003; Eisenhardt, 2002).

4.3 CATEGORIES OF PROFESSIONAL SOFTWARE WORK

Chapters Two and Three engaged with the key debates on the knowledge economy, software work and work intensity. However, despite the contributions of existing literature surrounding professional software work, there have been no attempts by researchers to explain the type of roles software professionals may be engaged in and the detailed nature of tasks. This section therefore presents a taxonomy of professional software job roles which has been developed by the author to provide insight into the nature of activities and tasks performed by individuals. This taxonomy has also been applied within this research to create the 'Activities and Codes' list and role descriptions on the 'About You' form as part of the work diary study (see Section 4.4.3). In this sense, the taxonomy has permitted exploration into the day-to-day activities carried out by software professionals', variations in work experiences and relationships to work intensity.

4.3.1 Heterogeneity, Complexity and Variability Within Professional Software Work

Existing research has tended to focus on the development life cycle as a means of discussing professional software work (Andrews et al, 2005; Marks and Lockyer, 2005; Marks et al, 2002; Barrett, 2001; Marks et al, 2001; Beirne et al, 1998), rather than outlining the activities performed by individuals themselves. Difficulties in classifying professional software roles can potentially be attributed to the heterogeneous nature of the software profession, in terms of varying job titles and roles, differing levels of involvement at stages in the software life cycle, depending on company size, organisational type and project nature and the intangible nature to work. Job titles and descriptions may contribute to further difficulties in understanding professional software roles, in that they may refer to individual status in the organisation or relationships with other parts of the workforce or company (Sharpe, 1998). In this sense, it is argued that:

> …software work, like other mind-work, does not readily lend itself to this sort of narrow definition of work and tidy division of people into discrete "job descriptions". Even today, categories as broad and poorly defined as coder, programmer and analyst are largely arbitrary and routinely crossed in the practice of writing a program. They are social divisions, as between manager and managed, more than technical divisions of labour and yield when the realities of the production process demand co-operative effort (Kraft, 1977: 65).

The complexity of professional software work is further emphasised by Beirne et al (1998) who suggest that professional software roles may be specialised or generalised in nature, depending on organisational focus, size or project nature. Specialisation may be considered important in some organisations, due to the recognition that individuals cannot be expected to have professional knowledge in all areas and tasks may exceed capabilities (Andrews et al, 2005). Indeed, Alvesson (2004) suggests this may lead to some individuals possessing ‘guru’ status due to the importance given to their experience, skills and knowledge. However, observations in the mid-1970s by Kraft (1977: 78) suggesting that “relations between mind-worker and detail worker are not as rigid as in other occupations” can be witnessed in more recent discussions of professional software work, with Baetjer (1998) arguing that software professionals may have a variety of responsibilities, due to difficulties in fragmenting software roles. In this sense, professional software work appears to be an extremely complex and variable process, with differing levels of specialisation and generalisation according to individual and organisational focus and variations in end products, according to individual skills, abilities and creative style (Gallivan, 2002).

However, whilst professional software roles may be heterogeneous, complex and variable in nature, the author argues it is possible to provide a general understanding of the type of roles and activities performed by software professionals. Crucially, the examination and collation of information from a variety of internet and paper-based career sources has enabled the author to develop a taxonomy as a framework for understanding professional software job roles. Taxonomy can enable researchers to classify groups and identify commonalities between members, as well as key differences (Leal and Powers, 1997). In addition, taxonomy can be applied to inform research methods and enable examination of particular phenomenon. For example, the taxonomy of professional software job roles developed by the author was used to create the ‘Activities and Codes’ list and the role descriptions on the ‘About You’ form as part of the work diary approach (see section 4.4.3), enabling exploration into the day-to-day work carried out by software professionals. Furthermore, application of the taxonomy to the work diary method facilitated the examination of specialisation, generalisation, levels of experience, variations in work experience, and relationships to work intensity.

4.3.2 Taxonomy of Professional Software Job Roles

The following section will present a classification of professional software work, with an overview of the generic title area 'Software Engineer' which the four main job roles (Systems/Business Analyst; Designer; Developer; and IT Consultant) stem from. The key characteristics of each job role are classified in terms of role description, typical working conditions, the specific tasks and activities performed and how these intersect with the development life cycle. Information has been collated from a variety of internet and paper-based career sources, namely: Careers Scotland; Computer Economics Limited and Remuneration Economics Survey, 2008; Hobson 2007 Get Science and IT; Jobs 4U Careers Database; Plan IT Plus; AGCAS Occupational Profile; Target IT 2007; and Yardley, 2004. The salary scales provided represent average rates in 2009.

Software Engineer

Software engineer can be seen as a 'catch-all' title for professional software job roles, in that this title area comprises the duties and activities encompassed in each of the four job roles. Software engineers tend to research, analyse, design, develop, test, implement, install and maintain software. The scope and breadth of software engineering activities can vary considerably, depending on company size, organisational type and project nature, with involvement in some or all stages of the life cycle.

Software engineers work with clients and colleagues in order to assess software specifications, budgets, deadlines and proposals for software and what the software should actually look like. This requires an understanding of client business requirements and existing systems, in order to develop new systems or to identify areas for enhancements. Computer code is created to make the new system or enhancements, according to design specifications and is tested to ensure the smooth running of software. Documentation, instructions and manuals need to be completed to detail the operation of the software. Training and support are other areas, in order to train clients to use the new program or system and also to support the system, dealing with any issues or problems which may arise.

As emphasised earlier, the generic title ‘software engineer’ can be split into four job roles: Systems/Business Analyst; Designer; Developer; and IT Consultant. The heterogeneous and complex nature of professional software work means that the tasks and activities associated with these four job roles may vary according to factors such as organisational type, company size, project nature and size of installation.

Systems/Business Analyst

Systems/Business Analysts work with their employing company, project leaders and clients to discuss IT requirements, in order to design and produce IT specifications for software projects. The role of an analyst can extend beyond these areas to include other activities, such as development. They may also be known as consultants, programmers or project managers. Analysts often start in more technically-oriented roles before becoming analysts and this job role tends to involve a mixture of business, technical and sales areas. They liaise with other software professionals, sales teams and clients throughout the life cycle and tend to be the point of contact between the software team and customers. This requires both interpersonal skills for client interactions and levels of technical knowledge for software team interactions. Analysts may require knowledge of software and hardware applications and need to be up-to-date with new developments in technology.

Systems/Business Analysts tend to work general office hours Monday to Friday and some overtime when deadlines are approaching. Salary scales can range from between £21,500 - £27,000 as an average starting salary, rising to £35,000 - £44,000 or above £50,000, depending on expertise and seniority. Salaries can vary according to location, company size and job demand. Work tends to be organised around project teams and analysts can either be office based or work on client sites. Travel to client sites may be a feature of this job role and some analysts may also be employed on short-term contracts.

The following diagram provides an outline of the typical tasks and activities performed by Systems/Business Analysts and how these intersect with the development life cycle:

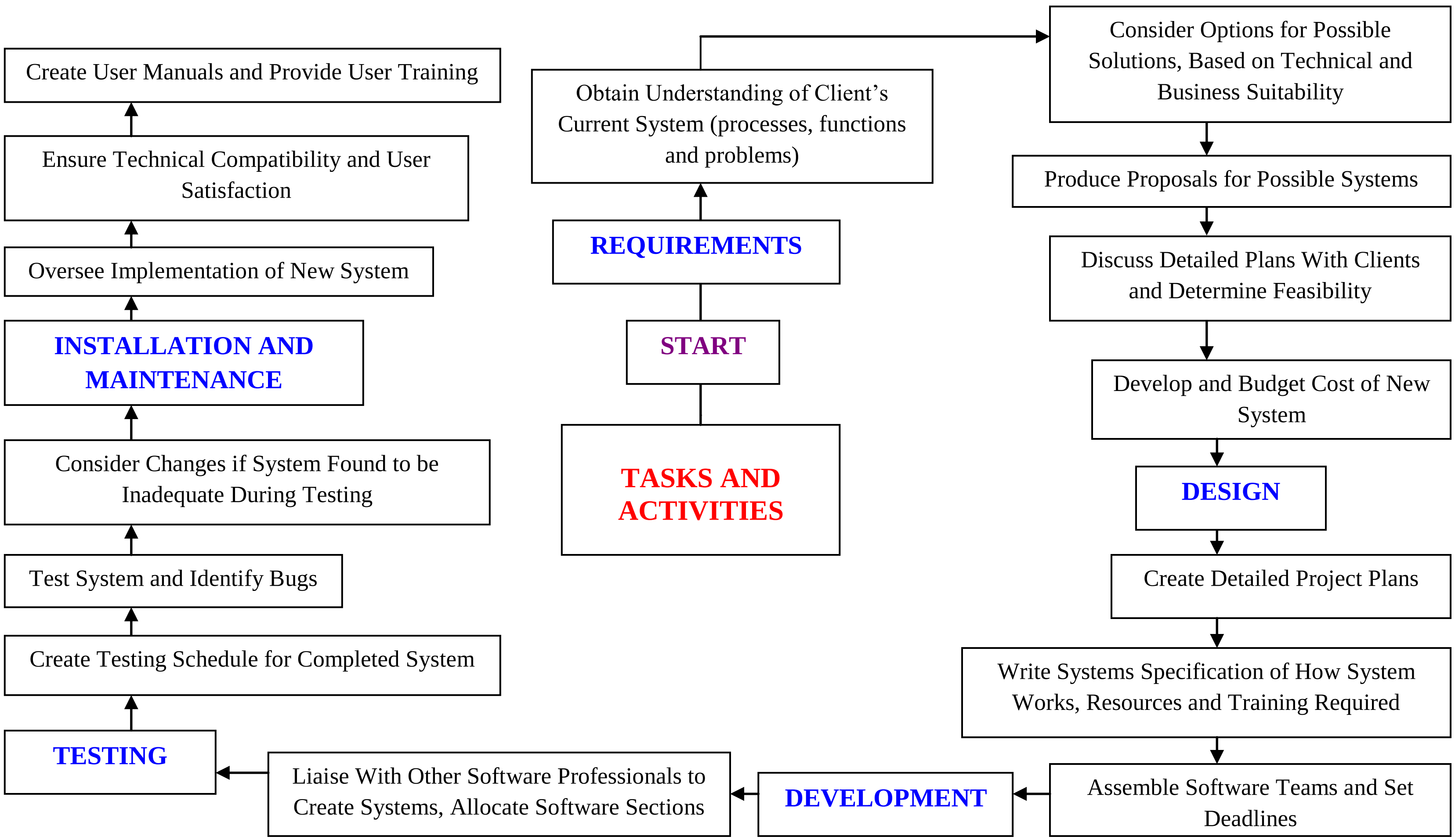


DIAGRAM 4.1 SYSTEMS/BUSINESS ANALYST

References: Careers Scotland; Hobson Get 2007 Science and IT; Jobs 4U Careers Database; Plan IT Plus; Prospects AGCAS Occupational Profile; Target IT 2007; Yardley (2004).

Designer

Designers take requirements and specifications for new systems and design them completely. Systems Designers are involved in the planning and designing of systems which integrate hardware, software and technologies. This job role requires extensive technical knowledge and research into possible technical and design approaches, as well as interpersonal skills for client and project team interactions.

Systems Designers generally work office hours Monday to Friday, with some overtime when deadlines are approaching. Salary levels tend to vary according to company size, location or job demand and can range from approximately £20,000 - £30,000 for junior designers, £29,000 - £37,000 for those with 3 - 5 years experience and between £41,000-£51,000+ for those with 10-15 years experience.

The following diagram provides an outline of the typical tasks and activities performed by Designers and how these duties intersect with the development life cycle:

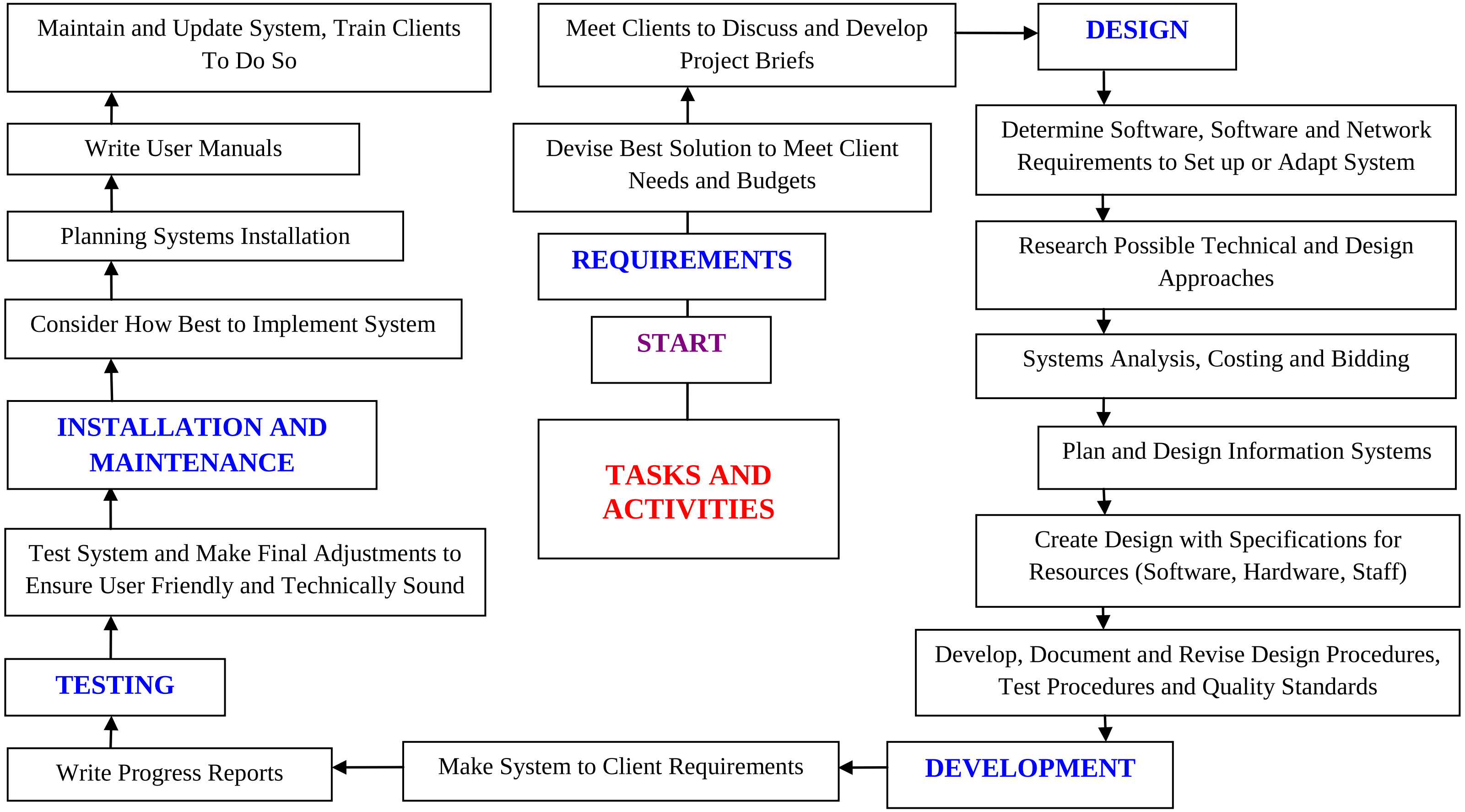


DIAGRAM 4.2 DESIGNER

References: Careers Scotland; Jobs 4U Careers Database; Plan IT Plus; Prospects AGCAS Occupational Profile; Yardley (2004)

Developer

Developers translate requirements and design specifications to make software programs. They may also write specifications, design, develop, test, implement and support software. Developers tend to work closely with analysts, designers and sales teams to discuss IT problems and requirements. They require extensive technical knowledge and interpersonal skills, due to client and project team interactions. At a junior level, Developers may write and test smaller parts, whilst more experienced Developers may write the main parts of software. They may also be called programmers, analysts or engineers.

Developers typically work traditional office hours, Monday to Friday, with some overtime when approaching deadlines. Salaries can range between £19,000 - £23,500 for junior Developers, £31,000 - £34,000 for 3-4 years experience, £40,000 - £44,000 for 10-15 years and £63,000+ depending on experience or seniority. Work tends to be structured around office-based project teams.

The following diagram provides an outline of the typical tasks and activities performed by Developers and how these duties intersect with the development life cycle:

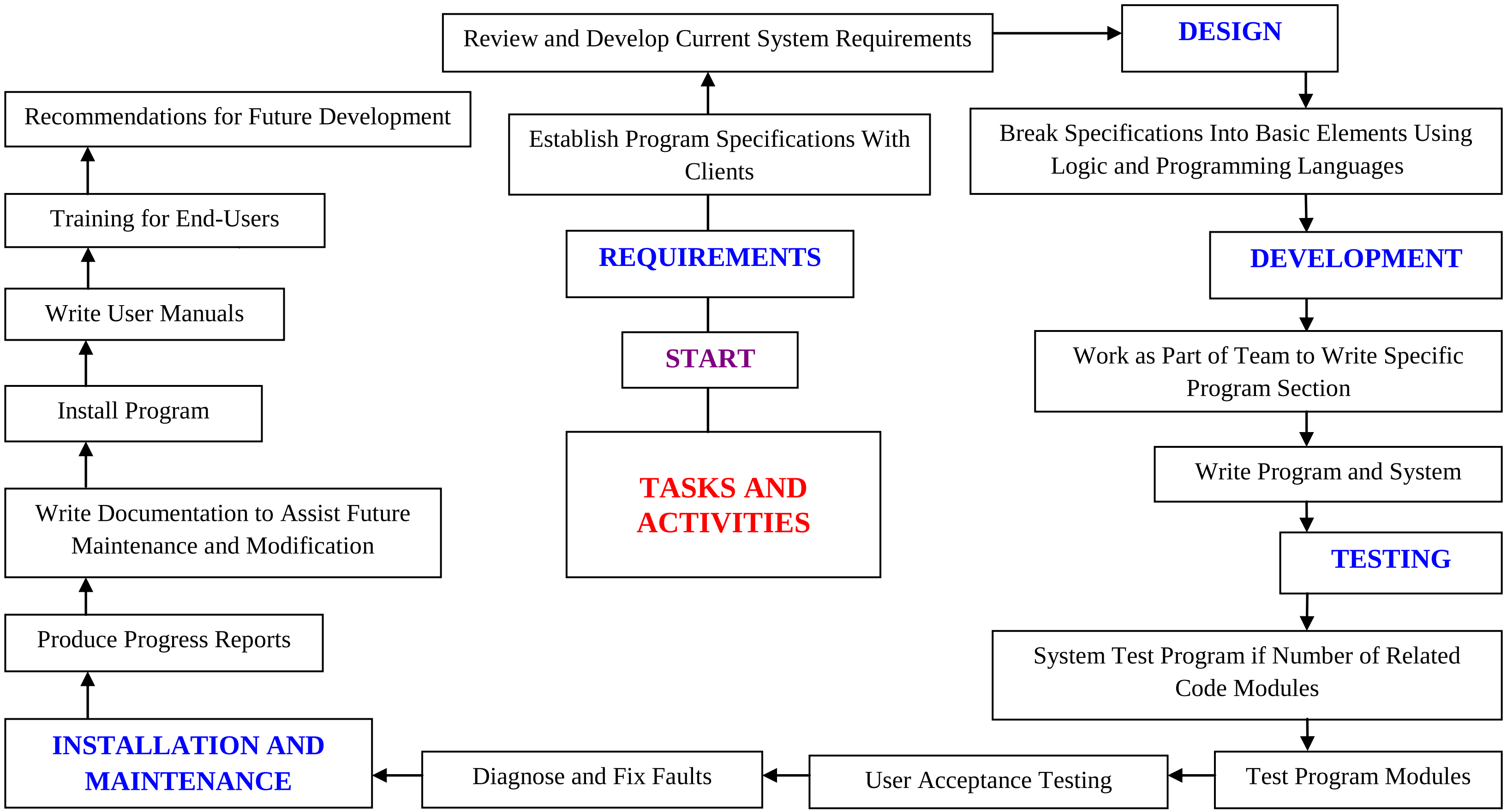


DIAGRAM 4.3 DEVELOPER

References: Jobs 4U Careers Database; Prospects AGCAS Occupational Profile; Yardley (2004).

IT Consultant

Consultants develop software and IT solutions for clients. The term 'consultant' covers those working as company-employed software professionals with 'consultant' as a job title or those self-employed as a traditional consultant. Consultants can be involved at any stage of the development life cycle, such as achieving contracts; project management; devising specifications; forming project teams to make software; design; development; testing; and support. Consultants may liaise with all levels of the client organisation and project teams, requiring extensive technical knowledge and interpersonal skills.

Consultants tend to work general office hours Monday to Friday and some overtime when deadlines are approaching. Salary scales can range from £23,000 - £32,000 as an average starting salary, to between £50,000 - £100,000, depending on expertise and experience. Work tends to be structured either around project teams or freelancing and consultants may also be based on client premises.

The following diagram provides an outline of the typical tasks and activities performed by Consultants and how these intersect with the development life cycle:

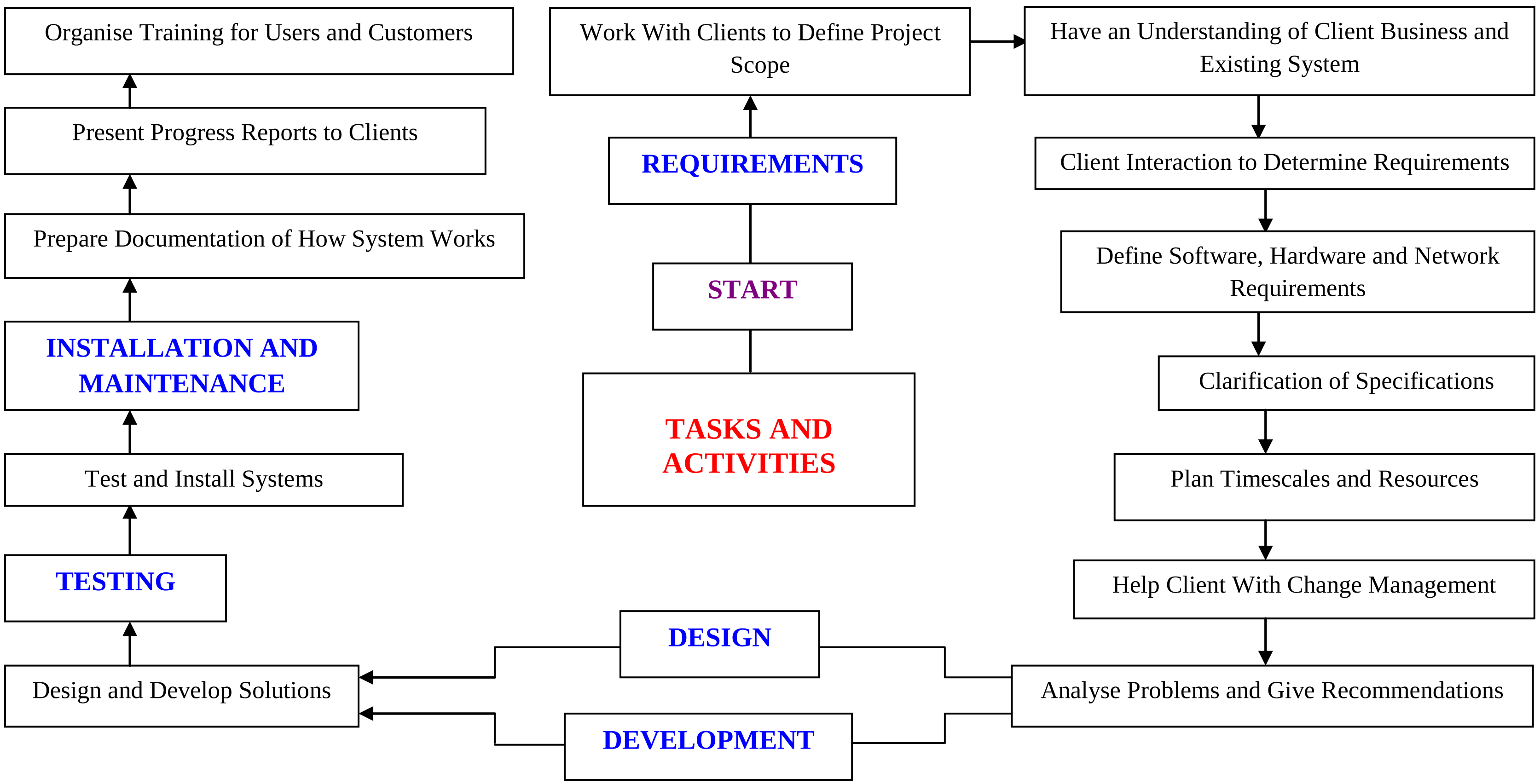


DIAGRAM 4.4 IT CONSULTANT

References: Prospects AGCAS Occupational Profile; Target IT 2007.

In summary, this taxonomy makes an original contribution by providing insight into the types of roles software professionals may perform and the detailed nature of tasks and activities. This taxonomy also raises important questions with regards to experiences of work intensity, in terms of how software professionals are affected by the development life cycle, whether stages occur sequentially or overlap in practice, levels of specialisation and generalisation and the nature of roles and tasks performed. Furthermore, the taxonomy has informed the design of the work diary method utilised within this research, permitting examination of the day-to-day activities and roles performed by software professionals and relationships to work intensity. The following sections provide further insight into the methodological approach utilised within this research with regards to research methods, case study information, sampling strategy, data recording and data analysis.

4.4 QUALITATIVE RESEARCH

Qualitative research methods are deemed to be most appropriate to achieving the aims and objectives of this research, through their emphasis on obtaining insight into the experiences and perceptions of individuals and focus on conceptualisation and theory to explain phenomenon (Bryman, 2008; Rossman and Rallis, 2003; Eisenhardt, 2002; Maxwell, 2002). Furthermore, the utilisation of multiple methods, including observation, documentation, work diaries and semi-structured interviews to explore phenomena at a variety of levels can enable the development of a holistic and in-depth understanding of work intensity:

> More interesting and a better justification for triangulation are results which are different in focus and level, which means that they complement each other or even contradict at first sight. In this case, you should not so much question the confirmability of your results; rather you should look for (theoretical) explanations on where these differences come from and what this diversity may tell you about

> your research and the issue you study. Then triangulation becomes more fruitful – as a strategy for a more comprehensive understanding and a challenge to look for more and better explanations (Flick, 2009: 450-451).

4.4.1 Observation

Observation of office surroundings, layout, activities, interactions, rituals and day-to-day events can provide valuable insight into the context surrounding the experiences of professional software workers, a descriptive framework within which to understand data and indicate areas which may be of potential interest (Rossman and Rallis, 2003). For example, as emphasised by Gillham (2008: 6):

> In that novel setting everything is going on presumably as normal: we notice differences, of course, but our main preoccupation is how things work in the social sense, what the 'rules' are: because every social setting operates in a different way.

The type of observation can vary, from actual participation through dual researcher and employee roles, observation with an overt, explicit role and negotiated access time, sporadic and interrupted involvement in terms of time in and out of the organisation, or complete observer, with no sustained interaction with those under study (Gold, as quoted in Flick, 2009; Easterby-Smith et al, 2004; Junkers, 1960). Observation can also be structured, through the monitoring and measurement of particular types of behaviour or events at set times, or unstructured and holistic, through the description of general settings, events, activities and interactions (Rossman and Rallis, 2003; Yin, 2003). Observer role and level of structure may depend on overall research purpose, access, researcher comfort and time (Easterby-Smith et al, 2004). The observational role selected for this research was overt, holistic in nature and made explicit to individuals through the provision of participant information sheets.

It is argued that specifying the purpose and type of observation can help to ensure overall comfort and understanding for participants. However, whilst some individuals may become accustomed to the presence of the researcher and behave naturally, others may potentially become influenced through observation and alter their behaviour accordingly (Flick, 2009; Bryman and Bell, 2003). In addition, whilst observation can provide valuable contextual and descriptive information, sole reliance on this research method can be of limited value, through consideration of present day-to-day experiences, interactions and activities, to the neglect of past situations. It is therefore essential to gather data through other complementary research methods, in order to provide a more comprehensive understanding of past and present work contexts, processes and experiences.

4.4.2 Documentation

Documentation can include official documents in the public domain, such as yearly reports, mission statements, advertisements, press releases, public relations material and private company documents, such as company newsletters, induction manuals, organisational charts, minutes of meetings, memos and correspondence (Flick, 2009; Bryman, 2008; Yin, 2003). Advancements in information and communication technologies (ICTs) have also encouraged the development of company internet pages for public viewing and company intranets for internal use (Bergmann and Meier, 2004). Studying public and private company documentation can provide valuable insight into organisational approach, in terms of the formal perspectives and realities presented by organisations. Indeed, it is useful to consider the context, purpose, producer, recipient and function of documentation to identify underlying themes or perspectives being promoted (Flick, 2009; Bryman, 2008). Documentation can potentially represent realities which may differ from the day-to-day, informal experiences of individuals:

> Their ‘official’ character, often a rather self-conscious one, will not display the informal realities which make up the origins of most of what one finds from current observations (Gillham, 2008: 101).

In this sense, it is advisable to obtain data through additional qualitative methods, in order to understand the true nature of reality and contexts represented by documents. For example, as summarised by Bryman (2008: 527):

> Atkinson and Coffey's central point is that documents need to be recognised for what they are – namely, texts written with distinctive purposes in mind, and not as simply reflecting reality. This means that, if the researcher wishes to employ documents as a means of understanding aspects of an organisation and its operations, it is likely that he or she will need to buttress an analysis of documents with other sources.

4.4.3 Work Diaries

The work diary method can assist in addressing two research aims: the need to gain an understanding of activities and their duration and the temporal linking of activities with particular outcomes (Gershuny, 2004). Diaries were therefore employed within this PhD research in order to obtain information at the individual level, enrich understanding of the relationship between activities and experiences of work intensity and allow for the analysis of differences between individuals and responses to phenomena (Bonke, 2005; Gershuny, 2004; Bolger et al, 2003; Conway and Briner, 2002). Crucially, diary research has been cited as being especially valuable where contextual information is considered important, making the work diary method especially appropriate to the emphasis on context within this research (Wiseman et al, 2005; Bolger et al, 2003).

The interest in studying everyday life and routines began in the 1950s and 1960s, largely motivated by the rise in behaviourism and consideration of changes in behaviour, feelings and thoughts (Wheeler and Reis, 1991). Self-recording techniques were first used by Hinrichs (1964), where participants were asked to self-record and complete a daily schedule of work activities, using alarm clocks set for designated times. Fixed-format diaries were also used by Wheeler and Nezlek

(1977) to provide recorded information on social interactions and experiences. The study of everyday life through diaries became of interest in academic psychology in the 1980s, with diaries being used to study a range of psychological issues such as mood and emotional states (King and Wilson, 1992). The work diary approach has also been used to study the relationship between activities, events and phenomena. For example, Grebner (as quoted in Klumb and Perrez, 2004) utilised the work diary method to record stressful events over a seven-day period, in order to examine the relationship between working conditions and well-being. In addition, Totterdell et al (2006) studied psychological strain through a seven-day diary method, where participants logged activities and experiences, according to work demands, job control, social support and job strain. In this sense, the work diary method can be seen to be especially applicable to this PhD research, due to the emphasis on sociological, organisational behaviour and employment relation elements within the field of study.

It is crucial that researchers have an understanding of potential work diary designs and select the most appropriate method, in order to fulfil overall research purpose and objectives (Breakwell and Wood, 2000). Diaries can take various forms, such as the traditional paper diary, encompass the use of devices or be on handheld and electronic data tools, such as handheld computers (Bolger et al, 2003). Diary studies can be split into three main categories: interval-contingent; signal-contingent; and event-contingent (Gershuny, 2004; Bolger et al, 2003; Wheeler and Reis, 1991).

Interval-contingent recording has been the most commonly used method of diary recording, requiring participants to report daily on experiences at regular, pre-determined intervals. The intervals for assessment, in terms of whether they should be fixed or variable, and overall spacing, are important considerations (Bolger et al, 2003). Intervals which are too narrow or short may place a greater burden on participants, whilst wider intervals may obscure natural cycles. It is therefore crucial that a time frame is selected which is most likely to reveal processes of research interest. Interval-contingent recording was regarded as being most applicable to this PhD research, due to its applications to studying the prevalence of certain events or phenomena and assessing everyday experience across a general time period (Wheeler and Reis, 1991).

Signal-contingent recording, which involves participants being signalled by the researcher, pagers, alarm clocks or wristwatches to provide diary entries at fixed or variable time points was not selected due to the recognition that this method could be intrusive and disruptive to participants, costly, due to the need for devices and the potential for technical problems if equipment broke down or needed replaced. It was also recognised that participants may have needed additional training for using devices (Bolger et al, 2003; Wheeler and Reis, 1991). Event-contingent recording, which requires participants to provide diary reports each time an event, fitting definitions set out by the researcher occurs, was further deemed inappropriate to this research. The predominant focus on rare phenomena and direct pin-pointing of pre-defined events within event-contingent recording was considered to be incongruous with the exploratory and inductive dimensions to this research (Bolger et al, 2003).

Initial inspiration for design of the work diary packs utilised within this research came from the Future of Work project (see Baldry et al, 2007), which used simplified work diaries as part of the research process. Work diaries utilised within this research covered a seven-day, twenty-four hour period for any work activities performed. A continuous time log and diary-related questions were deemed to be the most useful at eliciting rich data, through providing information on the type and number of activities individuals were engaged in and surrounding context, with regards to job description, tenure and employment status (Gershuny, 2004). Each diary pack (see Appendices 3 to 7) contained a consent form, detailed instructions for completing the diary, an 'Activities and Codes' list, the diary itself and an 'About You' page to provide contextual information surrounding data. The 'Activities and Codes' list contained a list of activities with a corresponding code for participants to place in a relevant timeslot. Crucially, the taxonomy on professional software job roles (see Section 4.3) provided valuable insight into the tasks and activities pertaining to professional software work and was instrumental in the development of the 'Activities and Codes' list. Informal discussions and observations during fieldwork provided information on extra activities that software professionals were engaged in, enabling their inclusion in the ‘Activities and Codes’ list. The diary also contained two other boxes, ‘Intensity of Day’ and ‘Main Causes of Intensity’. Participants were asked to use a scale of 1 to 4 (1 being ‘Not at all Intense’ and 4 being ‘Very Intense’) and place the relevant number in the ‘Intensity of Day’ box daily to show how intense they found the day. Participants were advised that, each day, if they experienced a particular activity or activities as causing feelings of intensity (fitting the definition and descriptions provided), they should place the relevant code(s) for these activity/activities in the ‘Main Causes of Intensity’ box. On the ‘About You’ page, participants were asked to provide information on their current job title, select a job description from a choice of five which best fitted their role, indicate their length of tenure, gender, employment status, how typical the week was and events which may have impacted on work that week. The diaries were therefore designed to measure more than one activity and address factors in different ways.

The work diaries were pilot tested on two professional software workers secured through personal contacts, allowing for small modifications, such as the inclusion of additional activities and increased box space for noting codes. During the work diary periods in both companies, new items were periodically added to the ‘Activities and Codes’ list as fieldwork demonstrated the need to add new codes. This was to ensure that data being recorded in the diaries consistently reflected the work being performed. All diary participants were notified of additions as they occurred to ensure they were kept up-to-date.

Diaries can be advantageous, as individuals may be familiar with what a diary entails and if provided with appropriate instructions and guidance, can generate data without the researcher being present (Breakwell and Wood, 2000). Diaries may also provide an enhanced overview of general working activities and conditions, with the inclusion of activities such as breaks, telephone calls, e-mails, meetings and discussions with colleagues (Bonke, 2005). However, keeping a work diary can be time-consuming and therefore requires a level of commitment on the part of participants. If diaries are complicated and overly time-consuming to complete, participants may become fatigued with record keeping and consequently become less thorough in reporting (Wheeler and Reis, 1991; Verbrugge, 1980). It is therefore essential that diaries are designed to be simple to follow and complete. Furthermore, clear, concise instructions and guidelines, as well as pre-printed dates and times for responses may also help guide participants in diary completion.

Researcher presence and contact during the diary recording process is an important factor in the accurate and continual completion of diaries. Participants may make entries based on their own interpretation of instructions, omit relevant information or even be discouraged from participating fully, without researcher contact and support (Mariño et al, 1999; Verbrugge, 1980). If the researcher has to contact respondents to clarify information, the data may then become subject to retrospective bias (Verbrugge, 1980). It should therefore be emphasised that participants make entries at the time of activities, rather than retrospectively.

During the work diary process, participants may be prone to reporting activities or events which are deemed acceptable, omitting those which are regarded as undesirable and subjecting the process to elements of social desirability bias (Breakwell and Wood, 2000). Participants may also report the activities regarded as important as being longer than they are or omit recording of brief activities (Higgins et al, 1985). Individuals who participate may be more stable, less stressed and anxious than those who do not take part (Waite et al, 1998). These issues can be addressed through the utilisation of other research methods, such as observation and interviews to validate data.

Habituation and enhanced understanding of phenomena are all possible effects which can potentially occur during diary data collection (Bolger et al, 2003; Wheeler and Reis, 1991). Self-reporting requires introspection and monitoring of daily working patterns at a level which participants may not be accustomed to, leading to greater awareness of the phenomena being measured and possible reactance. Habituation may also occur, whereby participants skim sections or omit responses at relevant times, due to familiarity with the diary process (Bolger et al, 2003). Repeated diary exposure can also potentially result in participant entrainment with regards to the phenomena under study. Thus, participants may develop a more complex understanding of the phenomena than necessary or alter their conceptualisation to fit with those measured in the diary (Bolger et al, 2003). To counteract these issues, diaries can be designed to measure more than one activity or address factors in different ways, making it more difficult for participants to alter their behaviour.

4.4.4 Semi-Structured Interviews

Interviews can be defined as "…conversations with a purpose" (Burgess, as quoted in Mason, 1996: 39) and are considered to be beneficial when insight into the knowledge, perspectives and experiences of individuals is necessary to increase understanding of phenomena (Rossman and Rallis, 2003; Mason, 1996). It is important to consider philosophical approach, overall purpose and research questions, in order to identify the most appropriate type of interview (Hopf, 2004). Semi-structured interviews were seen to be compatible with the structured yet exploratory approach taken in this research, through the application of an interview guide comprised of questions and themes, combined with the ability to vary question sequence and ask follow-up questions (Bryman and Bell, 2003). Indeed, the semi-structured approach can be seen to aid the investigation of pre-defined themes, whilst also providing insight into participant perspectives and uncovering unanticipated areas of interest (Easterby-Smith et al, 2004).

A purely structured approach, with standardised questions, themes and set sequencing of questions was discounted due to its predominant focus on pre-defined areas, hypothesis testing and establishment of relationships between variables at the expense of exploratory work and detailed understandings (Bryman and Bell, 2003; Miller and Glassner, 1997). Unstructured interviews were also rejected on the grounds that the absence of any structure would make it difficult to adequately explore pertinent themes and ultimately answer established research questions (Hopf, 2004; Yin, 2003).

Semi-structured interview scripts were structured around categories and sub-categories originating from the determinants of work intensity developed within this research, including work organisation, culture and leadership style, teamwork, individual factors and contextual factors. Three groups were identified as being necessary for interview, in order to provide full insight into professional software workers' experiences of work intensity: individual software professionals, project managers and senior managers/directors. Software professionals were selected due to their status as the focus for this PhD research and the need to obtain insight into individual experiences and perspectives on work intensity. Interviews with project managers were perceived to be important, through allowing experiences of work intensity amongst software professionals to be examined from an alternative perspective and to compare how this related to responses from software professionals themselves. It was also recognised that project managers may have had initial experience as software engineers before entering management grades, providing additional insight into software workers' experiences of work intensity over time. Senior managers and directors were interviewed in order to consider perspectives on work intensity from higher management levels and how this compared to the experiences of software professionals.

Interview scripts were piloted on one professional software worker secured through personal contacts in order to determine clarity of definitions, applicability, comprehensiveness, sequencing of questions and time frame. Face-to-face semi-structured interviews (see Appendices 8 and 10) lasting between one and a half hours to two hours were conducted with individual software professionals and project managers in private meeting rooms on company premises. Paper-based semi-structured interview scripts (see Appendix 9) were also e-mailed to three Indian workers from InSoft for completion as a result of differences in location and time zones. Face-to-face semi-structured interviews with senior managers and directors lasted between thirty and forty-five minutes to take account of reduced time frames given for speaking to these individuals. A condensed semi-structured interview script, based around areas which had been identified by software professionals and project managers as influencing experiences of work intensity, was utilised with senior managers/directors (see Appendix 11).

It is argued that interviewers require levels of skill and awareness when conducting interviews. For example, ability to make 'real time' decisions on content, sequencing of interview questions and follow-up questions can be seen to have an influence on the effectiveness of interviews:

> The intellectual task is to try to assess, on the spot, the relevance of each part of the interaction to your research questions, or to 'what you really want to know'. Although you are likely to have some sort of *aide-memoire* to remind you about the topics and issues you are interested in, you nevertheless need to be able to make connections between relevant issues quickly, and to spot and follow up issues which may be relevant, but which you had not anticipated (Mason, 1996: 45).

Interviewers should also avoid imposing their own frame of reference onto the encounter and instead facilitate the perspective of interviewees through techniques such as 'probing' (Easterby-Smith et al, 2004; Holstein and Gubrium, 1997). However, it can be deemed appropriate to ask for clarifications, in order to ensure correct interpretation of information provided by interviewees. Furthermore, it is argued that respondents may potentially have agendas which may be unrelated to the research or may attempt to present inaccurate information as a result of social desirability, meaning that clarifications may also serve to determine credibility of information (Flick, 2009; Yin, 2003).

Awareness of and sensitivity to emotional, interpersonal and political dimensions can also be seen to be important when managing interview interactions. For example, identification of characteristics such as work roles, seniority and power relationships can inform understanding of how individuals may interact or behave and allow for interactions to be structured accordingly (Baker, 1997). Researcher knowledge on company background, cultural inferences, work activities and jargon can also be seen to be crucial when conducting interviews, in order to fully make sense of information and assist in the eliciting of rich data (Rossman and Rallis, 2003; Baker, 1997). Furthermore, development of knowledge in these areas may be crucial to obtaining acceptance and co-operation from participants (Hermanns, 2004).

4.5 EVALUATING PERCEPTIONS OF WORK INTENSITY

Existing studies on work intensity have tended to measure physical as opposed to mental effort. This shortfall can be attributed to the difficulty in measuring mental effort consistently and reliably, as a result of its intangible nature. For example, it can be difficult to measure and evaluate mental effort associated with professional software work, in that whilst workers may appear to have spent a significant amount of time on one task, this may require a significant amount of mental effort which may not be easily observable or measurable. Green (2006) suggests that the problem of measuring and evaluating mental work effort can be addressed through establishing social norms based on people's perceptions of the work effort. This can be validated through gaining insight from other peers and influential groups, such as colleagues and project managers. Self-reporting measures and perceptions can be utilised by asking individuals to consider present and past work experiences, in order to provide valuable insight into work intensity. Subjective estimates of intensity can also be captured through the application of questions relating to required effort (how the job itself affects individuals) and discretionary effort (Bielby and Bielby, 1998). It is essential when asking for subjective estimates to clarify concepts such as work intensity, work intensification and work effort to ensure a common understanding of concepts amongst respondents. For example, Green (2006) suggests that respondents may perceive work intensity to relate to longer working hours, rather than the effort expended during the time working, without adequate definitions being supplied. Effort norms may also change over time, meaning workers may become accustomed to new norms, encouraging potentially different responses to effort questions without definitions on effort and work intensity (Green, 2006).

Other researchers have created frameworks and scales with possible variables influencing experiences of intensity. For example, Penn et al's (1994) Effort Pressure Sources Index, consisting of eight forms of pressure (machine/assembly line; clients/customers; supervisors/managers; colleagues; individual discretion; pay incentives; reports and appraisals; self-set targets) provides a useful framework for exploring factors contributing to intensity. Fiksenbaum et al (2010) present a 15 item scale for work intensity, with questions on a variety of areas including work characteristics (tenure, organisational level, organisational size); job-related areas (flow of work, scope of responsibility, work hours); personal characteristics (age, gender, educational level, marital and parental status); and outcome variables (such as work engagement, well-being, work-life conflict, psychosomatic symptoms and life symptoms).

Boisard et al (2008), through analysis of the results of the Third European Survey on Working Conditions, also identify a variety of organisational, job-related and personal variables for consideration when studying work intensity. These include hierarchical structure (relating to direct control by managers or supervisors); horizontal structure (relating to work done by other colleagues); demand constraints (requests from customers and users); automatic constraints (speed of machines or product movement); targets; procedural autonomy (control over order of tasks, work methods and periods of working); temporal autonomy (control over breaks, holidays and working hours); intensity of cognitive elements of the job (complex problems, learning and solving unanticipated problems); degree of social support at work; occupation; employment status (whether full-time, part-time, contracting or temporary); and demographics (sex, age, gender).

Critical evaluation and review of these frameworks allowed for the selection of elements which were most applicable to this research. Crucially, the data gathered from the work diaries carried out within this research, coupled with frameworks suggested by Penn et al (1994), Boisard et al (2008) and Fiksenbaum et al (2010) enabled professional software work to be broken down into constituent activities and possible determinants of work intensity, resulting in an index of possible factors (see

Appendix 13). The data attained through the work diaries and utilised within this index represents an original contribution, through allowing dimensions to be considered that existing frameworks have not acknowledged. Furthermore, the work diaries have provided insight into the day-to-day work carried out by individuals and intensity of work in relation to activities performed. Possible determinants of work intensity were split into categories and sub-categories under the headings of Organisation of Work; Culture and Leadership Style; Team Work; Individual Factors; and Contextual Factors, which allowed for a full evaluation of perceptions of work intensity. This index was also used to inform interview questions.

4.6 SPECSOFT: SPECIALIST SOFTWARE FIRM

Company Overview

SpecSoft was a Glasgow-based, small-medium sized specialist software firm. The firm was established in 1988 and in 2008 employed 140 individuals, with the main office in Glasgow and three offices in the United Kingdom. The firm focused on niche markets based on industry sectors at the time of research and was split into five divisions: energy and utilities; telecommunications; oil and gas; travel and transport; and the public sector. The firm had a variety of clients from each of these sectors and delivered a range of software solutions, including programme management, consultancy services, applications development, training and support. In 2006, the firm was bought by a large global company but was still able to operate largely independently. The divisions, following the commencement of fieldwork, were restructured into business units, along with the amalgamation of the energy and utilities and telecommunications divisions. At the time of research, the human resource department was being restructured, due to the company take-over and had begun to focus largely on recruitment. It should be noted that research at SpecSoft and InSoft was conducted between 2007 and 2009, prior to the current recession. Experiences documented by participants therefore related to events which occurred during and before the research period, such as ‘September the 11th’ and the ‘Year 2000 Bug’.

Overall access was granted by the managing director and field research was carried out in two project teams at the firm. These two project teams were proposed by the human resource manager as being representative of the short and long-term types of projects being carried out at the firm, Project Team One being engaged in a long-term project and Project Team Two being involved in short-term projects. An understanding of the building layouts, working environments, interactions and activities in SpecSoft was generated through regular presence in the office. Project managers of both teams allowed three full days per week to be spent for as long as necessary with each project team to observe and interact with individuals while they carried out their day-to-day work activities. During this time, a desk and PC were provided on the office floor alongside the particular project team being followed. Researcher access was also granted by the project manager of Project Team One to attend progress meetings and Technical Lead meetings. In Project Team Two, informal meetings on the normal office floor were observed. However, the researcher was not invited to attend formal meetings which occurred, such as the weekly meeting to keep division members updated on progress.

PROJECT TEAM ONE

Project Overview

Project Team One, comprised of fourteen team members (excluding one development Technical Lead who had recently departed and one team member located in Aberdeen) and one project manager, was engaged in a large-scale project in the travel and transport sector. The overall project had been running for seven to eight years and dealt with forecasting person flows and demand conversions, that is, how many members of staff would be required and at what times for work shifts, trolley movements and vehicle locations. The main project had been accepted, with a number of related projects running at the time of research, based on the same code.

One related project (Project A) involved functional enhancements to the original project and had been running for the past year, whilst another related project (Project B), involving support, was on-going. Project A was broken up into phased sections (Releases), in order to make it more manageable. Consequently, customers were able to receive information fairly regularly and view the work in stages, rather than waiting until project completion for modifications. The project was split into four releases following the development life cycle. Different releases and life cycle stages occurred simultaneously and overlapped. For example, when Design on Release 1 was completed, it moved to Release 1 Development. Part-way through Release 1, Release 2 Design then commenced. At the time of fieldwork, Releases 1 and 2 were complete, Release 2 went live shortly after the completion of field research and Release 3 Design and Development had been completed. Release 3 Testing and Release 4 Design on-going. The following diagram provided by the project manager illustrates the Releases and the overlapping of stages:

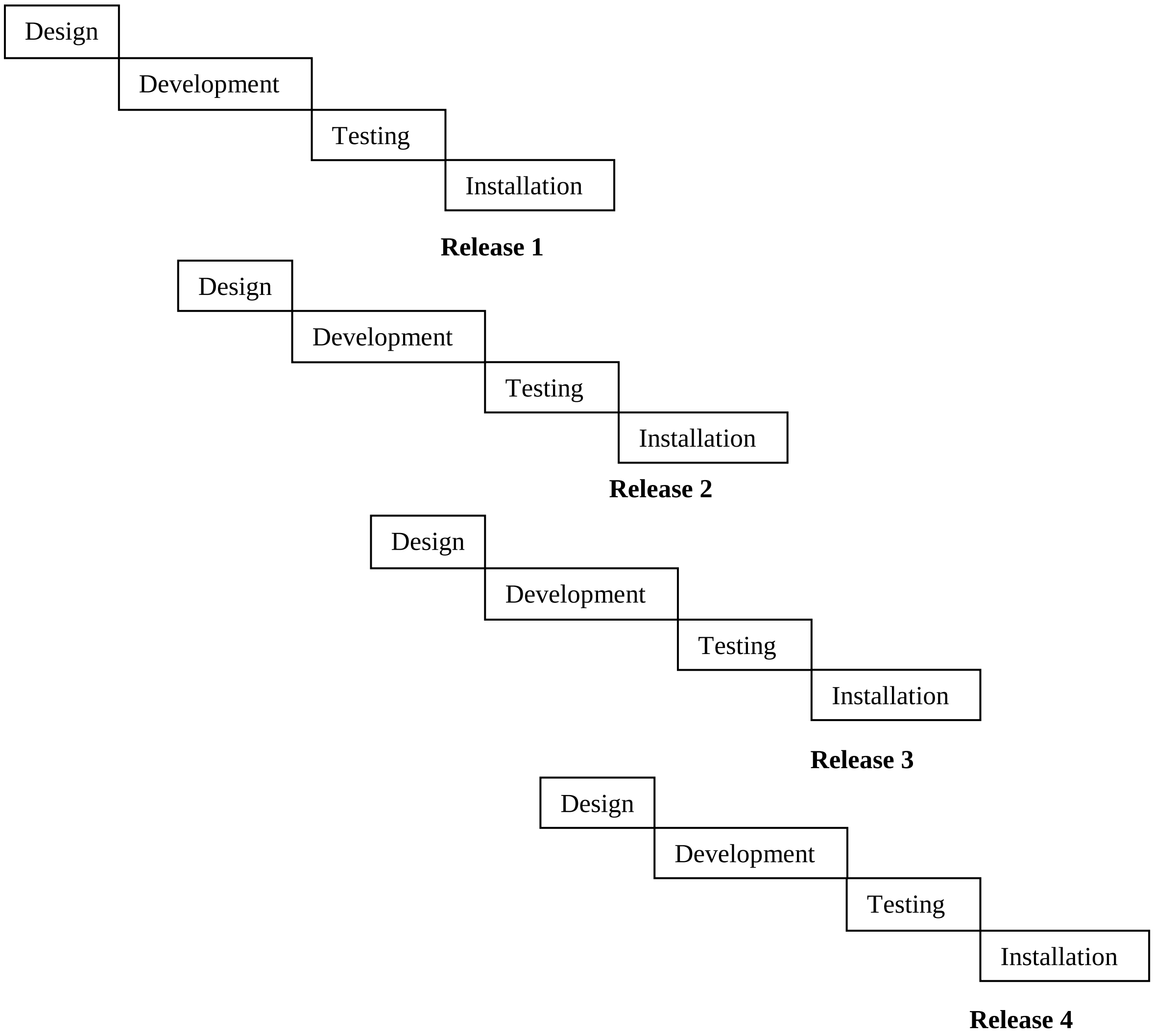


DIAGRAM 4.5 SPECSOFT PROJECT TEAM ONE PROJECT RELEASES

Project Team Structure

The project had an overall project manager and five Technical Leads (one Design Technical Lead, three Development Technical Leads and one Test Technical Lead), who reported to the project manager. Team members in Design, Development and Testing reported to the appropriate Technical Lead. The project manager was responsible for reporting to the client, senior management and managing the project team. Technical Leads were tasked by the project manager to divide and allocate tasks to themselves and team members. They were responsible for their team, in terms of monitoring the progress of work, resolving problems, checking the quality of work, ensuring team members met targets and timescales and reporting this information to the project manager. Reporting took the form of weekly Technical Lead meetings with the project manager and the submission of progress reports. The following diagram depicts the team structure for this project team, based on information provided by the project manager:

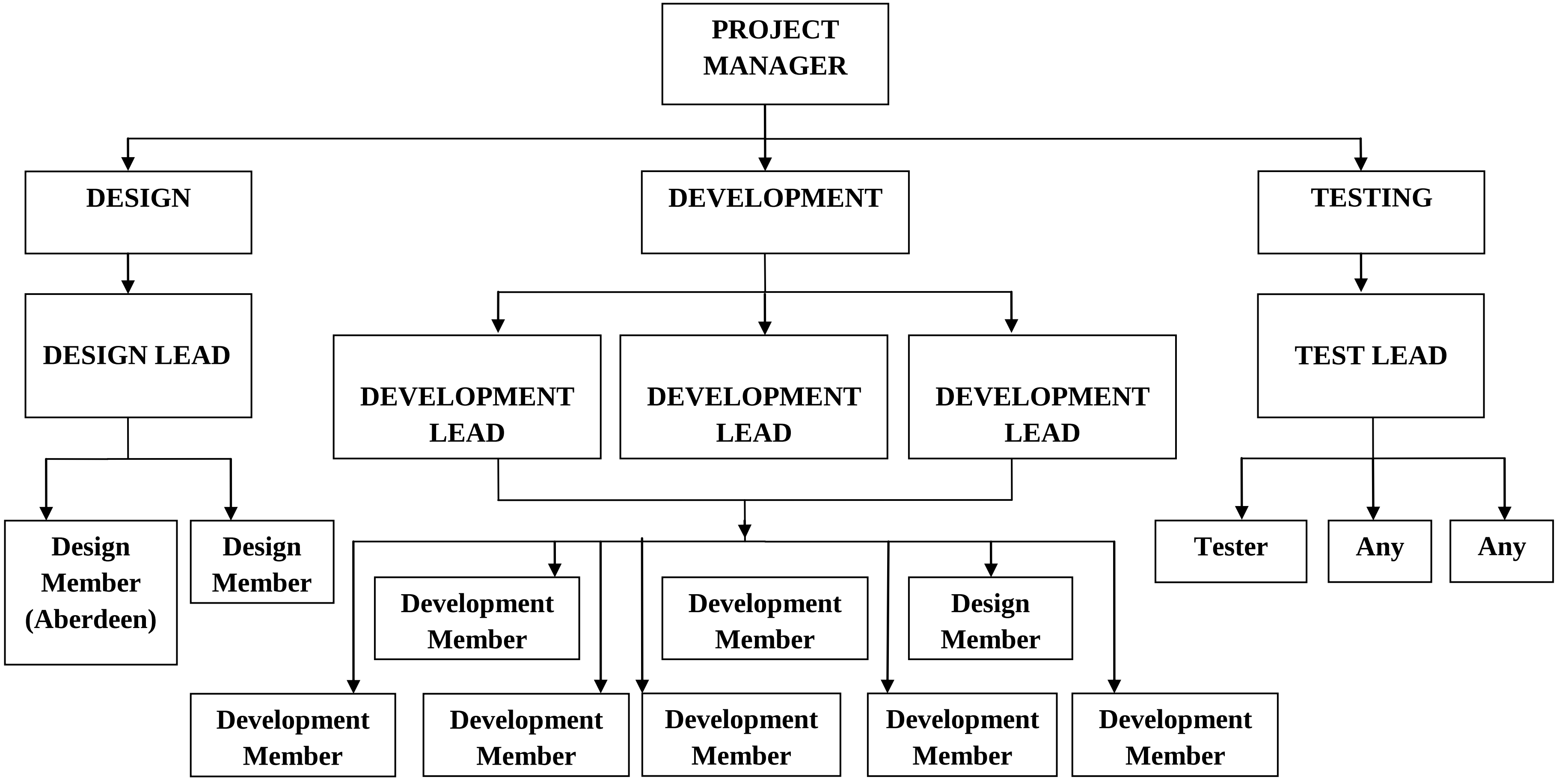


DIAGRAM 4.6 SPECSOFT PROJECT TEAM ONE STRUCTURE

Project Documentation

The project manager in Project Team One allowed extensive access to project documentation which provided valuable insight into project nature, structure and the roles and responsibilities of project teams and clients. Project charters included a detailed breakdown of sub-areas in each development life cycle stage for every release. Start and finish dates for each development life cycle stage per release were clearly outlined. In addition, project charters outlined additional tasks relating to the development life cycle such as training, user guides, documentation updates and provided overall costs of hours, resourcing and manpower costs for the customer. Project plans provided detailed information with regards to project aims, objectives, tasks, timescales and monitoring tools. These project plans also outlined project team member skills and the roles and responsibilities of project teams and clients. Documentation and information accessed via the company intranet and public company internet pages provided general insight into the travel and transport division as a whole.

PROJECT TEAM TWO

Project Overview

Project Team Two, comprised of fifteen individuals and one overall project manager, was engaged in various projects in the telecommunications sector. Ten of these fifteen were included in fieldwork. Five individuals were not included: two who commenced employment during the fieldwork; one who worked in another division during the fieldwork; and two who worked in project management functions, rather than software development. During fieldwork, two individuals from the energy and utilities division also worked on projects in the telecommunications division as a result of the recent amalgamation of the two divisions and are included in the ten mentioned above.

During the research period, the telecommunications division had twelve different projects running for three main clients. Project duration was highly variable, with projects lasting anything from a few weeks, to several months, or even a couple of years. Individuals could be working on a number of projects simultaneously, whilst project team size and composition was dependent on project nature, skills and knowledge and member availability. The projects largely followed the development life cycle but teams were trying to move to a more iterative approach, which would involve breaking a project down into staggered phases. It was hoped that this iterative approach would assist individuals in managing their workload and work effort to meet smaller regular deadlines, rather than one major customer deadline. This approach was also emphasised as providing clients with regular information on project work and allowing for frequent assessment of necessary changes, rather than being conducted on project completion.

The project manager in this team stated that telecommunications should be regarded as an extremely fast-paced industry as a result of regulatory changes and the need to constantly keep up with new technology. One consequence was that clients had to make constant changes to products and had to react quickly to be competitive, causing difficulties with planning their workload. This meant that the project team could be managing several projects simultaneously and constantly adjusting work priorities.

Project Team Structure

The telecommunications division was split into three main areas: Software, Consulting and Service Management (support), with an overall project manager presiding over all three areas. The project manager was also involved in the Consulting section as a domain consultant with industry-specific knowledge. Individuals could also be performing several roles across these three main areas. The Software section dealt mainly with software creation, whilst the Consulting section focused on traditional consulting areas of management, delivery, analysis and specialisms. The Service Management section was essentially the dedicated support function for clients. Due to the variability in project size and length, individuals tended to report directly to the project manager. Projects had designated Technical Leads acting as the main technical authority and with responsibility for monitoring work tasks, targets and checking work quality. The individual performing the Technical Lead function could vary from project to project, depending on project nature, technical knowledge and expertise. Due to project variability, meetings tended to be informal discussions between team members. A weekly progress meeting also kept division members updated on projects and progress. The following diagram depicts the structure of this division, based on documentation and team member information provided by the project manager:

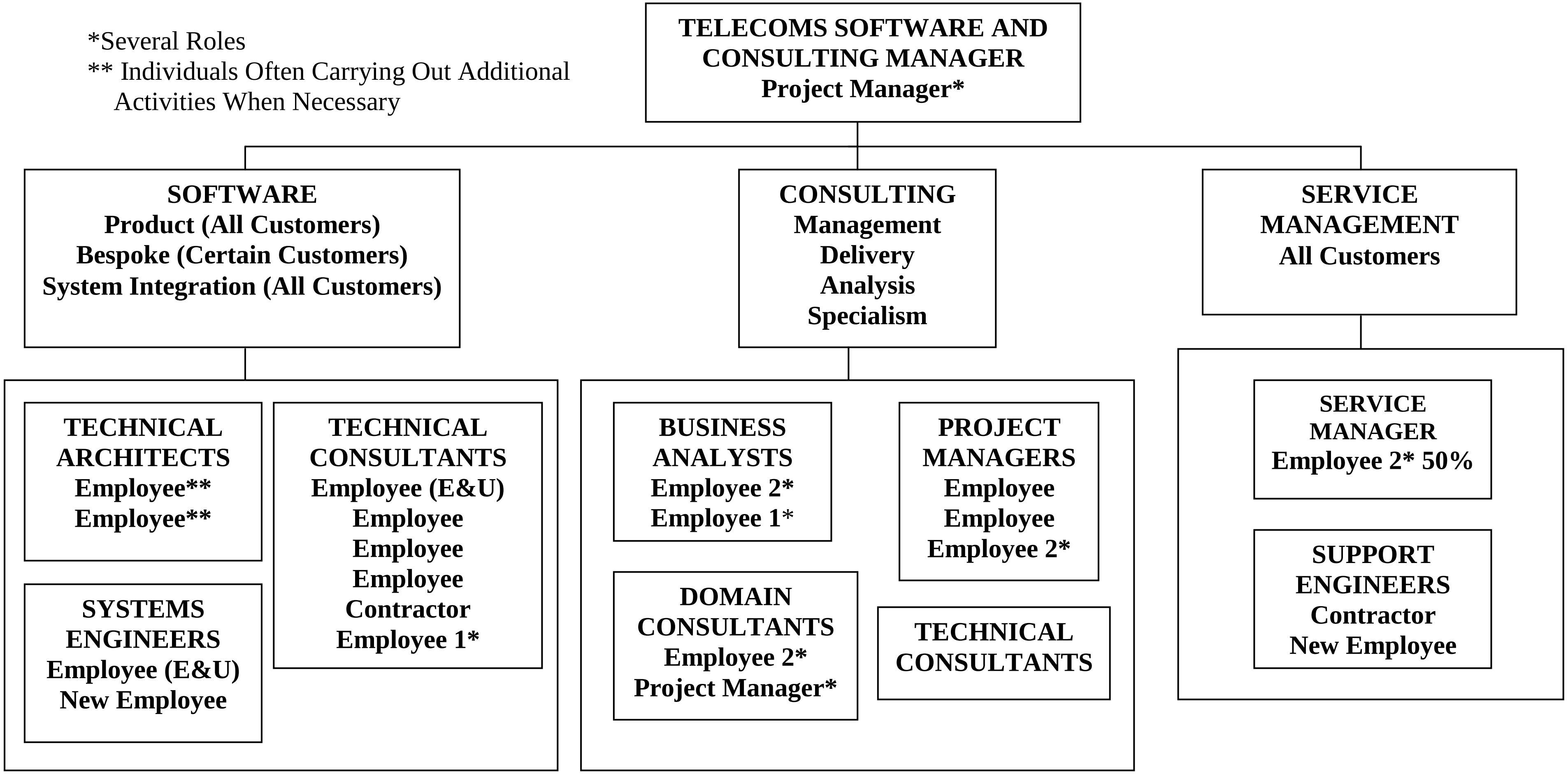


DIAGRAM 4.7 SPECSOFT PROJECT TEAM TWO STRUCTURE

Project Documentation

Project documentation available in Project Team Two was constrained to an extent, in that the project manager provided the researcher with selected documents to analyse. Project documentation included information on the main services offered by the telecommunication division, areas of expertise and information on rates and chargeability to clients for different levels of team members. Documentation also placed emphasis on the structured, short interval life cycle methodologies utilised by the division. Support documentation provided insight into the support and maintenance services available to clients and how these services operated. Documentation and information accessed via the company intranet and public company internet pages offered additional general insight into the telecommunications division.

4.7 INSOFT: IN-HOUSE IT DEPARTMENT IN A LARGE FIRM

Company Overview

InSoft was a leading provider of communication solutions and services and operated world-wide, with services including networked IT services, telecoms services, broadband and internet products and services. The company was split into six business lines, one of which covered the in-house IT software section. Two internal divisions existed within the in-house software section, one responsible for the design and delivery of technology and the other for the running of systems. The main functions of the in-house software section included: Business Analysis and Requirements Gathering; services to drive design and development; Architecture; Security; and the customisation of packaged software. The client base covered mainly the internal business as well as some external customers. The in-house IT section had originally been separate from the rest of the company and had operated as a largely autonomous software centre, with offices in Glasgow and the rest of the United Kingdom. These software centres had competed for company projects, largely as a software house would compete for external contracts. However, as a result of restructuring, these software centres moved to the central Glasgow office in 1998 to amalgamate with the rest of the company.

Access to the general office floor was InSoft was heavily constrained due to bureaucracy and confidentiality procedures. On the first day of field research, access was granted to spend two days based in a meeting room to carry out informal discussions with individuals, in order to secure initial contacts. After these two days, the company did not permit the researcher to have a continual presence in the office, apart from entering the in-house software floor for discussions and meetings which had been set up with individuals. Observations of layout, working environment, interactions and activities at InSoft therefore occurred when entering the building and office floor to attend these appointments.

Project Overviews

Project teams at InSoft were dispersed geographically across the United Kingdom and in India, with only some team members located in the central Glasgow office. The dispersal of team members across sites made it initially difficult to identify prospective participants and quantify the actual number of Glasgow-based team members. The method for gathering participants therefore differed between the two case study companies, as a result of differences in access permissibility and organisational structure. Access limitations and the geographical dispersal of team members at InSoft meant that initial contacts provided recommendations for other individuals who could be contacted, resulting in a 'snowball' approach. This contrasted to the approach taken at SpecSoft, where the open nature of access and positioning within two on-site project teams provided the opportunity to talk directly to individuals on a team-wide basis. Individuals working in Business Analysis and Requirements Gathering, Performance and Architecture, Development, Creating Tool Aids, Delivery Management and Application Support were secured, as well as two project managers.

Access to Indian offshore workers was especially challenging, with one InSoft manager taking responsibility for e-mailing work diary packs and interview scripts to Indian workers, removing the ability for the researcher to contact potential participants directly in the first instance. Three Indian offshore workers engaged in software engineering (Business Analysis, Development and Testing respectively) were secured, in order to provide a complementary perspective to that of UK-based software engineers at InSoft. One participant attempted to secure additional Indian offshore participants but was unsuccessful. This individual explained that this was a consequence of individuals having to account for time spent across the working day and any activities deemed non-productive being subject to scrutiny.

The majority of projects at InSoft were on-going and long-term in nature. Each development life cycle section (Analysis, Design, Development and Testing) typically had its own team, with areas and roles further embedded in these. In addition, the work from one project was essentially only one amongst a wide range of projects which fed into the overall in-house IT function, as the following diagram (sketched by one project manager) demonstrates:

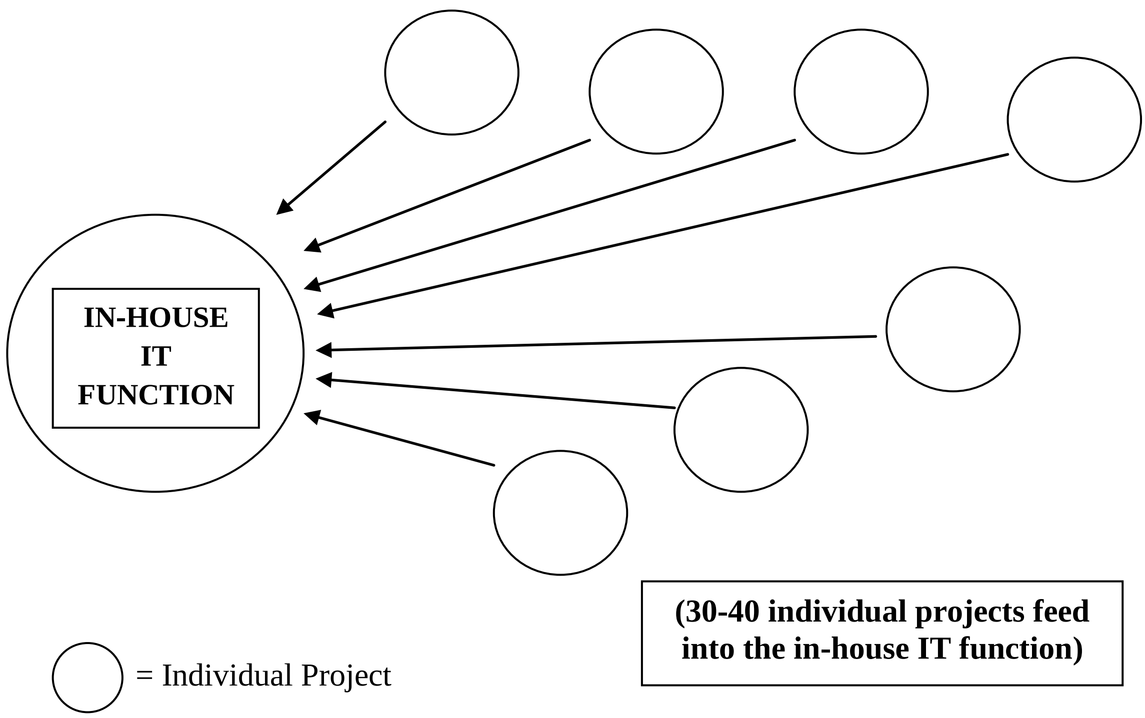
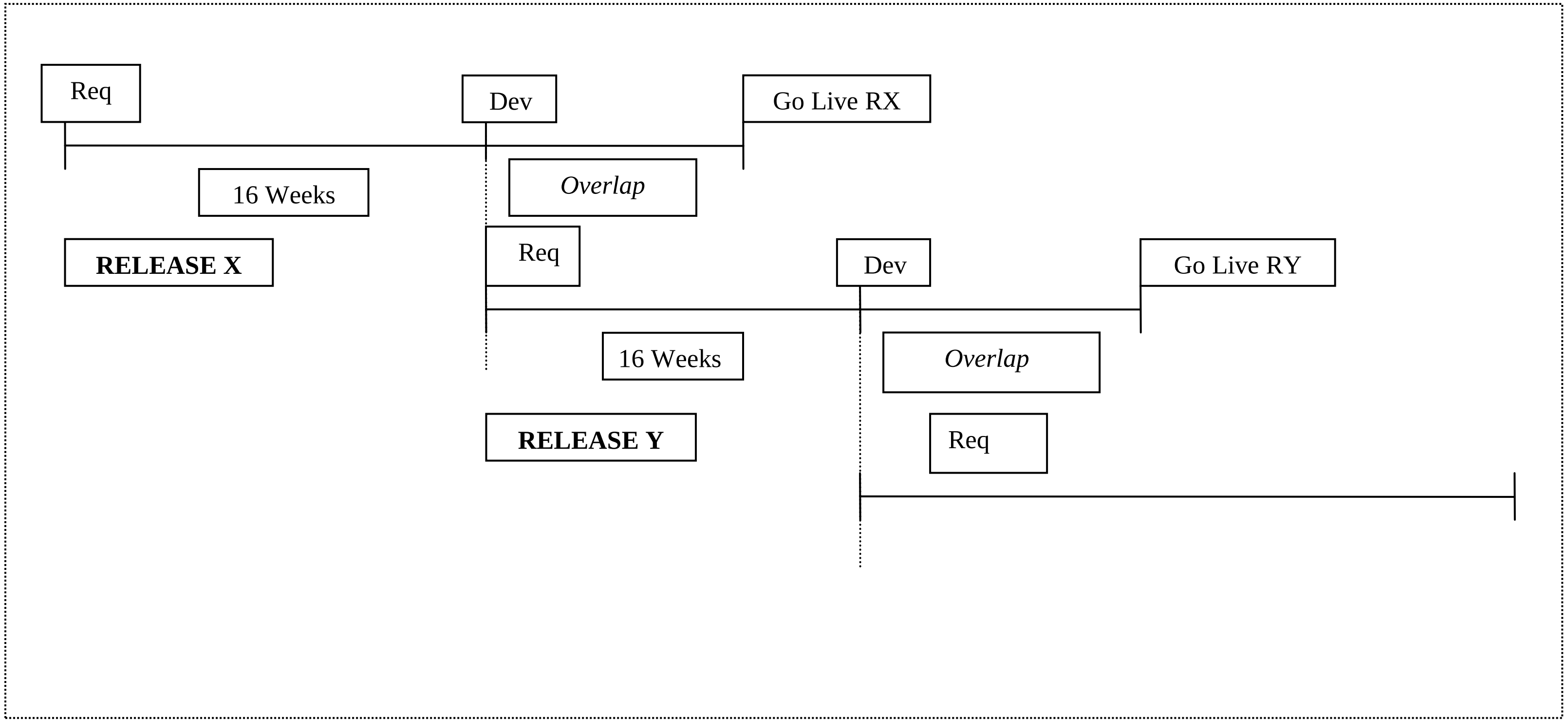


DIAGRAM 4.8 OVERVIEW OF INSOFT IN-HOUSE IT FUNCTION

The majority of the projects at InSoft were structured around the ‘Agile’ methodology (as discussed in Chapter Two), with phased, overlapping releases and regular feedback to clients. The ‘Agile’ methodology had been adopted by the company in order to release products and services into the market more quickly and to reduce the time spent resolving faults. It was hoped that ‘Agile’ would help to improve customer feedback and action, enable more flexibility in responding to changing business needs and enhance customer experiences. As stated in company information:

> (‘Agile’) It offers customers innovative products and services in much shorter timescales, functionality they are really looking for, a chance for them to get involved and have an influence over what’s being developed and overall much greater satisfaction from choosing and using XXXX.

The Business Analysis section tended to be structured around a number of releases in sixteen week sections. At particular times, these sections would overlap, raising the interesting research question of whether this particular structure might create workflow pressure points which, in turn, may contribute to feelings of work intensity. The following diagram illustrates the use of phased releases and the overlapping between stages for the Business Analysis teams:



Req	=	Requirements
Dev	=	Development
RX	=	Release X
RY	=	Release Y

DIAGRAM 4.9 INSOFT BUSINESS ANALYSIS TEAM RELEASE STRUCTURE

Project Team Structure

Project teams at InSoft were geographically dispersed across the United Kingdom and India. Teams did, however, appear to fit traditional structures common to software engineering, with project managers, Technical Leads and software engineers. The roles of project managers and Technical Leads were very similar to those at SpecSoft. Project managers were responsible for reporting to the client (often internal in this case), setting the direction of the team and ensuring work was on target. Technical Leads often acted as the point of contact between the team and the project manager and had responsibility for allocating tasks to team members and monitoring progress. Geographical dispersal of team members led to conference call meetings, extensive e-mail use, instant messenger and phone calls to maintain contact. Project teams would periodically set up workshops for team members to meet face-to-face every six to eight months.

Since it was not possible to follow teams geographically on-site and on location, access to team structures derived from the adoption of snowballing techniques. Diagrams charting project team structures and team member locations were drawn by the researcher during interviews to illustrate information provided by interviewees. The following diagrams demonstrate the multiplicity and dispersed nature of teams at InSoft:

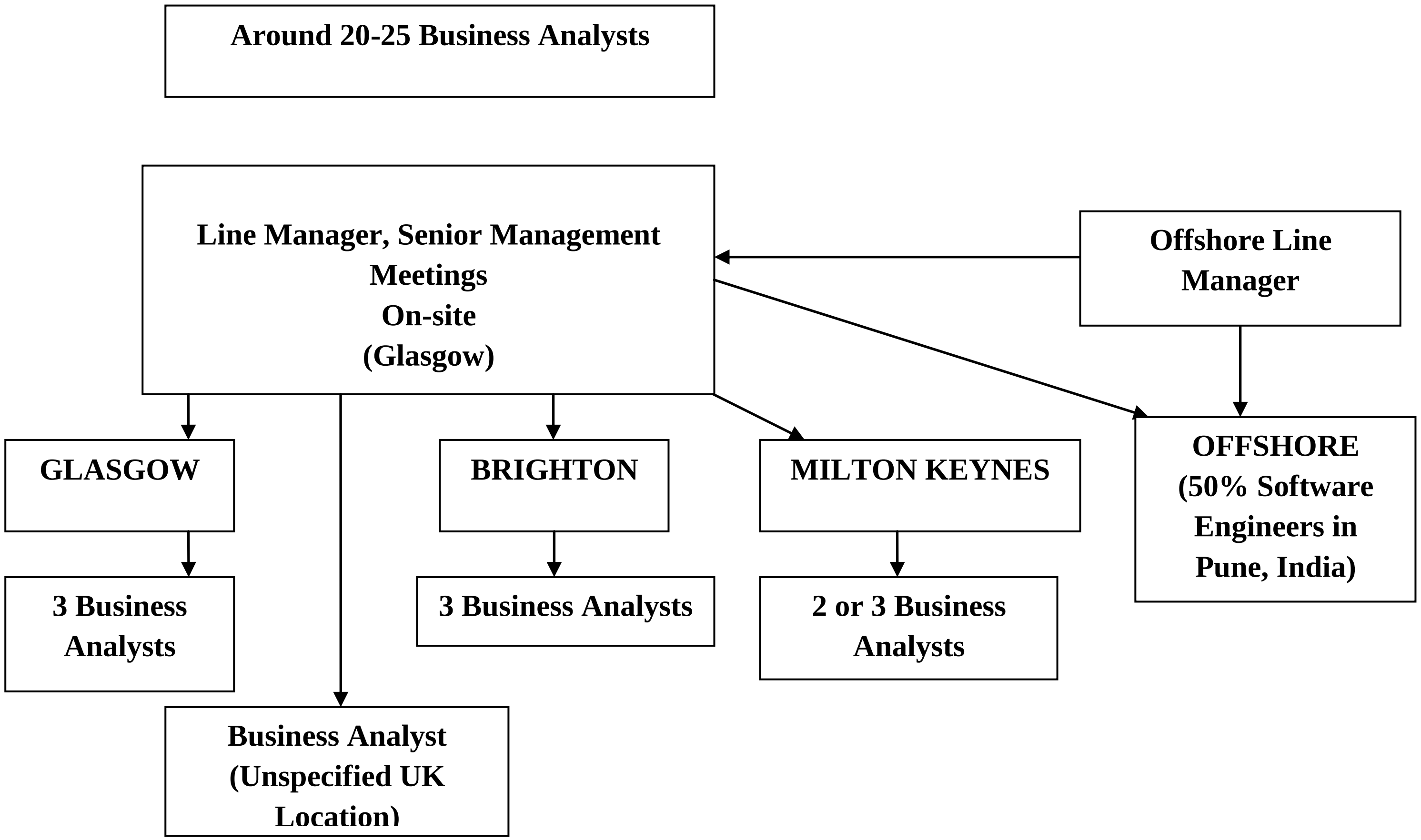


DIAGRAM 4.10 EXAMPLE OF ONE INSOFT BUSINESS ANALYSIS PROJECT TEAM STRUCTURE

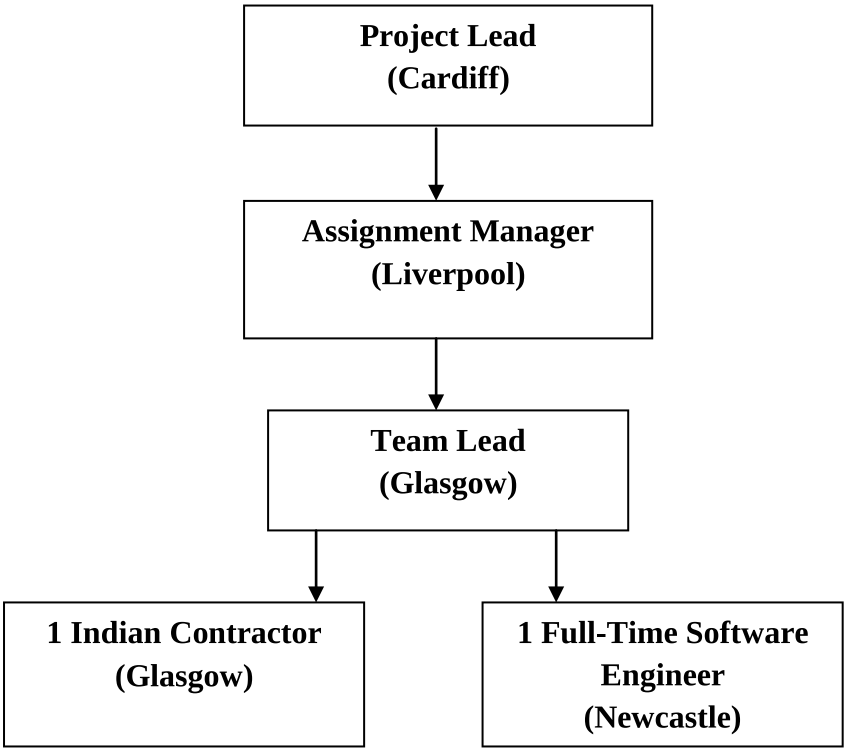


DIAGRAM 4.11 EXAMPLE OF ONE PERFORMANCE AND ARCHITECTURE PROJECT TEAM STRUCTURE

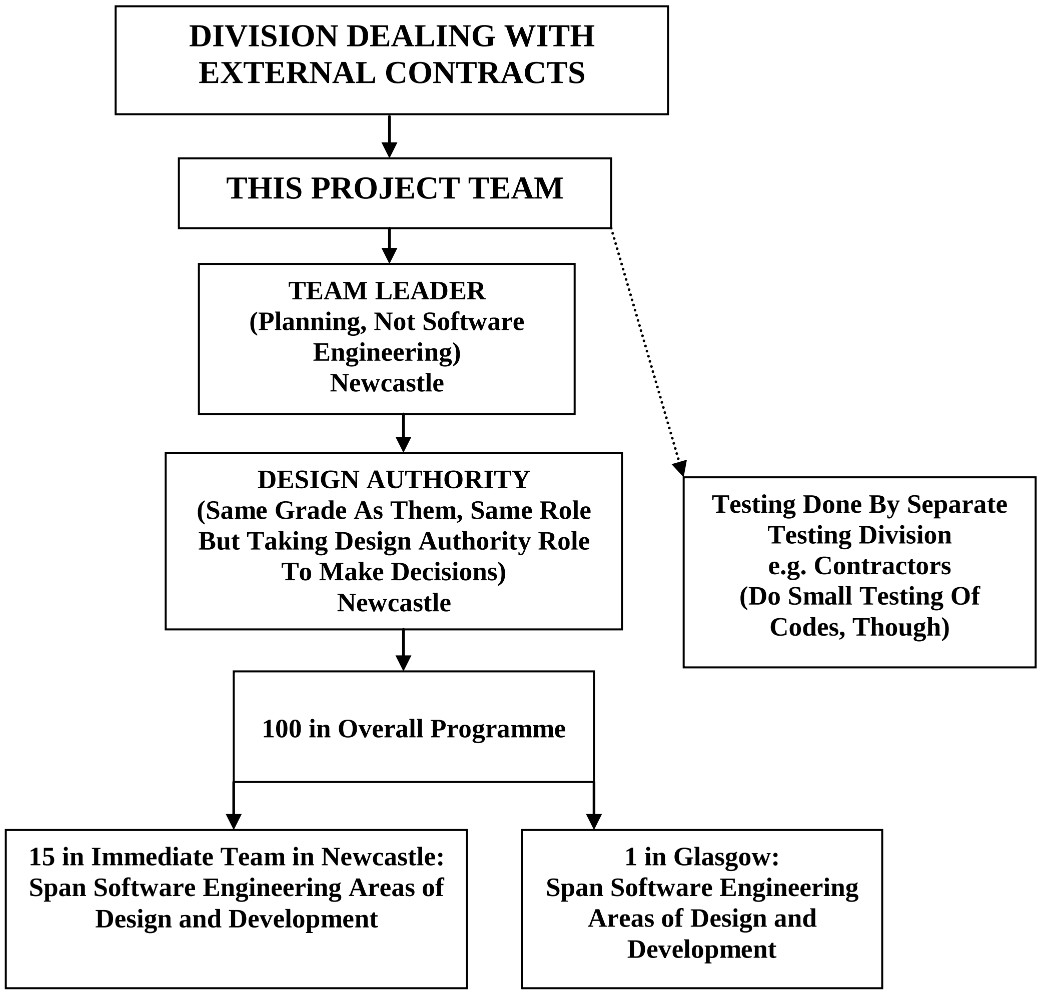


DIAGRAM 4.12 EXAMPLE OF ONE INSOFT EXTERNAL CONTRACTS PROJECT TEAM STRUCTURE

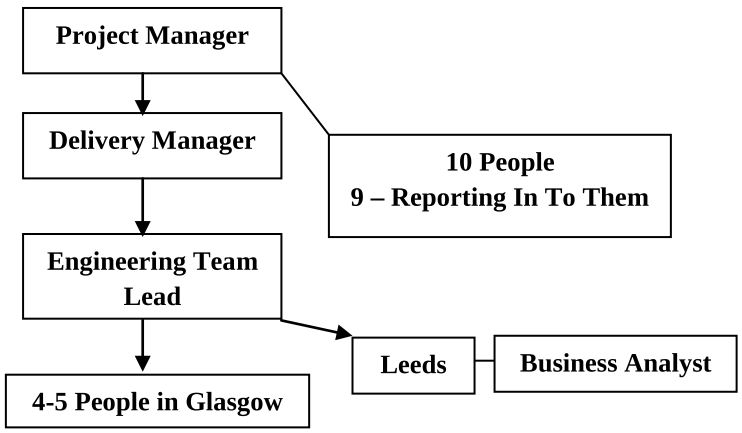


DIAGRAM 4.13 EXAMPLE OF ONE INSOFT ENGINEERING PROJECT TEAM STRUCTURE

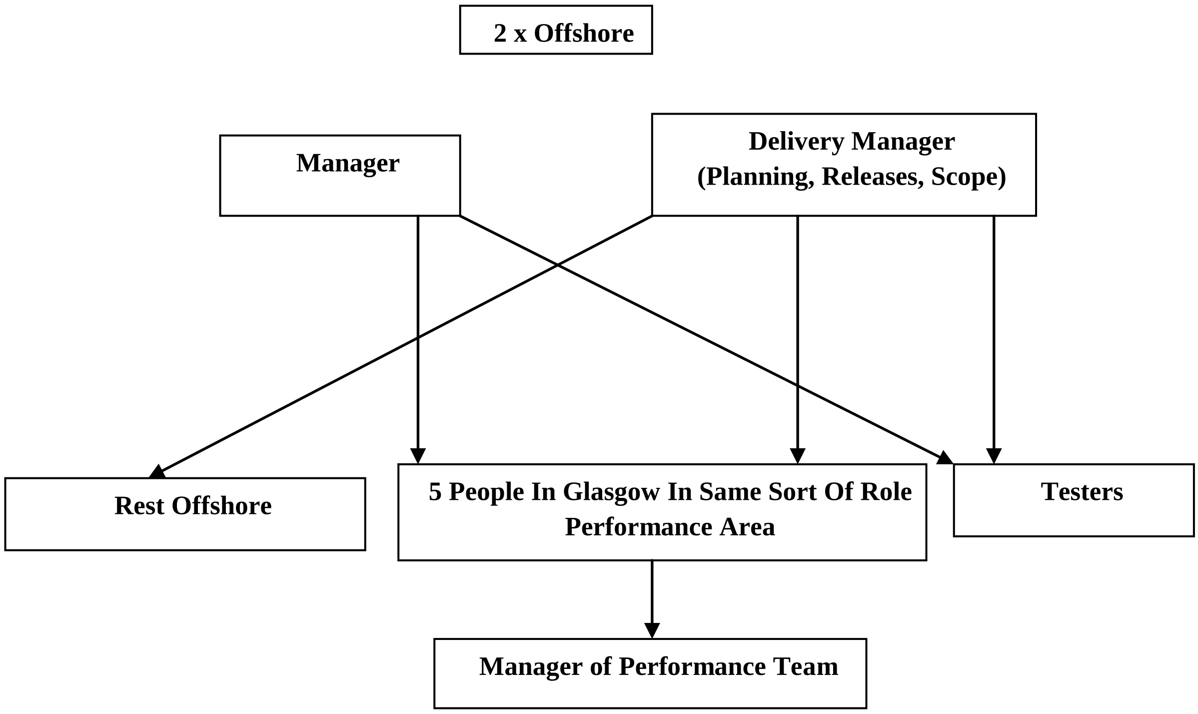


DIAGRAM 4.14 EXAMPLE OF ONE INSOFT SOFTWARE AIDS AND TOOLS PROJECT TEAM STRUCTURE

Project Documentation

Access to documentation at InSoft was heavily constrained due to bureaucracy and confidentiality procedures. InSoft did not permit access to private company documentation or intranet pages, therefore company information was attained through internet company sources, informal discussions and interviews with individuals. Informal discussions and interviews enabled the researcher to obtain an understanding of project nature, roles, responsibilities and the company in general.

4.8 SAMPLE

Participants were selected purposively, based on variations in gender, tenure, roles, responsibilities and characteristics of interest, in order to obtain an in-depth understanding of experiences of work intensity (Flick, 2009; Bryman, 2008; Rossman and Rallis, 2003). An iterative approach to sampling was adopted in order to determine the need for additional participants and further insight into phenomena (Bryman and Bell, 2003). Indeed, it is argued that an iterative approach can be beneficial through allowing additional insight, alterations and refinements to be applied to theoretical frameworks and enabling the attainment of a comprehensive understanding of work intensity. It was deemed appropriate to conclude sampling when incremental learning had minimal impact on the development of new insights, categories and improvements to theory (Flick, 2009; Eisenhardt, 2002).

Information sheets were given to all participants, outlining research purpose, research methods and assurances of confidentiality and anonymity (see Appendices 1 and 2). Consent forms were also signed by individuals to confirm that participation in work diaries and interviews was voluntary (see Appendices 3 and 12).

Work Diaries

Within SpecSoft Project Team One, all fourteen team members completed a work diary, leading to a 100% response rate. The principal characteristics of participants were as follows[1]:

- Design (team designation), Technical Lead (job title), male, full-time, permanent, 5 years tenure
- Design, Junior Team Member, female, full-time, permanent, 1 year tenure
- Development, Technical Lead, male, full-time, permanent, 6 ½ years tenure
- Development, Technical Lead, male, full-time, permanent, 9 years tenure
- Development, Senior Team Member, full-time, permanent, 16 years tenure
- Development, Senior Team Member, male, full-time, permanent, 1 year tenure
- Development, Average Team Member and Support Manager, female, full-time, permanent, 8 ½ years tenure
- Development, Average Team Member, male, full-time, contractor, 7 months tenure
- Development, Average Team Member, male, full-time, permanent, 1 year tenure
- Development, Junior Team Member, male, full-time, permanent, 2 years 5 months tenure
- Development, Junior Team Member, female, full-time, permanent, 8 months tenure
- Development, Junior Team Member, male, full-time, permanent, 4 months tenure
- Testing, Technical Lead, male, full-time, permanent, 2 years 10 months tenure
- Testing, Average Team Member, male, full-time, permanent, 9 months tenure

Within SpecSoft Project Team Two, ten individuals were provided with work diaries to complete, with six returned fully completed:

[1] Job titles of 'Junior' (little experience), 'Average' (average experience), 'Senior' (high level of experience) and 'Technical Lead' (high level of experience and responsibility) have been used to describe the level of individuals within project teams, in order to avoid using actual job titles applied within SpecSoft.

- Software and Consulting Sections (SCS), Technical Lead, male, full-time, permanent, 10 years tenure
- Software Section (SS), Technical Lead, male, full-time, permanent, 7 years tenure
- Software and Consulting Sections, Senior Team Member, male, full-time, permanent, 7 years tenure
- Software and Consulting Sections, Average Team Member, male, full-time, permanent, 3 years 4 months tenure
- Software and Consulting Sections, Junior Team Member, male, full-time, contractor, 2 ½ years tenure
- Software Section, Junior Team Member, male, full-time, permanent, 14 months tenure

At InSoft, work diary participants were generated through snowballing techniques, as discussed earlier. Ten out of twelve work diaries were completed by software engineers at InSoft. One diary could not be included in the study due to inaccuracy in information (codes within each day slot did not match up to codes in the 'Intensity of Day' box), whilst another was sent to an off-site developer and was not completed. Company contacts were responsible for securing offshore Indian software professional work diary participants, with three individuals being selected and two diaries being completed. Individuals participating in the work diary study worked within Business Analysis, Performance and Architecture, Application Support, Development, Engineering and External Contracts. The characteristics of these participants were as follows:

- Business Analyst, male, full-time, permanent, 10 years tenure
- Business Analyst, female, part-time, permanent, 13 years tenure
- Architecture Technical Lead, male, full-time, permanent, 12 years tenure
- Software Engineer, male, full-time, permanent, 12 years tenure
- Engineering Technical Lead, male, full-time, permanent, 8 years tenure
- Software Consultant, male, full-time, Indian on-site contractor, 1 year tenure
- Support Technical Lead, male, full-time, Indian on-site contractor, 2 years tenure

- Software Engineer, male, full-time, permanent, 19 years tenure
- Senior Developer, male, full-time, permanent, 4 years tenure
- Tester, female, part-time, permanent, 23 years tenure
- Business Analyst, female, full-time, Indian offshore contractor, 7 ½ years tenure
- Developer, male, full-time, Indian offshore contractor, 2 years 6 months tenure

Semi-Structured Interviews

From SpecSoft Project Team One, six out of the overall fourteen project team members were selected for interview, based on factors such as variability in tenure, gender, role within the team, nature of activities performed, experiences of work intensity and areas of interest. The individuals selected included: one male Design Technical Lead, five years tenure, with an extremely varied number of roles and responsibilities; one female Design Junior Team Member, one year tenure, working under the Design Technical Lead; one female Average Development Team Member/Support Manager, eight and a half year tenure, with a number of responsibilities and projects and who mentioned 'volume of work' as contributing to experiences of work intensity; one male Development Senior Team Member with the longest tenure at sixteen years; one male Average Development Team Member with contractor status and a seven month tenure; and one male Average Test Team Member, with a nine month tenure. The project manager was also interviewed to provide a comparison to the responses of team members, bringing the number of Project Team One interviewees to seven. In addition, the project manager had a ten year tenure at the company, meaning it was useful to gain insight into the experiences of this individual. The Test Technical Lead was asked for an interview but was unable to, due to a high volume of work tasks.

From SpecSoft Project Team Two, five out of the six work diary participants were interviewed. The interviewees selected included: one male SCS Technical Lead with a ten year tenure; one male SCS Senior Team Member with seven years tenure; one male SCS Average Team Member with just over three years tenure and who worked mainly in Business Analysis; one male SS Junior Team Member, the most junior with fourteen months tenure; and one male SS Technical Lead with seven years tenure and a large volume of tasks. One contractor working within SCS as a Junior Team Member was asked for an interview but was unable to participate, due to time constraints and deadlines. An interview with the project manager was scheduled, re-scheduled and finally cancelled, due to time constraints on part of the project manager. Interviews were also conducted at SpecSoft with the two directors for each company division (travel and transport and telecommunications, respectively), one of whom was also the acting company director.

At InSoft, interviews were conducted with seven software professionals working across the various areas discussed earlier. The software professionals selected were: one male Business Analyst with ten years tenure; one female Business Analyst with thirteen years tenure; one male Architecture Technical Lead with twelve years tenure; one male traditional Software Engineer with twelve years tenure, one male Engineering Technical Lead with eight years tenure; one Indian on-site Support Technical Lead contractor with two years tenure; and one traditional Software Engineer with nineteen years tenure. Company contacts were responsible for securing offshore Indian workers for filling in interview scripts, with three of these being completed. These included: one offshore female Business Analyst, one offshore male Developer and one offshore male Tester. Two interviews were carried out with project managers, in order to provide insight from the project management perspective and to compare and contrast responses to software professionals. An interview was also conducted with the senior manager of the in-house software section at the company. Three additional interviews at InSoft were planned but did not occur due to lack of response from one respondent and work diary issues for two others, in terms of inaccurate information in one diary and lack of diary completion for another.

4.9 RECORDING OF DATA

Detailed handwritten field notes documenting impressions, experiences, encounters and comments were written on each day spent in the case study organisations. Charts and diagrams were also produced to make sense of information and areas of interest. Field notes were typed up on the same day or the day after to ensure accuracy of information and were organised thematically, in order to aid examination of information and identify areas of interest.

Work diary data was initially presented in pie chart form in order to provide a picture overview of the types and number of activities performed by individuals. Pie charts were produced for each individual, representing each day in the work diary in terms of activities, percentage of time spent on these and information on the ‘Intensity of Day’ and ‘Main Causes of Intensity’. In addition, a further pie chart was devised per individual detailing activities from the overall work diary week and percentage of time spent on these. This information was then utilised to create tables summarising information for each grouping of individuals (SpecSoft Project Team One, SpecSoft Project Team Two, InSoft software professionals, InSoft offshore Indian software professionals). Each table allowed comparisons to be drawn between individuals with regards to job description selected on the ‘About You’ form, actual job title, length of tenure, total number of activities logged per day, ‘Intensity of Day’ ratings, activities noted as being the ‘Main Causes’ of intensity and typicality of data and events which may have impacted on work for that particular week.

Detailed handwritten notes were taken immediately and typed up after each interview, documenting findings, areas of interest, observations and interpretations. It was mentioned previously that diagrams charting project team structures and team locations were drawn by the researcher during interviews to illustrate information provided by interviewees. Interviewees also made sketches of project life cycles and forms of work organisation to use as discussion aids during interviews. Interviews were taped on a digital voice recorder, with permission for this granted prior to the interview. Recording the interviews allowed accurate representations of conversations to be attained and allowed for closer focus on information during the interview. For example, as stated by Mason (1996: 45):

> At any one time you may be: listening to what the interviewee(s) is or are currently saying and trying to interpret what they mean; trying to work out whether what they are saying has any bearing on

> ‘what you really want to know’; trying to think in new and creative ways about ‘what you really want to know’; trying to pick up on any changes in your interviewees demeanour and interpret these…reflecting on something they said 20 minutes ago; formulating an appropriate response to what they are currently saying; formulating the next question which might involve shifting the interview onto new terrain; keeping an eye on your watch and making decisions about depth and breadth given your time limits.

Furthermore, recording of interviews enabled the production of interview transcripts, which allowed for further examination of data and support of findings (Silverman, 2004; Bryman and Bell, 2003). Whilst interview transcripts remain authentic and unamended, digression has been removed from participant quotes in Chapters Five and Six, purely for the purposes of stylistic presentation.

4.10 DATA ANALYSIS

The utilisation of qualitative research methods represents the challenge of how to present data in a format which is comprehensive and easy to analyse, due to the large volumes of data accrued (Bryman, 2008). Indeed, data analysis is considered to be one of the least explained processes in qualitative research, with approaches depending on research purpose, objectives and methods (Yin, 2003). However, general guidelines to aid the organisation of qualitative data can be identified, including coding, establishing themes, or arraying information into matrices, tables or flow charts (Rossman and Rallis, 2003; Eisenhardt, 2002; Miles and Huberman, 1994).

It was emphasised earlier that data analysis occurred incrementally throughout the research process, in order to make sense of data, identify emerging themes and manage data effectively (Bryman and Bell, 2003). Data from each individual collection method (field notes, diagrams, observation, documentation, work diaries, interviews) was examined at each stage to identify arising themes and areas of interest. Data was then combined following each incremental stage in the research process (SpecSoft Project Team One, SpecSoft Project Team Two, InSoft software professionals) to draw comparisons and identify emerging themes. Following this, case study write-ups of both SpecSoft and InSoft, including company background, team structures and data findings enabled sources, linkages and mechanisms surrounding work intensity to be mapped. Case study write-ups allowed for the identification of similarities, connections and differences between the two companies and the development of theoretical and conceptual frameworks to explain experiences of work intensity.

CONCLUSION

This chapter has outlined the methodological approach taken in this research with regards to philosophical position, nature of enquiry, the categorisation of professional software job roles, qualitative research methods and strategies taken to record and analyse multiple sources of data. It has been argued that the main tenets of the Critical Realist philosophy are consistent with this research, due to the identification of an objective reality and the recognition of structure, agency and conflictual individual realities within professional software work. Critical Realism is also held to be particularly appropriate to this research due to the emphasis on establishing causal linkages between objects and identifying mechanisms, structures and relationships, in order to provide an in-depth, stratified explanation of work intensity. In addition, the taxonomy of professional software job roles (which has been developed by the author) has been applied within this research to explore the nature of activities and tasks performed by individuals, variations in work experiences and relationships to work intensity. The selection of two in-depth qualitative case studies is argued to have been beneficial through allowing for typological theorising, interpretation of meanings in context and the development of comprehensive theoretical and conceptual explanations of work intensity. Furthermore, the utilisation of multiple sources of evidence, as well as clear clarification and documentation of research aims, design, methods, data recording and analysis procedures have been deemed essential in increasing the objectivity, validity, reliability and transparency of research.

CHAPTER 5 SPECSOFT: SOURCES OF INTENSITY

INTRODUCTION

SpecSoft was housed across two large buildings in a business park on the outskirts of a residential area. The reception desk in the entrance of one of the buildings had a small seated area for guests, a table for company information and a corridor giving access to the company divisions. Project Team One (PT1) was based behind the reception area across two open plan spaces. Project Team Two (PT2) was based in the second large building in an upstairs wing over an open plan area.

This chapter identifies sources of intensity at SpecSoft and documents the consequent experiences of its software professionals. Despite the different sectors (travel and transport; telecommunications) in which these project teams operated, common causes and patterns of intensity were discernable. A generally applicable model of market dynamics, technological developments, firm characteristics, internal organisational factors, work organisation, immediate determinants and factors which help offset work intensity has been applied in order to explain the distinct, yet inter-related, layers which shape experiences of work intensity. Market dynamics, including de-regulation, privatisation, 'booms and busts' and the impact of global events framed experiences of intensity at SpecSoft. Technological developments in computer speed and communication mediums were considered to have intensified professional software work, through enabling functions to occur more rapidly and in parallel. Firm characteristics such as specialist software firm status and small company size meant that SpecSoft focused on securing client contracts, offering tight deadlines, utilising lean staffing levels and software professionals performing multiple roles. These characteristics meant that clients and deadlines predominantly shaped experiences of intensity at the internal organisational level. In addition, leadership style had implications for software professionals' experiences work intensity, depending on the personal style utilised by leaders and the extent to which leaders understood the realities of project work. Forms of work organisation at SpecSoft in terms of the physical proximity of project team members and utilisation of the ‘Agile’ methodology had implications for software professionals’ experiences of work intensity. Market dynamics, technological developments, firm characteristics, internal organisational factors and work organisation had implications for software professionals’ experiences of work intensity on a daily basis. Several factors were identified as immediate determinants of work intensity, namely carrying out support activities, volume of work, specialist knowledge, interruptions to work, internal motivation and breaks. Factors such as managing relationships with clients, willingness by project managers and Technical Leads to consider team member perspectives and opinions, personal pro-activeness and level of experience were identified as helping to offset software professionals’ experiences of work intensity.

This chapter concludes by summarising sources of intensity at SpecSoft in diagrammatic form, in order to illustrate the hierarchy of factors shaping experiences of intensity at the individual (i.e. software professional) level.

5.1 MARKET DYNAMICS

All participants perceived that market dynamics had implications for experiences of work intensity. For example, de-regulation and privatisation were considered to have intensified work for software professionals operating within the Utilities sector:

> When I joined Utilities, they were public sector, so it was a paced kind of intensification. You were busy but you weren’t stressed, you weren’t under a lot of pressure. But then, in 1990, they were privatised, so suddenly you had the pressure of privatisation. You had investors looking for returns on investments, you had to deliver profit, and it was a whole new pressure. Change becomes the pressure. 1994 we had de-regulation, so industrial customers could choose where they bought their electricity. Year 2000 brought a change to how electricity was traded. So you had all these kinds of changes, which each brought a pressure, because it had to be done by a given timetable. And that brings intensification. So it’s a competitive change, we need more profit for our shareholders or

> we're losing market share. Or it's a regulatory change and we're responding to some of the regulation. Change is a factor. Some industries are undergoing more change than others (Senior Manager).

Global events and 'booms' or 'busts' in the market were also reported to have implications for work intensity and the experiences of individuals. Senior management reported that the 'Year 2000' bug and 'September 11th' had created downturns in the aviation industry and reductions in client project financing, impacting on the travel and transport division. Following these events, concerns within the aviation industry to improve security and resources had latterly created project opportunities for the division. Furthermore, these events had similar implications for the energy, utilities and telecoms division through reducing project opportunities and stimulating new areas for work:

> After 2000 and 'September the 11th', there was a big downturn in all markets with regards to spending in IT. Energy and utilities in particular were struggling and we felt it in our division. The energy and utilities markets are driven by regulation and when there are changes, you get a lot of work. In the Year 2000, there were no changes and the work dried up completely. It's now taken an upward turn, and we should be focusing on things like renewable energy. There's a lot of drive for the more green aspects and there's a lot of work in that. And all of that can make your job more intense, 'cause when we had no work in energy and utilities, we all felt it for a while. So it can affect you big style (PT2 Technical Lead)[2].

Reductions in available project work as a result of market downturns were seen to encourage company downsizing, which could intensify work experiences for

[2] In order to preserve the anonymity of participants, this chapter has avoided using actual job titles applied within SpecSoft. The titles 'Junior Team Member' (little experience), 'Average Team Member' (average experience), 'Senior Team Member' (high level of experience) and 'Technical Lead' (high level of experience and responsibility) have instead been applied to describe the position of individuals within PT1 and PT2.

remaining individuals when work became available. However, buoyant market conditions and opportunities for project contracts could equally impact on intensity. For example, it was suggested that SpecSoft had accepted more work than was realistically achievable:

> When the market is booming, then there's more work. A company like us will take on probably more than it can do, which makes it intense, because then you've got lots of things to do but you don't have enough people to do them. So we would go out to tender with clients and say "We'll deliver this software to you for this much money" and we'll maybe put out a number of tenders. We'd be expecting that we'd maybe get two, maybe three. We'll put out maybe six tenders and say we get all six, we've got to deliver all of them. But we don't have the resources, we don't have the people to do all the work, which would then mean that we'd have to get in contractors, who would then have to be trained up. All of this makes it a lot more busy. In that sort of change over period, you've got lots of people trying to do two jobs, so that can make a big difference (PT1 Technical Lead).

In sum, there is agreement that broader economic conditions provide a framework for increasing work intensity. Both buoyant market conditions *and* downturn appear to have implications for experiences of work intensity through affecting opportunities for project work and levels of resourcing.

Staffing Arrangements

Staffing arrangements were commonly considered by both project teams to have variable impacts on work intensity, depending on the degree to which teams were under-staffed or too large. Senior management acknowledged that insufficient staffing levels contributed to experiences of intensity for software professionals at SpecSoft. However, it was reported that the general reduction in investment across the software industry meant that staffing levels had to be tighter and individuals had to be utilised more effectively:

> There's very little spare resource. Now, I think that's something that's changed in the industry. I look across any company and they don't have what they call 'bench'. People don't operate with 'bench' anymore. No industry carries a lot of spare resource. Before the dot.com boom, everyone was getting external investment and it didn't matter, they weren't burning their own money. There wasn't this same understanding of "Actually, you need to get these people utilised" (Senior Manager).

From one standpoint, PT1 and PT2 members reported that insufficient staffing levels increased levels of responsibility and volume of work, contributing to experiences of work intensity. For example, teams across the business were constantly vying for resources, meaning that existing team members had to juggle competing work tasks:

> I don't think I've worked on a project – for a long time at least – with the required number of developers. The development team is never as big as it could be, or should be. We're always fighting for resources across the business and that's probably the major bugbear. You never get the time to do the other things that you think you should be doing 'cause you're constantly doing something else, as if you're on a treadmill (PT2 Technical Lead).

However, whilst more resources could help greatly in terms of managing workload, members of PT1, PT2 and the PT1 project manager commented that this could create other complications. For instance, increases in team size could give rise to difficulties in overall management and coordination of project teams. Similarly, one PT1 Junior Team Member recognised that new team members could require coaching and clarifications from existing members such as Technical Leads, interrupting the flow of normal work tasks:

> Having brand new people creates a lot of questions, interruptions and that breaks your train of thought. When there are a lot of people on the project, that really affects the Team [Technical] Leads. There are more people to handle, to distribute tasks and

explain things to, more people to send work to and see how they're doing. That creates more work.

Individuals further emphasised that team members could be moved over to other projects or functions ascribed higher priority, heightening experiences of work intensity for remaining team members. Senior management recognised that moving software professionals to projects which were ascribed a higher priority could intensify work for remaining team members. Nonetheless, it was emphasised that this action was unavoidable, due to the need to secure new project work or retain existing clientele:

> You don't always know what work you're going to win. Sometimes a customer will phone up in a panic saying, "Can you help me out here?". Now, that customer's always given me a huge bit of work so you can't say no. You don't want them to go somewhere else, but you don't have anyone spare to do it. So you're trying to free someone out of a project in order to get them on to help out that customer and you're then putting pressure on the project for the people who are left behind. So I think because there's rarely any spare, it does put pressure on people.

To summarise, under-staffing, overly large project teams and rotation of team members can be identified as contributing to experiences of work intensity for software professionals. However, lean staffing levels and rotation may be unavoidable due to economic pressures and the need to secure and retain project work.

5.2 TECHNOLOGICAL DEVELOPMENTS

Improvements in Computer Speed and Technological Mediums

Senior management suggested that professional software work had intensified over time as a result of developments in technology. Crucially, increases in computer speed and the advent of technological mediums such as e-mail were identified as enabling functions to occur more quickly and in parallel:

> I've been in the industry since computers were invented and it's changed a great deal. Software systems are being recompiled every hour by lots of programs and it's all getting done right away. In my day, a computer took up a whole room and it needed water cooling. Software work was by written typed up memo. You used to write it out longhand and it would get sent to the typing pool, then you'd get it back and correct it. It would take days. If you look at now, it's all e-mail, it's all faster cycle times. As well as things happening more quickly, there's a higher bandwidth of things happening at the same time. I think these are the two main factors in intensification. Going back in time, because things were slow, you tended to focus on one activity, doing that through. I would say there's more bandwidth now, so people are doing several activities generally at the same time (Senior Manager).

In addition, four PT2 members stated that technological mediums such as e-mail and instant messenger could impact on intensity, through diverting attention from work tasks, breaking 'train of thought' and being less conducive for collaborative project team discussions. Consequently, one PT2 Average Member reported that measures such as switching off alerts or checking e-mails at pre-defined times were necessary to manage the interruptive nature of these technological mediums:

> MSN [messenger] most definitely gets switched off or put on 'busy' or 'appear offline' if I really need to get something done. That's to dissuade both colleagues, clients or personal clients I may have on that from talking to me. E-mail can really lead to a lot of intensity. I'm a bit pernickety in the way I use it. I switch off the alerts and I only check it when I want to. Especially in IT, people expect you to answer e-mails almost immediately because they think you have your e-mail open all day long. So that adds to the intensity, 'cause then they phone you and say, "Why's he not replied to e-mail?".

> E-mail and all these various communications definitely impact on your intensity and the interruptative nature of your job 'cause if you get interrupted, it feels more intense when you get back to the thing you should be doing.

Improvements in computer speed appear to have increased volume of work and contributed to work intensification within professional software work. In addition, technological mediums such as e-mail and instant messenger can be seen to have implications for software professionals' experiences of work intensity, through interrupting concentration and diverting attention from work tasks. Section 5.6 in this chapter will explore more widely the implications that volume of work and interruptions have for software professionals' experiences of work intensity.

5.3 FIRM CHARACTERISTICS

Organisational Type

Several members across both teams perceived that differences in chargeability and core organisational focus could contribute to varying levels of intensity between specialist software firms and in-house IT departments. Individuals who had worked for in-house IT departments were able to reflect on their past experiences. For example, one PT2 Technical Lead argued that levels of accountability, responsibility and visibility could be greater within specialist software firms such as SpecSoft by virtue of the fact that projects were being charged to and developed for external clients:

> I have worked in an in-house IT department within a large company. I'd say that there's more space for people to disappear, you know, or people can work less intensely without it being immediately visible to the people that employ them. Whereas [in SpecSoft], every hour that you work is accounted for, in terms of the project that you charge against and things that you have to deliver against. So, it is more intense in a specialist software house.

Similarly, one PT1 Senior Team Member stated that whether or not software was the core organisational focus could influence experiences of work intensity:

> Where computing isn't effectively your core subject, like in banking, then the computer systems are there as a service, so you have a wee bit more leeway, and it's not quite as intense as in a specialist software company.

These observations contrasted with senior management and project management perceptions that in-house IT departments and specialist software firms could face different types of pressures which could lead to experiences of intensity in both types of organisation. For example, one senior manager remarked that in-house IT departments could experience intensity through other organisational members de-valuing the quality of internal services:

> I think a large in-house IT department sometimes suffers from the politics of the organisation. I think there are pressures but they're different in each area. We deal with some in-house IT ones and I can see them, where their business is saying "Oh, our internal IT department are rubbish" and so the intensity there is that people don't rate them. Whereas at least if you're a supplier to an organisation, you've specifically been chosen to deliver that, so you are the choice, as opposed to, in an internal one, you are actually sort of put on them, in terms of, "You must use the internal". So I think there are different pressures, depending on the organisational type.

The project manager of PT1 agreed that whilst specialist software firms could be affected by the unpredictability of available projects, in-house IT departments could be faced with the threat of outsourcing of work:

> I would imagine there are different types of impactors, depending what environment you're in. In a big company with its own internal IT department, you're going to be working potentially on the same things for years to come, so that could be de-motivating.

> Likewise, a lot of big organisations get taken over and outsourced, so your job might be at threat. In our organisation, you're always wondering what the next job's going to be, because we typically do shorter term pieces of work, and if you look forward six months, nobody knows what they're going to be doing.

In this sense, differences in client base, accountability, core organisational focus and importance attributed to retaining software engineering functions can influence experiences of work intensity.

Company Size

There was virtual unanimity amongst software professionals that working within smaller organisations was likely to be more intense than working in larger organisations due to greater levels of visibility, leaner staffing levels and variability in roles. For example, one PT1 Average Team Member observed that were a software professional in this lean and visible environment to slacken work pace or reduce effort levels, such behaviours would be quite apparent:

> I think the smaller the company is, the harder it is to hide. [SpecSoft]'s quite a small company and it's quite visible what you're doing on a day-to-day basis. If you're in a huge company, nobody really knows who you are. Directors don't know your names and that kind of stuff, so it probably doesn't matter if you're sitting doing naff-all, no-one will really notice. Whereas in here, if you're sitting doing naff-all, it would be noticed – not just by people around you but by the management as well.

Senior management and the PT1 project manager confirmed these perceptions, commenting that individual contribution to projects and the 'bottom-line' within smaller companies was likely to be more visible and to increase experiences of work intensity.

Members of both project teams stressed that staffing resources could be tighter within smaller organisations, meaning that individuals could be responsible for carrying out a multiplicity of tasks. Indeed, the work diaries revealed that software professionals at SpecSoft were engaged in diverse tasks embracing the work of systems/business analysts, designers, developers and consultants. For instance, one work diary showed that a PT1 Technical Lead (see Appendix 14) was engaged in a wide range of activities ranging from 'Meeting With Clients' (6%), 'System Test' (2%), 'Develop a Small Section of Software' (1%) and 'Research Possible Technical and Design Approaches' (1%) . In addition, the work diary for one PT2 Senior Team Member (see Appendix 22) illustrated that this individual was responsible for a variety of tasks, such as 'Provide Support' (25%), 'Develop a Small Section of Software' (9%), 'Unit Testing' (5%) and 'Obtaining an Understanding of the Client's Current System' (2%). The phrase 'jack of all trades' was therefore utilised widely by software professionals to refer to the variety of tasks and responsibilities that individuals had to perform:

> It is necessary to be a 'jack of all trades'. You can't focus on working on one particular piece of the software development life cycle at [SpecSoft]. Currently, I'm bringing up-to-date all the design documentation associated with a project that I've been working on. I've also written the test specification and I've run through the test specification. I have done design on it, I've done implementation. I've coached and mentored other members of staff and I've delivered software. I've also been involved with the customer in discussing and agreeing requirements and planning aspects of it. It's across the whole software life cycle (PT2 Technical Lead).

Individuals identified that carrying out multiple roles, rather than being able to rely on designated design, development and testing sections, could contribute to greater levels of intensity for software professionals within smaller companies such as SpecSoft:

> When you're in a smaller company, you can't have a test team and a development team and a documentation team and whatever some of these big teams have. You're just kind of doing it all yourself, so it will make you feel more intense, 'cause you've got so many different tasks to do. Just from chatting to other people and knowing what other companies are like, you know, there's always the test team and we don't have that. We just do all that ourselves, as well as the development and stuff. I think that can make it feel more intense because we are smaller, we can't have those separate teams (PT2 Average Team Member).

From the perspective of members of SpecSoft, small company size was therefore deemed to contribute to increased work intensity.

Industry Type

Industry sector proved to be an additional influence upon work organisation and experiences of work intensity at SpecSoft. The telecommunications sector stood out as having a notable impact on experiences of intensity within PT2. For instance, software professionals and the project manager emphasised that telecommunications was a particularly fast-moving and competitive industry, as a result of regulatory changes, constantly evolving new technologies and market saturation. Consequently, clients had to make regular changes to products and react quickly to remain competitive, making it difficult to plan workloads in advance. Members of PT2 stated that this dynamic impacted on the telecommunications division in terms of speed of response, technology adoption, simultaneous management of several projects and prioritising of work:

> [In the] telecoms industry, there's a lot of money to be made from the products that you can develop. The problem is the telecoms industry is *so* fast-moving it's frightening, 'cause everybody wants to beat their competitors to get something advertised and to market quicker to achieve a bigger market share. With that comes the pressure of getting things delivered and with that comes work

intensity. It just gets placed on us as we're the development team (PT2 Technical Lead).

The following diagram (5.1) represents the interplay of characteristics of the telecommunications market, pressures on clients and the impact on PT2:

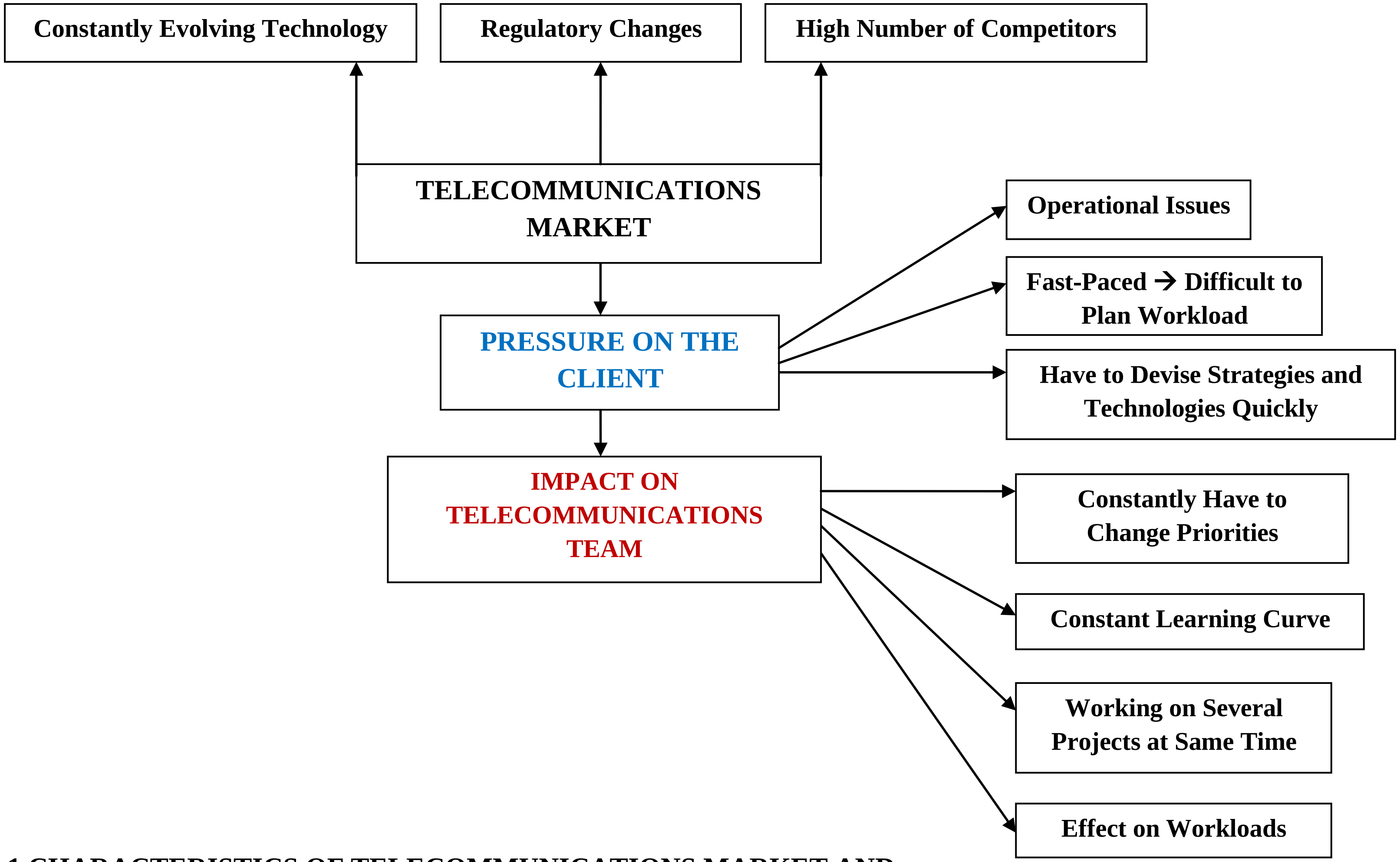


DIAGRAM 5.1 CHARACTERISTICS OF TELECOMMUNICATIONS MARKET AND IMPLICATIONS FOR CLIENTS AND SPECSOFT PROJECT TEAM TWO

5.4 INTERNAL ORGANISATIONAL FACTORS

Clients

All PT1 and PT2 members, including project managers, reported that clients impacted on intensity of work through their indecision, lack of clarity and unwillingness to compromise over deadlines. One PT1 Technical Lead was emphatic in regarding clients as a source of intensity, through their continuous changes to priorities and frequent, sudden requests for alterations to work already in progress:

> They [clients] have a huge influence. In the design process, it can be extremely frustrating. The most recent design process that we went through, we had a number of requirements that the client had stated "These are the ones that we want for this release". So we did a bit of initial work saying "This is what we're going to do for each of these requirements" and went into the first design meeting with that document, presented each one of the requirements and in each case, got "That's good, that's bad, we need to make changes to this". So that was all good. But there was also the fact that they said "Well, we also want to talk about *these* requirements" when we were already two weeks into the design process. In the same timescales, we're having to design the extra requirements and they won't decide which of the original requirements they want to drop. It leaves you in a difficult position of having to design everything, including these extra things.

One PT1 Junior Team Member further observed that client indecision impacted on other software professionals working elsewhere in the IT industry:

I don’t think it’s confined to the project and this particular client. I know from friends of mine who work in IT and meet clients – exactly the same thing happens to them. It just seems to be an industry-wide characteristic that clients don’t know what they want until you’ve delivered a system and then they say “We wanted it slightly different”.

Others stated that levels of technical knowledge could influence the role, influence and power exerted by clients. Two individuals within PT2 (one Senior Team Member and one Technical Lead) stated that the influence and power exerted by clients could decrease on certain projects as project team members acquired more knowledge of the system over time. For instance, the complexity and age of systems could make it more difficult for clients to employ other organisations to maintain, modify or upgrade these systems. Retention of knowledge over systems could therefore place PT2 members in a position of power for the securing of future work:

I think they’re more reliant on us than we are on them. As [the system] becomes older, the ability of other companies to come in and do work to it, it’s harder. As an application gets older and gets more things added to it, it becomes more of a spaghetti junction, more complex and therefore harder to maintain, modify and upgrade. We’re in more of a position of strength from that perspective. Also, because our customer is trying to go into new revenue streams and because there’s new regulatory and business requirements coming in from their business, they’re certainly throwing work at us. We’re in more of a position of power with that. They’re in less of a position of power than they were three years ago (PT2 Technical Lead).

PT1 members, however, reported that clients had significant influence due to the amount of technical knowledge they possessed on the systems and software being created. Indeed, PT2 acknowledged that if clients possessed significant technical knowledge, this could create tensions between clients' preferred approach and that of the team members involved in the project.

Whilst senior managers and the PT1 project manager recognised that clients could greatly contribute to experiences of intensity, views on the impact of clients' technical expertise contrasted with the experiences documented by software professionals. For example, knowledgeable and experienced clients were perceived as being more likely to have realistic expectations of tasks, timescales and costs:

> We like to deal with organisations that know about IT. So, the envelope of expectation is narrow, 'cause they know how long things should take, how difficult they are and what they should cost. If you're dealing with more of a naïve customer who's maybe read computing and they've read that somebody's done something or other, they think, "Oh, I can get this in a couple of months and it'll cost £50k". These are harder people to deal with, because the expectation is far away from the reality. So, that's why [SpecSoft] elects only to work with blue chip customers, because they're experienced and more realistic (Senior Manager).

Senior managers also remarked that client role and involvement could be determined by pressures and agendas within client organisations themselves. For instance, clients could become more involved as a result of internal pressures to deliver systems on time:

> I've seen some systems where we've been left alone for a wee while, we've maybe had a wee blip and the customer's wants to get involved. That's because a customer's got a career too. If you deliver something that's not what they asked for, or if you're going to be a month late and you don't tell them until the day before it's due, you've just given them a career bad mark, because they've then got to go and say to their boss, "Actually, you know that

> system I told you you're getting tomorrow, you're not getting it". So I think a lot of the customer's involvement is they're under pressure and they're looking for "How can I make sure?" (Senior Manager).

Levels of client involvement were recognised by senior managers as being further shaped by the degree to which projects were central or peripheral to client objectives and agendas. In this sense, the types of projects selected could potentially create different forms of intensity for software professionals at SpecSoft:

> We've always gone for – and I think it's because we want to be best –projects that are fairly critical to our customer's business. If it's fairly critical to the customer's business, they're [the customer] going to want quite a bit of involvement and the risk of it going wrong is big to the customer. So that creates it for us. If we were doing some hum-drum stuff, I guess the pressure, the routine work, could easily get offshored. But because we go into demanding areas, then I think that creates the pressure (Senior Manager).

Clients and their demands can therefore be recognised as impacting in diverse and important ways on the degree of work intensity experienced by software professionals. Possession of technical knowledge and the extent to which projects are central to objectives appears to influence the role, influence and power of clients.

Deadlines

Senior management suggested that competitive, financial and efficiency concerns had encouraged clients to change the nature of contracts for software projects. For example, one senior manager noted that whilst budgets and timescales had been flexible under contracts in the past, clients increasingly placed emphasis on terms and conditions, fixed budgets and set timescales in order to deliver projects as efficiently as possible. Consequently, these pressures could be seen to impact on experiences of work intensity. For example, whilst all participants emphasised they had autonomy to plan out work, set work pace and provide estimates of how long tasks would take (according to experience), these flexibilities operated within the constraints of deadlines. Crucially, deadlines were identified by all PT1 and PT2 members as being the pivot around which all work activities were managed, requiring tasks to be constantly juggled and necessitating increases in work effort as deadlines approached. One PT2 Technical Lead illustrated the challenge of managing the realities of project work within the framework of immovable deadlines:

> If you've got a deadline for one development task, that deadline essentially can't be moved because it means the whole project deadline's going to move. And then you get support calls in, or you get some other responsibility, or the development task's been underestimated, or it's more complex than you first thought. It can be really intensive, that's what is always at the back of your mind, always, always, always. You're always kind of monitoring your own progress.

Junior Members echoed similar experiences of deadlines causing work intensity. For instance, one PT2 member commented in their work diary, "I had a deadline for Friday, so the intensity increased as I approached it". Indeed, the 'Intensity of Day' column clearly illustrated that work intensity increased for this individual from 2 (A Little Intense) to 3 (Intense) across the work diary week, as a result of this deadline.

Deadlines have emerged as the main mechanism around which all software activities are planned and structured. Thus, deadlines can be identified as a major cause of intensity through requiring software professionals to juggle work tasks and increase effort levels in order to meet these targets.

Leadership Style

Many software professionals commented that SpecSoft operated with a small company ethos, where it was possible to "walk into one of the directors' offices and just start chatting to them". It was also suggested that the informal and collaborative culture at SpecSoft had largely stemmed from the approach taken by the original managing director, who placed emphasis on taking responsibility and helping others if they needed help:

> For most of the time I've been here, a *lot* of the culture has stemmed directly from XXXX [old managing director] and the other founding directors who have now left. What they were looking for was for people to be professional and supportive. XXXX used to do this talk, which is quite illustrative. It was company culture talk and XXXX would give it to people within the first two or three weeks of them joining the company so they didn't form any bad habits. The thing used to illustrate it was "Fill The Kettle". The idea was that if you went into the kitchen and the kettle was empty, fill it. Or if you're the last person to use the kettle, fill it. It was XXXX's way of illustrating "Take on the problems, own the issues". "If you leave a mess in the kitchen" was another one. "You need something done – you do it". "You see someone else who's having a bit of difficulty, give them a hand" (PT1 Project Manager).

Despite the informal and collaborative approach identified at SpecSoft, the style of leadership (at Technical Lead, line management, project management, head of division and director levels) was reported by members of both project teams as potentially affecting work intensity. For instance, one PT2 Technical Lead suggested that individuals working with project teams on a daily basis (such as Technical Leads, other team members or project managers) were more likely to understand the realities of project work:

> If you look at leadership at the technical and sales director level, I think that they can directly impact on your work and they have a more ‘Just get it done’ approach. Whereas the level below them - the people you’re working with directly day in and day out - recognise the constraints that you’re working under, how long it will actually take to do things and what other things you’re doing. As a result, there’s less pressure on you to complete that work in a shorter time frame, because they recognise what’s realistically possible and they’ve also got a better view of long-term what you’ve to achieve. So, I do think it has an effect on the intensity of how you have to work (PT2 Technical Lead).

In addition, facilitative, constructive styles of leadership were seen to be more effective at managing individuals than traditional control approaches:

> This is going to make me sound quite sensitive but I definitely do work easier and better overall for a more engaging manager, rather than somebody who’s just saying, “Get it done”. In terms of intensity, I think your day-to-day work is more enjoyable if you’re getting feedback that you’re doing the right thing. I’d work better for a manager who’s got better feedback skills, better communication skills and who remembers to do both sides – tell you when you’re not doing quite what you’re supposed to be doing but remind you when you’ve done exactly what was asked (PT1 Average Team Member).

Project managers and senior management supported comments made by PT1 and PT2 individuals, suggesting that software professionals’ experiences of work intensity could be dependent on the personal style and approach adopted by leaders. Indeed, senior management recognised that leadership style could influence work experiences and the general ethos of project teams:

> There's a phrase that goes "Speed of the leader, speed of the team". It's almost like the persona of the manager tends to become the characteristic of the department. If the manager is very driven towards achieving things quickly, then you'll find that pressure goes down the whole department. If the boss is interested in quality but not too bothered about time-keeping, surprisingly, that then becomes the characteristic. It's almost like the deed of the manager is more important than what's said or written. In all the organisations I've been in, the passion of the boss can actually create the characteristic of the work group.

The approach taken by leaders can therefore be seen to influence the attitudes and behaviours embodied by software professionals and the ethos of project teams. Leadership style appears to affect experiences of work intensity, with facilitative and supportive styles being more effective at managing software professionals than direct control methods.

5.5 WORK ORGANISATION

Physical Proximity

The evidence suggests that the physical layout of teams and the configuration of work teams at SpecSoft were factors of some significance. PT1 members and the project manager were interspersed with other members of the travel and transport division (who were involved in different projects) in an open plan area in clusters of four, in order to improve interactions and communications in the overall sector. Individuals worked at their individual work stations and engaged in informal work-related and non-work related discussions with other team members.

PT2 members and the project manager were seated in an open plan area in clusters of four. There were low partitions between clusters but people could be easily observed. The atmosphere of PT2 was informal and with more audible noise than was detected with PT1. PT2 members and the project manager claimed that the telecommunications divisions had quite a ‘macho’ culture, with joking, ‘ribbing’ and an element of bravado.

It was widely held by PT1 and PT2 members that the close physical proximity of team members could aid the work process, through allowing individuals to discuss work tasks and devise solutions more readily. Crucially, individuals claimed that if team members were dispersed and located elsewhere, this could create difficulties in communicating, monitoring work progress and coordinating project team work, contributing to experiences of intensity. Indeed, one PT2 Technical Lead emphasised the importance of face-to-face communication and discussion within project teams, due to the collaborative nature of work:

> I don’t know whether it’s just me being the Technical Lead and being responsible for things and having to make sure that everybody is following what we had set out but I would rather everybody was just here. You have discussions five, ten times a day when you’re really heavily into development. Certainly in design and development stages, it is a collaborative task. To have somebody completely remote at the end of the phone is not as good as when you’re doing it face-to-face. I like to work with white boards, visual aids, and bring people into discussions. If everyone’s just sitting over there, you can shout them over and you can get a better discussion than having someone at the end of the phone.

The experiences of these software professionals therefore suggests that close physical proximity can aid the professional software work process and is important in reducing experiences of work intensity.

'Agile' Methodology

Both project teams had adopted the 'Agile' methodology in an attempt to make work more manageable, allow individuals to meet smaller regular deadlines, and provide customers with more regular feedback. PT1 had split the project into phased releases following the development life cycle, meaning that different releases and life stages occurred simultaneously and overlapped. Similarly, PT2 was moving towards a more iterative approach, which involved breaking projects down into staggered phases.

Perspectives on the impact of the 'Agile' approach on intensity varied between PT2 and PT1. Comments made by one PT2 Senior Team Member supported senior management views that 'Agile' allowed for more frequent feedback which could be taken into account throughout the software life cycle. In addition, the 'Agile' approach of setting several shorter deadlines as opposed to one absolute deadline was regarded by one PT2 Technical Lead as allowing work effort to be distributed more effectively, helping to manage experiences of intensity:

> Whether the deadline you've been set is realistic or unrealistic, there's a tendency in software engineering to work slowly at the start and then increase your intensity as you head towards that deadline. We try to manage that by setting much shorter deadlines. If you say, "Right, we're going to deliver the project in its entirety in three months, but we'll break it into two week trunches", we'll hit a milestone every two weeks because people are never far away from a clearly defined milestone. The amount of effort they expend is more widely distributed across the piece, rather than the first two months being a bit of a holiday and then the last month being everybody working until twelve o'clock at night.

However, comments made by the PT1 project manager illustrated some of the repercussions of distributing work across several shorter cycles. For example, it was reported that the overlapping and simultaneous working on different releases and life cycle stages contributed to a constant level of intensity for PT1:

> With this project, it's been pipelined the whole way through, so there's never been a rest and recovery period for anybody throughout the project. The pressure has come and gone, it's not been hugely intense every day all the way through the two years but there's been constant pressure and it's been unrelenting. On a scale of 1 to 10 where 10's unbearably about to crack, 5 is where you'd maybe expect people to be and 1 is you're coasting, I would say everybody has been between 5 and 10 the whole way through. No-one's ever dropped into that kind of coasting. They're not getting that recharge opportunity and that's hard work. With it being a two year project, it's wearing, you know, people get tired (PT1 Project Manager).

Consequently, the PT1 project manager attempted to move team members to other areas of the life cycle at times to try and alleviate project fatigue.

'Agile' clearly has both positive and negative repercussions for software professionals' experiences of work intensity through affecting the distribution of work effort, deadlines and the pacing of release structures.

5.6 IMMEDIATE DETERMINANTS

Support Function

PT1 had a support rota whereby individuals assigned to support would have to take support calls and immediately shift their attention to these tasks. However, despite this rota, all PT1 individuals were periodically engaged in support during office hours. Similarly, whilst a separate Services (support) function existed in PT2, other

PT2 members were still engaged in providing support to clients. Four out of six PT2 members identified 'Providing Support' in the work diaries as contributing to work intensity. Support did not feature heavily as a factor contributing to work intensity in the work diaries for PT1 as members noted they were heavily involved in system testing during that time. However, in the interviews which followed, PT1 members identified support as an activity which could contribute to work intensity. Crucially, all PT1 and PT2 members reported in the interviews that the unpredictable nature of support calls from clients, in terms of when they could arise or how long they could take to resolve, could detract attention from work tasks and give rise to experiences of intensity:

> During the period after our FAT [Factory Acceptance Testing], when we are bug-fixing and preparing for the next phase, all our time is spent supporting their testing. In one week, I had thirteen hours of basic time I spent supporting their calls. And that was thirteen hours of time I didn't spend doing what I was supposed to be doing. You have to put more and more hours in and then you start your test preparation phase already constrained for time. It gets you frustrated, because you realise that you either need more resource, you need more time or you need to cut back on the number of phone calls that your client gives you and you can't do any of those. You're in this vicious circle. You have to juggle many different things and you can be juggling three different tasks at one time (PT1 Average Team Member).

Support activities therefore impacted on experiences of work intensity through detracting attention from other tasks and reducing overall work time in which to meet immovable deadlines.

Volume of Work

Data from the work diaries and the interviews showed that intensity for PT1 and PT2 members was located in the volume of tasks individuals had to perform, as opposed to the difficulty, complexity or nature of tasks. For example, whilst 'System Testing' was one of the most frequently cited activities in the work diaries as contributing to intensity for PT1 members, individuals commented that intensity originated from the accumulation of tasks that had to be constantly juggled, rather than from the nature of 'System Testing' itself. Moreover, working on different projects simultaneously was also identified by one PT1 Average Team Member in their work diary as contributing to intensity, through increasing volume of work and the need to juggle tasks. Furthermore, one PT2 Average Team member noted in their work diary that intensity on four out of five days was caused by "Having to carry out many different tasks and constantly being distracted onto other tasks" and "Having too much to do in one day". The data in the work diaries clearly illustrated that members of PT1 and PT2 had to manage a number of different, competing tasks with differing levels of importance, depending on seniority and levels of expertise. These observations suggested that further exploration into volume of work was necessary in the interviews conducted with software professionals, project managers and senior management. Diagram 5.2 depicts the demands placed upon software professionals as a consequence of having to perform several tasks simultaneously and navigate between these:

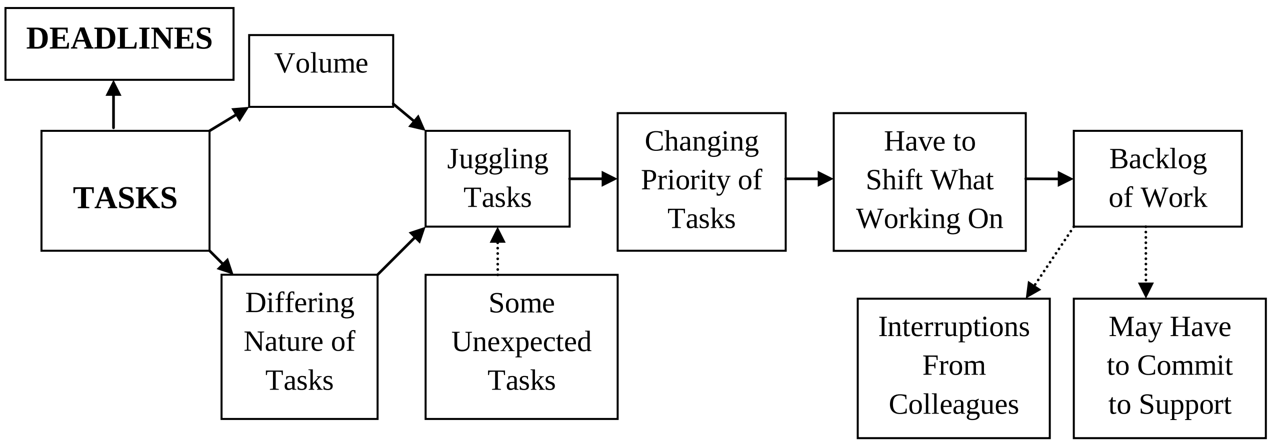


DIAGRAM 5.2 JUGGLING OF WORK TASKS

Interestingly, the work diaries illustrated that the number and variability of tasks performed by PT1 and PT2 members appeared to increase as individuals progressed within SpecSoft. For instance, at the beginning of their careers in the company, PT1 and PT2 members appeared to perform traditional 'programming' functions, such as development and writing code. Indeed, one PT2 Junior Team Member (see Appendix 21) was largely engaged in 'Task Analysis' (23%), 'Provide Support' (18%), 'Develop a Small Section of Software' (15%) and 'Database Administration' (3%) for the work diary week. However, with more experience, roles expanded to include a variety of areas such as business and system analysis; design; client interactions; development; documentation; testing; support; training for users; coaching; and recruitment and selection. For example, one PT2 Technical Lead (see Appendix 20) carried out 25 different activities across the overall week, which ranged from 'Develop Small Section of Software' (11%), 'Coaching Others' (9%), 'Create Test Schedule' (9%), 'Recruitment and Selection' (6%), 'Research Possible Technical and Design Approaches' (3%), 'Training for Users' (2%) and 'Meeting With Clients' (1%). The interviews supported the observations made in the work diaries that individuals with greater experience typically performed a greater variety of tasks. Crucially, it was commonly held by individuals within both PT1 and PT2 that work was typically more intense for more experienced individuals, due to the increase in responsibilities and allocation of more complex tasks.

Critically, all of the interviewees in PT1 and PT2 perceived that Technical Leads experienced higher levels of intensity, due to the volume of tasks and responsibilities carried out. These perceptions fitted a statement by the project manager of PT2 that Technical Leads were individuals with the most technical knowledge and often ended up with the greatest number of activities as a result. PT1 and PT2 Technical Leads provided confirmation, stating that this level required numerous tasks and responsibilities to constantly be juggled, contributing to experiences of intensity:

> In the role *I'm* doing, because you're Technical Lead, you're not always just doing the one thing. So, you're not always just developing. You're doing development, people constantly ask you

> questions, you've got customers asking you questions back and forward, you've got to get technical specs out the door. The TA [Technical Architect] role can be quite difficult. When you add support into the mix, it's just a nightmare, because not only am I support manager, I've got to make sure I get progress reports out on time, I've got to do support calls. And then to make it even worse, you might even be on more than one project at a time. So, it can be quite intense (PT2 Technical Lead).

The experiences of Technical Leads contrasted to that of Junior Team Members, who cited greater levels of buffering and lower levels of responsibility as reducing levels of intensity:

> I don't feel [intensity] to be honest. If you're a Technical Lead on a project, you've got to bear the responsibility so you may feel it a lot more. But because I'm just a [Junior Team Member], a lot of that's taken off me. It's not passed down to me, which is good (PT2 Junior Team Member).

Evidence from the work diaries for PT1 and PT2 indeed demonstrated that Technical Leads experienced greater volume of work and higher levels of responsibility. For example, individuals working at Technical Lead level within PT1 (see Appendices 14 and 15) tended to have more and a greater variation in tasks carried out in total across the overall week period, compared to those working at Junior, Average and Senior levels (see Appendices 14 to 19). Indeed, one Design Technical Lead carried out 21 different activities, compared to 14 by the Design Team Member (see Appendix 14). In Development, the two Development Technical Leads had 19 and 18 different tasks respectively (see Appendix 15), compared to 4, 5, 5, 10, 10, 13 and 15 for the Junior, Average and Senior Team Members (see Appendices 16, 17 and 18). The only Development Team Member with a higher number of activities (23 different activities) had the additional responsibility of Support Manager (see

Appendix 18). In Testing, the Test Technical Lead had 24 different activities, compared to 9 by the Average Team Member (see Appendix 19).

Similarly, within PT2, the two Technical Leads had higher numbers of work activities logged across the overall week period at 18 and 23 respectively (see Appendix 20), compared to 11 and 5 for Junior Team Members (see Appendix 21), and 15 for one Senior Team Member (see Appendix 22). There was slight variability in experience level to task ratio within PT2, as one Average Team Member had the second highest rate of overall activities logged at 23 (see Appendix 22). However, this discrepancy was explained by the number of different roles being performed by this individual within the project team.

Volume of work, opposed to complexity of work, can therefore be identified as contributing to intensity for software professionals at SpecSoft. More experienced software professionals, particularly Technical Leads, are likely to be subjected to higher levels of intensity through an increase in work volume, responsibilities and allocation of more complex tasks.

Specialist Knowledge

PT1 and PT2 members reported that interdependencies could lead to experiences of intensity both for knowledgeable individuals and team members lacking specialist knowledge in particular areas. Individuals with more expertise could experience interruptions from other team members requiring assistance, impacting on overall time available to perform other tasks and increasing intensity of work. However, if more knowledgeable individuals were unavailable at certain points, this could similarly intensify work for team members who lacked such expertise through increasing workloads, time estimates and effort levels to meet timescales. This collective perception was supported by the project manager of PT1:

> I would say that interdependency can have a huge impact – both on the individual and on the people who are waiting to speak to him

> [experienced individual]. The individual themselves can end up being the focus of a lot of interaction with people which prevents them doing what maybe they are ideally planning to do that day. But, correspondingly, for the group of four or five people who are all sitting one day waiting to get a bit of XXXX, if they're fifth in line for that conversation, they could be sitting there for a half day very distressed that their stuff's going late because they haven't managed to talk to XXXX.

Thus, for those software professionals with specialist knowledge, the possession of such expertise may increase the demands placed upon them and contribute to work intensity. However, at the same time, less experienced individuals who are seeking the support of a specialist may experience intensity through being unable to access such expertise.

Interruptions

Constant interruptions from colleagues, clients and support calls contributed to experiences of intensity for members of PT1 and PT2 at all levels through affecting concentration, disrupting work rhythms and reducing the time to meet impending deadlines. 'Informal Discussions With Colleagues' to provide clarifications or answer questions was one of the most frequently cited activities by PT1 in their work diaries as contributing to work intensity, suggesting that this area required further exploration in the interviews. Indeed, it should be noted that the work diary method was integral to uncovering the presence of interruptions within professional software work. For example, two PT1 members commented in their work diaries that intensity was caused by "continuous questions by team and colleague making things awkward!" and "amount of time spent investigating issues and helping others with their work". Individuals from both project teams provided further insight in the interviews which followed by commenting that it was difficult to concentrate on tasks without being interrupted continuously by colleagues:

> Interruptions can make it very difficult, because you think to yourself, “I’ve got this one really important task that I need to get through today, it’s something I need to spend my whole day on” and you never do. A lot of us sit and listen to music quite a lot when we’re working. I’ll pop my headphones on while I’m coding or whatever just for some background noise. What I realised recently is that I’m lucky if two or three songs can go by without someone tapping me on the shoulder to speak to me. So that means you’re interrupted, on average, every fifteen minutes, and that’s a *ridiculously* interruptive nature to work under. I was chatting about it with one of my colleagues yesterday and I was saying, “This is just ridiculous, you can’t work like that” (PT2 Average Team Member).

Constant interruptions from colleagues especially impacted on those working as Technical Leads in both project teams, due to their levels of expertise. Indeed, Technical Leads reported that whilst helping colleagues was an important part of the role, interruptions impacted on time to perform other tasks and greatly contributed to their experiences of intensity:

> Other people having problems can really impact on the intensity of your work because in a Technical Lead role, you can have an awful lot of people coming to ask you questions about their work. In a day, you might have ten people talking to you about their problems and each of them is a relatively quick chat. But every single time it breaks your concentration. If that’s your main role, that’s fine. But if you’re also utilised seventy or eighty percent on your own tasks and you’re meant to be spending seventy percent of your time developing something, then it can really add to the intensity of your work (PT1 Technical Lead).

Furthermore, another Technical Lead from PT2 drew attention to the impact of constant interruptions by commenting in the work diary that “Regular customer interruptions, colleague support and recruitment activities have all meant spending less time on development than hoped”. PT2 also emphasised that clients and support calls could give rise to further interruptions. For instance, one PT2 Technical Lead reported that clients could often interrupt work rhythms through asking team members to investigate problems or add extra tasks to projects, with no changes to existing deadlines.

Crucially, it should be noted that software professionals were unable to avoid interruptions or simply put up a ‘do not disturb’ sign, in that daily interactions with team members, project managers and clients were necessary due to the collaborative and interdependent nature of work. In addition, individuals were not immune to interruptions from project team members, managers or clients based elsewhere, due to the availability of technological mediums such as e-mail, instant messenger and phone calls. The impact of these constant interruptions on software professionals was summarised by one PT1 Average Team Member:

> You’ve got your own task to complete and, again, this comes back to the thing about being able to concentrate on fixed chunks of work for extended periods of time, uninterrupted. If you’re trying to do a chunk of work, concentrate on it and you keep getting interrupted, “How do you do this?”, ‘”Where, where would you put this?”, “How is this done?”, then that interrupts your own work and that can’t help but affect you.

In summary, interruptions contributed greatly to software professionals’ experiences of work intensity at SpecSoft through affecting concentration, disrupting work rhythms and reducing the time in which to complete tasks. However, the experiences of software professionals illustrate that interruptions from colleagues, project managers and clients cannot be avoided due to the collaborative and interdependent nature of professional software work. In addition, insufficient staffing levels can be seen to further exacerbate the impact of interruptions through increasing volume of work to be completed with set deadlines.

Internal Motivation to Work

When asked what they liked and disliked about work, all members of both PT1 and PT2 appeared to like the problem-solving, analytical and technical side of professional software work, especially writing code. For example, as illustrated by one PT1 Technical Lead:

> In general, I really enjoy solving technical problems. I get an awful lot of satisfaction from doing that and just writing. If you write something that's technically difficult and then you get to the end and you see the end result and it works well and the client's happy, that's great. I even enjoy testing of code because sometimes if you see something that's really good, you just get a great feeling about it.

Individuals across both PT1 and PT2 disliked support (detracting attention from other tasks), volume of work, juggling tasks and documentation. PT1 members additionally disliked client ambiguity, unclear requirements and phone calls with clients, whilst PT2 individuals disliked conducting interviews for recruiting new members of staff (detracting attention from work), unrealistic timescales and working with older technologies.

All interviewees in both project teams appeared to value the importance of quality, commitment, pride in one's work, willingness to 'go the extra mile', supporting work colleagues and enthusiasm for work. These characteristics seemed to instil an internal motivation in individuals to perform and undertake work to a high standard. Indeed, one senior manager perceived that software professionals could even create experiences of intensity for themselves through their own personal drive, regardless of pressures placed on them by others:

> It's almost like, you can choose to make it more intense by personal choice, rather than by someone above you putting the pressure on you in a sense…"what *I* want to achieve". And you set your own aspirations. So, although there's maybe not someone above you pressing you, because you want to achieve something, you can create the aspirations and therefore the intensity below you.

Performance appraisals, status reporting and code reviews (the only forms of metrics applied) were viewed and utilised by members of both project teams as part of their normal work routine and were not identified as sources of intensity. Indeed, one PT2 Junior Team Member suggested that code reviews were beneficial in allowing team members to evaluate the contents of their own work more effectively:

> Well, basically a code review is where you've worked on a piece of work and the Technical Lead or someone with more experience or whatever will basically look through your code and ensure that everything is as expected. It gives you the confidence that you're doing your work correctly and if there's anything that you need to change, you can.

PT2 members had also initiated informal workshop sessions themselves to examine performance at the end of projects in order to identify what had been effective and areas for improvement, demonstrating the internal motivation to work.

Internal motivation to work therefore has implications for work intensity through encouraging software professionals to perform work to a high standard and exert greater effort levels when necessary. Performance metrics can be viewed as part of the normal work routine and bear no impact on experiences of work intensity at SpecSoft.

Breaks

Individual approach to taking lunch or tea breaks and the impact on work intensity was similar in both project teams. All members of PT1 and PT2 stated that they could be less likely to take a lunch break or other breaks when working on tasks which were challenging and intellectually stimulating, as this could interfere with their 'train of thought'. Crucially, taking a break in this situation could increase intensity due to the amount of time needed to access the same thought processes:

> If you've got momentum, you don't want to break the train of thought, you don't want to move away from your desk. You want to keep doing what you're doing because there's a fear that when you come back, you might have lost that momentum and things might be a wee bit harder. If you're into it, the amount of work you're completing, it feels like a breeze, it feels like little effort. It might be technically quite difficult but you've got a momentum going and you're answering those questions (PT2 Technical Lead).

Nevertheless, all PT1 and PT2 members equally recognised that it was important to take a lunch break or other breaks, in order to have time away from a problem and create a fresh outlook:

> I'll know in the morning how intense my afternoon's going to be, so I may work through my lunch to try and reduce the intensity. But if you don't take a break as well, you're not working at your full capability anyway. I think it's good sometimes to just take a break, refresh, pull yourself together, give your brain a chance to recharge and think about something else. I think you can get so focused and into what you're doing sometimes that it's quite hard to step back and look at the big picture (PT2 Junior Team Member).

Breaks can therefore influence experiences of work intensity depending on the nature of tasks and the approach taken by software professionals themselves.

Contractors

There was unanimity amongst software professionals and agreement from the PT1 project manager that contractors were not likely to be treated in the same way as permanent team members with regards to work tasks. Contractors tended to be hired for their specific knowledge and were expected to work on a specific task, rather than the project team having to train a permanent member of staff in particular areas. For example, one PT2 Junior Team Member with contractor status (see Appendix 21) carried out five activities in total during the work diary week (the lowest total in PT2), predominantly focusing on ‘Develop a Large Section of Software’ (73%), as well as ‘Travelling To/From Work’ (14%), ‘Lunch’ (7%), ‘Planning/Timetabling Work for the Day’ (4%) and ‘Fixing Faults’ (2%). This was in comparison to the work diary for one permanent PT2 Junior Team Member (see Appendix 21), who carried out 11 activities in total, ranging from ‘Task Analysis’ (23%), ‘Provide Support’ (18%), ‘Develop Small Section of Software’ (15%) and ‘Informal Discussion With Colleagues’ (7%).

Contractors could also be hired to carry out tasks that were unpopular with permanent employees. Two PT1 members and two PT2 members perceived that work could be less intense for contractors due to lack of permanent employment status. For example, one PT2 Technical Lead professed that contractors could experience less pressure, compared to permanent employees who could feel more responsibility:

> I think if you’re a full-timer, as opposed to a contractor, it can be more intensive. Most contractors work core hours and that’s it, on the basis that as soon as they’re out of hours, they’re not getting paid, so therefore they’re not working. That’s just the way contractors are and if I was a contractor, I would be doing the exact

> same. Whereas if you're full-time, then you might have to work a lot more hours and not necessarily get paid for it. But you've got – or certainly I do feel – a lot more responsibility because you're a full-timer to make that extra effort.

Furthermore, permanent team members such as Technical Leads could potentially face interruptions to work rhythms in order to provide background information on systems to contractors, contributing to experiences of intensity:

> The particular project that we've been on, there's been quite a high turnover of contractors working on it. The Technical Leads in particular spend an awful lot of time explaining to contractors what it is they actually have to do and what the big picture is. They [contractors] might be able to cut the code quicker but actually understanding the system is more difficult (PT1 Senior Team Member).

Contractors may therefore be expected to experience lower levels of work intensity compared to permanent employees due to differences in personal investment, responsibility for tasks and knowledge on systems.

5.7 FACTORS HELPING TO OFFSET WORK INTENSITY

Managing Relationships With Clients

Factors such as relationships with the client, ensuring clients adhered to specifications and recording information in e-mails were commonly recognised by both project teams as helping to alleviate the pressures exerted by clients and reducing work intensity. For example, PT1 members and the project manager stated that the level of intensity exerted was dependent on the relationship established with the client. Thus, if a good relationship existed, there could be room for compromise or discussion on issues, whereas if this relationship was poor, discussions could be difficult:

> You've got the ones that expect you to jump right now and you really want to please them, 'cause you'll know they'll just moan the face off you, or it'll affect the relationship. I've got other ones that I've got a really good relationship with. I think if you've got a good relationship with a client, they'll be more accepting of you saying, "Well, I can't do it immediately, can you wait for it?", as long as they know that they're eventually going to get there. Whereas some clients just want everything done immediately and you don't have that same relationship where you can sit and say, "Well, I've actually got all this other stuff on", you don't think they'll be as accepting of that explanation (PT1 Average Team Member).

One PT2 Technical Lead suggested that the impact of clients on intensity could be managed more effectively with experience and through relationships built up over time:

> When you first start out as a graduate, you worry about what people think, you're conscious about the fact that you're working for a customer and they're essentially paying your wages. And you're very unsure about the relationship you have with them. As you get more experienced, you get more comfortable in what it is that you're giving to them and what it is that you're offering to them and you become more comfortable with how it works. You build up relationships as well over time. A lot of our customers I've known for a few years and that helps enormously, opposed to somebody that you've just met.

PT2 members stated that measures to manage client indecision were necessary to prevent tasks escalating and profits from suffering. For instance, one PT2 Senior Team Member stressed that the team was placing greater emphasis on clients clearly defining specifications at the outset and ensuring clients adhered to these. Indeed, it was argued that early clarification and agreement on specifications could assist PT2 in presenting the case for extra financing or deadline extensions if clients later requested changes. Documenting information in e-mails when communicating with clients was further recognised as an important tool for securing accurate records of discussions and preventing digression from established objectives:

> We use e-mail a *lot* between us and the customer, because it's great to have something written down. You wouldn't believe how many times where the customer will say something and they'll come back six months later, have changed their mind and swear blind that they've been saying that all along. So we always insist on e-mails to confirm things, you've got a record of it (PT2 Technical Lead).

Establishing relationships with clients, ensuring clients adhere to specifications and keeping records of client discussions can therefore help software professionals manage the extent to which clients contribute to work intensity.

The Approach Taken By Project Managers and Technical Leads

Individuals from both project teams expressed that experiences of intensity could be dependent on levels of monitoring, approaches to planning projects and the amount of time given for tasks. Crucially, project managers had the potential to increase or reduce software professionals' experiences of work intensity, through their power over deadlines, planning of workloads, allocation of tasks and the ability to buffer workers from client complaints and interference. In addition, Technical Leads could influence software professionals' experiences of work intensity, due to their role in allocating tasks to team members, monitoring work progress, resolving problems and reporting information to project managers. However, members of PT1 and PT2 noted that Technical Leads were generally supportive due to the collaborative nature of work. In addition, intensity could be greatly influenced by the extent to which a leader (project manager or Technical Lead) was supportive and willing to consider team member perspectives and opinions.

In this sense, the approach taken by project managers and Technical Leads can have implications for software professionals' experiences of work intensity.

Personal Pro-Activeness

Senior management provided insight into the role software professionals themselves could play in managing the impact of deadlines on intensity. For instance, drawing client attention to delays in providing information could potentially alleviate the pressures software professionals faced in meeting existing timescales:

> Some customers will pose a deadline where you think, 'That's not achievable. We're so good at working in partnership with the customer but sometimes we're not good at saying, "Actually, you were three days late in giving us that information so that deadline needs to slip three days because of it". That would take the pressure off us but sometimes we don't create ourselves that space. So, sometimes the customer does put more pressure on us but sometimes we put the pressure on ourselves (Senior Manager).

In addition, senior management and the PT1 project manager commented that software professionals themselves had to recognise the existence of client downtime schedules and the importance of meeting deadlines:

> Sometimes we think, "If it slips away, it's only a week in a six month project". What we don't know is there are downtime schedules and if we miss that week, it could be three months before the customer can put that in again. If we find something like that

> out late and don't understand the customer's contingency, we've created our own intensity. Or if two weeks into a project we're a week late and we think, "That's okay, we'll catch it up later". Actually, the time to catch it up is as soon as you're late, because then you've cleared it and you're on an even keel. Whereas from the next six weeks, if you're always a week behind, you've created that intensity yourself (Senior Manager).

In this sense, pro-activeness and self-awareness can help software professionals manage personal experiences of work intensity.

Level of Experience

Whilst deadlines were a predominant source of intensity for all individuals, Senior and Technical Lead members of PT1 and PT2 recognised that level of experience could help to manage the onerous impact of deadlines. For instance, experience could help software professionals recognise that it was rarely catastrophic if work did not go exactly to plan:

> They're [deadlines] probably the main impact, in that you have a milestone to work to. It's very rarely catastrophic if you don't make one, but that's something you learn through time as well. When you're a new developer you're like, "Oh no, I've got to hit this deadline" and then you work a lot of overtime. With experience, you realise that the world isn't going to end if you don't meet it. I think, with experience, you learn that one day isn't going to make a difference. If you think an extra day or working a weekend can make you meet that deadline, you do it, but it's not the end of the world if you don't make it (PT2 Senior Team Member).

Experience can therefore help software professionals manage the extent to which deadlines contribute to experiences of work intensity.

CONCLUSION

This chapter has identified sources of intensity at SpecSoft and documented the experiences of software professionals. De-regulation, privatisation, market buoyancy, downturn and world events shaped experiences of work intensity through affecting availability of project work and acceptance of project contracts. In addition, staffing arrangements had implications for volume of work, levels of responsibility and project team coordination within SpecSoft. Technological developments, in terms of improvements in computer speeds and technological mediums, can be seen to have intensified professional software work. Notably, these developments can be seen to have increased work volume for software professionals over time by enabling functions to occur more quickly and in parallel. In turn, specialist software firm status, small company size and concentration on industry niches impacted on levels of visibility, responsibility and encouraged emphasis to be placed on lean staffing levels and multi-tasking.

At the internal organisational level, clients, deadlines and leadership influenced experiences of work intensity at SpecSoft. Clients clearly contributed to experiences of work intensity for software professionals through their unwillingness to compromise over deadlines, indecision and changing of priorities. Furthermore, the need to deliver projects quickly and competitively to external clients meant that deadlines appeared to be tight and largely immovable. Deadlines therefore represented a key source of intensity for individuals through the need to juggle tasks and exert greater levels of work effort in order to meet targets. In addition, the extent to which leaders understood the realities of project work influenced software professionals' experiences of work intensity. Facilitative and constructive leadership styles were also recognised as being more effective at managing professional software work.

Forms of work organisation further affected experiences of work intensity at SpecSoft. Close physical proximity clearly aided the professional software work process and reduced experiences of work intensity. In addition, 'Agile' represented an important step-change to the professional software labour process at SpecSoft through changing the distribution of work effort and management of release structures.

The combination of market dynamics, technological developments, firm characteristics, internal organisational factors and work organisation contributed to software professionals' experiences of work intensity at the immediate level. Unpredictability of support activities detracted attention, increased work volume and reduced work time to complete existing workloads. Volume of work, as opposed to difficulty and complexity of work, clearly contributed to experiences of work intensity, particularly for more experienced individuals. Specialist knowledge and heterogeneity within project teams clearly impacted on experiences of intensity both for more knowledgeable individuals and those requiring assistance. Interruptions can be identified as a significant finding due to their impact on software professionals' concentration, volume of work, and ability to complete prescribed tasks. In addition, internal motivation to work appeared to encourage individual self-supervision and management of work effort, influencing the approach individuals adopted with regards to breaks.

Factors such as managing relationships with clients, willingness by project managers and Technical Leads to consider team member perspectives and opinions, personal pro-activeness and level of experience can be identified as helping to offset software professionals' experiences of work intensity.

The following diagram (Diagram 5.3) illustrates the hierarchy of factors which shape experiences of intensity for software professionals at SpecSoft:

MARKET DYNAMICS

De-Regulation
'Booms' or 'Busts
World Events
Downturn or Buoyancy of Markets
Staffing Arrangements

TECHNOLOGICAL DEVELOPMENTS

Improvements in Computer Speed
Technological Mediums

FIRM CHARACTERISTICS

Specialist Software Firm
Small Company Size
Different Business Sectors

INTERNAL ORGANISATIONAL FACTORS

Clients
Deadlines
Leadership Style

WORK ORGANISATION

Physical Proximity
'Agile' Methodology

IMMEDIATE DETERMINANTS

Support
Volume of Work — Juggling Tasks; Level of Experience
Specialist Knowledge
Interruptions
Internal Motivation to Work
Breaks

FACTORS HELPING TO OFFSET WORK INTENSITY

Managing Relationships With Clients
The Approach Taken By Project Managers and Technical Leads
Personal Pro-Activeness
Level of Experience

DIAGRAM 5.3 SOURCES OF INTENSITY AT SPECSOFT

CHAPTER 6 INSOFT: SOURCES OF INTENSITY

INTRODUCTION

InSoft was based in a large building in a city-centre business park. The entrance of InSoft opened on to an expansive reception area, with two seated areas for guests and barriers through which to enter or exit. Beyond the barriers were the company floors (around six to eight in total) accessible by lifts, a cafeteria, a shop and two large seated areas. The lifts and each floor level had clear, glass-sided panels, making it possible to view individuals at work. Company mantras in large letters were hung from the ceiling on various levels referring to corporate service, customer standards, confidence in products and teamworking. The in-house IT section of InSoft was dispersed across one large open-plan level which included office space, meeting rooms, an informal seating area and a kitchen. There was a constant low noise on entering the open plan office floor which came from phone calls, conversations and lifts.

This chapter will provide an in-depth discussion of factors which contributed to experiences of intensity for software professionals at InSoft. As in Chapter Five, a generally applicable model of market dynamics, technological developments, firm characteristics, internal organisational factors, work organisation, immediate determinants and factors which help offset intensity has been utilised to explain the layers which shape experiences of work intensity. The broader context of privatisation and increased competition necessitated the development of responsive business strategies at InSoft. Technological developments in computing speed and communication methods were identified as intensifying professional software work, through increasing work volume and enabling activities to occur simultaneously. In addition, technological mediums, such as phone calls, e-mails and instant messenger contributed to daily experiences of work intensity through creating interruptions to work rhythms. However, the directness and personal utilisation of technological mediums by individuals appeared to influence the extent to which these methods generated interruptions and contributed to work intensity.

Firm characteristics, such as organisational type, large company size and concentration in the telecommunications sector had implications for responses to market pressures. Leadership style was identified at the internal organisational level as placing greater emphasis on managerially-oriented software engineering functions, stimulating the separation of ‘high value’ planning activities and ‘low value’ execution areas with implications for work organisation. The geographical dispersal of project teams across the UK and India was significant in contributing to experiences of intensity due to cultural differences, communication difficulties and poor management of agreements with offshore contractors. ‘Agile’ methodology was identified as both engendering and helping to manage intensity, due to the incremental nature of releases.

Work organisation contributed to daily experiences of intensity in the form of immediate determinants, with volume of work and possession or inaccessibility of specialist knowledge impacting on software professionals. In addition, whilst deadlines, clients and breaks influenced software professionals’ daily experiences of intensity, the impact of these factors varied depended on job role, association with client deliverables and the particular stage in the development life cycle. Factors such as internal knowledge of the business, the approach taken by project managers and Technical Leads and level of experience were further identified as helping to offset experiences of work intensity for InSoft software professionals.

Finally, this chapter summarises sources of intensity at InSoft in diagrammatic form, in order to illustrate the hierarchy of factors shaping experiences of intensity at the individual (i.e. software professional) level.

6.1 MARKET DYNAMICS

De-regulation, privatisation, financial pressures and broader societal trends were considered by software professionals, project managers and senior management to have impacted on the in-house IT department. De-regulation of the telecommunications industry had increased levels of competition, placing pressure on InSoft to develop products more quickly and within tighter timescales. Privatisation had prompted InSoft to focus on developing faster, efficient and cheaper processes in order to manage costs and raise profits. Increasing consumer demand for products and services to be delivered at lower costs within shorter timescales had further encouraged the development of more responsive business models at InSoft.

6.2 TECHNOLOGICAL DEVELOPMENTS

Improvements in Computer Speed and Technological Mediums

Senior management suggested that developments in technology had intensified professional software work and increased overall volume of work. For example, developments in computer speed and the advent of technological mediums such as e-mail, mobiles and laptops were identified as enabling functions to occur more quickly and in parallel, consequently increasing the demands placed upon software professionals:

> We all have mobile devices, we've all got remote access cards for laptops. Things come to you quicker and they can come to you at any time of day or night. That makes it very different to when I started [as a Software Engineer]. When we made a release of software, we cut it to tape and posted it to someone. There was as a several day delay while it made it to the other end of the country and someone unpacked it, loaded it up and started to test it. That

> delay isn't there now. You tell someone "We think we've fixed this bug. We've uploaded it onto the test server. Do you want to start testing it?". That is good and bad, because you get more immediate feedback on what you do and the actual throughput as an organisation is much improved. But I would say it's much more intense for the workforce.

Developments in ICTs can therefore be recognised as providing the potential for activities to be performed with greater speed, increasing work volume and subjecting professional software work to intensification.

Interruptions

Software professionals at InSoft habitually worked on different tasks and projects, due to the geographical dispersion of teams. Informal face-to-face discussions were therefore not commonplace at InSoft and did not emerge as a factor contributing to intensity. In contrast, interruptions came mainly from technological mediums such as e-mails, phone calls and instant messenger, which were used to communicate with dispersed team members and contributed to experiences of intensity. The work diary method provided valuable insight into the frequency with which InSoft software professionals and offshore workers utilised technological mediums. For example, one offshore Business Analyst (see Appendix 24) spent 16% of the working week 'Receiving Phone Call(s)', whilst one Senior Developer (see Appendix 25) spent 10% on the same activity. Crucially, phone calls were considered by the majority of InSoft software professionals as being the most intrusive form of interruption, requiring an immediate answer and resulting in loss of one's 'train of thought'.

E-mails and instant messenger were most commonly used to maintain contact with other team members. For instance, the work diaries illustrated that one offshore Indian Business Analyst (see Appendix 24) and one Engineering Technical Lead (see Appendix 28) devoted considerable time in the overall working week to 'Reading/Replying/Composing Work E-Mail(s)' (30% and 24% respectively) and

‘Using Messenger’ (17% and 14% respectively). However, software professionals possessed varying perceptions on the impact of e-mail and messenger on experiences of intensity. For example, some software professionals stated that e-mails and messenger were less intrusive and allowed them to choose when to respond, helping to manage impacts on intensity:

> IM [Instant Messenger]’s not so bad, because they [team members] are not in the same office as you and they don’t know if you’re actually sitting at your desk or not. I personally feel free to ignore IM until I have time to deal with it. The same with e-mail. It’s not immediate, in that you’re not forced to answer it like phone calls. I think e-mail and IM from that point of view are a good thing, because you can prioritise, you can force them to wait (Software Engineer).

Conversely, other software professionals felt under pressure to respond to e-mails and instant messenger promptly, which contributed to intensity. Indeed, senior management reported that technological mediums such as e-mail had increased the frequency of interruptions and intensified work for software professionals:

> I think it comes back to the technology. They interrupt you. We always had telephones, so it was never completely without its interruptions but you used to generally do your tasks in a fairly serial order. We didn’t even have company-wide e-mail when I started; we had e-mail within a unit. Largely speaking, when you sat down to do your task for the day, apart from some technical interrupts, queries from your colleagues – who, at that point, were usually with you – you carried on during the day. Now, there’s a lot more interruptions and different paths taken or “We told you to do that. Can you do this instead?”. It’s a lot more chaotic.

In summary, the geographical distribution of project teams at InSoft meant that software professionals had to utilise technological mediums regularly, in order to communicate with team members. Phone calls, e-mails and instant messenger could create interruptions to work rhythms and contribute to experiences of intensity for individuals. However, at the same time, the extent to which individuals experienced intensity appeared to be influenced by personal utilisation and directness of methods.

6.3 FIRM CHARACTERISTICS

Organisational Type

InSoft had previously performed software engineering activities across the development life cycle, including requirements gathering, design, development and testing. However, senior management reported that the expansion of the software industry and the availability of packaged software had enabled InSoft to purchase and customise software packages, rather than produce systems internally. Crucially, these trends had triggered changes in the software engineering function at InSoft, removing the need for all software activities to be performed internally and reducing levels of technical expertise:

> In the time I've been in the industry, it has changed quite dramatically. For firms that are not pure software engineering, the nature of work has changed quite a bit because a lot more functionality you buy off the shelf. Some of what we ask people to do is arguably less technical than it was fifteen years ago. We used to write stuff down to the letter, we'd write the graphical user interface, we would write lines of code. You don't do any of that stuff anymore; you call on a package. The pooling together of a number of software packages is quite a different role (Senior Manager).

In this sense, whilst software engineering and technical skills can be recognised as the main areas of expertise within specialist software firms, technological developments can be deemed to have enabled InSoft to place less emphasis on internal technical expertise, due to the core organisational focus not including software engineering.

Company Size

Senior management and software professionals across all areas argued that company size could have an impact on resourcing, visibility and general decision-making, resulting in work intensity. Participants commented that the effort exerted by software professionals within smaller companies could potentially be linked more visibly to overall profitability and success. However, this was deemed to be in contrast to the situation in larger organisations, where it could be difficult to ascertain the relationship between worker effort and overall organisational outcomes. Smaller organisations were also judged to have fewer available resources, meaning that software professionals could be responsible for a wider array of tasks and consequently experience greater levels of intensity:

> I wouldn't like to work for a small company, because I think things would be more intense. I think rather than having a specific role, you'd have more general roles, which would increase the complexity and it would also increase the amount of time you've got to spend doing it. There's more spread of the workload in a big company. You'd have to be more of a 'jack of all trades' in a much smaller company (Business Analyst).

In addition, participants perceived that lower levels of resourcing within smaller firms could give rise to interdependencies within teams, making work more intense for those with specialist skills.

Nevertheless, experiences of intensity were identified as being equally prevalent for software professionals working within larger organisations due to greater bureaucracy and an emphasis on formal procedures. These conditions could make it difficult to introduce new ideas or understand processes, increasing the time in which to perform tasks:

> My experience of smaller companies is that it's perhaps a little easier to get things done quickly. You don't have to work out how you do it, you're pretty much aware or somebody will be able to tell you exactly how to do it and therefore you can lash at it. Whereas here, there's a bit more research involved, a few more hoops to jump through and that can increase time pressures (Software Engineer).

These observations were supported by project managers, who reported that set procedures could make it difficult to deal with situations promptly, which consequently contributed to frustration and delay for software professionals:

> A lot of edicts have come out of [InSoft] in the last few years where it's absolutely the word of law that you cannot work on something unless there is agreed financial authorisation for it. We get things that go wrong in systems and if it was a machine that put milk in cartons, they would be standing there watching all the milk pouring away and the cartons falling. They're not able to do anything because they have to wait. For teams that work on platforms, that would be extremely frustrating (Project Manager).

In different ways, company size was therefore deemed by participants to have implications for intensity. Lower levels of resourcing and greater multi-tasking were identified as contributing to intensity for software professionals working within smaller organisations, whilst bureaucratic procedures were seen to be a source of intensity for those based within larger organisations.

6.4 INTERNAL ORGANISATIONAL FACTORS

Leadership Style

Software professionals, project managers and senior management reported that market dynamics and firm characteristics had brought changes in leadership style and strategy. For example, one project manager claimed that IT agendas were increasingly being driven by sales and marketing teams who were motivated by profit margins and time to market. Leaders were also considered to place greater emphasis on managerial-software related functions which were deemed to be more profitable, as opposed to technically-oriented roles. Consequently, all participants observed that there were few routes for software professionals interested in areas such as development and programming to pursue:

> [InSoft] is looking more for project management skills, delivery management, higher level design and solution building tasks. The sort of skill where I'm developing things is not sought after at all just now. The people who decide what other people are doing tend to have gone up the management chain and they value the skills of managing and organising things. When they see skills such as developing software they think "Oh, anyone can do that. We'll contract people in at a really cheap rate because they'll save us money. We don't need our own people to do that because it's not as important (Software Engineer).

In order to progress, software professionals therefore had the option of moving into planning and managerial roles, such as Business Analysis, Requirements Gathering and Design, or had to ensure they became a 'guru' in their chosen technical software engineering area. Transformations in the in-house IT function therefore had ramifications in terms of the nature of software engineering activities performed within InSoft.

In addition, software professionals and project management stated that higher levels of leadership at InSoft were ‘dictatorial’ in nature, adopting a demand and control approach. For example, one software professional provided insight into the ‘dictatorial’ leadership style at higher levels referred to by participants:

> It gets more dictatorial the higher up you go. We’ve got certain targets that have to be met by Christmas time and I heard one of our very senior managers saying on a webcast to thousands and thousands of [InSoft] employees “This is not a democracy. You will do…”. I was sitting watching this webcast and saw about a dozen people going, “What?!?!”. So that’s what you’ve got to deal with. It’s very, very different from your line manager who lets you get on with your day-to-day work (Business Analyst).

However, all participants observed that leadership styles at InSoft varied between the ‘dictatorial’ approach adopted at higher levels to the facilitative, supportive styles evident at project management and Technical Lead levels. It should be noted that Technical Leads played an important part in managing and monitoring project team work. Thus, whilst project managers were responsible for the overall management of project teams, Technical Leads were in charge of allocating tasks to team members, providing estimates, monitoring work progress and reporting information to project managers. In this sense, the leadership style adopted by project managers and Technical Leads was deemed to have implications for software professionals’ experiences of work intensity.

Crucially, the emphasis leaders placed on managerially-oriented software functions to the detriment of technically-oriented areas had implications for the type of software engineering activities which were performed within InSoft and shaped experiences of work intensity. In addition, whilst leadership styles differed between those operating at higher levels and project management, supportive and facilitative approaches were seen to be more effective in managing software professionals’ experiences of intensity.

Separation of 'High Value' and 'Low Value' Software Work

Leadership preference for managerially-oriented roles had prompted the separation of 'high value' planning areas such as Business Analysis, Requirements Gathering, Documentation and Design from 'low value' execution areas such as Development and Programming. Official information on the company internet site stated that software professionals at InSoft were engaged in the most profitable and revenue generating work, with 'lower value' work being outsourced or offshored, following industry best practice:

> [InSoft's] IT professionals are being provided with the most up-to-date technology and skills training to carry out the most profitable work, while lower-value work is outsourced. More than 2,400 IT professionals have moved into revenue-generating ICT work. On top of this, [InSoft] now has a core set of strategic IT suppliers to replace contractors and non-strategic suppliers. This is supported by adopting offshoring as industry best practice with the use of high calibre, low cost resources from around the world.

The separation of software engineering roles into 'high value' and 'low value' areas led participants to conclude that software engineering at InSoft in the technical sense was a 'dying breed'. Indeed, the work diaries were instrumental in confirming these perceptions, with only four InSoft work diary participants (see Appendix 25 'Senior Developer'; Appendix 26 'Software Engineers'; Appendix 28 'Engineering Technical Lead') out of twelve still fully involved in 'low value' areas such as development and programming.

The work diaries helped identify that InSoft software professionals were largely focused on tasks pertaining to their particular job role (as detailed in the taxonomy in Section 4.3), rather than performing a variety of roles across the development life cycle. Individuals working as Software Engineers were generally engaged in different types of tasks from those working as Business Analysts. For example, Business Analysts were largely engaged in tasks based around reading/replying/composing work e-mails, meetings, making and receiving phone calls, obtaining understanding of clients' systems and interactions with clients and other software engineers. Indeed, one permanent, full-time Business Analyst (see Appendix 23) spent the majority of the working week engaged in 'Reading/Replying/Composing Work E-Mail(s)' (31%), 'Meeting with Colleagues' (14%), 'Planning for Meetings' (11%), 'Making Phone Call(s)' (9%) and 'Using Messenger' (6%). Similarly, one offshore full-time Business Analyst (see Appendix 24) was focused on 'Reading/Replying/Composing Work E-Mail(s)' (30%), 'Using Messenger' (17%), 'Receiving Phone Call(s)' (16%), 'Meeting With Colleague(s)' (4%) and 'Meeting With Manager(s)' (3%).

Whilst Business Analysis can be recognised as an important part of software engineering in terms of analysing business needs and gathering requirements, Business Analysts at InSoft did not appear to have exposure to technically-oriented software engineering tasks such as development, writing code, testing or providing support. This was in contrast to individuals working in more technical roles who dealt with tasks relating to 'Developer' or 'Software Engineer' job roles in the taxonomy, such as development, fixing faults, testing and writing progress reports. For instance, one permanent, full-time Senior Developer (see Appendix 25) was engaged in 'Fixing Faults' (21%), 'Provide Support' (21%), 'Reading/Replying/Composing Work E-Mail(s)' (14%), 'Develop Specific Section of Software' (8%) and 'Re-Testing of Bugs' (5%) across the work diary week. In addition, one full-time permanent Software Engineer (see Appendix 26) was involved in 'Develop Specific Section of Software' (40%), 'Fixing Faults' (20%), 'Keeping Up-To-Date With Technology/Advances' (5%) and 'Informal Discussion With Colleague(s)' (2%).

Work activities performed by the three Technical Leads varied slightly, due to the different areas these individuals worked within (see Appendices 28 and 29). However, the work diaries for Technical Leads from Architecture, Engineering and Support exhibited similarities, in that these individuals performed a wide range of activities and possessed the highest overall number of activities out of all work diary participants (22 each respectively). For instance, the Engineering Technical Lead (see Appendix 28) was engaged in 'Reading/Replying/Composing Work E-Mail(s)' (24%), 'Informal Discussion With Colleague(s)' (8%), 'Fixing Faults' (3%), 'Research Possible Technical and Design Approaches' (3%), 'Coaching Others' (2%), 'Document and Code Reviews' (2%) and 'Define Requirements and Specifications (1%).

A recurring theme in the interviews involved rumours concerning the possibility of InSoft's re-introduction of offshored development work to the in-house IT department. However, most of this information appeared to have been generated simply from what individuals had 'heard' from others. Certainly, one project manager commented that no reference had been made as to how development work could actually be re-introduced, making it difficult to see how this could occur in practice:

> There's talk of trying to set up something again that is like a regional centre. They like the idea of it, but in practice, who's going to fund it, who's going to run it and are people willing to contribute to it? If you don't get these three things, it won't happen.

Indeed, software professionals, project managers and senior management emphasised that very few individuals within InSoft were still involved in technical areas of software engineering, with potential repercussions for the future of the in-house IT section in general:

> Recently there's been a lot of outsourcing of work, which I've always said is a terrible thing. We're outsourcing all of our development, so we don't have any developers in [InSoft] anymore. We have what they call 'higher level' designers. But we can't train people to be higher level designers without letting them go through learning about development and how to do the 'lower level' thing.

> So, basically, we're doing ourselves out of a job. I've expressed that to people high up but nothing happens (Software Engineer).

Transformation of the in-house IT department and the separation of 'high value' and 'low value' software activities can therefore be seen to have had implications for the nature of roles performed in-house. In addition, this broader division of labour influenced forms of work organisation at InSoft, shaping experiences of intensity for both on-site and offshore software professionals.

6.5 WORK ORGANISATION

Geographical Dispersion of Teams

Project teams at InSoft were extremely fragmented and geographically dispersed, with team members located across the United Kingdom and in India. Most teams had only one or two team members located in the central Glasgow office, with other team members located elsewhere. Many software professionals had difficulty in understanding and explaining team structures and identifying the geographical location of members, due to their complexity. Indeed, at times, individuals found it hard to identify the actual number of individuals within the team:

> Now, I've said twenty [team members]. I'll be honest, it varies from week to week. It's very difficult to keep track. Even our line manager XXXX sometimes doesn't know how many people he's got in his team. You've got XXXX at the very top and he basically line manages all the Glasgow people. All the offshore section have got a girl called XXXX in charge of them. They report to her and she reports to [Glasgow line manager]. So with that extra leg, sometimes things get lost in translation (Business Analyst).

On-site individuals were spread out across a large open-plan area in rows of desks with sections of six which were split into three desks facing each side. Flexi-desking was in practice for half of the office floor, where individuals shared the same desks and had to clear these out at the end of each day, storing work belongings in trays. However, some individuals reported that they continued to use the same desk each day. Company information stated that the firm had introduced flexible working practices such as home and remote working for InSoft employees in order to improve competitiveness, utilise resources more efficiently and enable employees to achieve a more effective work-life balance:

> From time to time, it will be essential that teams are based together – but our priority is simply to have the right people on a project working in harmony. That doesn't necessarily require them to physically be in the same place, as long as they're networked and communicating.

However, difficulties in communicating with individuals elsewhere as a result of geographical dispersion were recognised by on-site individuals, offshore workers and project managers as impacting on experiences of work intensity. For instance, participants reported that the time difference between UK and India contributed to intensity. UK-based workers and offshore Indian team members had a four hour time block to converse (at the beginning of the day for UK workers; second half of the day for Indian offshore individuals), making work particularly intense during this time period:

> The offshore aspect is probably the most difficult, there's no getting away from it. You've got a window of about four hours to speak to the offshore team when you're both in the office. You have to make the most of it and when you've got a lot of work on, you find that four hours can be choc-a-bloc. I've said it for ages now, we should get development onshore or all the engineering team should be in one location, be it offshore or onshore, but nobody listens to

> me. It's difficult and the time factor's the biggest. It makes the first half of your day far more intense (Business Analyst).

In addition, on-site software professionals emphasised that it could be difficult to communicate and secure prompt responses from geographically dispersed team members. Offshore Indian software professionals echoed these concerns in the interview scripts, commenting that it was necessary to set up formal conference calls or wait until UK times to resolve issues, rather than being able to speak directly with team members. Differences in time zones further impacted on those individuals responsible for managing team members offshore. For instance, UK-based Technical Leads reported that it could be challenging to manage and allocate work tasks effectively across different time zones:

> I did have a contractor working for me for a while from India and trying to manage that relationship was really hard. There was very little overlap in time. I was giving him fairly low level things, because it was hard to find anything of consequence when the people he needed to speak to were never available when they were in the office. I think the person being offshore raised the intensity, as it became harder to try and manage that relationship (Engineering Technical Lead).

Of all the participants, one on-site Indian contractor working as a Support Technical Lead stated that working across time zones was manageable through having structured conference calls and e-mails. However, the contrasting experience of this individual could potentially be attributed to the sharing of nationality and cultural background with offshore team members and having an increased understanding of culture and work process.

Software professionals at InSoft reported that offshore workers were often trained on-site to ensure they had the appropriate skills to perform work activities offshore and to coach other team members. However, Technical Leads and project managers perceived that working styles differed between Scottish and Indian workers, despite attempts to encourage universal approaches. For example, Scottish workers were deemed to place greater emphasis on problem-solving, logic and possess long-term views on work and careers. Indian workers, in contrast, were considered to be more short-term in their planning and organising, to regularly change jobs and place emphasis on rank and achievement, rather than earnings. Indian offshore workers were also cited as failing to mention when problems arose. This short-term approach was seen to have negative repercussions for InSoft team members, giving rise to unexpected tasks close to deadlines and resulting in experiences of intensity. Technical Leads and project managers therefore emphasised that overall control of activities remained on-site with InSoft, in order to ensure that work was managed and monitored effectively.

Language barriers and communication difficulties were further identified by software professionals and project managers as hindering the work process and contributing to intensity, both for on-site and offshore workers. For example, UK-based software professionals across all areas stated that issues had to be explained carefully to offshore team members, in order to avoid misunderstandings. Concurrently, offshore Indian workers mentioned that junior team members could experience difficulties in understanding UK accents and the detail of conversations, requiring other offshore individuals to translate and provide explanations. Crucially, verbal misunderstandings as a result of communication difficulties were identified by one project manager as potentially being detrimental for work outcomes:

> If you're all sitting together, then the communication's cut down. That kind of closeness is completely lost [with geographically dispersed teams]. It's much more difficult and in terms of the quality of work, there can be lots of mistakes. There's a team who thought I was saying on the phone the first of August; I was saying the fifth of August. They were running around trying to do things for the first of August. It's purely a verbal communication problem.

Furthermore, difficulties in managing agreements between InSoft and Indian contractor companies contributed to experiences of intensity. Software professionals commented that offshoring should ideally increase levels of resourcing and allow work to be carried out by workers during and outside UK times. However, in practice, ineffective management of offshoring contributed to intensity. One Business Analyst commented that team members often followed idiosyncratic processes and procedures as a result of differing political agendas between InSoft and offshore contractor companies, creating internal team difficulties. In addition, despite official agreements established between InSoft and Indian contractor companies that Indian offshore workers would work to UK times, one Software Engineer reported that offshore workers would often call outside specified times or would complete work according to Indian time zones:

> We were always told when this started that they would work UK time. But less and less that seems to be the case. An example being, we do this on-call rota. It's not just support work, as part of that we build releases and those releases are then deployed by people in India. We build the release to try and get it available to these guys in India, supposedly for first thing UK time. But we've been finding that they're coming in early India time, maybe eight am, which is about half past three in the UK. So, if we've not finished the build by then, they're hassling us at three o'clock in the morning when we've been trying to get this thing to work for the past three hours.

Geographical dispersion and offshoring were therefore claimed to ideally expand levels of resourcing and enable work to be performed during and outside normal UK times. However, in practice, difficulties in communicating across time zones, cultural differences, language barriers and ineffective management of offshoring contributed to experiences of intensity at InSoft, both for on-site and offshore software professionals.

‘Agile’ Methodology

Software professionals, project managers and senior management all reported that InSoft had introduced the ‘Agile’ methodology in an attempt to release products and services into the market more quickly, reduce the amount of time spent resolving faults and increase levels of customer feedback. These perceptions corresponded with official company information which stated that ‘Agile’ methods could allow InSoft to respond to changing business needs and enhance customer experience more effectively:

> It [‘Agile’] offers customers innovative products and services in much shorter timescales, functionality they are really looking for, a chance for them to get involved, have an influence over what’s being developed and experience much greater satisfaction from choosing and using [InSoft].

Software professionals reported that the ‘Agile’ approach could have both positive and negative repercussions for experiences of intensity. For example, the ‘Agile’ methodology could allow client feedback to be reviewed and incorporated throughout, helping to manage intensity:

> On development projects previously, we used a traditional ‘waterfall’. Clients would come in at the start and say, “This is what we want”. We’d go away and then they’d go, “Oh, that wasn’t quite what we thought” and then we’d have to go through the whole thing again. They were there at the beginning and at the end and that was it. Whereas with [‘Agile’], because they’re so collaborative the whole way through, things get done much faster and they get what they want. It’s a huge improvement. I suppose in some way it’s harder work. But you fail faster, so if something’s not going to work, you find out an awful lot earlier and you have the chance to change it (Business Analyst).

Indeed, project managers stated that whilst ‘Agile’ had potentially intensified professional software work through increasing levels of client interaction, software professionals could obtain an improved understanding of client requirements and deliver superior end results.

Moreover, the distribution of work effort across shorter, incremental releases under ‘Agile’ led many software professionals to conclude that ‘Agile’ had changed the time span of intensity. Technical Leads and Software Engineers suggested that whilst intensity could occur more frequently under incremental release structures, experiences could potentially be easier to cope with due to shorter spells of activity:

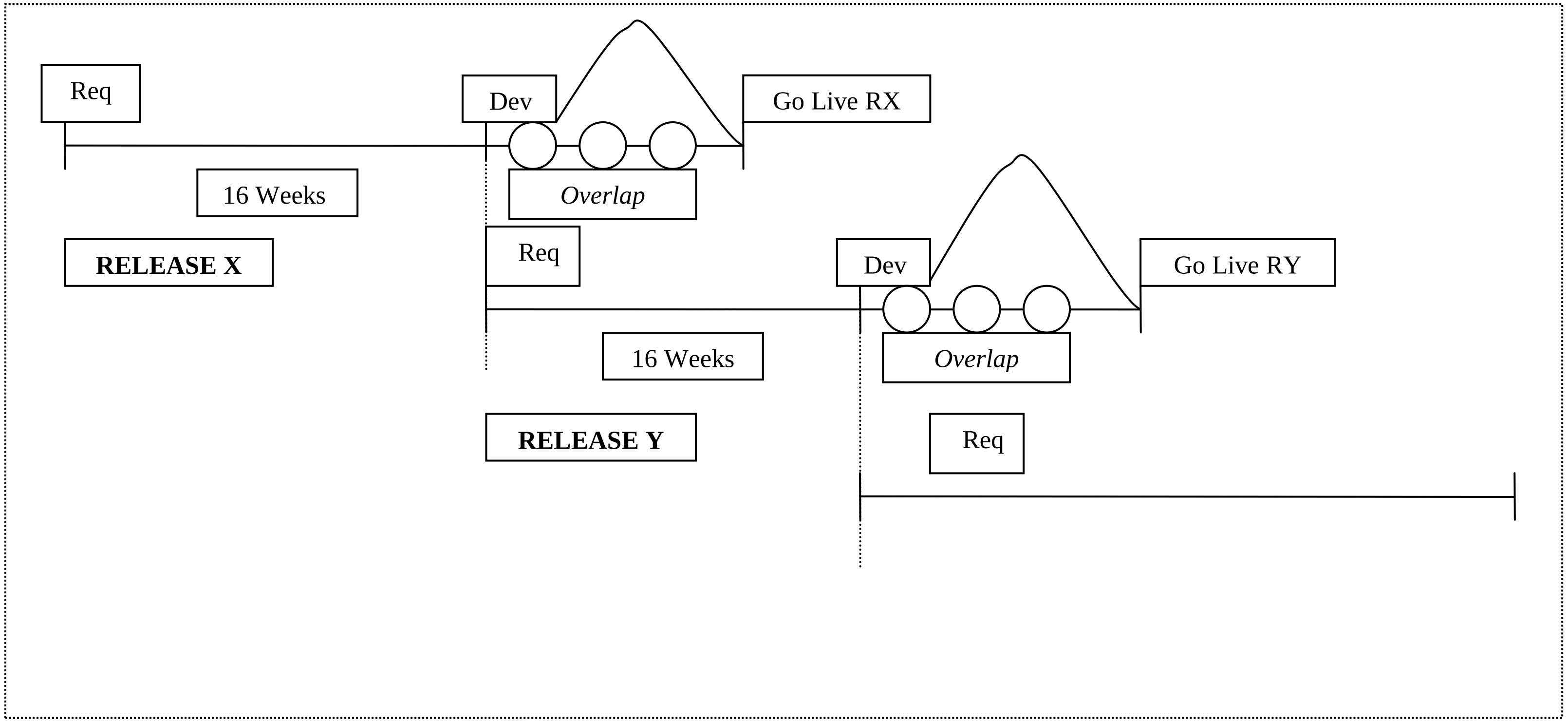
> It used to be that we had maybe six months to develop something. So that period of going along nice and easy might have been a bit longer, just working as normal and then the last couple of weeks would be intense. I think ‘Agile’ works better. I don’t know if it’s less intense – there’s probably more intensity but it’s for shorter periods of time. If it is intense, you know that it’s only going to be for a month, because we’ve got a month’s development window. The break you get isn’t as long, so you maybe have a couple of weeks rest, but I think you can cope with it better because it’s over a bit more quickly (Engineering Technical Lead).

However, work volume could increase under the ‘Agile’ approach as individuals moved through the different release stages, giving rise to negative experiences of intensity. For example, whilst individuals could influence the day-to-day management of workloads, the ‘Agile’ methodology set out a strict pattern of incremental deadlines. Furthermore, the frequent and incremental nature to releases under ‘Agile’ meant that new tasks could continuously be added to existing workloads, compressing work time in which to complete tasks:

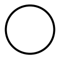
> The trouble is that if anything does go wrong, it’s added to our workload. That can be a pain, because it’s constantly “We need

> you to finish something here, here and here". If anything comes in here, it's a pain, because you've got a deadline two weeks away, instead of having a deadline three months away and if something did come in, you would still have time to do all this other stuff. So, if something comes in, it can be really intense, because there's no time in there to put your other stuff aside (Engineering Technical Lead).

Indeed, one project manager commented that software professionals could be working on releases in parallel as they progressed through the life cycle, increasing experiences of intensity due to the sheer volume of work. The following diagram (sketched by one project manager) illustrates the overlapping of releases and peaking rates of intensity, as experienced within the Business Analysis team:



Req = Requirements
Dev = Development
RX = Release X
RY = Release Y

 = Pressure Points

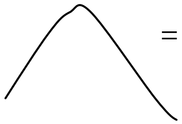
 = Peaking Rate of Intensity

DIAGRAM 6.1 INSOFT BUSINESS ANALYSIS TEAM RELEASE STRUCTURE

Utilisation of the 'Agile' methodology at InSoft can therefore be seen to have mixed consequences for software professionals' experiences of intensity. Crucially, whilst 'Agile' can be seen to have helped manage intensity, negative repercussions can equally be identified through the continuous expansion of workloads.

Staffing Arrangements

Staffing arrangements were widely held by software professionals, project managers and senior management to affect the intensity of work. Participants reported that under-staffing or overly large team size could equally impact on experiences of intensity, in that whilst smaller project teams could increase overall workloads, larger project teams could be difficult to manage. In addition, software professionals across Business Analysis, Software Engineering and Technical Lead areas reported that offshore team members were often moved on to other projects or companies. Crucially, individuals stated that the constant influx of new, inexperienced project team members could have negative repercussions both for on-site UK and offshore team members. For example, UK software professionals could have to provide greater levels of assistance to new team members, creating interruptions to work rhythms and experiences of intensity. Technical Leads responsible for managing and overseeing project work could also experience difficulties in managing and overseeing project work as a result of the continual rotation of offshore team members:

> The team I was involved in was the only team that was onshore, so people would come over for maybe about a year. That was an issue, because every three or four months, there would be somebody from the team leaving and someone new coming in. The experience of people in the team fluctuated all the time, so that was quite hard to manage. If someone new comes in to the team, they don't know the way we work and they don't know technical things. It slows everything down and myself and others in the team have to double-check and show them how to do things. It was a pain, but

> they live six thousand miles away, they're here for about a year and they want to go home, so there was nothing we could do (Engineering Technical Lead).

Similar experiences were documented by offshore Indian workers, with one Indian Business Analyst stating that the resignation or re-allocation of project team members could contribute to experiences of intensity. In this sense, the rotation of project team members and fluctuating levels of experience and knowledge within teams can be seen to have implications for experiences of intensity at InSoft.

6.6 IMMEDIATE DETERMINANTS

Volume of Work

InSoft software professionals across Business Analysis, Software Engineering and Technical Lead positions perceived that high work volume and changing priorities, rather than specific activities in themselves, contributed to experiences of work intensity. It should be noted that the work diary study enabled valuable insight into the volume of work performed by permanent InSoft software professionals and offshore workers. Indeed, observations made within the work diaries that volume of work could contribute to work intensity suggested that this issue required further exploration in the interviews. For example, some comments in the work diaries referred directly to the volume of work and its impact on intensity, with two Business Analysts stating that "For Monday, Tuesday, Thursday, it was really just the volume of work, as opposed to one thing, that was causing the intensity" and "Time pressure. Too much to do in time available, so I didn't get done what I'd planned".

Volume of work especially impacted on intensity for those operating as Technical Leads and for Indian offshore workers. For example, individuals working as Technical Leads (see Appendices 28 and 29) tended to have a greater number of activities to perform overall compared to other software professional participants (see Appendices 23 to 27). Crucially, the three Technical Leads each performed 22 activities in total across the work diary week, compared to the Business Analysts (15, 20 and 20 activities, with the latter being an offshore worker), the Developers (19 and 12 activities each), the Software Engineers (8 and 12 activities, respectively) and the Tester (8 activities). The effects of increased workload levels on experiences of intensity were documented by one Engineering Technical Lead:

> If the volume's high, it's probably always negative [effect on intensity]. Because of the amount of different tasks, I would probably have to work late just to get those done, 'cause there's a certain deadline there and we can't go over that. The only thing that can move is me doing more work.

Indian contractors also demonstrated higher levels of work volume in the work diaries, compared to their UK counterparts (see Appendices 23 to 29). For example, one offshore Developer and one on-site Consultant performed 19 and 20 activities across the work diary week respectively, in contrast to UK individuals operating as Developers (12 activities), Testers (8 activities) and Software Engineers (8 and 12 respectively). Crucially, offshore Indian contractors perceived it was assumed by InSoft that they would take on greater workloads and work longer hours, stating that these demands contributed to experiences of work intensity:

> Offshore working is assumed to be extended working hours and taking on work equivalent to two or more similar resources in the UK. Due to heavy expectations and increasing customer demands, it has become very intense and difficult (Offshore Indian Business Analyst).

In addition, offshore Indian workers commented that it was often necessary to stay late or start earlier to collaborate with UK-based teams, even though hours were officially Monday to Friday between 9am and 6:30pm. This was certainly the case for one Indian offshore Developer, who noted working hours in the work diary as being 8am to 8pm for four days and 8am to 10pm for one day. Indeed, the vast majority of InSoft software professionals perceived that contractors could experience greater levels of intensity compared to permanent InSoft employees as a result of higher expectations and emphasis on greater workloads. For example, one Business Analyst claimed that whilst permanent InSoft workers had to meet objectives, UK contractors were expected to deliver improved results and Indian contractors were required to achieve equivalent objectives in shorter timescales:

> The [UK contractor company] guys and the [Indian contractor company] guys probably get a harder time than they deserve, purely because they're contractors and more is expected of them. The [InSoft] guy's expected to deliver, [UK contractor company] expected to do fifty percent more and [Indian contractor company] expected to do it in half the time.

However, project managers at InSoft presented a contrasting perspective, reporting that offshore workers often failed to mention when they were experiencing difficulties with volume of work, making it difficult to ease workloads:

> It doesn't sit very comfortably with someone like me. If I am tasking someone with a piece of work and it's too much for them and they can't do it, I expect them to be able to say to me, 'XXXX, I need you to prioritise on that, because I can't do everything'. That's very valid and some people do come and say that, but they're more likely to be an [InSoft] permanent employee. I have to try and understand the workload of an Indian sub-contractor, because if I ask him to do something, he will very likely not feel like he can turn round and say no and will work through the night to finish it, without me realising that he had something else on that I wasn't aware of (Project Manager).

Furthermore, even though software professionals across all areas concurred that volume of work impacted on intensity of work, individuals exhibited differences with regards to the juggling of work tasks. For instance, those working in less technical roles such as Business Analysis did not emphasise juggling of tasks as an issue. However, Technical Leads and Software Engineers engaged in technically-oriented areas such as development identified juggling tasks, managing workloads and changing priorities as contributing to experiences of intensity. Indeed, participants stated that they had to constantly juggle and switch between tasks as they progressed through life cycle stages, a sequence that contributed to work intensity:

> Switching between tasks always increases intensity. There are times when you have to stop one task for a while and go and do something else and that's useful. But most of the time when you're switching between tasks, you have to *stop* thinking about that and *start* thinking about that. As well as the thinking phase, if it's a design you're doing, you have to bring up the design tools on the computer, so there's the *time* to do that. Task switching is always an overhead (Software Engineer).

The requirement to juggle tasks was further regarded by two project managers (who had originally worked as Software Engineers) as being a characteristic of technically-oriented software engineering roles. For example, one project manager summarised the process of 'thrashing' that software engineers were subject to, as a result of competing work tasks:

> In computing, there's something called 'thrashing'. You have a task that takes five minutes but it takes you a minute to remember where you are and you've actually only got three minutes. So you go to do it, you only get two minutes of it done, you put it down, you do the next thing, you come back again. You chop and change and you think, "There are ten things to do. What one will

> I do first? I'll do that". At the end of the day, you find out that the thing you said you would do first is a half-written e-mail that you didn't actually send (Project Manager).

Work volume can therefore be identified as contributing to intensity for software professionals at InSoft, particularly for Technical Leads and those located offshore.

Specialist Knowledge

All software professionals and project managers believed that there was a link between specialist knowledge and work intensity. Possession of specialist knowledge impacted on software professionals across all areas through having to help colleagues, to guide contractors or to mentor less experienced team members. For example, software professionals who had previously worked for the autonomous software centre could be interrupted by colleagues requiring help in historical technical areas:

> People have problems with their PC or something technology-wise and will come to round to my desk and say "Oh, can you come and have a look at something". You get draws on your time that way and it does pull my focus a bit. People know I used to do that as a job and take advantage of the fact that it's easier for me to come and help them than it is to phone a helpdesk to solve their problem. I don't mind helping someone else but I think it's where there's a perception that you're going to drop everything to come running (Architecture Technical Lead).

Indeed, individuals whose roles within the in-house IT department had become less technical were considered by one Technical Lead to be more likely to request help with technical problems. This was due to perceived differences in reasoning, problem-solving and initiative between individuals working in more managerially-oriented areas and those continuing to operate in technical capacities. In addition, one Business Analyst stated that contractors frequently required background information on systems, resulting in interruptions to work rhythms, increased volume of work and experiences of intensity for permanent InSoft software professionals:

> Whenever there's a new delivery, it gets tested from end to end and there's a specific team to do that. I think they're trained testers but they don't understand the system we're using. They don't know what all the processes and procedures are and they've never had any training in them. That contributes to the intensity of work, because you have to explain things as they haven't seen the system. Part of it's an interruption and part of it's just the volume of work to go through, which takes a long time.

Furthermore, one Software Engineer perceived that if individuals did not possess high levels of experience, knowledge and confidence in their abilities, demands could be increased for more knowledgeable team members, who were obliged to provide guidance to less experienced individuals. For instance, one Engineering Technical Lead stated that offshore Indian team members required greater levels of mentoring and closer task supervision due to the general nature of their skill sets:

> It [specialist knowledge] had a big influence on the intensity of my work, because I would either have to do the work myself or mentor someone. I'd have to check everything. They would have problems and I'd have to sit down and work them through it. So that would add to my work stack. I couldn't hand it off to someone else as I would have to actually do it or supervise it and that had a big impact.

Nevertheless, whilst specialist knowledge could impact on more experienced team members, project management provided insight into the experience of software professionals with less expertise. For example, less experienced team members could experience intensity through having specialist knowledge “bottlenecked away from them” and having to rely on those with greater levels of expertise.

Documentation containing background information on specialist areas was recognised by two software professionals (one Software Engineer and one Business Analyst) and project managers as helping to mediate experiences of intensity for software professionals. Documentation could potentially allow less experienced team members to access information on specialist areas themselves, reducing interruptions to work rhythms for individuals with greater expertise. However, despite the perceived benefits of documentation, participants acknowledged that experience played an important part in developing expertise, meaning that guidance from more experienced workers was necessary and continued to have implications for experiences of intensity.

Specialist knowledge can therefore be recognised as shaping experiences of intensity, both for more experienced software professionals tasked with mentoring other team members and for less experienced individuals who are reliant on those possessing such expertise.

Part-Time Workers

Some software professionals believed that part-time workers experienced greater levels of intensity than those working full-time. For instance, one Architecture Technical Lead reported that it was sometimes assumed that part-time workers would complete similar volumes of work to full-time staff, even though they were contracted to work part-time hours. Meetings could also potentially be scheduled for when part-time workers were not contracted to work, an occurrence that was evidenced by one part-time worker in their work diary. Indeed, one part-time worker who was contracted to work twenty-two and a half hours a week in reality worked thirty-one hours in the work diary week, with twenty activities logged in total (see Appendix 23). This contrasted with one full-time worker in a similar role who worked forty-seven hours across the work diary week and logged fifteen activities in total (see Appendix 23). Nevertheless, whilst one part-time worker acknowledged that having a reduced number of hours in which to perform work tasks could contribute to intensity, management support could help to alleviate increases in contractual hours and workloads:

> I guess being female and part-time probably does have an impact. I can't really do much in the way of overtime or anything, so I've *got* to fit everything in. But I'm quite lucky, because my managers are well aware of the fact that I get a lot of support for not doing more than my contracted hours. So I suppose that balances it out (Business Analyst).

Indeed, whilst one full-time Business Analyst noted intensity rates of 2, 2, 3, 3, 3 for the work diary week, the part-time worker noted comparatively lower rates of 2, 1, 1, 1, 1[3].

Thus, at InSoft, part-time workers were perceived to experience greater levels of intensity, due to assumptions that workloads could exceed normal contractual hours. However, whilst part-time workers themselves acknowledged that working fewer hours could reduce the amount of time available to complete tasks, support from management was seen to be key in helping to alleviate intensity.

Internal Motivation to Work

Software professionals at InSoft varied with regards to what they liked and disliked about their work, depending on their job role. Business Analysts liked the variety in their work, working with users and the sense of accomplishment when software satisfied user needs. Software Engineers enjoyed development, problem-solving, viewing end results and the flexibility that InSoft granted over working hours and work location. Technical Leads liked technical challenges, the variety in their work

[3] Participants were asked to use a scale of 1=Not All Intense, 2=A Little Intense, 3=Intense and 4=Very Intense and to place the relevant number in the 'Intensity of Day' box daily to show how intense they found the day.

and the additional autonomy and discretion accorded to them as a result of being in charge of a team.

However, Business Analysts disliked being the first point of contact for user complaints on software and the different agendas set by InSoft and the offshoring contract company. Software Engineers disliked office politics, bureaucracy, the undervaluing of technical skills and lack of advancement opportunities. Technical Leads disliked the pressure of having to deliver software to tighter deadlines, the volume of work and found it difficult to manage team members across cultures.

Performance metrics were used in the form of performance appraisals and target setting at InSoft, which individuals were then assessed against. However, these metrics were not perceived to contribute to work intensity. Rather, performance metrics merely required software professionals to spend time gathering information for appraisal forms and exert greater levels of work effort when seeking promotion.

Software professionals at InSoft can therefore be seen to vary in their likes and dislikes, depending on whether they are involved in Business Analysis, Software Engineering or Technical Lead roles. Performance metrics appear to have no impact on experiences of work intensity.

Breaks

There was virtual unanimity amongst InSoft software professionals and project managers on the importance of taking breaks, especially at lunchtime. Many software professionals commented that whilst breaks could reduce the time in which to complete tasks, they provided necessary relaxation time and helped reduce experiences of intensity:

> Taking my lunch break is essential. You might argue that I would get more work done if I worked through my lunch break but I suspect it would make my day more intense, because it would be

> constant pressure all through the day. Similarly, taking a tea break can relax me a bit and make the day less intense. My experience is that no matter how much work you've done, there's always more to do. I would say breaks make the day less intense than they would be otherwise (Software Engineer).

Indeed, one Business Analyst emphasised that if work could not be completed in the allotted time, this could imply that the workload was too sizeable and it would be necessary to speak to a manager to deal with this situation.

However, approaches to breaks taken by offshore Indian workers contrasted to that of on-site UK software professionals. Breaks taken by offshore Indian workers generally tended to be shorter, with lunch breaks around twenty to thirty minutes compared to the hour available for UK counterparts. Tea breaks were also recognised as being less feasible by one offshore Indian Business Analyst due to "the workload in the afternoon". In addition, whilst offshore workers recognised that breaks were important for health reasons, taking breaks during periods of concentration could distract attention and contribute to experiences of intensity:

> Some breaks like lunch breaks are essential. In our occupation, we have to stare at the computer for long hours, which is not too good for the health of eyes or the back. On the other hand, taking a break may sometimes distract attention and should be avoided when one needs to concentrate on a particular activity for a long period (Offshore Indian Developer).

Project managers recognised the difference in approaches taken by InSoft employees and offshore Indian workers, stating that this was a result of diverging professional cultures. For instance, whilst InSoft employees were encouraged to take breaks and time in lieu for extra hours work, Indian offshore companies placed emphasis on offshore workers working longer hours and taking fewer breaks:

> Those who are not permanent [InSoft] employees are employed by an Indian vendor. Their culture is to work many more hours than they are contracted to, working days are significantly longer and they're expected to travel more. They don't get the same lunch breaks or the same annual leave as [InSoft] employees. I try to treat every member of the team equally but the professional culture often has a greater influence than I do (Project Manager).

It is useful to consider the contrasting approaches UK workers and Indian offshore contractors adopted with regards to breaks and the consequent implications for intensity. Thus, whilst UK workers recognised the importance of breaks in helping to manage intensity, Indian offshore software professionals' approach to breaks was influenced by the nature of activities being performed, levels of concentration and professional working culture.

Clients

Whilst some software professionals within InSoft were engaged in external contracts, internal clients were identified as the main group dictating priorities and project importance. One Engineering Technical Lead engaged in development stated that internal clients could request last minute changes or require explanations if work did not directly fit expectations under the 'Agile' methodology:

> They [clients] have a huge impact, because they come in at the last minute and say "Oh, we don't like that. We want to change that". They have no understanding of the impact of a small change. If we screw up, if we don't deliver on time or if there's a problem, then the customer doesn't like it and I have to explain why things are causing a problem (Engineering Technical Lead).

In addition, internal clients could impact on intensity for Business Analysts through requesting changes to requirements. However, it should be noted that factors such as internal knowledge of the business, whether work was client-facing and stage in the development cycle influenced the extent to which internal clients contributed to work intensity for software professionals. Section 6.7 therefore explores factors that helped offset work intensity for software professionals in greater detail.

Deadlines

Software professionals across all areas stated that they were able to exercise control over how they planned and managed their work on a day-to-day basis. Software Engineers, Technical Leads and offshore Indian workers argued that greater expenditure of work effort was necessary when approaching deadlines and could contribute to experiences of intensity:

> Deadlines impact a lot. You think something's going to take you a certain amount of time and then you have problems. At the start of a design or delivery sprint, everything's kind of okay and then near the deadline, it always comes up. It's just the deadline won't move. You need to get the work done, so you'll probably work harder. Everything goes up and then falls down again (Engineering Technical Lead).

However, factors such as the nature of role performed, interim project deadlines, phase in the development life cycle and level of experience helped manage the extent to which deadlines contributed to work intensity for InSoft software professionals. Section 6.7 therefore discusses these aspects in greater detail.

6.7 FACTORS HELPING TO OFFSET WORK INTENSITY

Internal Knowledge of the Business

Within Business Analysis, good relationships could help to manage the setting of priorities and overall interactions. Business Analysts with greater experience were also able to make the business case for requirements and dictate priorities to internal clients due to possession of an in-depth understanding of the internal business:

> When I took over the XXXX area, my boss basically said "It's yours, do with it what you want". So I control my work to the extent that, while it's normally the customer who tells the BA [Business Analyst] what requirements they want, it's the other way round with us. I'm telling the customer what I think we should be doing. It's a good relationship and I've got a lot of scope in there (Business Analyst).

In this sense, internal knowledge of the business appears to help software professionals manage the impact of clients and reduce experiences of work intensity.

The Approach Taken By Project Managers and Technical Leads

Leadership styles adopted by project managers could potentially increase or reduce software professionals' experiences of intensity due to their influence over workload size, task nature and levels of support. For example, project managers could choose to support software professionals during project difficulties or expose software professionals to client complaints:

> You get very different leadership styles and that directly impacts on the intensity of work. Some projects run to time, some projects don't. You'll get people leading who will take that fully on the chin when the team say "Can't do it, can't do it" and the project manager

> saying, “Right, just carry on, do it as fast as you can and I will deal with what’s coming”. And you get people who just step back and you’ve got the unhappy customer practically screaming at all levels of the hierarchy (Senior Manager).

Project managers could also protect team members from constant client interference and requests:

> Sometimes it helps that someone’s there to stop you getting constant requests from the customer. Sometimes your customer can affect your intensity just by constantly bombarding you with requests. If these aren’t things that have necessarily been agreed beforehand and in all the budgeting and so on, it can be quite hard (Software Engineer).

Furthermore, if workers were under pressure from overlapping releases under the ‘Agile’ methodology, project managers possessed the ability to place pressure on other management levels to specify priorities and re-locate particular work tasks. In addition, Technical Leads could potentially alleviate software professionals’ experiences of intensity, due to their influence over the time estimates given for tasks and the allocation of work to team members.

In this sense, project managers and Technical Leads can potentially offset experiences of intensity for software professionals, due to their power over task estimates, task allocation, workloads and client interference. Crucially, facilitative and supportive leadership styles at project management and Technical Lead level can be recognised as helping software professionals manage the intensity of work.

Focusing on Non-Client Based Work

The move from client-facing work to producing tools for other software professionals within InSoft meant that one Software Engineer was removed from the ‘direct chain’ to the customer. Crucially, this individual stated that experiences of intensity had lessened, due to the non-essential nature of work being performed and developers being less likely to view themselves as 'clients':

> What I produce is *not* something that the customer will ever use. I produce things that aid the development or testing or the product. Being out of that direct chain has reduced the intensity. They [tools and aids] enable us to deliver the product but they're not actually built in. They're nice to have but if they don't have it, they can still cope. My 'clients' are the developers or the testers, so they're perhaps not so good at putting pressure on me because they don't see themselves as clients. So from my point of view, that makes life a lot easier.

In addition, less emphasis was placed on deadlines when producing tools and aids for other software professionals to use in comparison to internal client-based project work, helping to reduce experiences of intensity:

> Back when I was working on a structured project as part of a development team, it would be more hard and fast, "We need to have it in testing by the fifteenth of August". But at the moment, it's a bonus when things come out. It's not something that they're absolutely waiting on (Software Engineer).

In this sense, the extent to which professional software roles are associated with client deliverables can therefore be seen to shape experiences of intensity. Crucially, non-client centred roles can potentially help to reduce experiences of intensity for software professionals, due to the non-essential nature of activities.

Stage in the Development Cycle

Particular stages in the development life cycle were viewed as inducing variable levels of intensity, depending on the required levels of client involvement and the type of deadlines. For example, one Software Engineer suggested that experiences of intensity could decrease during particular stages, such as lower level design and development and increase when designs or releases were delivered to clients for review:

> While I was doing design, they were reviewing designs and we would demonstrate things to them. They had a lot of influence on how things looked and what they thought about it. Now we're into the lower level design and development phases, we go to them for approval of changes and queries. It's basically all been agreed, so we're working through it just now. The customer's getting reports from our managers on how things are going but they're not seeing what it's like yet. Once we deliver something and they use it, we might get more requests for changes or them saying "I didn't think it would work that way".

In addition, whilst work could be more intense when approaching official release deadlines, interim project team deadlines to mark team progress were recognised by one Software Engineer as being less intense:

> Normally, approaching the deadline is a fairly intense time, as you're trying to get things finished and working by then. It's not been in this one. It's partly because it's kind of an artificial deadline, it's getting part of the work done and it's not a delivery to the customer. It's just to say "We've done this part and we're moving on to the next part". Maybe that's got an effect on it. We're not giving something out that the customer then has to use (Software Engineer).

Indeed, senior management suggested that the phase in the development life cycle and immediacy of release dates could influence the extent to which deadlines impacted on software professionals' experiences of intensity.

The specific stage in the development life cycle therefore appears to influence the impact that deadlines and clients have on software professionals' experiences of intensity. Notably, intensity may be reduced at stages in the development life cycle when clients are less involved or where deadlines represent informal interim markers of project team progress, rather than official release dates.

Level of Experience

Software professionals operating as Technical Leads and in more senior capacities emphasised that their greater experience could enable greater levels of influence to be exerted over task estimates. Thus, whilst it was not always possible to change deadlines, individuals could liaise with management to discuss how to address difficulties in managing unrealistic deadlines:

> The development team have got two days left to do something and I think they're going to struggle. We can't change the deadline. However, if I say to my boss "I don't think we're going to finish it in time", we'll discuss how we're going to fix it. My influence over something that isn't going to meet a deadline has changed. It's just with experience and they trust me now. Whatever I say is probably going to be right (Technical Lead).

In this sense, level of experience and ability to influence task estimates can play an important role in offsetting experiences of work intensity for software professionals.

CONCLUSION

This chapter has provided an in-depth discussion of sources of intensity at InSoft and the consequent experiences of software professionals. Privatisation and increased competition placed pressure on InSoft to develop more responsive business models, in order to manage efficiency, reduce costs and raise profits.

Technological developments, such as improvements in computing speed and the emergence of technological mediums had intensified professional software work at InSoft, through increasing overall work volume, enabling functions to occur more quickly and allowing these to overlap. In addition, technological mediums such as phone calls, e-mails and instant messenger created interruptions to work rhythms and contributed to daily experiences of work intensity for software professionals. However, it is important to note that the directness and individual utilisation of technological mediums influenced the extent to which these mediums created interruptions and contributed to work intensity.

InSoft's firm characteristics can be seen to have influenced the nature of strategies adopted in response to market dynamics, with implications for internal organisational factors influencing experiences of intensity. Crucially, the leadership emphasis on managerially-oriented software engineering activities had implications for the nature of software work performed within the in-house IT department and resulting experiences of intensity.

The division of software engineering labour at InSoft between 'high value' and 'low value' areas had implications for software professionals' experiences of intensity with regards to work organisation. Crucially, geographical dispersion of project teams across the UK and India and the rotation of project team members can be identified as contributing to experiences of intensity for both on-site and offshore software professionals. Utilisation of the 'Agile' methodology can be recognised as giving rise to positive and negative experiences of intensity for software professionals at InSoft through changing the distribution of work effort.

Furthermore, forms of work organisation at InSoft shaped the daily immediate determinants of intensity. Volume of work contributed to experiences of intensity for all software professionals, particularly for Technical Leads and offshore Indian workers. Juggling of work tasks can also be identified as a further dimension surrounding the volume of work for those individuals operating in technically-oriented roles. In addition, specialist knowledge further shaped experiences of intensity, both for more experienced software professionals responsible for mentoring and assisting other team members and for those reliant on receiving this help.

Approaches to breaks depended on personal approach, nature of activities, levels of concentration required and professional culture. In addition, whilst clients and deadlines contributed to intensity for software professionals, the extent to which they impacted clearly depended on the nature of role being performed, stage in the development life cycle and association with client deliverables. In this sense, factors such as internal knowledge of the business, the approach taken by project managers and Technical Leads, focus on non-client based work, stage in the development cycle and level of experience were identified as helping to offset experiences of work intensity.

Finally, the following diagram (Diagram 6.2) provides a heuristic depiction of the hierarchy of factors that shaped software professionals' experiences of work intensity at InSoft:

MARKET DYNAMICS

De-Regulation and Privatisation
Competition

TECHNOLOGICAL DEVELOPMENTS

Improvements in Computer Speed
Technological Mediums
Interruptions

FIRM CHARACTERISTICS

In-House IT Department in a Large Firm
Telecommunications Sector

INTERNAL ORGANISATIONAL FACTORS

Leadership Style
Separation of ‘High Value’ and ‘Low Value’ Work

WORK ORGANISATION

Geographical Dispersion of Project Teams
‘Agile’ Methodology
Staffing Arrangements

IMMEDIATE DETERMINANTS

Volume of Work — Level of Experience; Juggling Tasks; Contractors
Specialist Knowledge
Part-time Workers
Breaks
Clients
Deadlines

FACTORS HELPING TO OFFSET WORK INTENSITY

Internal Knowledge of the Business
The Approach Taken By Project Managers and Technical Leads
Focusing on Non-Client Based Work
Stage in the Development Life Cycle
Level of Experience

DIAGRAM 6.2 SOURCES OF INTENSITY AT INSOFT

CHAPTER 7 COMPARISON OF SPECSOFT AND INSOFT FINDINGS

INTRODUCTION

This chapter will compare and contrast findings at SpecSoft and InSoft, in order to identify similarities, patterns and differences in software professionals' experiences of work intensity between the organisations. This comparison will contribute to a comprehensive analysis of the factors shaping software professionals' experiences of intensity.

7.1 TECHNOLOGICAL DEVELOPMENTS

Technological developments in computer speed, processing power and communication mediums were perceived by senior managers in both companies to have enabled software functions to occur more quickly, speeding up cycle times, increasing volume of work and intensifying professional software work. The availability of technological mediums such as e-mails, instant messenger and phone calls also meant that software professionals at both SpecSoft and InSoft were not immune to interruptions from project colleagues, project managers or clients based elsewhere. However, the extent to which technological mediums contributed to work intensity for software professionals in both SpecSoft and InSoft depended on the directness and personal utilisation of methods.

7.2 FIRM CHARACTERISTICS

Market trends such as de-regulation, privatisation, competition, downturn, buoyancy and world events created a framework which SpecSoft and InSoft had to respond to in order to remain competitive. Crucially, differences in organisational type and company size influenced the strategies and responses adopted by the two companies with regards to staffing arrangements, the physical proximity of project team members, the importance of internal software engineering expertise and the emphasis placed on deadlines and raising profits.

Organisational Type

At SpecSoft, software engineering formed the core area of expertise due to its specialist software firm status. The nature of software engineering activities being performed, physical proximity of project team members and emphasis placed on technical skills therefore remained unchanged within SpecSoft. However, SpecSoft tightened finances, downsized, made project teams leaner and accepted additional projects in response to market pressures. PT2 at SpecSoft further adapted to the fast-pace and changing nature of the telecommunications industry by constantly updating products, keeping up-to-date with technological developments and through juggling several projects simultaneously.

InSoft's status as an in-house IT department within a large firm operating beyond the strictly defined boundaries of software engineering generated different responses to market dynamic pressures. Indeed, whilst InSoft software professionals had previously performed activities across the development life cycle, developments in technology, globalisation and the upsurge in specialist software firms led to less emphasis being placed on internal software engineering expertise. InSoft therefore chose to amalgamate the in-house IT department with the rest of the company, separate software engineering activities into 'high value' and 'low value' areas, offshore 'low value' work and geographically disperse project teams. Crucially, the move towards managerially-oriented software roles and decisions to offshore development activities had important implications for experiences of work intensity at InSoft.

Company Size

Software professionals at SpecSoft and InSoft uniformly considered intensity to be greater within smaller organisations as a result of increased visibility, effort being more directly linked to company success, leaner staffing levels and greater multi-tasking. Indeed, individuals perceived that it could be more difficult to establish the link between individual effort and organisational outcomes within larger organisations. However, the experiences of InSoft participants suggested that large company size could equally contribute to experiences of work intensity. For example, bureaucracy made it difficult to introduce new ideas, understand processes and deal promptly with situations, contributing through frustration to experiences of work intensity.

7.3 INTERNAL ORGANISATIONAL FACTORS

Firm characteristics can be seen to have influenced the strategies adopted at the internal organisational level with regards to leadership style, client base and software engineering roles and had implications for experiences of work intensity.

Leadership Style

Technical expertise, interpersonal skills, problem-solving and logic were recognised by software professionals at both SpecSoft and InSoft as being important skills for performing work effectively. These views were shared by project managers at both companies who had worked as software engineers in the early stages of their careers. However, higher level leaders at InSoft who had progressed through managerial routes placed greater emphasis on managerially-oriented and interpersonal dimensions as opposed to technical elements, with implications for the type of activities performed in-house. In addition, leadership emphasis on managerial software-related functions meant that leaders often failed to consider how systems could technically be designed, developed and implemented in practice. This lack of understanding contributed to difficulties and experiences of intensity for technically-oriented software professionals tasked with producing these systems.

Leadership style was identified at both companies as shaping experiences of intensity. Participants noted differences in styles between director and senior management levels, which were more target driven, and immediate levels such as project management and Technical Leads. Crucially, project management power over deadlines, planning of workloads and management of project teams meant that the approach taken at this level had important implications for software professionals' experiences of work intensity. In addition, Technical Leads could further influence experiences of work intensity due to their role in allocating tasks to team members, providing estimates, monitoring work progress and reporting information to project managers. Crucially, supportive and facilitative leadership styles, particularly at project management and Technical Lead levels, were recognised at SpecSoft and InSoft as being effective at managing software professionals' experiences of work intensity.

Clients

Software professionals at SpecSoft and InSoft revealed variation in experiences of work intensity as a result of differences in client base. For example, SpecSoft carried out projects for external clients and perceived that levels of accountability, responsibility and visibility were likely to be greater when providing services for such clients. Indeed, clients greatly impacted on experiences of intensity at SpecSoft through their indecision, lack of clarity, changing of priorities and unwillingness to compromise. In contrast, whilst InSoft had some external clients, the majority of software professionals provided services to internal clients. Factors such as internal knowledge of the business, whether work was client-facing and stage in the development life cycle influenced the extent to which internal clients contributed to work intensity for software professionals at InSoft. Internal clients could contribute to intensity for software professionals at Technical Lead level through requesting last minute changes or interrupting the work process with queries. However, individuals working within Business Analysis reported reduced experiences of intensity due to the ability to dictate priorities and make the business case for requirements. Software professionals working within areas deemed non-essential and removed from the direct chain to customers similarly observed lower levels of intensity. The impact of clients on experiences of intensity at InSoft further depended on stage of the development life cycle. For example, whilst increased client involvement during design and delivery stages could heighten experiences of work intensity, reduced client involvement for lower level design and development helped to decrease the incidence of intensity.

At SpecSoft, the influence and power exerted by clients was considered to be dependent on levels of technical knowledge possessed. Software professionals perceived that clients with significant technical knowledge could create tensions between clients' preferred approaches and that of software professionals. This contrasted to the position held by project managers that more knowledgeable clients could have more realistic understandings and expectations of tasks, timescales and costs. In addition, the role and influence of clients at SpecSoft was perceived to be influenced by the criticality and importance of projects.

Software Engineering Roles

Experiences of intensity at SpecSoft and InSoft were influenced by the nature of software engineering roles performed and level of responsibility. Software professionals at SpecSoft were engaged in a wide variety of tasks across the development life cycle, covering the four job roles set out in the taxonomy (see Section 4.3). The emphasis placed on multi-tasking contributed to work intensity for individuals through increasing volume of work and necessitating juggling of work tasks. Senior Team Members and Technical Leads in particular tended to have a wider variety of activities to perform and greater levels of responsibility, increasing experiences of work intensity. This contrasted with the experience of Junior Team Members, who carried out fewer and less varied activities and reported lower levels of work intensity as a result.

InSoft software professionals had previously carried out activities across the development life cycle. However, the division of labour between 'high value' and 'low value' software engineering activities meant that individuals had become more specialised in their roles and were focused on a specific job role such as

‘Systems/Business Analyst’, ‘Designer’ or ‘Developer’, as detailed in the taxonomy (see Section 4.3). InSoft software professionals appeared to experience lower levels of intensity in comparison to those at SpecSoft, as a result of the internal and specialised nature of software engineering activities performed. It should be noted that the majority of InSoft software professionals had been with the company for several years, meaning that there was little variability in terms of tenure and experience level. This factor made it difficult to fully explore the relationship between experience and work intensity.

7.4 WORK ORGANISATION

Strategies adopted at the internal organisational layer had implications for approaches to work organisation within SpecSoft and InSoft. Deadlines, ‘Agile’ methodology, the physical proximity of project team members, and staffing arrangements were identified as the main work organisation elements with implications for software professionals’ experiences of intensity.

Deadlines

Software professionals at SpecSoft and InSoft were able to exercise autonomy over how tasks were performed technically, deal with problems through self-determined means and influence the duration, scheduling and distribution of personal work time. However, autonomy occurred within the boundary of set deadlines which formed the basis for structuring, organising, managing and controlling all work activities. Deadlines were identified at SpecSoft as a predominant factor contributing to software professionals’ experiences of intensity through requiring individuals to manage and juggle competing work tasks and exert greater levels of work effort when approaching deadlines. At InSoft, however, the impact of deadlines varied depending on the nature of role performed and whether work was client-focused. For example, whilst individuals working within Software Engineer areas and in Technical Lead capacities documented similar experiences to those at SpecSoft, those operating in non-critical and non-client centred functions reported lower levels of intensity, due to less emphasis being placed on deadlines.

‘Agile’ Methodology

Both SpecSoft and InSoft had adopted the ‘Agile’ methodology through the perception that an incremental approach was more appropriate to managing and organising professional software work activities of a complex, ambiguous and intellectual nature. Indeed, software professionals at both companies observed that ‘Agile’ helped to manage experiences of work intensity, through improving understanding of client requirements, increasing levels of customer feedback and distributing work effort more evenly. However, ‘Agile’ could equally contribute to experiences of work intensity, with increased customer involvement creating interruptions to the work process. In addition, the incremental release structure under ‘Agile’ increased workloads as individuals progressed through the development life cycle. The ‘Agile’ methodology can therefore be recognised as having positive and negative implications for experiences of work intensity at both companies.

Physical Proximity

Software professionals at SpecSoft emphasised that close physical proximity of project team members helped reduce work intensity through providing individuals with the ability to engage in work-related discussions, knowledge-sharing, brainstorming and enabled them to coordinate work more effectively. This situation contrasted with that at InSoft, where project team members were distributed across the UK and India and interacted through the utilisation of technological mediums such as e-mail, instant messenger and phone calls. InSoft software professionals reported that geographical dispersion contributed to experiences of work intensity through making it difficult to communicate effectively with team members, secure prompt responses and allocate, manage and coordinate tasks. In addition, whilst InSoft claimed that offshoring could allow cost savings and ‘round the clock’ services to be achieved, differences in time zones, cultural differences and ineffective management of offshoring demonstrated the consequences for experiences of work intensity in practice.

Staffing Arrangements

Both insufficient staffing levels and overly large project teams were recognised at both SpecSoft and InSoft as having the potential to increase experiences of work intensity. For example, insufficient staffing levels at SpecSoft increased workloads, responsibilities and volume of work. In addition, while neither company reported excessive staffing levels, unwieldy project teams were considered to be difficult to coordinate and necessitate greater levels of coaching, interrupting work rhythms for those providing assistance. Rotation of project team members was similarly identified at both companies as influencing experiences of intensity for remaining individuals. For example, team members at SpecSoft were often moved on to other projects demanding higher priority, requiring fewer individuals to perform the same volume of work. At InSoft, UK software professionals experienced interruptions to work rhythms through experienced offshore individuals being moved off onto other projects and new team members requiring greater levels of coaching and guidance. Crucially, fluctuations in staffing levels and rotation of project team members can therefore be seen to have important implications for software professionals' experiences of work intensity.

7.5 IMMEDIATE DETERMINANTS

Forms of work organisation at SpecSoft and InSoft had implications for experiences of work intensity at the individual level. Work volume, specialist knowledge, interruptions and internal motivation to work were recognised as more immediately determining software professionals' daily experiences of work intensity.

Volume of Work

Software professionals from both SpecSoft and InSoft unanimously stated that volume of work, as opposed to the difficulty and complexity of work itself, contributed to daily experiences of work intensity. Technological developments were considered to have increased volume of work and intensified professional software work. In addition, adoption of the 'Agile' methodology was regarded to have increased volume of work and encouraged greater levels of client involvement in both companies which contributed to experiences of work intensity. Juggling of tasks in order to manage overall volume of work further contributed to experiences of intensity for software professionals at SpecSoft and those engaged in technically-oriented roles at InSoft. Furthermore, as discussed earlier, level of experience influenced volume of work at SpecSoft, with software professionals operating at Senior and Technical Lead levels performing a wider variety of tasks and experiencing greater levels of intensity, in comparison to Junior Team Members.

Specialist Knowledge

More experienced software professionals at both SpecSoft and InSoft had to mentor and assist less experienced team members, a responsibility that contributed to work intensity for both groups. For example, as a result of mentoring, those possessing specialist knowledge reported interruptions to work rhythms, increased volume of work and reduced time to perform designated tasks. For those being mentored, the reliance on those with specialist knowledge meant that workloads and effort levels to meet timescales could increase when help was postponed or unavailable.

Documentation was recognised at InSoft as potentially helping to mediate experiences of intensity for software professionals at all levels, through providing less experienced team members with necessary information and reducing interruptions to more knowledgeable individuals. However, despite the benefits to be attained through availability of documentation, mentoring and guidance from experienced team members was still considered essential at both SpecSoft and InSoft due to the tacit and intangible nature of professional software activities.

Interruptions

Interruptions from colleagues, technological mediums, clients and support calls were identified as contributing to daily experiences of intensity for software professionals at SpecSoft and InSoft. For example, whilst regular informal on-site discussions at SpecSoft allowed individuals to obtain clarifications, brainstorm and develop skills and knowledge, these interactions gave rise to frequent interruptions across the working day. Crucially, these interruptions contributed to experiences of intensity for software professionals through disrupting concentration, affecting the management of work tasks and reducing time in which to perform tasks to deadlines. Interruptions were particularly prevalent for Technical Leads due to their possession of greater levels of technical expertise.

Informal, face-to-face discussions were not commonplace at InSoft due to the geographical dispersion of project teams. Thus, project team interactions mainly occurred via technological mediums such as phone calls, instant messenger and e-mails. Phone calls were recognised as the most intrusive communication method through requiring an immediate response. The directness of phone calls therefore served to interrupt work rhythms and disrupt 'train of thought', contributing to experiences of intensity for software professionals. E-mails and instant messenger were recognised at both companies as helping to reduce or increase experiences of intensity, depending on whether individuals felt the need to respond promptly or chose when to respond.

Clients were identified as a common source of interruptions and contributed to experiences of work intensity. For example, clients could request additional tasks to be added to workloads without altering existing deadlines, request clarifications or ask problems to be investigated. Crucially, interruptions from clients could disrupt work rhythms, increase the juggling of tasks and reduce the time available in which to perform work to deadlines.

Support activities further contributed to experiences of intensity at SpecSoft. Indeed, software professionals cited the unpredictability and uncertainty of when support calls would come through and how long they would take to resolve as detracting attention from other work tasks and impacting on time to perform work for deadlines. However, in contrast, support was not an aspect which was identified within InSoft as contributing to intensity.

Internal Motivation to Work

Software professionals at SpecSoft and InSoft demonstrated differences in their likes, dislikes and internal motivation to work. Individuals at SpecSoft clearly enjoyed problem-solving, analytical and technical aspects to work, especially writing code. SpecSoft software professionals particularly revealed internal motivation to work, valuing quality, commitment, pride in work and supporting colleagues. Indeed, the instigation of formal workshop sessions by software professionals themselves to evaluate project performance and identify improvements illustrated the internal motivation to work. At InSoft, those performing Software Engineer and Technical Lead roles mirrored the preferences at SpecSoft for technical challenges, problem-solving and creativity. However, Business Analysts at InSoft cited the sense of accomplishment when software satisfied the needs of users, variety and working with the customer as being enjoyable aspects to work.

Software professionals at SpecSoft disliked support, client ambiguity, carrying out interviews and writing documentation. The factors identified by software professionals at InSoft as being less enjoyable aspects of work varied across job roles. For example, while traditional Software Engineers disliked office politics, bureaucracy and the undervaluing of technical skills, Technical Leads at InSoft were averse to managing team members across cultural divides, volume of work and pressures to deliver more in shorter timescales. Business Analysts at InSoft, in contrast, disliked differences in agendas between offshore companies and InSoft and user criticism.

Breaks

Software professionals at SpecSoft recognised that while breaks were important to create a fresh outlook and help manage intensity, breaks could equally interfere with concentration when working on challenging or intellectually stimulating tasks and contribute to experiences of intensity. At InSoft, there was a contrast in the approaches taken to breaks by offshore Indian contractors and on-site permanent employees. Offshore Indian contractors exhibited a similar approach to taking breaks as witnessed at SpecSoft, through recognising that whilst breaks could be helpful, they could interrupt ‘train of thought’ when working on challenging tasks. However, on-site permanent software professionals at InSoft were pragmatic on the need to take breaks, emphasising that breaks were important and helped reduce experiences of work intensity. Differences in professional cultures could explain the contrasting approaches taken by offshore contractors and permanent employees at InSoft. Crucially, whilst permanent employees at InSoft were encouraged to take breaks and time in lieu for extra hours work, the professional cultures within Indian offshore worker companies placed greater emphasis on working longer hours and taking fewer breaks.

Contractors

Contractors at SpecSoft were hired for specific knowledge, to work on particular tasks or to carry out activities unpopular with permanent employees. Software professionals therefore perceived that contractors experienced reduced levels of intensity in comparison to permanent employees, due to lower levels of personal responsibility and lack of permanent employment status. However, contractors at InSoft tended to work in areas which were not typically performed in-house, such as ‘low value’ coding and development. Software professionals and project managers at InSoft perceived that contractors experienced greater levels of intensity compared to permanent employees, due to larger workloads and higher expectations. The differing experiences of contractors at both companies could be attributed to differences in organisational type and the nature of software engineering activities being carried out. Thus, whilst contractors within SpecSoft could be seen to possess software engineering skills akin to permanent employees, the separation of ‘high value’ on-site activities from ‘low value’ offshore activities at InSoft meant that the activities performed by contractors were distinct in nature.

Part-Time Workers

At SpecSoft, neither PT1 or PT2 had any workers on part-time contracts, meaning that it was not possible to examine the experiences of part-time workers. Software professionals at InSoft suggested that part-time workers could be assumed to complete the same volume of work as full-time workers. Indeed, part-time workers themselves stated that the reduced amount of time to perform tasks could contribute to experiences of work intensity. However, support from management was held to be crucial in helping to manage and reduce part-time workers' experiences of work intensity.

Software professionals at both organisations perceived that women were treated equally and given the same opportunities as men. Gender therefore did not appear to influence experiences of work intensity for either male or female participants. However, caution must be exercised in drawing this conclusion due to the small numbers of female workers and therefore respondents at both organisations.

7.6 FACTORS HELPING TO OFFSET WORK INTENSITY

The Approach Taken By Project Managers and Technical Leads

Software professionals at both companies perceived that the approach taken by project managers and Technical Leads could influence experiences of work intensity. Project managers could offset experiences of intensity through providing protection from client interference, specifying priorities, addressing workload difficulties and re-allocating excess work. In addition, measures by Technical Leads to take team member perspectives and opinions into account when allocating tasks or providing estimates could further help to manage software professionals' experiences of intensity. Alternatively, project managers and Technical Leads could contribute to experiences of intensity through neglecting to perform these functions.

Managing Relationships With Clients

Participants from both SpecSoft and InSoft appreciated the importance of establishing and managing relationships with clients, in order to manage experiences of intensity. In addition, SpecSoft participants emphasised the need to agree specifications early on, ensure clients adhered to specifications and keep records of discussions, in order to prevent clients changing priorities and contributing to intensity.

Level of Experience

Experienced and senior software professionals at SpecSoft and InSoft, particularly those operating at Technical Lead level, were able to exert influence over task time estimates through more accurate understandings of how long tasks would take. In this sense, whilst individuals were unable to alter deadlines, level of experience could help towards establishing more realistic task completion timescales and had implications for daily experiences of work intensity.

Personal Pro-Activeness

At SpecSoft and InSoft, senior managers and software professionals recognised that personal pro-activeness and self-awareness could help software professionals alleviate their own experiences of work intensity. For example, senior managers at SpecSoft suggested that software professionals themselves could prevent tasks from accumulating and manage their expenditure of work effort by recognising the importance of meeting deadlines. In addition, actively drawing client attention to delays in providing information could help software professionals reduce pressures to meet unrealistic timescales. At InSoft, individuals emphasised the importance of making project managers aware of work overload and communicating difficulties in a timely fashion, in order to alleviate experiences of work intensity.

CONCLUSION

This chapter has identified patterns, similarities and differences in experiences at SpecSoft and InSoft in order to inform the development of an in-depth understanding of work intensity. Technological developments, such as improvements in computing speeds and technological mediums were deemed to have intensified professional software work. Most notably, these developments enabled functions to be performed more quickly, sped up cycle times and increased volume of work for software professionals. In addition, market dynamics created a framework which the two companies had to respond to in order to remain competitive.

Differences in firm characteristics clearly affected the strategies adopted by SpecSoft and InSoft in response to market dynamics. Organisational type influenced the importance the two companies placed on internal software engineering expertise. Company size further shaped the visibility of effort, staffing arrangements, the need for multi-tasking and levels of bureaucracy within both organisations.

At the internal organisational layer, leadership style shaped work intensity, with facilitative and supportive project management styles being deemed most effective at managing professional software work in both companies. While external clients greatly contributed to experiences of work intensity at SpecSoft, internal clients had varying impacts on intensity at InSoft, depending on whether work was client-facing and stage in the development life cycle. In addition, whilst levels of intensity at InSoft were lower due to the specialised nature of activities, work intensity at SpecSoft was greater, as a result of performing a wide variety of tasks across the development life cycle.

Deadlines were identified as a common form of work organisation at both companies. However, whilst deadlines greatly contributed to experiences of intensity at SpecSoft, the impact of deadlines at InSoft depended on the nature of role performed and whether work was client-focused. Both companies agreed that the 'Agile' methodology could have positive and negative implications for experiences of work intensity. SpecSoft and InSoft also shared the view that staffing arrangements, in terms of insufficient staffing levels, unwieldy project teams and rotation of team members, could contribute to work intensity. However, whilst close physical proximity of project team members helped reduce intensity at SpecSoft, the geographical dispersion of project team members contributed to experiences of work intensity at InSoft.

Volume of work, specialist knowledge and interruptions from colleagues, clients and technological mediums were commonly identified at SpecSoft and InSoft as factors contributing to daily experiences of work intensity. Differences in the likes and dislikes identified at SpecSoft and InSoft can be seen to relate to the differing roles performed by software professionals. The contrasting approaches taken to breaks at SpecSoft, InSoft and by Indian offshore contractors can be seen to be influenced by the professional cultures within these organisations. Furthermore, differences in organisational type and levels of internal software engineering expertise can explain the contrasting experiences of contractors at SpecSoft and InSoft. In addition, whilst no members of PT1 and PT2 at SpecSoft were employed on part-time contracts, part-time workers at InSoft appeared to be particularly susceptible to experiences of work intensity as a result of the reduced amount of time in which to perform tasks.

Finally, particular factors were identified as helping offset experiences of work intensity for software professionals within both SpecSoft and InSoft. These included project managers and Technical Leads adopting facilitative and supportive leadership styles, software professionals managing and establishing relationships with clients, level of experience, personal pro-activeness and self-awareness.

CHAPTER 8 DISCUSSION: AN EXPLANATION OF WORK INTENSITY

INTRODUCTION

The utilisation of a contextually-based, in-depth comparative case study approach has enabled the identification and explanation of the linkages, mechanisms and relationships which influence and shape software professionals' experiences of work intensity. This chapter will therefore return to the research questions set out in Chapter Three to address the research aim of studying work intensity in the software industry, its incidence and its impact on professional software workers. These research questions are:

- How do firm characteristics/internal organisational factors (organisational type, company size, product market, skills and knowledge development opportunities, leadership style) influence experiences of work intensity?
- To what extent are software professionals subject to experiences of work intensity as a result of the way work is structured, designed, organised, managed and controlled?
- In what ways are software professionals affected by associated internal organisational factors?
- How are software professionals affected by the way their work is structured, designed, organised, managed and controlled?
- What are software professionals' perceptions and experiences regarding intensity/intensification?

This chapter will demonstrate several original contributions made by this study. It will identify aspects which are consistent with, or challenge, existing accounts on professional software work and it will locate the research findings within the narratives of the broader knowledge economy. This chapter is structured around four main themes deriving from these research questions which can be identified as having implications for work intensity. Section 8.1 focuses on the implications that markets and firm characteristics have for software professionals' experiences of work intensity. Market dynamics, competitive pressures and changes in external investment are shown to influence internal organisational strategies adopted within the software industry. This section also demonstrates that organisational type and company size can influence the nature of activities performed in-house, internal expertise, work pace and levels of visibility.

Section 8.2 outlines the main organisational dynamics within professional software work which influence work intensity. Notably, these concern volume of work, the supersedence of 'Agile' over the 'waterfall cycle' and the impact that clients can have on the software work process. Moreover, this section challenges the perspective that technological developments have necessarily de-skilled professional software work.

Section 8.3 provides insight into the interactive nature of professional software work and its implications for work intensity. This section argues that specialist knowledge within project teams and the dynamics of interruptions can contribute to experiences of work intensity for individuals. In addition, this section emphasises that work intensity can be further shaped by work location, physical proximity and the use of technological mediums.

Section 8.4 outlines the control and mediation strategies which influence software professionals' experiences of work intensity. This section argues that leadership background, leadership style, deadlines, normative control, performance metrics and breaks can influence software professionals' experiences of work intensity. In addition, this section explores variations in how individuals respond to workplace pressures, in terms of how relationships with clients are managed, the approach taken by project managers and technical leads, personal pro-activeness and level of experience.

Finally, this chapter summarises sources of intensity in diagrammatic form, in order to illustrate the hierarchy of factors shaping software professionals' experiences of intensity.

8.1 MARKETS AND FIRMS

Market Dynamics

This study has shown that market dynamics, in terms of de-regulation, privatisation, competition, developments in technology and changing consumer demand have had implications for those working within the software industry. The trends documented within this research correspond with issues viewed more widely within the new economy, such as increased competition and pressures on companies to devise quicker, cheaper and more efficient processes (Hornby and Clarke, 2002; Webster, 2000). In addition, regulatory changes, competition, globalisation, downsizing and outsourcing have clearly had implications for IT budgets, priorities, strategies and influence within client organisations (Intellect Software and IT Services Report, 2009; Quintas, 2004). Most notably, the testimonies of software professionals and evidence from managers and organisations has revealed that these factors have influenced strategies concerning the management of costs and deadlines and the nature of contracts and resourcing, with implications for software professionals' experiences of work intensity.

The experiences of software professionals documented here suggest that companies are under increasing pressure to keep abreast of technological developments, meet consumer demand and offer services at lower costs, in order to remain competitive. Certainly, Dyson et al (1996) suggest that developments in ICTs have encouraged greater competition between companies in the new economy with regards to addressing customer needs, lowering prices and offering new technologies. Indeed, the evidence from this study confirms that pressures to launch software more quickly to marketplaces and respond competitively to customer demand have impacted on software professionals through the setting of increasingly aggressive, tight deadlines (O'Riain, 2006; Perlow, 1997; Kunda, 1992).

Furthermore, changes in external investment within the software industry as a result of competitive, financial and efficiency concerns have encouraged companies to increasingly utilise leaner project teams and rotate team members, in order to manage costs. Indeed, Voss-Dahm (2005: 133) has argued that the economic environment can have implications for levels of resourcing within software project teams:

> The economic environment in which a project is implemented is a major factor in determining the degree of latitude in work organisation. The fewer human and material resources a project uses, the greater profit it generates.

This research has ascertained that under-staffing, unwieldy project team size and the rotation of project team members contribute in different ways to work intensity for software professionals. Lean staffing levels can increase workloads, responsibilities and volume of work, whilst increases in project team size can create difficulties in managing and coordinating project team work. The ability to rotate between project teams can allow software professionals to continuously broaden their technical expertise, as well as allow organisations to supplement staffing levels when necessary (Marks and Lockyer, 2004). However, this research offers an alternative perspective by demonstrating that the rotation of project team members can have adverse implications for software professionals' experiences of work intensity. Critically, this study illustrates that the rotation of project team members can increase workloads for remaining individuals if teams become understaffed, contributing to experiences of intensity. Rotation can also interrupt work rhythms and increase workloads for project team members responsible for coaching and assisting new members, further giving rise to experiences of work intensity. Indeed, observations by Applebaum et al (2004) that employee exit from organisations can result in increased workloads, limited resources and tight deadlines for remaining individuals mirror the consequences of rotating software project team members.

This research therefore makes an important contribution through demonstrating the implications that decisions concerning staffing arrangements, competition and deadlines can have for software professionals' experiences of work intensity.

Firm Characteristics

This research has demonstrated that organisational type and company size have implications for the type of software engineering activities performed in-house, the importance of internal software engineering expertise, the work contractors are engaged in, work pace and the emphasis placed on deadlines. These findings support more general claims that work intensity may be influenced by whether an organisation is public or private, company size, business cycle stage or the global business situation (Burchell and Fagan, 2004; Green, 2001). Consideration of context has therefore made a marked contribution to our understanding of firm-related factors which may affect software professionals' experiences of work intensity. This contribution is all the more significant due to the largely de-contextualised nature of the existing research on software professionals.

Organisational type, in terms of the important distinction between companies which operate as specialist software firms or as in-house IT departments within large firms, appears to be a major factor influencing the strategies adopted internally. Crucially, this research contributes through identifying that the emphasis an organisation places on internal software engineering expertise has implications for the software engineering activities performed internally and software professionals' experiences of work intensity. The evidence indicates that organisations which have a core business type beyond software engineering may choose to retain functions deemed of crucial importance in-house and outsource functions deemed to be outside of internal expertise. In addition, globalisation, developments in ICTs and the ability to re-locate software engineering activities have had implications for the nature of internal software engineering expertise within companies where core organisational type resides outside of software engineering. Critically, these factors may encourage greater specialisation of software engineering activities and division of labour within these organisations. The decision to create division of labour may elicit from capital's need to produce more efficiently through specialisation (Rueschemeyer, 1986). Indeed, specialisation may be important in some organisations, particularly where it is perceived that individuals cannot have knowledge in all areas and tasks may exceed capabilities (Andrews et al, 2005).

Nevertheless, this research demonstrates that the move towards reducing the emphasis on internal expertise within in-house IT departments may differ from the situation within specialist software firms, where software engineering and technical skills form the main areas of expertise. Indeed, this study has identified that market dynamics, globalisation and technological developments have had different implications for specialist software firms. Crucially, specialist software firms are more likely to place emphasis on securing project work, juggling existing projects, offering aggressive deadlines and keeping up-to-date with technology, rather than altering the nature of activities performed by individuals. Software professionals may therefore continue to perform more generalised software engineering functions, despite moves towards division of labour within some organisations. Certainly, difficulties in fragmenting software engineering functions and flexibility in work structures can be argued to have encouraged multi-skilling and interdependency between roles within professional software work (Baldry et al, 2007; Baetjer, 1998). In this sense, whilst division of labour may be a conducive strategy within some companies, multi-skilling and flexibility can arguably enable others to respond effectively to unpredictability, uncertainty, changing market conditions and consumer demand (Baldry et al, 2007; Elger, 2001; Piore, 1986).

However, it should be noted that emphasis on multi-tasking and generalisation have been shown within this research to increase volume of work for software professionals and contribute to experiences of work intensity. Experiences of intensity can be particularly prevalent for more senior software professionals, such as Technical Leads, due to the greater volume of work performed within this role. Vitally, the experiences of software professionals match trends viewed more generally in the knowledge economy, where multi-tasking and flexibility are perceived to have expanded workloads and necessitated greater expenditure of work effort, particularly where fewer workers are held responsible for the same quantities of work (Green, 2006, 2001; Thompson and McHugh, 2002; Tomaney, 1990).

In addition, this study contributes to our knowledge by providing valuable insight into the implications that organisational type can have for the nature of activities that contractors are engaged in and resulting experiences of work intensity. The evidence suggests that contractors working for specialist software firms may be hired to carry out specific tasks and have lower levels of responsibility, thereby reducing experiences of intensity. This stands in contrast to permanent employees who may be personally invested in tasks and experience greater intensity as a result. However, contractors who are recruited to carry out software engineering activities which have been outsourced or offshored by in-house IT departments may have similar experiences of intensity to permanent employees working in specialist software firms, due to increased responsibility for tasks. Furthermore, findings from this research tentatively suggest that offshore contractors may be expected to accept greater workloads and work longer hours than permanent employees, exacerbating their experiences of work intensity. Indeed, the ability of software contractors to avoid temporal constraints may depend on the extent to which they are able to exert control over work rhythms or choose which hours or days they wish to work for clients (Barley and Kunda, 1992).

Moreover, this study has identified that company size has implications for work intensity, offering insight into a contextual, institutional element which has been unexplored within existing accounts on software professionals. Most notably, company size affects levels of visibility, decisions concerning staffing arrangements, focus on specialisation or generalisation and levels of bureaucracy. Intensity appears to be greater for software professionals working within smaller firms due to increased visibility of effort, utilisation of leaner project teams to manage costs, emphasis on multi-tasking and greater generalisation of activities. However, despite suggestions that knowledge work occupations may experience lower levels of bureaucracy (Newell et al, 2002; Spender, 1998; Alvesson, 1995; Quinn, 1992), this research has shown that software professionals, particularly those within larger firms, continue to be subject to bureaucratic constraints. Crucially, bureaucracy may contribute to experiences of work intensity, through making it difficult for software professionals to introduce new ideas, understand processes and deal with situations promptly. The discovery within this research that bureaucratic conditions continue to exist for those working within knowledge-intensive areas and have implications for work intensity therefore challenges existing pre-conceptions that knowledge occupations in the ‘new’ economy have completely transformed in terms of work organisation.

It is important to note that whilst market dynamics and firm characteristics have important implications for those working within the software industry, these features alone cannot fully explain work intensity. Crucially, this study identifies that organisational dynamics, interaction within the work process and mediation and control strategies play an equally important role in influencing software professionals’ experiences of work intensity. In this sense, this research makes an important contribution through explaining these distinct, yet still inter-related layers and providing a comprehensive understanding of work intensity within professional software work.

8.2 ORGANISATIONAL DYNAMICS

This section outlines the main organisational dynamics within professional software work which have implications for software professionals’ experiences of work intensity. Firstly, this section challenges the perspective that technological developments have necessarily de-skilled professional software work, by arguing that these trends have instead influenced the type and range of activities performed across the development life cycle. Secondly, it is argued that technological advances have increased volume of work for software professionals and contributed to experiences of work intensity. Finally, this section discusses the implications that developments in structured methodologies and clients have had for software professionals’ experiences of work intensity.

Challenging the De-Skilling Debate

It has been argued that technological developments within the information society have contributed to the mechanisation, automation and de-skilling of work, which may have affected professional occupations (Castells, 2000; Lyon, 1996). This research confirms that improvements in computer speeds, processing power, memory and greater hardware miniaturisation have had implications for professional software work (Baetjer, 1998; Lavoire et al, 1993, 1991; Friedman and Cornford, 1989; Kraft, 1977). Synergies between telecommunications, computer networking, computer capacity and new micro-electronic devices have further stimulated technological developments (Castells, 2000). In addition, this study confirms that professional software work is a separate occupation and distinct from the less skilled areas such as routine IT work, problem-solving/help desk and maintenance work. Furthermore, technological developments and the availability of packaged software have also enabled organisations to purchase and customise software packages, rather than producing systems internally (Quintas, 1994).

However, this research challenges the perspective that technological developments have necessarily de-skilled professional software work. It is important to make the distinction between 'intensification' and 'de-skilling', in that the former did not contribute to the latter within this study. Crucially, the evidence suggests that technological developments have had implications for the type and range of activities that software professionals may perform internally across the development life cycle, rather than de-skilling the nature of work itself. In addition, the experiences documented within this research demonstrate that division of labour can encourage the development of specialist expertise and concentration in particular aspects of the software development life cycle, rather than downgrading or de-skilling the nature of work. Moreover, the discovery that software professionals at InSoft in possession of specialist technical knowledge frequently helped colleagues shows that difficulty in predicting how packaged software interact and behave with systems continues to necessitate technical expertise and skill within in-house IT functions (Quintas, 1994).

Volume of Work

This research has identified that technological advances have intensified professional software work by speeding up cycle times, allowing more activities to occur in parallel and increasing volume of work for software professionals. This research therefore supports claims that new technologies may increase workloads, speed up activities for individuals more generally and stimulate increases in work effort (Bittman et al, 2009; Green, 2004, 2001; Edwards et al, 1998; Gallie et al, 1998;

Green and McIntosh, 1998; Sennett, 1998). In addition, the re-configuration of time and space can be seen to have raised competitiveness, shortened work times and intensified working arrangements for software professionals (O'Riain, 2000).

Crucially, by utilising dimensions of work effort devised by Wichert (2002), this research has identified that volume of work (quantitative aspects), as opposed to the difficulty and complexity of work (qualitative aspects), is a key contributor to daily experiences of work intensity for software professionals. This discovery is particularly novel, in that explorations of dimensions of work effort have been absent from existing studies on professional software workers. Software professionals within technically-oriented roles may juggle work tasks, in order to manage competing work tasks and overall work volume. However, this study illustrates that juggling tasks can further subject software professionals to experiences of work intensity, due to the need to constantly shift attention between tasks, according to changing priorities.

Moreover, this research has established that level of experience can influence the breadth of activities software professionals carry out and resulting experiences of work intensity. The recognition that level of experience has implications for work intensity highlights an aspect which is missing from current accounts. Crucially, the evidence indicates that more experienced software professionals, particularly those operating at Technical Lead level, tend to perform a wider range of activities and have greater volume of work to manage, contributing to greater levels of work intensity. This can be seen in comparison to junior team members, who are more likely to focus in particular areas and perform fewer activities overall, encouraging lower levels of work intensity.

'Agile' Methodology

The supersedence of 'Agile' over the 'waterfall cycle' that was witnessed during the period of this research suggests that the methodologies adopted by the software industry may evolve over time in response to factors such as market dynamics, competitive pressures and client demands. The evidence shows that the 'Agile' methodology is an increasingly attractive alternative to the ‘waterfall’ approach for organisations, by ostensibly helping software professionals manage the increased complexity of systems, ensuring software meets requirements and allowing software to be released into the market more quickly. Whilst existing research on professional software work has focused on the ‘waterfall cycle’ approach, the discovery of ‘Agile’ within this research makes an important contribution through drawing attention to the current evolution for structuring professional software work activities. Indeed, the ‘Agile’ methodology clearly signifies an important step-change for the structuring of professional software work activities, with implications for software professionals’ experiences of work intensity. At the level of the individual software professional, this research has shown that ‘Agile’ can affect the distribution and time span of activities and levels of client involvement, with positive and negative implications for work intensity.

This research demonstrates that the incremental distribution of work across releases under the ‘Agile’ approach can help software professionals manage the intensity of work, through preventing the need for increased effort levels at the end of longer cycles to meet deadlines. In addition, this incremental structure can enable client feedback to be attained through the software development process, helping to improve software professionals’ understanding of clients’ requirements. The benefits of ‘Agile’ have been clearly documented by software professionals engaged in this research:

> Whether the deadline you’ve been set is realistic or unrealistic, there’s a tendency in software engineering to work slowly at the start and then increase your intensity as you head towards that deadline. We try to manage that by setting much shorter deadlines. If you say, “Right, we’re going to deliver the project in its entirety in three months, but we’ll break it into two week trunches”, we’ll hit a milestone every two weeks because people are never far away from a clearly defined milestone. The amount of effort they expend is more widely distributed across the piece, rather than the first two months being a bit of a holiday and then the last month being

> everybody working until twelve o'clock at night (SpecSoft PT2 Technical Lead).

Nevertheless, evidence from this study convincingly demonstrates that under the 'Agile' approach, new tasks can continuously be added to workloads and stages may overlap as software professionals progress through the development life cycle, increasing work volume and contributing to work intensity. Moreover, increased client involvement throughout the development life cycle can provide clients with the ability to interrupt the software work process with questions, clarifications and requests for changes to work already underway, contributing to work intensity for software professionals. Indeed, the testimonies of software professionals within this study have illustrated the negative aspects of the 'Agile' methodology:

> The trouble is that if anything does go wrong, it's added to our workload. That can be a pain, because it's constantly "We need you to finish something here, here and here". If anything comes in here, it's a pain, because you've got a deadline two weeks away, instead of having a deadline three months away and if something did come in, you would still have time to do all this other stuff. So, if something comes in, it can be really intense, because there's no time in there to put your other stuff aside (InSoft Engineering Technical Lead).

The documented mixed consequences of 'Agile' for the professional software work process and for experiences of work intensity within this research therefore poses the question of what future methodologies may have in store for these workers.

Clients

Client involvement can be confirmed to be an increasingly important aspect of professional software work, in order to help software professionals define problems, make clarifications and ensure that software meets requirements (Alvesson, 2004; Beirne et al, 1998). However, customer presence may bring the pressures of markets and competition directly into the workplace, with implications for software professionals (O'Riain, 2010). For example, client indecision, unanticipated crises and customer use of systems may introduce elements of unpredictability to the software work process (O'Riain, 2010; O'Carroll, 2008). Nonetheless, despite these contributions, exploration into the implications that clients may have for software professionals' experiences of work intensity has been missing from current accounts.

This research advances our understanding by demonstrating that clients can impact in diverse and important ways on the degree of work intensity experienced by software professionals. Most notably, this research has discovered that whether clients are external or internal, levels of client technical and business knowledge and the criticality of software can influence software professionals' experiences of work intensity. Clients can be identified as a predominant source of intensity for software professionals engaged in work for external clients or priority work for internal clients. Crucially, client indecision, changing of priorities and regular, sudden requests for alterations to work already in progress can destabilise and interrupt the software work process, contributing to work intensity. Moreover, levels of client interference and involvement may be all the more prevalent where projects are critical to the client organisation. In contrast, software professionals engaged in internal work which is deemed non-essential to the organisation may experience reduced levels of work intensity, due to less priority and importance being given to these tasks.

Possession of technical knowledge, the extent to which projects are central to objectives and stage in the development life cycle have been identified within this research as influencing the role, influence and power of clients. Certainly, Beirne and Ramsay (1988) have suggested that clients may have little participation due to lack of technical knowledge, regulated involvement according to knowledge, power and resources or be fully involved throughout the process. This study convincingly demonstrates that clients' possession of significant technical knowledge can create tensions between the preferred approaches of software professionals and clients themselves, acting to destabilise the overall work process. However, levels of client influence may diminish over time as software professionals obtain greater knowledge of systems over time, allowing workers to exert greater power as a result. In addition, this research supports claims that different stages in the development life cycle may involve varying levels of client interaction, with implications for software professionals' experiences of work intensity. Certainly, experiences of work intensity were found in this study to decrease during lower level design and development and increase when designs or releases were delivered to clients for review. Finally, this research demonstrates that internal business knowledge held by software professionals, particularly those working within in-house IT departments, can provide these individuals with greater power to make the business case for requirements.

8.3 INTERACTION IN THE SOFTWARE WORK PROCESS

This section argues that the interactive and collaborative nature of professional software work has important implications for work intensity. Notably, the evidence convincingly demonstrates that specialist knowledge, the dynamics of interruptions, work location, the physical proximity of project team members and the use of technological mediums influence software professionals' experiences of work intensity.

Specialist Knowledge

This research confirms that software project teams can be typified by heterogeneous skills and responsibilities, differing individual responsibilities and degrees of specialisation (Marks and Lockyer, 2004; Tam et al, 2002; Kraft, 1977). Project team structures can clearly enable software professionals to supplement each others expertise, support collaboration and assist the integration of work activities (Tam et al, 2002; Baetjer, 1998; Baldry et al, 1998; Walz et al, 1993; Kraft, 1977). However, the discovery within this research that interdependencies and the presence of specialist knowledge can contribute to work intensity demonstrates the failure of existing studies to acknowledge the negative repercussions of heterogeneity within project teams. Crucially, evidence from this research illustrates that whilst more knowledgeable individuals such as Technical Leads may experience interruptions to work rhythms due to the need to coach colleagues, less experienced team members may be subject to increases in workload and effort levels to meet timescales when help is unavailable.

Documentation has been identified within this research as being a useful source of information and guidance for software professionals, helping to mediate experiences of work intensity for those providing and requiring assistance. However, the evidence also indicates that the tacit, intangible and intellectual nature of professional software work prevents the full codification and formalisation of knowledge. In this sense, knowledge work appears to be difficult to express and codify due to its tacit and intangible nature and its relation to experience, intuition, dynamism and relation to context (Newell et al, 2002; Bird, 1995). Furthermore, sharing knowledge which is tacit, informal and specialised, or attempting to embed informal work practices, may be particularly difficult where software project teams are geographically dispersed (Boreham et al, 2008). In this sense, interactions between project team members continue to be necessary to enable individuals to fully convey and share tacit knowledge that is attained through experience, intuition and abilities (Newell et al, 2002; Thompson and McHugh, 2002).

Interruptions

This research clearly demonstrates that interactions with project team members, project managers and clients are necessary due to the collaborative and interdependent nature of professional software work. Professional software work in its current form therefore differs significantly from the historical conditions identified by Kraft (1977), where individuals worked in private, quiet places, free from distractions. Crucially, this research maintains that software professionals need to interact with colleagues, project managers or clients on a daily basis, either face-to-face or via technological mediums, to ask for assistance, request clarifications, problem-solve or brainstorm. Clients may also ask for additional tasks to be added to workloads without altering deadlines, request clarifications or ask for problems to be investigated.

This study confirms that interruptions are a normal part of the work process within professional software work and may arise due to office layout, the need for collaboration, client unpredictability and the use of technological mediums (Boisard et al, 2008; O'Carroll, 2008; Voss-Dahm, 2005; Perlow, 1997; Kunda, 1992). In addition, the unpredictable and spontaneous nature of interruptions within professional software work clearly affects thought processes, disrupts work rhythms and fragments schedules (Perlow, 1997). Crucially, this research makes an important contribution by demonstrating that interruptions contribute to work intensity for software professionals:

> You've got your own task to complete and, again, this comes back to the thing about being able to concentrate on fixed chunks of work for extended periods of time, uninterrupted. If you're trying to do a chunk of work, concentrate on it and you keep getting interrupted, "How do you do this?", '"Where, where would you put this?", "How is this done?", then that interrupts your own work and that can't help but affect you (SpecSoft PT1 Average Team Member).

Moreover, the discovery that interruptions contribute to work intensity for software professionals challenges the position presented by Boisard et al (2008) that interruptions that may occur within occupations in general are not necessarily disruptive.

Perlow (1997) has argued that organising interactions according to priority or scheduling time blocks for 'quiet time' may help software professionals manage the disruptive and unpredictable nature of interruptions. In addition, Perlow (1997) has suggested that individuals who are willing and supportive may experience interruptions to a greater extent, compared to those who attempt to manage their interactions. However, this research challenges these perspectives by arguing that in reality, it may be difficult for software professionals to strictly manage and prioritise interactions. Crucially, the evidence convincingly demonstrates that the collaborative, interactive, interdependent and unpredictable nature of professional software work can make it difficult for individuals to plan and schedule interactions.

Moreover, particular individuals may be less able to control and prioritise their availability to project team members. For instance, this research has shown that interruptions are especially prevalent for software professionals possessing higher levels of expertise, due to greater levels of responsibility and the need to provide coaching and assistance to less experienced team members.

This research has also demonstrated that the approach taken by project managers can impact on software professionals and their experiences of work intensity. Crucially, project managers may fail to formally recognise the presence of interruptions or take into consideration the effect these can have on software professionals' ability to manage existing workloads or meet deadlines, exacerbating experiences of work intensity. These observations build upon the position adopted by Perlow (1997), who has suggested that organisational practices and management styles may influence the approach individuals take with regards to managing their working time and interactions.

In summary, this research therefore challenges existing perspectives by arguing that interruptions may be unavoidable and necessary, due to the collaborative, interactive and unpredictable nature of professional software work. In addition, this study makes an important contribution through identifying that levels of experience, specialist knowledge and project management approach can influence the extent to which software professionals experience interruptions and the impact on work intensity.

Location and Physical Proximity

The evidence from this study confirms that globalisation and developments in ICTs have helped companies move beyond the national context, provided greater access to production capabilities and markets and enabled greater flexibility over choice and location of labour (OECD Information Technology Outlook, 2006; McGrath-Champ, 2005; Tomkins, as quoted in Callinicos, 2001; Castells, 2000). Globalisation appears to present companies with potential benefits, such as cost savings, the ability to take advantage of attractive conditions elsewhere and to offer 'round the clock' services

(Upadhya, 2009; Aneesh, 2006; OECD Technology Outlook, 2006; O'Riain, 2006; McManus and Floyd, 2005; May, 2002; Carnoy et al, 1993). In addition, the outsourcing or offshoring of particular functions can arguably enable companies to take advantage of innovative services, fill in internal skill gaps, receive services required on a short-term basis and externalise activities which are deemed less important to central operations (Key Note Market Report, 2008c; Aneesh, 2006; McManus and Floyd, 2005; Arora et al, 2001). Furthermore, the availability of telecoms networks, computers, computing devices and mobile communication devices has arguably enabled greater networking and global integration, making time and space increasingly irrelevant in the new economy (Stehr, 2004; Castells, 2000).

Newell et al (2002) have argued that the utilisation of technological mediums and teleworking can be effective if team members share the same reference points and possess a common language. However, this study has established that interactions with team members are an essential part of professional software work due to the intangible, creative and tacit nature of work, making it difficult to communicate through purely technological means. Crucially, professional software work can be confirmed to be a collaborative work process, requiring individuals to interact with project team members in order to problem-solve, supplement expertise, bring together new skills and knowledge, make clarifications and ask questions (Baldry et al, 2007; Marks and Lockyer, 2004; Swart and Kinnie, 2003; Tam et al, 2002; Kofman and Senge, 1993). The ability to interact and communicate easily with project team members can be deemed essential for knowledge-intensive occupations, such as professional software work, due to the intangible, creative, tacit and complex nature of work:

> Teamwork is critical. Since neither problems or solutions can be defined in advance, frequent and informal conversations help ensure that insights and discoveries are put to their best uses and subjected to quick, critical evaluation (Reich, 1991: 179).

In addition, this study confirms that globalisation and ICTs have had implications for work location, the physical proximity of project team members, interactions and methods of communication (Osnowitz, 2010; O'Riain, 2006, 2000; Benner, 2002). Crucially, this research makes an original contribution through demonstrating that these developments have had implications for software professionals' experiences of work intensity. Notably, this study demonstrates that close physical proximity of project team members within professional software work can help to reduce work intensity, through allowing issues to be resolved promptly, aiding problem-solving, brainstorming and preventing misunderstandings. In contrast, the geographical dispersion of project teams can clearly contribute to work intensity for software professionals, through creating difficulties in communicating and dealing with situations promptly, monitoring overall team progress effectively and managing work across time zones. Indeed, Aneesh (2006) and O'Riain (2000) have argued that differences in time zones may hinder communications, the ability to promptly resolve problems and individual capacity to effectively organise daily work activities. In addition, the finding that offshore workers completed work outside of specified times at InSoft demonstrates the importance of following agreed timings if benefits are to be secured from distributing work globally:

> …this new timing of organisational flows must follow the day-and-night pattern strictly if it is to function properly as the team in India must finish the task during their daytime (Aneesh, 2006: 85).

Notably, the evidence of work diaries and interview scripts from Indian offshore contractors have provided unprecedented insight into the experiences of work intensity at a remote global location and how these compared to on-site worker experiences. For instance, the documented experiences of software professionals distinctly show that language barriers and communication difficulties can give rise to misunderstandings within geographically dispersed project teams, with detrimental consequences for work outcomes. Indeed, as detailed in Chapter Six:

> If you're all sitting together, then the communication's cut down. That kind of closeness is completely lost [with geographically dispersed teams]. It's much more difficult and in terms of the quality of work, there can be lots of mistakes. There's a team who

> thought I was saying on the phone the first of August; I was saying the fifth of August. They were running around trying to do things for the first of August. It's purely a verbal communication problem (InSoft Project Manager).

In addition, this study confirms that attempts by software professionals to screen information from project managers located elsewhere may create future difficulties, such as tasks being added to existing workloads close to deadlines (O'Riain, 2000).

Moreover, whilst companies operating globally may attempt to reproduce universal corporate cultures, this research demonstrates that cultural differences and contrasts in working styles may impact adversely on the work process and contribute to work intensity. Indeed, the difficulties experienced by globally dispersed teams within this research can be seen to echo the obstacles diverse occupational communities may face in creating common understandings (Bechky, 2003). In this sense, the ability to communicate via technological means may not compensate for the benefits to be secured from face-to-face interactions (Boreham et al, 2008; Baldry et al, 2007; Aneesh, 2006; O'Riain, 2006, 2000).

The evidence gathered from this research has also revealed that poor management of offshoring relationships and adherence to uniform procedures can further contribute to experiences of work intensity for software professionals working within geographically dispersed project teams. Indeed, companies may devote greater time to establishing contracts with providers, to the detriment of managing relationships and contracts effectively (Key Note Market Report Computer Services, 2008). Effective management of offshoring relationships and the establishment of uniform procedures, working patterns and objectives are therefore crucial in helping to offset software professionals' experiences of work intensity. Notably, the difficulties faced by software professionals at InSoft demonstrates the importance of establishing clear communication patterns, ensuring mutual cultural understandings, instilling quality standards and explicitly define working procedures (Huws and Flecker, 2004). In addition, combining the use of technological mediums with periodic travel to work sites may potentially help to prevent unnecessary misunderstandings (Huws and Flecker, 2004).

Technological Mediums

This study supports claims by Poster (1990) that the information age has allowed communications to occur through a variety of mediums, such as oral exchanges, written and printed exchanges and electronic exchanges. Indeed, the software professionals who were the subjects of this study interacted with colleagues, project managers and clients directly or used technological mediums, such as phone calls, e-mails and instant messenger to communicate. In addition, experiences at InSoft illustrated that project teams whose members are spatially dispersed are more likely to utilise technological mediums to communicate on a daily basis as a result of difficulties in interacting face-to-face. Crucially, this research has identified that technological mediums have implications for software professionals' experiences of intensity, depending on the directness and personal utilisation of methods.

The evidence from this research demonstrates that phone calls are the most intrusive method of communication for software professionals. Most notably, phone calls clearly disrupt thought patterns due to the need to provide an immediate response and contribute to work intensity for individuals. O'Carroll (2008) has argued that e-mails may interrupt work rhythms for software professionals, through ambiguity over content or difficulty in predicting the time it may take to read and write messages. However, this research makes an important contribution through demonstrating that the extent to which e-mails and instant messenger give rise to intensity depends on the personal utilisation of methods by software professionals themselves. For example, the results from this study indicate that e-mails and instant messenger can enable software professionals to choose when to respond, reducing the extent to which these methods intrude in the work process and helping to manage experiences of intensity. Concurrently, these mediums can contribute to work intensity for those individuals who perceive that they should respond immediately.

The discovery that technological mediums have implications for the intensity of professional software work makes a marked contribution to existing literature which has predominantly focused on how these methods have revolutionised society, provided flexibility over coordinating and controlling schedules and increased intensity of work more generally (Bittman et al, 2008; Golder and Feisler, 2007; Felstead et al, 2005; Haddon, 2004; Ling, 2004; and Katz and Aakhus, 2002).

8.4 CONTROL AND MEDIATION STRATEGIES

This section outlines the strategies within firms which can be seen as an attempt to manage the organisational and interactional processes outlined in parts 8.2 and 8.3 and their influence on work intensity. In addition, this section explores variations in how individuals respond to workplace strategies and implications for experiences of work intensity.

Leadership

Transformations in the era of dot.com companies can be confirmed to have stimulated changes in organisational structures, leadership style and skill sets (Edward and Wacjman, 2005; Woodfield, 2000). This research confirms that software professionals, particularly those performing activities across the development life cycle, require a variety of technical, interpersonal, business and political skills in order to perform their work effectively (Lerouge et al, 2005; Sonnetag, 1994; Curtis et al, 1988). In addition, software professionals may need to develop their skills and knowledge continuously, in order to keep up-to-date with the expansion and rapidly changing nature of technology in the software industry (Couger et al, 1992). Sonnetag (1995) has acknowledged that whilst clients may place emphasis on software professionals' possessing business knowledge, good communication skills and orientation to users, software professionals may consider technical knowledge, good working style in structuring problems, adopting an individual approach whilst operating to team standards and interpersonal skills as being more desirable. However, whilst it beneficial to consider the outlook of clients, this research argues that it is essential to include project management and senior management perspectives in the discussion.

Crucially, this research draws attention to the implications that project management and senior management perspectives can have for the type of software engineering activities performed within organisations and how these can shape experiences of work intensity. Most notably, this study demonstrates that software professionals, project managers and senior management may place differing levels of importance on technical skills, depending on career background and progression. Project managers and senior management may be more inclined to place importance on technical skills if they have had previous experience working as software engineers. However, project managers and senior management who have progressed through managerial routes may be more likely to place greater emphasis on managerially-oriented software engineering activities such as business and design, to the neglect of technical areas.

This research makes an important contribution through identifying that leadership style can shape experiences of work intensity, depending on the approach taken at director, senior management, project management and Technical Lead levels. Facilitative, supportive and co-operative leadership styles can be confirmed to be the most appropriate methods for managing software professionals and most notably, help to manage experiences of work intensity. Crucially, supportive and co-operative styles can facilitate software professionals' ability to collaborate, exercise their judgement, problem-solve and carry out their work effectively within an intangible, intellectual and creative work process (Thompson and McHugh, 2002; Spender, 1998). This confirms general observations that management styles which focus on coordinating, facilitating and supporting work activities may be more suited to knowledge work occupations, rather than traditional command and control structures (Mathews, as quoted in Thompson and McHugh, 2002; Newell et al, 2002; Thompson and McHugh, 2002; Spender, 1998).

Furthermore, this study adds value by demonstrating the complexity of supervision within professional software work. For example, whilst project teams may formally have an overall project manager and Technical Leads to oversee and monitor work, team members can also informally play a part in supervising the software work process, through their own internal motivation.

Skills and Knowledge Development Opportunities

The experiences of professional software workers within this research serves as an illustration of trends in the new economy more widely for individuals being increasingly responsible for managing their own careers and employability (Baldry et al, 2007; Herriot and Pemberton, 1996; Heckscher, 1995). Indeed, whilst organisations may provide access to training and develop opportunities and support individuals through the process of performance appraisals with goal setting, this research confirms that career management appears to rest primarily with software professionals themselves (Ituma and Simpson, 2006; Arnold, 2005; Baruch, 2004; Cohen and Mallon, 1999; Cappelli; Rousseau, 1995). Tensions in managing employer and personal aspirations are therefore not readily evident, generally as a result of individuals being largely in charge of their own careers and employability. This discovery challenges the position held by Marks and Lockyer (2004) and Marks et al (2002) that software professionals may experience difficulties in balancing individual aspirations, team requirements and employer objectives. Furthermore, suggested moves in the economy from traditional contracts focused on long-term employment and promotion from within to self-managed, 'boundaryless' careers have not been completely substantiated by this research, in that many individuals in both companies had longer tenures. In this sense, skills and knowledge development opportunities and pursuit of aspirations can be seen to have negligible influence on software professionals' experiences of work intensity due to individual responsibility for these areas.

Deadlines

Conventional wisdom suggests that knowledge work occupations are characterised by non-bureaucratic working conditions, flexibility, facilitative management styles, autonomy, project team forms of work organisation and opportunities for development (Baldry et al, 2007; Baldry et al, 2005; Cappelli, 2000; Alvesson, 1995; Kunda, 1992). This research confirms that the intangible and intellectual nature of professional software work requires an autonomous, flexible and facilitative approach to managing workers. In addition, the evidence confirms that software professionals have the ability to exercise operational, technical and time autonomy over their work (Barrett, 2005; Voss-Dahm, 2005; Newell et al, 2002). Autonomy can be identified as an important characteristic of professional software work, due to the need for individuals to deal with client indecision, ambiguity, manage workloads and depart from planned schedules when necessary (Voss-Dahm, 2005; Gibbons et al, 2002; Newell et al, 2002; Warhurst and Thompson, 1998; Friedman and Cornford, 1989). However, autonomous working conditions can be confirmed to encourage individuals to control themselves in the economic interests of the firm, securing their responsibility to work harder and longer to complete tasks to deadlines (Rasmussen and Johansen, 2005). Crucially, this research has shown that autonomy tends to occur within the constraints of deadlines, supporting the position that deadlines form the main mechanism around which all professional software work activities are structured, organised, managed and controlled (Baldry et al, 2007; O'Riain, 2010, 2006; Andrews et al, 2005). In this sense, deadlines are equally recognised to be an integral part of the software work process, meaning that these workers can be subject to structure and control akin to any other occupation.

Working time can be confirmed to be an important aspect of professional software work and may be influenced by the interplay between markets, clients and organisations (O'Riain, 2010). Indeed, this study illustrates that the pressures faced by clients to deliver projects more efficiently and cost-effectively may be passed on to workers in general, stimulating the need for tight deadlines and sustained speed of working (Boisard et al, 2008). Furthermore, this research shows that it is increasingly necessary for companies employing software professionals to apply aggressive deadlines, in order to compete more effectively in the marketplace. Certainly, O'Riain (2006) argues that the application of deadlines as a means of exerting control over professional software work can be justifiable on the grounds of enabling companies to remain competitive. This research has therefore established that aggressive deadlines and client pressures to meet these can form a powerful form of temporal control within the work process, with implications for software professionals' experiences of work intensity. Crucially, deadlines are a key contributor to work intensity for software professionals engaged in providing critical software for external or internal customers. Indeed, these observations confirm that deadlines can regulate professional software workers' expenditure of effort and work time, enabling control to be exerted over a work process which is otherwise intangible, ambiguous and creative in nature:

> The deadline is the mechanism by which management brings the intensification of time into the heart of the team. It is also an attractive mechanism of control since direct authority over the work process is undermined by the expertise of the employees and the need for rapid communication and cooperation. In contrast, time can be regulated through the use of the deadline with only a limited managerial presence and with relatively little ongoing exercise of managerial authority (O'Riain, 2006: 13).

Certainly, Sharone (2004) has argued that software professionals may naturally increase their expenditure of work effort, in order to meet impending deadlines.

However, this research has noted that deadlines may be less prevalent for software professionals engaged in non-critical or less client-centred work. In this sense, the criticality of software to the client, whether work is client-centred and immediacy of release dates can be identified as influencing the extent to which deadlines contribute to software professionals' experiences of work intensity.

Normative Control

The intrinsically satisfying attributes of professional software work have also been identified to encourage normative forms of control, meaning that individuals can influence their own experiences of work intensity. Professional software work can be confirmed to be challenging, stimulating and intrinsically satisfying, with individuals enjoying elements such as problem-solving, creativity, technical challenges and writing code. For example, this was illustrated in Chapter Five:

> In general, I really enjoy solving technical problems. I get an awful lot of satisfaction from doing that and just writing. If you write

> something that's technically difficult and then you get to the end and you see the end result and it works well and the client's happy, that's great. I even enjoy testing of code because sometimes if you see something that's really good, you just get a great feeling about it (SpecSoft PT1 Technical Lead).

In addition, this research illustrates that professional software workers may experience satisfaction from realising potential and achieving assigned tasks, traits seen to characterise those working within knowledge-intensive occupations (Tampoe, 1993). Crucially, the intrinsically satisfying attributes of professional software work can encourage normative forms of control, such as internal motivation to work, self-supervision, loyalty to organisational objectives and the management of behaviour and attitudes (Alvesson, 2004; Kunda, 1992; Boreham, 1983). Normative control can motivate software professionals to expend greater effort levels in response to the cyclical nature of work, removing the need for direct managerial supervision (Green, 2005; Thompson, 2003; Lockyer et al, 2001; Gallie et al, 1998; Deetz, 1995). Thus, this research supports the position of Baldry et al (2007) by arguing that flexibility and the need to accept additional responsibilities may be considered by software professionals to simply be part of the work process. Certainly, flexibility and self-management can be seen in general to have encouraged individuals to have greater personal investment in their work, contributing to work intensity (Green, 2004; Thompson, 2003; Gallie et al, 1998).

Identification with the occupation, team and organisation can be further identified as motivating software professionals to adapt themselves to the cyclical demands of work. Indeed, it has been claimed that project team structures may be more effective at securing and extracting value from individuals than traditional control methods (Sharpe, 1998). However, this research challenges the perspective that project team structures necessarily encourage peer control and increased surveillance (Green, 2006, 2004, 2001; Findlay et al, 2000; Springer, 1999; Baldry et al, 1998). For example, this study convincingly demonstrates that whilst Technical Leads are responsible for monitoring progress and reporting this to project management, this process tends to be supportive in nature as a result of the collaborative nature of work. In this sense, internal motivation to work, heterogeneity of knowledge and the cyclical nature to work appear to influence expenditure of work effort, rather than peer pressure.

Performance Metrics

Performance metrics such as performance appraisals, performance-related pay, status reporting, log sheets and code reviews can be confirmed to be part of the professional software work process. Existing research has suggested that performance metrics may represent attempts by management to exert control over the labour process and influence how individuals manage their work time and effort (Sharone, 2004; Beirne et al, 1998; Friedman and Cornford, 1989). For instance, Sharone (2004) has argued that self-managed performance grading structures may encourage software professionals to self-impose long working hours or increase expenditure of work effort, in order to portray images of professional competence. However, this study challenges these positions by arguing that performance metrics appear to be viewed as part of the normal work process by software professionals and do not contribute to experiences of work intensity. This research has demonstrated that software professionals perceive that these metrics are necessary to keep track of project work, work chargeability to clients and progress in reaching deadlines. In addition, code reviews can enable software professionals to evaluate personal work content and identify improvements, fitting observations by Friedman and Cornford (1989) that these form an important part of the learning and mentoring process. Finally, these mechanisms can provide software professionals with information on the work process, enabling individuals to develop skills, knowledge and more fully understand systems created by other individuals.

Software professionals can therefore influence their own experiences of intensity, regardless of the pressures placed on them by others, as a result of internal motivation, autonomous time management and the extent to which they identify with the occupation, team and organisation.

Breaks

This research contributes through providing insight into the approach that software professionals take with regards to breaks, serving to further illustrate internal motivation to work and implications for experiences of work intensity. Crucially, whilst O'Carroll (2008) has acknowledged that knowledge workers may have 'fuzzy holes' in the working day when they switch between tasks or take breaks, the effects of taking or not taking breaks on software professionals' experiences of work intensity has been unexplored within existing literature. This research shows that software professionals are less likely to take breaks when working on challenging or intellectually stimulating tasks, demonstrating personal investment in tasks. Indeed, Baldry et al (2007) and Perlow (1997) suggest that breaks may become shorter or be taken less frequently in response to impending deadlines. Notably, this study shows that taking breaks during periods of concentration can increase experiences of work intensity. However, it should equally be recognised that breaks can help manage experiences of work intensity through aiding relaxation and helping to create a fresh outlook. Furthermore, the approach software professionals take with regards to breaks may depend on nature of roles, levels of responsibility and professional cultures. Indeed, Chapter Six illustrated the influence that professional cultures had on approaches to work at InSoft:

> Those who are not permanent [InSoft] employees are employed by an Indian vendor. Their culture is to work many more hours than they are contracted to, working days are significantly longer and they're expected to travel more. They don't get the same lunch breaks or the same annual leave as [InSoft] employees. I try to treat every member of the team equally but the professional culture often has a greater influence than I do (Project Manager).

Certainly, Upadhya (2009) has suggested that software professionals in India may be expected to negotiate workloads, deadlines and the allocation of work time with managers. In this sense, it is useful for organisations to understand the internal motivations of knowledge workers, in order to identify favourable working conditions and conducive organisational forms (Alvesson, 2004).

Mediating Strategies

This research confirms that consent and accommodation strategies within the labour process may be necessary to harness and motivate the creative and reproductive power of workers (Boreham et al, 2008; Edwards and Scullion, 1982; Cressey and MacInnes, 1980), particularly in the case of software professionals. However, whilst Edwards (1979) suggests that consensual methods may minimise worker opportunities and resistance, the ownership of intellectual capital in the knowledge economy and the potential to withdraw labour can enable knowledge workers to exert greater influence and negotiation within the employment relationship (Robertson and Swan, 2004; May, 2002). Indeed, this research convincingly demonstrates that software professionals have the ability to exercise agency within the software labour process, due to the possession of desirable skills and knowledge and the intellectual, tacit nature to work. Crucially, the potential for worker agency within professional software work demonstrates the importance of considering both subjectivity and objectivity within this research. Indeed, Chapter Three illustrated that a purely objective approach could underestimate the knowledgeability and capability of workers and the ability of management to monopolise elements of knowledge (Wilson, 1988; Burawoy, 1978).

This research therefore makes a valuable contribution through identifying factors which may mediate software professionals' experiences of work intensity. These include software professionals' ability to manage relationships with clients, the approach taken by project managers, pro-activeness and levels of experience.

This research demonstrates the benefits to be accrued from software professionals establishing good working relationships with clients. For example, forming good working relationships can help to encourage compromise and discussion on issues such as unrealistic timescales, potentially alleviating experiences of work intensity. Good working relationships can also potentially help with the retention of clients, with Alvesson (2004) suggesting that clients may prefer to work with individuals they have previously encountered. This research also illustrates that software professionals can avoid clients destabilising the work process by encouraging them to agree specifications early on and ensuring clients adhere to these. In addition, confirming details from client discussions and decisions via e-mail can allow software professionals to accurately retain and record information, further serving to prevent client digression from objectives and alleviating client pressures. Possession of a range of skills, spanning technical, interpersonal, political and business areas can also be seen to help software professionals in their ability to manage clients more effectively.

Project managers have been singled out in this research as the group with the potential to increase or reduce software professionals' experiences of work intensity, due to their power over deadlines, planning of workloads, allocation of tasks and the ability to buffer workers from client complaints and interference. Software professionals' experiences of work intensity can therefore be influenced by the type of leadership style adopted by project managers and the extent to which project managers are willing to consider individual perspectives and opinions. Technical Leads can play a further role in this process, due to their role in reporting issues to project managers and providing task time estimates. However, it is important to note that the extent to which professional software workers communicate difficulties to Technical Leads and project managers can shape experiences of work intensity.

This research makes a further valuable contribution through identifying that software professionals themselves can play an important role in managing personal experiences of intensity. Notably, the extent to which software professionals make Technical Leads or project managers aware of work overload, conflicting tasks and client interference can affect the ability of these layers of leadership to alleviate work intensity. Furthermore, the failure of project team members to communicate difficulties to project managers in a timely fashion can contribute to work intensity for other team members, through the sudden allocation of unexpected tasks and greater expenditure of work effort being necessary on approaching deadlines. Indeed, O'Riain (2000) has argued that screening problems from project managers may potentially create difficulties for software professionals at a later stage when deadlines are approaching. In addition, experiences documented within this research suggest that recognition of the importance of meeting deadlines by software professionals themselves can help prevent work tasks from accumulating, control the expenditure of effort and manage experiences of work intensity. Certainly, Perlow (1997) has suggested that working reactively, rather than pro-actively, may have a detrimental effect on the ability of software professionals to complete work to deadlines.

This research has identified that more experienced software professionals are subject to higher levels of work intensity, as a result of having to manage more responsibilities and larger workloads. However, it should be noted that greater experience can potentially help software professionals in learning how to manage work intensity, as a result of familiarity in dealing with situations. Indeed, Ackroyd (2004) and Archer (1998) have argued that individuals may vary in the extent to which they are complicit in reproducing structures or are capable of producing changes, depending on interests, powers, resources, constraints and the nature of relationships. This finding corresponds to more general observations made by Boisard et al (2008) that lack of familiarity in dealing with intense situations may contribute to intensity for younger, less experienced individuals. In addition, software professionals with greater levels of experience may possess more accurate understandings of how long tasks will take, enabling them to exert greater influence over task time estimates. In this sense, whilst software professionals may be unable to directly alter deadlines, the ability to encourage the setting of more realistic completion timescales for tasks can arguably help manage experiences of work intensity. This research therefore demonstrates the importance of examining occupational level when conducting research, in that this aspect may influence the resilience and vulnerability of individuals to work intensity (Boisard et al, 2008; Wichert, 2002).

CONCLUSION

This chapter has provided an in-depth explanation of the incidence and impact of work intensity on professional software workers. At its simplest level, professional software work can be confirmed as being subject to work intensity. Firstly, markets and firms have had implications for work intensity within the software industry. Notably, market dynamics can be seen to have affected IT budgets and priorities within client organisations (Intellect Software and IT Services Report, 2009; Quintas, 2004) and influenced the strategies adopted within companies, with implications for software professionals' experiences of work intensity. In addition, firm-related factors such as organisational type and company size have been identified as influencing the emphasis placed on internal software engineering expertise, the type of work that contractors are engaged in, work pace, deadlines, levels of visibility, staffing arrangements and focus on specialisation or generalisation. This research has also challenged the perspective that knowledge occupations are necessarily subject to lower bureaucracy (Newell et al, 2002; Spender, 1998; Alvesson, 1995; Quinn, 1992) by demonstrating that bureaucracy still exists within larger firms and has implications for software professionals' experiences of work intensity.

Secondly, this chapter has emphasised that organisational dynamics can be seen to influence work intensity. Notably, technological developments have influenced the type and range of activities software professionals may perform across the development life cycle, rather than downgrading or de-skilling the work itself. This study has also identified that technological developments have sped up cycle times, allowed activities to occur in parallel and increased volume of work, contributing to work intensity for software professionals. Recognition within this study that level of experience can influence the breadth of activities software professionals carry out and experiences of work intensity highlights a further aspect which is missing from current accounts. Discussion of 'Agile' within this research and recognition that this methodology can both manage and increase work intensity lends an important contribution to understanding present forms of work organisation within professional software work. In addition, this research has advanced our understanding by demonstrating that clients can impact in diverse and important ways on the degree of work intensity experienced by software professionals.

Thirdly, this chapter has shown that interaction within the software labour process has implications for work intensity. Crucially, this research has shown that interdependencies, the presence of specialist knowledge and the dynamics of interruptions can influence software professionals' experiences of work intensity. Moreover, this chapter has established that whilst outsourcing and offshoring may provide potential benefits to companies (Upadhya, 2009; Key Note Market Report, 2008c; OECD Technology Outlook, 2006; McManus and Floyd, 2005; May, 2002; Arora et al, 2001; Carnoy et al, 1993), the geographical distribution of project teams can contribute to work intensity for software professionals. Crucially, the evidence demonstrates that close physical proximity of project team members can help to reduce software professionals' experiences of intensity, through aiding problem-solving, allowing issues to be resolved promptly and preventing misunderstandings. This study has also made an original contribution through demonstrating that technological mediums such as phone calls, e-mails and instant messenger have implications for software professionals' experiences of work intensity.

Fourthly, this chapter has provided comprehensive insight into the strategies within firms and variations in workers' responses which influence work intensity. This research has drawn attention to the implications that leadership background and leadership style can have for software professionals' experiences of work intensity. Moreover, this research has made an important contribution through identifying that the criticality of software to the client, whether work is client-centred and immediacy of release dates can influence the extent to which deadlines contribute to work intensity. The intrinsically motivating attributes of professional software work and identity at occupational, team and organisational levels have been found to shape experiences of work intensity. This study has also provided valuable insight into the approach that software professionals take with regards to breaks and implications for work intensity, drawing attention to an aspect which has been unexplored within existing literature. Finally, this study has identified particular factors which can mediate software professionals' experiences of work intensity. These include software professionals actively managing relationships with clients, the utilisation of facilitative, supportive leadership styles by project managers and Technical Leads, personal pro-activeness and level of experience.

The following diagram (Diagram 8.1) summarises the hierarchical layers shaping software professionals' experiences of work intensity:

MARKETS AND FIRMS

Competition
Legislation
De-Regulation and Privatisation
Staffing Arrangements
Organisational Type
Company Size

ORGANISATIONAL DYNAMICS

Volume of Work
Level of Experience
'Agile' Methodology
Clients

INTERACTION IN THE WORK PROCESS

Specialist Knowledge
Interruptions
Work Location
Physical Proximity
Technological Mediums

CONTROL AND MEDIATION STRATEGIES

Leadership
Deadlines
Normative Control
Breaks
Managing Relationships With Clients
The Approach Taken By Project Managers and Technical Leads
Personal Pro-Activeness
Level of Experience

DIAGRAM 8.1 SUMMARY OF FACTORS AFFECTING EXPERIENCES OF WORK INTENSITY

CHAPTER 9 CONCLUSION

9.1 OVERVIEW

This chapter will begin by re-visiting the five research objectives composed at the beginning of this ‘logic of discovery’, in order to provide final reflections on the answers which this thesis has attempted to provide. This chapter will also summarise the main contributions from this PhD. Notably, this PhD has established that software professionals are subject to work intensity from diverse sources. Advances in ICTs, globalisation, flexibility and developments in the software industry have been identified as having implications for software professionals’ experiences of work intensity. In addition, this study has demonstrated that aspects of the software labour process, including deadlines, project team structures, specialist knowledge, interruptions, normative control, breaks and worker agency, have implications for software professionals’ experiences of work intensity. Furthermore, the development of a taxonomy of professional software job roles and the utilisation of the work diary study have provided a comprehensive understanding of the tasks and activities performed by individuals. Finally, this PhD has demonstrated that contextual aspects, such as market dynamics, firm characteristics and internal organisational factors can shape software professionals’ experiences of work intensity. This chapter concludes by outlining the limitations of this study and suggesting areas for future research.

9.2 RE-VISITING THE RESEARCH OBJECTIVES

The meta-narrative of the knowledge economy has brought forth contending perspectives on globalisation, developments in ICTs, the status of theoretical knowledge and flexibility over the choice and location of labour. For example, optimistic outcomes, including the up-grading of knowledge and skills, improved communications and greater satisfaction at work, have appeared alongside the negative consequences of rationalisation, de-skilling, efficiency, displacement, de-skilling, greater self-management and the expansion of responsibilities (Baldry et al, 2007; Kumar, 2005; Alvesson, 2004; May, 2002; Thompson and Warhurst, 1998).

Some existing literature has pointed to the potential for work intensity within the new economy. For example, flexibility, multi-skilling and empowerment may have stimulated the integration of tasks, expanded workloads, increased performance pressures and encouraged greater expenditure of work effort (Bittman et al, 2009; O'Riain, 2006; Green, 2006, 2004, 2001; Thompson and McHugh, 2002; Tomaney, 1990; Gallie et al, 1998; Green and McIntosh, 1998; Sennett, 1998). Furthermore, ICTs, privatisation and commercialisation may have changed the nature of competition and intensified markets in the new economy (Hornby and Clarke, 2002; Castells, 2000; Webster, 2000; Dyson et al, 1996).

Crucially, this thesis has argued that consideration of the key debates surrounding the knowledge economy should impart, or at least rest upon, those who create its infrastructure. Research on software professionals has therefore been deemed to be of particular interest, due to their role in creating a key technology – software – which has applications for virtually all work, business and social situations (Freeman and Perez, 1998; Quintas, 1994) and their presumed status as archetypal knowledge workers (Baldry et al, 2007; Baldry et al, 2005; Newell et al, 2002). Thus, the central aim of this PhD has been to study work intensity in the software industry, its incidence and its impact on professional software workers. To remind ourselves, the research objectives which have guided this research are as follows:

- To provide a focused examination of the tasks and activities performed by software professionals
- To understand how software professionals respond to and experience the labour process
- To explore contextual and internal organisational factors which may have implications for experiences of work intensity
- To establish whether or not professional software workers experience work intensity

- To the extent that software professionals do experience work intensity, to examine their experiences and perceptions of work intensity and its extent, character, causes and consequences.

9.3 CONTRIBUTIONS

One of the main objectives of this thesis has been to explore whether software professionals are necessarily immune from experiences of work intensity, by virtue of their central position within the new economy and their status as archetypal knowledge workers. First and foremost, this PhD therefore makes an original contribution through identifying that professional software workers are subject to experiences of work intensity. Crucially, this study has provided unprecedented insight into the linkages, mechanisms and relationships which influence and shape software professionals' experiences of work intensity.

9.3.1 Debates on the New Economy

This thesis makes an original contribution through demonstrating that developments in ICTs, globalisation and the emphasis on flexibility have had implications for software professionals and their experiences of work intensity. Crucially, globalisation and developments in ICTs have influenced methods of communication, the physical proximity of project teams and decisions to outsource or offshore work activities within professional software work. The discovery within this PhD that technological mediums can shape software professionals' experiences of work intensity demonstrates that technological developments can have implications even for workers who create a key technology within the new economy. In addition, this study confirms that the ability to outsource or offshore software activities has enabled organisations to make cost savings, take advantage of attractive conditions available elsewhere and offer 'round the clock' services (Upadhya, 2009; OECD Information Technology Outlook, 2006; McManus and Floyd, 2005; May, 2002; Carnoy et al, 1993). However, the evidence clearly indicates that the geographical dispersal of project teams can contribute to experiences of work intensity for software professionals. Notably, close physical proximity can help reduce experiences of work intensity by facilitating collaboration, interaction and communication between project team members. In this sense, it is important for companies to consider the physical proximity of individuals when forming software project teams, due to the implications that this can have for software professionals' experiences of work intensity.

This PhD has discovered that volume of work, as opposed to the difficulty and complexity of work, is a key contributor to daily experiences of work intensity for software professionals. This finding is particularly significant, in that explorations into dimensions of work effort have been absent from existing studies on professional software workers. Workloads within professional software work have clearly been affected by trends viewed within the new economy, such as the emphasis on flexibility, multi-skilling and empowerment (Green, 2006, 2001; Thompson and McHugh, 2002; Tomaney, 1990). Furthermore, technological developments have intensified professional software work through speeding up cycle times, allowing activities to occur in parallel and increasing volume of work. The finding that greater levels of experience can increase the breadth of activities software professionals carry out and raise experiences of work intensity demonstrates the importance of considering occupational level when conducting research. Indeed, this is an aspect which is missing from current accounts. Notably, this study has revealed that more experienced software professionals may be best placed to manage experiences of work intensity, due to greater familiarity in dealing with situations and the ability to exert influence over task time estimates.

This study makes an original contribution through demonstrating that conventional perspectives which suggest that work has been up-skilled, downgraded or de-skilled within the knowledge economy (Baldry et al, 2007; Kumar, 2005; Alvesson, 2004; May, 2002; Thompson and Warhurst, 1998) do not capture the reality of what has happened within professional software work. Crucially, the evidence demonstrates that technological developments have instead influenced the type and range of activities that software professionals perform across the development life cycle. Moreover, this study has shown that technical expertise and skills continue to be necessary even for software professionals tasked with implementing and customising packaged software, due to difficulty in predicting how packaged software will interact and behave with systems (Quintas, 1994). In this sense, existing perspectives on the knowledge economy are problematic, in that they tend to deal with absolutes. Indeed, this PhD has revealed that a more complex reality exists within professional software work, typified by changes in work performed across the development life cycle and the compression of work time, rather than the up-skilling, de-skilling or downgrading of work.

9.3.2 Developments Within the Software Industry

Professional software work has been subject to continuous change since its early inception during World War Two as the necessary complement to hardware (Kraft, 1979). Notably, professional software work has evolved from its beginnings as an unstructured, unregulated 'art' (Baetjer, 1998; Quintas, 1994; Kraft, 1979) to the structured discipline of 'software engineering' through the adoption of scientific, mathematical and engineering principles (Kraft, 1979, 1977). Critically, this research draws attention to the current evolution in approaches to structuring and managing professional software work activities by demonstrating the move towards the 'Agile' methodology in the two case study companies. Market dynamics, competitive pressures and client demands appear to have generated support for the 'Agile' methodology within the software industry, through the recognition that an incremental approach can help manage the increased complexity of systems, ensure software meets requirements and allow software to be released into the market more quickly. Crucially, the discovery of 'Agile' within this research makes a significant contribution, in that previous accounts on professional software work have failed to identify this methodology and its implications for the structure, organisation and management of professional software work activities.

Notably, this study has illustrated that the 'Agile' approach affects the distribution and time span of activities and increases levels of client involvement, with mixed consequences for software professionals' experiences of work intensity. 'Agile' can clearly help software professionals' manage experiences of work intensity, through preventing the need for increased effort levels at the end of longer cycles to meet deadlines and improving understanding of client' requirements. However, the evidence equally demonstrates that the simultaneous overlapping of releases and life cycle stages under 'Agile' can increase work volume, compress work time and reduce levels of 'downtime' for software professionals, contributing to work intensity.

Moreover, this study has shown that clients can impact in diverse and important ways on the degree of work intensity experienced by software professionals. For instance, clients are a predominant source of work intensity for software professionals engaged in work for external clients or priority work for internal clients, through their indecision, changing of priorities, and regular requests for alterations to work already in progress. However, software professionals engaged in internal work which is deemed non-essential to the organisation may experience reduced levels of work intensity.

These contributions are particularly significant, in that consideration of the 'Agile' methodology and the impact of clients are missing from existing accounts on professional software work. Furthermore, the evolutionary nature of professional software work challenges the idea that knowledge work occupations – and indeed the current knowledge economy - are necessarily static structures. This therefore begs the question of what future trends may have in store for knowledge workers such as software professionals and their experiences of work intensity.

9.3.3 Characteristics of Professional Software Work

Conventional wisdom has dictated that the 'new' economy produces, rests and is reliant on knowledge workers who are empowered, creative, autonomous and place emphasis on theoretical knowledge, symbolic and analytical skills, problem-solving and intellectual judgement (Pyöriä, 2005; Alvesson, 2004, 2001; Newell et al, 2002; Thompson et al, 2001; Frenkel et al, 1995). However, these attributes, while generally true, are qualified when the tasks and activities that software professionals perform are subjected to scrutiny. This PhD has therefore been concerned with addressing the tendency of existing research to focus on the development life cycle as a means for explaining professional software work, rather than detailing the actual tasks and activities performed by individuals. Crucially, this study has made an original contribution by developing a taxonomy of professional software job roles to provide insight into the roles, tasks and activities performed by individuals. This taxonomy has also enabled dimensions such as variations in work roles, level of experience and levels of specialisation and generalisation to be examined within this research in terms of relationships to work intensity.

Moreover, the work diary study utilised within this research has made a considerable contribution to our understanding of the daily work patterns and activities performed by software professionals. Notably, the work diaries supplied information on the hours software professionals worked across the week, start and finish times, specific activities performed each day, informal discussion or meetings with colleagues, clients and managers, the use of technological mediums and time taken for breaks. It should also be noted that the work diary data enabled the development of an index of possible determinants of work intensity to be devised, allowing for the scrutiny of dimensions which have not been included in existing frameworks, particularly with regards to knowledge work occupations. Furthermore, the work diary study enabled the identification of previously unexplored aspects of the professional software work process, such as work volume, level of responsibility, work roles and aspects of specialisation and generalisation, which clearly had implications for experiences of work intensity.

9.3.4 The Software Labour Process

Critically, this study has discovered that the software labour process can have implications for software professionals' experiences of work intensity. Deadlines can be confirmed to be an integral part of the software labour process and form the main mechanism around which all activities are structured, organised, managed and controlled (Baldry et al, 2007; O'Riain, 2010, 2006; Andrews et al, 2005). Notably, this study makes an original contribution through revealing that deadlines have implications for software professionals' experiences of work intensity – an aspect which has been unacknowledged within existing accounts. Crucially, deadlines are a key contributor to experiences of work intensity for software professionals engaged in providing critical software to external or internal customers. Concurrently, deadlines may be less prevalent for software professionals engaged in non-critical or less client-centred work, due to the absence of competitive pressures.

The findings from this PhD indicate that software project teams tend to be supportive and facilitative in nature, challenging the perspective that these structures necessarily encourage peer pressure and increased surveillance (Green, 2006, 2004, 2001; Thompson and McHugh, 2002; Findlay et al, 2000; Springer, 1999; Baldry et al, 1998). In addition, the revelation that specialist knowledge and interdependencies contribute to work intensity for software professionals demonstrates the failure of existing studies (Marks and Lockyer, 2004; Tam et al, 2002; Kraft, 1977) to acknowledge the negative repercussions of heterogeneity within project team structures.

The findings from this research confirm that the collaborative, supportive and interactive nature of professional software work subjects individuals to regular, daily interruptions from project team members, project managers and clients (Boisard et al, 2008; O'Carroll, 2008; Voss-Dahm, 2005; Perlow, 1997; Kunda, 1992). Crucially, this study makes an original contribution by demonstrating that interruptions highly contribute to experiences of work intensity for software professionals. Indeed, lest it be thought that interruptions as a source of intensity appear as mundane or insignificant, this factor was reiterated and emphasised by virtually all respondents. In addition, the findings from this study challenge suggestions by Perlow (1997) that software professionals can manage the disruptive and unpredictable nature of interruptions by scheduling time blocks for 'quiet time' or organising interactions according to priority. Crucially, the evidence convincingly demonstrates that the collaborative, interactive, interdependent and unpredictable nature of professional software work can make it difficult for individuals to plan and schedule interactions. Moreover, particular individuals, such as Technical Leads, may be less able to control and prioritise their availability to project team members, due to the need to provide coaching and assistance.

Furthermore, the evidence from this study contradicts suggestions that software professionals may see performance metrics as placing pressure on them to perform or representing management attempts to exert control (Sharone, 2004; Beirne et al, 1998; Friedman and Cornford, 1989). Crucially, this study has shown that software professionals themselves see these methods as an important part of the work process, through helping to keep track of progress and aiding the evaluation of personal work content. Furthermore, the evidence indicates that performance metrics do not contribute to experiences of work intensity for software professionals. In this sense, work intensity is less the outcome of prescriptive performance metrics than pressures arising from the cyclical nature of professional software work, methodologies such as ‘Agile’, work volume and internal motivation to work.

The intellectual, intangible and complex nature of professional software work clearly requires an autonomous, flexible and facilitative approach to managing workers. However, this study has illustrated that normative control is an aspect of the software labour process which can contribute to experiences of work intensity. For example, individuals may be inclined to self-supervise, accept additional responsibilities, place greater investment in work and increase expenditure of work effort due to the intrinsically satisfying attributes of the work itself (Green, 2005, 2004; Thompson, 2003; Lockyer et al, 2001; Gallie et al, 1998; Deetz, 1995). In addition, this research makes an original contribution by illustrating that the approach software professionals adopt with regards to breaks can have implications for experiences of work intensity. Crucially, the evidence indicates that taking breaks during periods of concentration can increase work intensity through interrupting ‘train of thought’. Conversely, taking breaks can also help software professionals manage work intensity by aiding relaxation and helping to create a fresh outlook. Moreover, this PhD has shown that worker agency plays an important part in influencing software professionals’ experiences of work intensity. Notably, personal pro-activeness, greater levels of experience and influence and the ability to manage relationships with clients can help software professionals to offset experiences of work intensity.

9.3.5 The Importance of Context

Consideration of context within this PhD has enabled differences in outcomes to be examined in greater detail, provided a setting for interpreting and understanding individual experiences and helped towards generating a comprehensive understanding of work intensity. Market dynamics, firm characteristics and internal organisational factors have been identified within this PhD as having implications for software professionals' experiences of work intensity. Notably, market dynamics such as competition, de-regulation, privatisation and developments in ICTs appear to have influenced budgets and priorities within client organisations, affected the strategies adopted within organisations and have had clear implications for software professionals' work intensity. In addition, firm characteristics such as organisational type and company size have been found to shape software professionals' experiences of work intensity. Indeed, this study has discovered that firm characteristics can influence the emphasis that organisations may place on internal software expertise, the type of software engineering activities performed in-house, the work contractors are engaged in, the emphasis placed on deadlines and overall work pace. Interestingly, the discovery within this research that bureaucratic conditions continue to exist for software professionals and have implications for work intensity challenges the stereotype that all knowledge workers have necessarily experienced transformations in work organisation in the new economy.

Moreover, the evidence convincingly demonstrates that internal organisational factors such as leadership background and leadership style have implications for software professionals' experiences of work intensity. Leadership background, in terms of whether leaders have progressed through technical or managerial routes, can affect the importance placed on technical skills and the type of software engineering activities performed within organisations, with implications for work intensity. Moreover, facilitative, supportive and co-operative leadership styles can be confirmed to be the most appropriate methods for managing knowledge workers such as software professionals (Newell et al, 2002; Thompson and McHugh, 2002; Spender, 1998) and, in particular, help to manage experiences of work intensity. However, the pursuit of aspirations and skills and knowledge development appear to have negligible influence on software professionals' experiences of work intensity, due to individual responsibility for these areas.

In summary, this study has therefore addressed the main objectives set out at the beginning of this 'logic of discovery'. Firstly, software professionals *do* experience work intensity. Secondly, this study has demonstrated that developments in ICTs, globalisation, flexibility and developments in the industry have had implications for software professionals' experiences of work intensity. Thirdly, the development of the taxonomy and application of the work diary study have provided comprehensive insight into the tasks and activities performed by software professionals. Fourthly, aspects of the software labour process such as deadlines, project team structures, specialist knowledge, interruptions, normative control and worker agency have been identified as having implications for experiences of work intensity. Fifthly, contextual aspects, including market dynamics, firm characteristics and internal organisational factors have been identified as influencing software professionals' experiences of work intensity. Finally, this PhD has provided detailed insight into software professionals' experiences and perceptions of work intensity.

9.4 LIMITATIONS

This PhD has stressed the importance of identifying and explaining entities, processes and mechanisms, opposed to purely establishing causal linkages and rejecting unobservables, in order to develop a comprehensive understanding of work intensity (Sayer, 2008; Mingers, 2004; Benton and Craib, 2001; Ackroyd and Fleetwood, 2000; Archer, 1998; Bhaskar and Lawson, 1998). In this sense, an in-depth, qualitative, case study research design was deemed to be most suited to achieving the central aim and objectives of this PhD. Consequently, this research has focused on analytical generalisation, whereby the identification of entities, processes and mechanisms has allowed for the development of theoretical and conceptual understandings (Danermark et al, 2002; Eisenhardt, 2002; Orum et al, 1991). This PhD has therefore not attempted to quantitatively measure law-like relationships or establish universalities (Yin, 2003). Hence, it is not possible to draw statistical inferences from the findings or to statistically generalise from the particularities of the two case studies to the wider population of firms employing software professionals (Yin, 2003). In addition, it is important to acknowledge that concepts or constructs may not always be applicable if an objective reality is subject to change (Outwaithe, 1987). However, the author takes the stance that iteration and re-conceptualisation of existing understandings is an important and natural part of research, in order to convey the nature of reality more effectively and with greater reality (Modell, 2009; Sayer, 2008).

Moreover, this research may be criticised on the grounds that it has focused on capturing the experiences and perceptions of software professionals, rather than attempting to objectively measure work intensity. However, it has been argued that professional software work is mental, intellectual, tacit and complex in nature, making it difficult to objectively measure and evaluate work intensity. In addition, it has been stressed that whilst work intensity can be viewed as an objective concept, its outcomes may depend on the interpretations, perceptions and responses of software professionals themselves. Consequently, this research has attempted to alleviate concerns through adopting a systematic approach in examining professional software workers and their experiences of work intensity. Notably, this has involved establishing social norms based on people's perceptions of work effort and validating assessments through insight from peers, project managers and senior managers (Green, 2006). In addition, multiple research methods, such as observation, documentation, work diaries and semi-structured interviews, have been utilised within this PhD, in order to help form a 'chain of evidence' and increase the validity of findings (Yin, 2003). Moreover, the author has been explicit on research purpose and been systematic in documenting and recording information on research design, data collection methods and data analysis processes.

Observation was utilised as a research method within this study, in order to provide insight into the context surrounding the experiences of software professionals and a descriptive framework to understand data (Rossman and Rallis, 2003). However, the overt, holistic and explicit nature of observation within this study may have potentially influenced the behaviour of some participants (Flick, 2009; Bryman and Bell, 2003; Rossman and Rallis, 2003). In addition, participation in the work diary study may have required introspection and monitoring of daily work patterns at a level which software professionals may have been unaccustomed to, encouraging greater awareness of work intensity and possible reactance (Bolger et al, 2003; Wheeler and Reis, 1991). Participants may also have been prone to reporting activities or events which were deemed acceptable, omitting those regarded as unacceptable and under or over-reporting activities, depending on their relevance to individuals (Breakwell and Wood, 2000; Higgins et al, 1985). Consequently, the work diaries were designed to measure more than one activity and address factors in different ways.

This study has procured work diaries and interview scripts from Indian offshore contractors, in order to provide insight into the experiences of work intensity at a remote location and how these compared to the experiences of on-site workers. However, there may be limitations in generalising to the Indian case, based purely on the experiences of three Indian offshore workers. In addition, whilst accounts from SpecSoft and InSoft suggested that gender did not influence experiences of work intensity, it is difficult to fully arrive at this conclusion due to the small number of female to male participants within both companies. Finally, it should be noted that few individuals at InSoft were employed on part-time contracts, making it difficult to examine the experiences of part-time workers in greater depth.

9.5 FUTURE RESEARCH CONSIDERATIONS

This PhD has provided valuable insight into the implications that current market dynamics, globalisation and developments in technology have had for company strategies and software professionals' experiences of work intensity. However, it should be noted that this fieldwork was conducted between 2007 and 2009, prior to the current recession. Thus, it would be interesting to return to the two case study organisations at a later stage to explore the implications that the financial crisis and future trends may have for the nature of contracts, deadlines, the location of project team members, internal software engineering expertise, staffing arrangements and consequences for software professionals' experiences of work intensity.

This study was conducted between 2007 and 2009, during which time both companies were transferring from the ‘waterfall cycle’ to the ‘Agile’ methodology. The move towards ‘Agile’ clearly had implications for the distribution and time span of activities and levels of client involvement, with mixed consequences for work intensity. Consequently, it would be fascinating to examine the implications that future methodologies may have for professional software workers’ experiences of work intensity.

Constraints of time and the focus on depth over breadth prevented the application of in-depth investigation which was employed at SpecSoft and InSoft, to other parts of the software industry, such as computer games firms and multimedia firms. It would therefore be beneficial to conduct a similar examination into the experiences of software professionals working within computer games firms and multi-media firms. Furthermore, it would be pertinent to examine whether the current recession, budgetary constraints and future austerity measures have had implications for experiences of work intensity for software professionals working within public organisations.

Few software professionals within this study were employed on part-time contracts. Future research focused on part-time software professionals could therefore permit more detailed comparisons in experiences of work intensity to be drawn between part-time workers and those employed on full-time contracts. In addition, the evidence from this study suggests that contractors working within specialist software firms may experience lower levels of intensity compared to those carrying out software engineering activities which have been outsourced or offshored by in-house IT departments. Future research focusing purely on software professionals with contractor status would therefore be particularly valuable, through permitting further examination into the issues raised within this PhD.

This study has demonstrated that globalisation and developments in ICTs have provided firms with greater flexibility in the choice and location of labour and the ability to outsource or offshore software activities. The observation that India is an attractive choice for offshoring professional software work activities, coupled with the evidence that Indian contractors experienced higher levels of work intensity, suggests that future research into work intensity within the Indian software industry would be of significant value. Finally, attention to how offshoring relationships can be managed more effectively, particularly after contracts have been established, could potentially help address the difficulties that software professionals within InSoft faced as a result of the geographical dispersal of project teams.

REFERENCES

Ackroyd S (2004) 'Methodology for Management and Organisations Studies: Some Implications of Critical Realism', chpt 6. In Fleetwood S and Ackroyd S (Eds) *Critical Realist Applications in Organisation and Management Studies,* Routledge, London pp. 137-163.

Ackroyd S (2002) 'Methodology for Management and Organisation Studies: Some Implications of Critical Realism', Extract from Chpt 6. In Ackroyd S and Fleetwood S (Eds.) *Critical Realism in Action in Organisation and Management Studies*, Routledge, London.

Ackroyd S and Fleetwood S (2000) *Realist Perspectives in Management and Organisations*, Routledge, London.

Ackroyd S and Fleetwood S (2000) 'Realism in Contemporary Organisation and Management Studies', chpt 1. In Ackroyd S and Fleetwood S (Eds) *Realist Perspectives in Management and Organisations*, Routledge, London pp. 3-25.

Adam A (2002) 'Exploring the Gender Question in Critical Information Systems', *Journal of Information Technology*, Vol. 17, No. 2 pp. 59-67.

Alvesson M (2004) *Knowledge-Work and Knowledge-Intensive Firms*, Oxford University Press, Oxford.

Alvesson M (2001) 'Knowledge Work: Ambiguity, Image and Identity', *Human Relations*, Vol. 54, No. 7 pp. 863-886.

Alvesson M (2000) 'Social Identity and the Problem of Loyalty in Knowledge-Intensive Companies', *Journal of Management Studies*, December, Vol. 37, No. 8 pp. 1101-1123.

Alvesson M (1995) *Management of Knowledge-Intensive Companies*, de Gruyer, New York.

Alvesson M and Sköldberg K (2009) *Reflexive Methodology: New Vistas for Qualitative Research*, Sage Publications Ltd, London.

Amakawe U P, Hall J C and Schor S M (2000) 'Knowledge Related Skills and Effective Career Management', *International Journal of Manpower*, Vol. 21, No. 7 pp. 566-579.

Andrews C K, Lair C D and Landry B (2005) 'The Labour Process in Software Startups: Production on a Virtual Assembly Line?' chpt 3. In Barrett R (Ed) *Management, Labour Process and Software Development*, Routledge, Oxon pp. 45-75.

Aneesh A (2006) *Virtual Migration: The Programming of Globalisation,* Duke University Press, United States.

Appelbaum S H, Shapiro B T, Danaks H, Gualtieri , Li L, Loo D, Renaud P and Zampieri N (2004) 'Internal Communication Issues in an IT Engineering Department: A Case Study', *Corporate Communications*, Vol. 9, No. 1 pp. 6-24.

Archer M (1998) 'Introduction: Realism in the Social Sciences', chpt 7. In Archer M, Bhaskar R, Collier A, Lawson T and Norrie A (Eds) *Critical Realism: Essential Readings*, Routledge, London pp. 189-203.

Archer M, Bhaskar R, Collier A, Lawson T and Norrie A (1998) 'General Introduction'. In Archer M, Bhaskar R, Collier A, Lawson T and Norrie A (Eds) *Critical Realism: Essential Readings*, Routledge, London pp. ix-xxiv.

Arnold J (2005) *Managing Careers Into the 21st Century*, SAGE Publications, London.

Arora A, Arunachalam V S, Asundi J and Fernandes R (2001) 'The Indian Software Industry', *Research Policy*, Vol. 30, No. 8 pp. 1267-1287.

Arthur M B, Claman P H and DeFillippi R J (1995) 'Intelligent Enterprise, Intelligent Careers', *Academy of Management Executive*, Vol. 9, No. 4 pp. 7-22.

Athreye S S (2005) 'The Indian Software Industry', chpt 2. In Arora A and Gambardella A (2005) (Eds) *From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland and Israel*, Oxford University Press, Oxford pp. 7-40.

Baetjer Jr H (1998) 'Software as Capital: An Economic Perspective on Software Engineering', *IEEE Computer Society Press*, Los Alamitos, CA.

Baker C (1997) 'Membership Categorisation and Interview Accounts', chpt 9. In Silverman D (Ed) *Qualitative Research: Theory, Method and Practice*, Sage Publications, London pp.130-143.

Balasubramanyam V N and Balsasubramanyam A (2000) 'The Software Cluster in Bangalore', chpt 13. In Dunning J H (Ed) *Regions, Globalisation and the Knowledge-Based Economy*, Oxford University Press, Oxford.

Baldry C, Bain P and Taylor P (1998) 'Bright Satanic Offices: Intensification, Control and Team Taylorism', chpt 9. In Thompson P and Warhurst C (Eds) *Workplaces of the Future*, MacMillan Business, Hampshire pp. 163-183.

Baldry C, Bain P, Taylor P, Hyman J, Scholarios D, Marks A, Watson A, Gilbert K, Gall G and Bunzel D (2007) *The Meaning of Work in the New Economy*, Palgrave MacMillan, Hampshire.

Baldry C, Scholarios D and Hyman J (2005) 'Organisational Commitment Among Software Developers'. In Barrett R (Ed.) *Management, Labour Process and Software Development*, Routledge, Oxon pp. 168-195.

Barker J R (1993) ‘Tightening the Iron Cage: Concertive Control in Self-Managed Teams’, *Administrative Science Quarterly,* Sep, Vol. 38, No. 3 pp. 408-437.

Barley S R and Kunda G (2004) *Gurus, Hired Guns and Warm Bodies: Itinerant Experts in a Knowledge Economy*, Princeton University Press, Oxfordshire.

Barrett R (2005) (Ed) *Management, Labour Process and Software Development: Reality Bites*, Routledge, Oxon.

Barrett R (2004) ‘Working at Webboyz: An Analysis of Control Over the Software Development Labour Process’, *Sociology*, Vol. 38, No. 4 pp. 777-794.

Barrett R (2001) ‘Labouring Under an Illusion? The Labour Process of Software Development in the Australian Software Industry’, *New Technology, Work and Employment*, Vol. 16, No. 1 pp. 18-34.

Baruch Y (2004) ‘Transforming Careers: From Linear to Multidirectional Career Paths’, *Career Development Journal*, Vol. 9, No. 1 pp. 58-73.

Baxter L F (2000) ‘Bugged: The Software Development Process’, chpt 3. In Pritchard C, Hull R, Churner M and Willmott H (Eds) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire pp. 37-48.

Bechky B A (2003) ‘Sharing Meaning Across Occupational Communities: The Transformation of Understanding on a Production Floor’, *Organization Science,* May/June, Vol. 14, No.3 pp. 312-330.

Beirne M and Ramsay H (1988) ‘Computer Redesign and ‘Labour Process’ Theory: Towards a Critical Appraisal’, chpt 9. In Knights D and Willmott H (Eds) *New Technology and the Labour Process*, The MacMillan Press Ltd, Hampshire pp. 197-229.

Beirne M, Ramsay H and Panteli A (1998) ‘Developments in Computing Work: Control and Contradiction in the Software Labour Process’. In Thompson P and Warhurst C (Eds) *Workplaces of the Future,* MacMillan Business, Hampshire pp. 142-162.

Bell D (1974) *The Coming of Post-Industrial Society: A Venture in Social Forecasting*, Penguin, Harmondsworth.

Benner C (2002) *Work in the New Economy: Flexible Labour Markets in Silicon Valley*, Blackwell Publishers Ltd, Oxford.

Bennett A and Smith-Gavine S (1988) ‘The Percentage Utilisation of Labour Index (PUL)’. In Bosworth D (Ed) *Working Below Capacity*, MacMillan, London.

Benton T and Craib I (2001) *Philosophy of Social Science: The Philosophical Foundations of Social Thought*, Palgrave, Hampshire.

Bergmann J R and Meier C (2004) 'Electronic Process Data and Analysis', chpt 5.8. In Flick U, von Kardoff E and Steinke I (Eds) *A Companion to Qualitative Research*, Sage Publications, London pp. 243-247.

Bhaskar R (1989) *A Critical Introduction to Contemporary Philosophy*, Verso, London.

Bhaskar R and Lawson T (1998) 'Introduction: Basic Texts and Developments', chpt 1. In Archer M, Bhaskar R, Collier A, Lawson T and Norrie A (Eds) *Critical Realism: Essential Readings*, Routledge, London pp. 3-15.

Bielby D D and Bielby W T (1998) 'She Works Hard for the Money: Household Responsibilities and the Allocation of Work Effort', *American Journal of Sociology*, Vol. 93 pp. 1031-1059.

Birchall and Lyons (1995) *Creating Tomorrow's Organisation: Unlocking the Benefits of Future Work*, Pitman Publishing, London.

Bird A (1996) 'Careers as Repositories of Knowledge: Considerations for Boundaryless Careers', chpt 9. In Arthur B and Rousseau D M (Eds) *The Boundaryless Career: A New Employment Principle for a New Organisational Era*, Oxford University Press, Oxford pp. 150-168.

Bittman M, Brown J E and Wajcman J (2009) 'The Mobile Phone, Perpetual Contact and Time Pressure', *Work, Employment and Society*, Vol. 23, No. 4 pp. 673-691.

Blacker F (1995) 'Knowledge, Knowledge Work and Organisations: An Overview and Interpretation', *Organization Studies*, Vol. 16. pp. 1021-1046.

Boisard P, Cartron D, Gollac M, Valeyre A, Besançon J B (2008) *Time and Work: Work Intensity*, European Foundation for the Improvement of Work and Living Conditions, Office for Publications of the European Communities, Luxembourg.

Boland R J (1978) 'The Process and Product of System Design', *Management Science*, Col. 24 pp. 887-898.

Bolger N, Davis A and Rafaeli E (2003) 'Diary Methods: Capturing Life as it is Lived', *Annual Review of Psychology*, Vol. 54 pp. 579-616.

Bonke K (2005) 'Paid Work and Unpaid Work: Diary Information Versus Questionnaire Information', *Social Indicators Research*, Vol. 70 pp. 349-368.

Boreham P (1983) 'Indetermination and Professional Knowledge: Organisation and Control', *The Sociological Review*, Vol. 31, No. 4 pp. 693-718.

Boreham P, Parker R, Thompson P and Hall R (2008) *New Technology at Work*, Routledge, Oxon.

Braverman H (1974) *Labour and Monopoly Capital: The Degradation of Work on the Twentieth Century*, Monthly Review Press, London.

Breakwell G M and Wood P (2000) ‘Diary Techniques’, chpt 21. In Breakwell G M, Hammond S and Fife-Schaw C F (Eds) *Research Methods in Psychology*, Sage Publications, London pp. 294-302.

Brinkley I, Fauth R, Mahdon M and Theodoropoulou S (2009) *Knowledge Workers and Knowledge Work: A Knowledge Economy Programme Report*.

Brown P and Hesketh A (2004) *The Mismanagement of Talent: Employability and Jobs in the Knowledge Economy*, Oxford University Press, Oxford.

Bryman A (2008) *Social Research Methods*, Oxford University Press, Oxford.

Bryman A and Bell E (2003) *Business Research Methods*, Oxford University Press, Oxford.

Bunting M (2004) *Happy Slaves: How the Overwork Culture is Ruling Our Lives*, Harper Collins, London.

Burawoy M (1978) ‘Towards a Marxist Theory of the Labour Process: Braverman and Beyond’, *Politics and Society*, Vol. 8, No. 3-4 pp. 247-313.

Burchell B (2002) ‘The Prevalence and Redistribution of Job Insecurity and Work Intensification’, chpt 3. In Burchell B, Ladipo D and Wilkinson F (Eds.) *Job Insecurity and Work Intensification*, Routledge, London.

Burchell B and Fagan C (2004) ‘Gender and the Intensification of Work: Evidence From the European Working Conditions Surveys’, *Eastern Economic Journal*, Fall, Vol. 30, No. 4 pp. 627-642.

Burke R J (2007) *Research Companion to Working Time and Work Addiction*, Edward Elgar, Cheltenham.

Burke R J and Cooper C L (2008) *The Long Work Hours Culture: Causes, Consequences and Choices*, Emerald Group Publishing, Bingley.

Burke R J, Koyuncu M, Fiksenbaum L and Acar F T (2009) ‘Work Hours, Work Intensity, Satisfactions and Psychological Well-Being Among Turkish Manufacturing Managers’, *Europe’s Journal of Psychology*, Vol. 2 pp. 12-30.

Callinicos A (2001) *Against the Third Way: An Anti-Capitalist Critique*, Polity Press, Blackwell Publishers Ltd.

Caloghirou Y, Constantelou A and Vonortas N A (2006) ‘By Way of an Introduction: Knowledge Flows: The Drivers for the Creation of a Knowledge-Based Economy’, chpt 1. In Caloghirou Y, Constantelou A and Vonortas N A (Eds.) *Knowledge Flows in European Industry*, Routledge, Oxon pp. 1-24.

Cappel J J (2001) ‘Entry-Level IS Job Skills: A Survey of Employers’, *Journal of Computer Information Systems*, Winter, Vol. 42, No. 2 pp. 76-83.

Cappelli P (2000) 'Managing Without Commitment', *Organisational Dynamics*, Vol. 28, No. 4 pp. 11-24.

Cappelli P (1999) *The New Deal at Work: Managing the Market-Driven Workforce*, Harvard Business School Press, Boston.

Carmel E and Bird B J (1997) 'Small is Beautiful: A Study of Packaged Software Development Teams', *The Journal of High Technology Management Research*, Vol. 8, No. 1 pp. 129-148.

Carnoy M, Castells M, Cohen S S and Cardoso F H (1993) *The New Global Economy in the Information Age: Reflections on our Changing World*, Pennsylvania State University Press, University Park.

Careers Scotland
www.careers-scotland.org.uk/CareerInformation/Occupations/ComputingandICT

Castells M (2000) *The Rise of the Network Society*, Blackwell Publishers, Oxford.

Castells M (1989) *The Informational City: Information Technology, Economic Restructuring and the Urban-Regional Process*, Blackwell, Oxford.

Castree N, N M Coe, Ward K and Samers M (2004) *Spaces of Work: Global Capitalism and Geographies of Labour*, Sage Publications Ltd, London.

Causer G and Jones C (1996) 'Management and the Control of Technical Labour', *Work, Employment and Society*, Vol. 10, No. 1 pp. 105-123.

Cheney P H, Hale D P, Kasper G M (1990) 'Knowledge, Skills and Abilities of Information Systems Professionals: Past, Present and Future', *Information and Management*, Vol. 19, No. 4 pp. 237-247.

Cheney P H and Lyons N R (1980) 'Information Systems Skill Requirements: A Survey', *MIS Quarterly*, Vol. 4, No. 1 pp. 35-44.

Chumer M, Hull R and Pritchard C (2000) 'Introduction: Situating Discussions About Knowledge'. In Pritchard C, Hull R, Churner M and Willmott H (Eds) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire pp. xv-xxx.

Churner M and Willmott H (Eds) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire pp.176-198.

Cohen L and Mallon M (1999) 'The Transition from Managerial Career to Portfolio Career: Perceptions of Boundarylessness', *Work Employment and Society*, Vol. 13, No. 2 pp. 329-352.

Cohen S and Zysman J (1987) *Manufacturing Matters: The Myth of Post-Industrial Economy*, Basic Books, New York.

Collin W G (1978) *Introducing Computer Programming,* National Computing Centre, Manchester.

Computer Economics Limited and Remuneration Economics Survey, 2008.

Conway N and Briner R B (2002) ‘A Daily Diary Study of Affective Responses to Psychological Contract Breach and Exceeded Promises’, *Journal of Organisational Behaviour*, May, Vol. 23, No. 3 pp. 287-302.

Couger J D (1982) ‘Evolution of System Development Techniques’. In Couger J D, Colter M A and Knapp R W (Eds) *Advanced Systems Development/Feasibility Techniques*, Wiley, New York pp. 6-13.

Couger J D, Opperman E B and Amoroso D L (1992) ‘Motivating IS Managers in the 1990s’, *Inside DPMA*, May pp. 5-9.

Couger J D, Zawacki R A and Opperman E B (1979) ‘Motivation Levels of MIS Managers Versus Those of Their Employees’, *MIS Quarterly*, Vol. 3 pp. 47-49.

Cressey P and MacInnes J (1980) ‘Voting for Ford: Industrial Democracy and Control of Labour’, *Capital and Class*, Vol. 11 pp. 5-33.

Curtis B, Krasner H and Iscoe N (1988) ‘A Field Study of the Software Design Process for Large Systems’, *Communications of the ACM*, Vol. 31 pp. 1268-1287.

Danermark B, Ekström M, Jakobsen L and Karlsson J C (2002) *Explaining Society: Critical Realism in the Social Sciences*, Routledge, Oxon.

Deetz S (1995) *Transforming Communication, Transforming Business: Building Responsive and Responsible Workplaces*, Hampton Press, Cresskill, NJ.

DiDio L (1997) ‘Look out for Techno-Hazing’, *Computerworld*, Vol. 31, No. 39 pp. 72.

Donaldson J R (1973) ‘Structured Programming’, *Datamation*, Dec, Vol. 19, No 12 pp. 52-54.

Donnelly R (2006) ‘How ‘Free’ is the Free Worker? An Investigation into the Working Arrangements Available to Knowledge Workers’, *Personnel Review*, Vol. 35, No. 1 pp. 78-97.

Drucker P (1993) *Post-Capitalist Society*, Butterworth-Heinemann, Oxford.

Dyson E, Gilder G, Keyworth J and Toffler A (1996) ‘Cyberspace and the American Dream’, chpt 3. In Webster F (2004) *The Information Society Reader*, Routledge, London pp. 31-41.

Easterby-Smith M and Thorpe R (2002) *Management Research: An Introduction*, Sage Publications, London.

Easterby-Smith M, Thorpe R and Lowe A (2004) *Management Research: An Introduction*, Sage Publications, London.

Easton G (2000) 'Case Research as a Method for Industrial Networks: A Realist Apologia', chpt 10. In Ackroyd S and Fleetwood S (Eds) *Realist Perspective in Management and Organisations*, Routledge, London pp. 205-219.

Edwards R (1979) *Contested Terrain: The Transformation of the Workplace in the Twentieth Century*, Heinemann, London.

Edwards P K, Collinson M and Rees C (1998) 'The Determinants of Employee Responses to Total Quality Management: Six Case Studies', *Organization Studies*, Vol. 19, No. 3 pp. 449-475.

Edwards P K and Scullion H (1982) *The Social Organisation of Industrial Conflict: Control and Resistance in the Workplace*, Blackwell, Oxford.

Edwards P and Wacjman J (2005) *The Politics of Working Life*, Oxford University Press, Oxford.

Eisenhardt K M (2002) 'Building Theories From Case Study Research', chpt 1. In Huberman A M (Eds) *The Qualitative Researcher's Companion*, Sage Publications Ltd, London pp. 5-35.

Elger T (1991) 'Task Flexibility and Intensification of Labour in UK Manufacturing in the 1980s'. In Pollert A (Ed) *Farewell to Flexibility?*, Blackwell, Oxford.

Elger T (1990) 'Technical Innovation and Work Reorganisation in British Manufacturing in the 1980s: Continuity, Intensification or Transformation?', *Work, Employment and Society*, May, Vol. 4 (Special Issue) pp. 67-101.

Felstead A and Jewson N (1999) 'Flexible Labour and Non-Standard Employment: An Agenda of Issues'. In Felstead A and Jewson N (Eds) *Global Trends in Flexible Labour*, MacMillan, London pp. 1-20.

Felstead A, Jewson N and Walters S (2005) *Changing Places of Work*, Palgrave MacMillan, Basingstoke.

Fielding N G and Fielding J L (1986) *Linking Data: The Articulation of Qualitative and Quantitative Methods in Social Research*, Sage, California.

Fiksenbaum L, Jeng, Kouncu M and Burke R J (2010) 'Work Hours, Work Intensity, Satisfactions and Psychological Well-Being Among Hotel Managers in China', *Cross Cultural Management*, Vol. 17, No. 1 pp. 79-93.

Fincham R, Fleck J, Proctor R, Scarborough H, Tierney M and Williams R (1994) *Expertise and Innovation*, Oxford University Press Inc, New York.

Findlay P, Marks A, McKinlay A and Thompson P (2000) 'In Search of Perfect People: Teamwork and Team Players in Scottish Spirit Industry', *Human Relations*, Vol. 53, No. 12 pp. 1549-1574.

Flecker J and Hofbauer J (1998) 'Capitalising on Subjectivity: The 'New Model Worker' and the Importance of Being Useful', chpt 6. In Thompson P and Warhurst C (Eds) *Workplaces of the Future*, Palgrave MacMillan, Hampshire pp. 104-123.

Flick U (2009) *An Introduction to Qualitative Research*, Sage Publications Ltd, London.

Flood P C, Turner T, Ramamoorthy N and Pearson J (2001) 'Causes and Consequences of Psychological Contracts Among Knowledge Workers in the High Technology and Financial Services Industries', *International Journal of Human Resource Management*, Vol. 12, No. 7 pp. 1152-1165.

Foray D (2006) *The Economics of Knowledge*, MIT Press, Paris.

Freeman C and Perez C (1988) 'Structural Crises of Adjustment, Business Cycles and Investment Behaviour'. In Dosi G, Freeman C, Nelson R, Silverberg G and Soete L (Eds) *Technical Change and Economic Theory*, Pinter Publishers, London pp. 38-66.

Frenkel S, Korczynski M, Donoghue L and Shire K (1995) 'Re-Constituting Work: Trends Towards Knowledge Work and Info-Normative Control', *Work, Employment and Society*, Dec, Vol. 9, No. 4 pp. 773-796.

Friedman A L (1977) *Industry and Labour: Class Struggle at Work and Monopoly Capitalism*, MacMillan, London.

Friedman A L and Cornford D S (1989) *Computer Systems Development: History, Organisation and Implementation*, Wiley, Chichester.

Gallie D, White M, Cheng Y and Tomlinson M (1998) *Restructuring the Employment Relationship*, Oxford University Press, Oxford.

Gallivan M A (2002) 'The Influence of Software Developers' Creative Style on Their Attitudes to and Assimilation of a Software Process Innovation', *Information and Management*, Vol. 40 pp. 443-465.

Gerring J (2007) *Case Study Research: Principles and Practices*, Cambridge University Press, New York.

Gershuny J (2004) 'Costs and Benefits of Time Sampling Methodologies', *Social Indicators Research*, Vol. 67 pp. 247-252.

Gibbons M, Limoges C, Nowotny H, Schwarzman S, Scott P and Trow M (2002) *The New Production of Knowledge*, Sage, London.

Gillham G (2008) *Observation Techniques: Structured to Unstructured*, Continuum International Publishing Group, London.

Gillham B (2000) *Case Study Research Methods*, Continuum, London.

Giordano L (1988) 'Beyond Taylorism: Computerisation and QWL Programmes in the Production Process', chpt 8. In Knights D and Willmott H (Eds) *New Technology and the Labour Process*, MacMillan Press Ltd, Hampshire pp. 163-196.

Glaser E M (1976) *Productivity Gains Through Work-Life Improvements*, Harcourt Brace Jovanovich, London.

Golden A and Geisler C (2007) 'Work-Life Boundary Management and the Personal Digital Assistant', *Human Relations*, Vol. 60, No. 3 pp. 519-551.

Green F (2006) *Demanding Work: The Paradox of Job Quality in the Affluent Economy*, Princeton University Press, New Jersey.

Green F (2004) 'Work Intensification, Discretion and the Decline in Well-Being at Work', *Eastern Economic Journal*, Fall, Vol. 30, No. 4 pp. 615-625.

Green F (2001) 'It's Been a Hard Day's Night: The Concentration and Intensification of Work in Late Twentieth-Century Britain', *British Journal of Industrial Relations*, Vol. 39, No. 1 pp. 53-80.

Green F and McIntosh S (2001) 'The Intensification of Work in Europe', *Labour Economics*, Vol. 8 pp. 291-308.

Greenbaum J (1976) 'Division of Labour in the Computer Field', *Monthly Review*, Vol. 28, No. 3 pp. 40-55.

Grindley K (1986) *Fourth Generation Languages: Volume 1, A Survey of Best Practice*, IDPM Publications, London.

Haddon L (2004) *Information and Communication Technologies in Everyday Life: A Concise Introduction and Research Guide*, Berg, Oxford.

Hague D (1991) 'Knowledge Society, University Challenge', *Marxism Today*, Sept pp. 12-17.

Hakim C (2000) *Work-Lifestyle Choices in the 21st Century: Preference Theory*, Oxford University Press, Oxford.

Handy C (1994) *The Empty Raincoat: Making Sense of the Future*, Hutchinson, London.

Hansen M T, Nohria N and Tierney T (1999) 'What's Your Strategy for Managing Knowledge?', *Harvard Business Review*, March-April pp. 106-116.

Harrison D and Easton G (2004) 'Temporally Embedded Case Comparison in Industrial Marketing Research', chpt 8. In Fleetwood S and Ackroyd S (Eds) *Critical Realist Applications in Organisation and Management Studies*, Routledge, London pp.194-210.

Heckscher C (1995) *White-Collar Blues: Management Loyalties in an Age of Corporate Restructuring*, Basic Books, New York.

Henwood D (2003) *After the New Economy*, The New Press, London.

Hermanns H (2004) 'Interviewing as an Activity', chpt 5.3. In Flick E, von Kardoff E and Steinke I (Eds) *A Companion to Qualitative Research*, Sage Publications, London pp. 209-213.

Herriot P and Pemberton C (1996) 'Contacting Careers', *Human Relations*, Vol. 49, No. 6 pp. 757-790.

Herriot P and Pemberton C (1995) *Competitive Advantage Through Diversity*, Sage, London.

Higgins C A, McClean R J and Conrath D W (1985) 'The Accuracy and Biases of Diary Communication Data', *International Journal of Structural Analysis*, Vol. 7, No. 2 pp. 173-187.

Hinrichs J (1964) 'Communication Activity of Industrial Research Personnel', *Personnel Psychology*, Vol. 17 pp. 193-204.

Hobsons (2007) *Get 2007 Science and IT*, CRAC.

Holstein J A and Gubrium J F (1997) 'Active Interviewing', chpt 8. In Silverman D (Ed) *Qualitative Research: Theory, Method and Practice*, Sage Publications, London pp. 113-129.

Hopf C (2004) 'Qualitative Interviews: An Overview', chpt 5.2. In Flick E, von Kardoff E and Steinke I (Eds) *A Companion to Qualitative Research*, Sage Publications, London pp. 203-213.

Hornby S and Clarke Z (2002) *Challenge and Change in the Information Society*, Facet Publishing, London.

Hull R (2000) 'Knowledge Management and the Conduct of Expert Labour', chpt 4. In Pritchard C, Hull R, Churner M and Willmott H (Eds) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire pp. 49-68.

Huws U (2003) *The Making of a Cybertariat: Virtual Work in a Real World*, Monthly Review Press, London.

Huws U and Flecker J (2004) *Asian EMERGENCE: The World's Back Office?*, IES Report 409, Brighton.

Hyman J, Baldry C, Scholarios D and Bunzel D (2003) 'Work-Life Imbalance in Call Centres and Software Development', *British Journal of Industrial Relations*, June, Vol. 41, No. 2 pp. 215-239.

Hyman J, Lockyer C, Marks A and Scholarios D (2004) 'Needing a New Program: Why is Union Membership so Low Among Software Workers?'. In Brown W, Healy G, Heery E and Taylor P (Eds.) *The Future of Worker Representation* pp. 37-61.

Ichniowski C, Kochan T A, Levine D, Olson C and Strauss G (1996) 'What Works at Work: Overview and Assessment', *Industrial Relations*, July, Vol. 35, No. 3 pp. 299-333.

Intellect Software and IT Services Report (2009): The Year Ahead.

Ituma A and Simpson R (2006) 'The Chameleon Career: An Exploratory Study of the Work Biography of Information Technology Workers in Nigeria', *Career Development International*, Vol. 11, No. 1 pp. 48-65.

Jacques R (2000) 'Conclusion – Theorising Knowledge as Work: The Need for a 'Knowledge Theory of Value'. In Prichard C, Hull R, Churner M and Willmott H (Eds) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire pp.199-215.

Jobs 4U Careers Database
http://www.connexions-direct.com/jobs4u/index.cfm?pid=13

Jones C and DeFillippi R J (1996) 'Back to the Future in Film: Combining Industry and Self-Knowledge to Meet Career Challenges of the 21st Century', *Academy of Management Executive*, Vol. 10, No. 4 pp. 89-103.

Jones D (2000) 'Knowledge Workers 'R' Us: Academics, Practitioners and 'Specific Intellectuals', chpt 10. In Pritchard C, Hull R, Churner M and Willmott H (Eds) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire pp. 158-175.

Junkers B H (1960) *Fieldwork: An Introduction to the Social Sciences*, Cambridge University Press, Chicago.

Katz J and Aakhus M (Eds) *Perpetual Contact: Mobile Communication, Private Talk, Public Performance*, Cambridge University Press, Cambridge.

Kautz K and Thaysen K (2001) 'Knowledge, Learning and IT Support in a Small Software Company', *Journal of Knowledge Management*, Vol. 5, No. 4 pp. 349-357.

Kelley R (1990) *The Gold Collar Worker: Harnessing the Brain Power of the New Workforce*, Addison-Wesley, Reading.

Keynote Market Report (2008a) *Computer Hardware*.

Keynote Market Report (2008b) *Computer Software*.

Key Note Market Report (2008c) *Computer Services*.

Key Note Market Review (2004) *The Computer Market*.

Khan M B and Kukalis S (1990) 'MIS Professionals: Education and Performance', *Information and Management*, Vol. 19, No. 4 pp. 249-255.

King R M and Wilson G V (1992) 'Use of a Diary Technique to Investigate Psychosomatic Relations in Atopic Dermatitis, *Journal of Psychosomatic Research*, Vol. 35 pp. 697-706.

Klumb P L and Perrez M (2004) 'Why Time-Sampling Studies Can Enrich Work-Leisure Research', *Social Indicators Research*, Vol. 67 pp. 1-10.

Knell J (2000) *Most Wanted: The Quiet Birth of the Free Worker*, A Futures Report, available at www.indsoc.soc.uk, The Industrial Society, London.

Knights D, Murray F and Willmott H (1993) 'Networking as Knowledge Work: A Study of Inter-organisational Development in the Financial Services Sector', *Journal of Management Studies*, Vol. 30 pp. 975-996.

Knights D and Willmott H (1989) 'Power and Subjectivity at Work: From Degradation to Subjugation in Social Relations', *Sociology*, Vol. 23, No. 4 pp. 535-558.

Kochen M (1984) 'A New Concept of Information Society'. In El-Hadidy B and Horne E E (Eds) *The Infrastructure of an Informational Society*, Elsevier, Amsterdam pp. 25-38.

Kofman F and Senge P (1993) 'Communities and Commitment: The Heart of Learning Organisations, *Organisational Dynamics*, Vol. 22, No. 2 pp. 5-22.

Kraft P and Dubnoff S (1986) 'Job Content, Fragmentation and Control in Computer Software Work', *Industrial Relations*, Spring , Vol. 25, No. 2 pp. 184-196.

Kraft P (1979) 'The Industrialisation of Computer Programming: From Programming to Software Production'. In Zimbalist A (Ed.) *Case Studies on the Labour Process*, Monthly Review Press, New York pp. 1-17.

Kraft P (1977) *Programmers and Managers: The Routinisation of Computer Programming in the United States*, Springer.

Kumar K (2005) *From Post-Industrial to Post-Modern Society*, Blackwell Publishing Ltd, Oxford.

Kunda G (1992) *Engineering Culture: Control and Commitment in a High-Tech Corporation*, Temple University Press, Philadelphia.

Lachman R and Aranya N (1986) ‘Job Satisfaction and Turnover Intentions Amongst Professionals in Different Work Settings’, *Organization Studies*, Vol. 7 pp. 279-293.

Lash S and Urry J (1993) *Economies of Signs and Space*, Sage, London.

Lavington S (1980) *Early British Computers*, Manchester University Press, Manchester.

Lavoire D, Baetjer H and Tulloh W (1993) *Component Software: A Market Perspective on the Coming Revolution in Software Development*, Patricia Seybold Group, Boston.

Lavoire D, Baetjer H and Tulloh W (1991) ‘Coping With Complexity: OOPS and the Economist’s Critique of Central Planning’, *Hotline on Object-Oriented Technology*, Nov, Vol. 3 pp. 6-8.

Leal R P and Powers T L (1997) ‘A Taxonomy of Countries Based on Inventive Activity’, *International Marketing Review*, Vol. 14, No. 6 pp. 445-460.

Lee P C B (2002) ‘Career Goals and Career Management Strategy Among Information Technology Professionals’, *Career Development International*, Vol. 7, No. 2 pp. 6-13.

Lerouge C, Newton S and Blanton J E (2005) ‘Exploring the Systems Analyst Skill Set: Perceptions, Preferences, Age and Gender’, *The Journal of Computer Information Systems*, Vol. 45, No. 3 pp. 12-23.

Ling R (2004) *The Mobile Connection: The Cell Phone’s Impact on Society*, Elsevier, Amsterdam.

Littler C R and Salaman G (1982) ‘Braverman and Beyond: Recent Theories of the Labour Process’, *Sociology*, Nov, Vol. 16 pp. 251-269.

Lockyer C, Marks A and Mulvey G (2001) ‘Programming Teamwork? The Impact of Location on the Identities of Software Workers in the Scottish Software Industry’, *Paper Presented to the 5th International Workshop on Teamworking (IWOT5)*, Leuven (Belgium), 10-11 September 2001.

Löwendahl B (1997) *Strategic Management in Professional Service Firms*, Copenhagen Business School Press, Copenhagen.

Lyon D (1996) *The Information Society: Issues and Illusions*, Polity Press, Cambridge.

Mariño R, Minichiello V and Browne J (1999) ‘Reporting of Events Using Diaries’. In Minichiello V, Sullivan G, Greenwood K and Axford R (Eds) *Handbook for Research in Health Sciences*, Addison-Wesley, Sydney.

Marks A, Baldry C and Lockyer C (2003) 'The Knowledge Worker and Social Mobility: An Analysis of the Perceptions of Class by Software Professionals', *Paper Presented at Labour Process Conference*, 1-17 April.

Marks A and Lockyer C (2005) 'Debugging the System: The Impact of Dispersion on the Identity of Software Team Members', *International Journal of Human Resource Management*, Feb, Vol. 16, No. 2 pp. 219-237.

Marks A and Lockyer C (2004) 'Producing Knowledge: The Use of the Project Teams as a Vehicle for Knowledge and Skill Acquisition for Software Employees', *Economic and Industrial Democracy*, May, Vol. 25, No. 2 pp. 219-245.

Marks A, Lockyer C and Mulvey G (2001) 'In Need of a New Language? Issues of Identity in Software Development Teams', *Paper Presented at 19th Annual Labour Process Conference*, Royal Holloway University, March.

Marks A, Scholarios D and Lockyer C (2002) 'Identifying a Profession: The Creation of Professional Identities Within Software Work', *Presented at the 18th Egos Colloquium*, Barcelona, Spain, July 4-6.

Maxwell J A (2002) 'Understanding and Validity in Qualitative Research', chpt 2. In Huberman A M (Ed) *The Qualitative Researcher's Companion*, Sage Publications Inc, London pp. 37-64.

May C (2002) *The Information Society: A Sceptical View*, Blackwell Publishers Inc, Malden.

McGrath-Champ S (2005) 'Globalisation's Challenge to Labour: Re-Scaling Work and Employment', *Economic and Industrial Democracy*, Vol. 26 pp. 323-334.

McInerney C and LeFevre D (2000) 'Knowledge Managers: History and Challenges', chpt 1. In Pritchard C, Hull R, Churner M and Willmott H (Eds) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire pp. 1-19.

McKinlay A (2000) 'The Bearable Lightness of Control: Organisational Reflexivity and the Politics of Knowledge Management, chpt 7. In Pritchard C, Hull R, Churner M and Willmott H (Eds) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire pp. 107-121.

McManus J and Floyd D (2005) 'The Global Software Industry', *Management Services*, Summer, Vol. 49, No. 2 pp. 26-31.

McManus J, Li M and Moitra D (2007) *China and India: Opportunities and Threats for the Global Software Industry*, Chandos Publishing, Oxford.

McNicoll I, Kelly U, Marsh R and McLay D (2002) *Defining and Identifying the Knowledge Economy in Scotland: A Regional Perspective on a Global Phenomena*, University of Strathclyde Publishing, University of Strathclyde.

Miles M B and Huberman A M (1994) *Qualitative Data Analysis: An Expanded Sourcebook*, Sage Publications, Thousand Oaks.

Miller J and Glassner B (1997) 'The 'Inside' and the 'Outside': Finding Realities in Interviews', chpt 7. In Silverman D (Ed) *Qualitative Research: Theory, Method and Practice*, Sage Publications, London pp.99-112.

Mingers J (2008) 'Management Knowledge and Knowledge Management: Realism and Forms of Truth', *Knowledge Management Research and Practice*, Vol. 6 pp. 62-76.

Mingers J (2004) 'Future Directions in Management Science Modelling: Critical Realism and Multimethodology', chpt 7. In Fleetwood S and Ackroyd S (Eds) *Critical Realist Applications in Organisation and Management Studies*, Routledge, London pp. 164-193.

Modell S (2009) 'In Defence of Triangulation: A Critical Realist Approach to Mixed Methods Research in Management Accounting', *Management Accounting Research*, Vol. 20 pp. 208-221.

Mokyr J (2002) *The Gifts of Athena: Historical Origins of the Knowledge Economy*, Princeton University Press, Oxford.

Naur P and Randall B (1969) *Software Engineering*, Report of the NATO Science Committee Conference, Garmisch, October 1968, NATO, Brussels.

Negroponte N (1995) *Being Digital*, Coronet/Hodder and Stoughton, London.
Newell S, Robertson M, Scarborough H and Swan J (2002) *Managing Knowledge Work*, Palgrave, Hampshire.

Nonaka I (1994) 'A Dynamic Theory of Organisational Knowledge Creation', *Organisational Science*, Feb, Vol. 5, No. 1 pp. 14-37.

Nonaka I and Takeuchi H (1998) 'A Theory of the Firm's Knowledge-Creation Dynamics'. In Chandler A D, Hagström P and Sölvell O (Eds) *The Dynamic Firm: The Role of Technology, Strategy, Organisation and Regions*, Oxford University Press, Oxford pp. 214-241.

O'Carroll A (2008) 'Fuzzy Holes and Intangible Time: Time in a Knowledge Industry', *Time Society*, Vol. 17, No. 2/3 pp. 179-193.

OECD Information Technology Outlook 2006.

O'Riain S (2010) 'The Missing Customer and the Ever-Present Market: Software Developers and the Service Economy', *Work and Occupations*, Vol. 37, No. 3 pp. 320-348.

O'Riain S (2006) 'Time-Space Intensification: Karl Polanyi, the Double Movement and Global Informational Capitalism', *Theory and Society*, Vol. 35, No. 5/6 pp. 507-523.

O'Riain S (2000) 'Net-Working for a Living: Irish Software Developers in the Global Workplace', chpt 6. In Burawoy M, Blum J A, George S, Gillie Z, Gowan T, Haney L, Klawiter M, Lopez S H, O'Riain S and Thayer M (Eds) *Global Ethnography: Forces, Connections and Imaginations in a Postmodern World*, University of California Press, London pp. 175-202.

Orum A M, Feagin J R and Sjoberg G (1991) 'Introduction: The Nature of the Case Study'. In Feagin J R, Orum A M and Sjoberg G (Eds) *A Case For The Case Study*, University of North Carolina Press, London pp. 1-26.

Osnowitz P (2010) *Freelancing Expertise: Contract Professionals in the New Economy*, Cornell University Press, USA

Outwaithe W (1987) *New Philosophies of Social Science: Realism, Hermeneutics and Critical Theory*, MacMillan Education, London.

Penn R, Rose M and Rubery J (Eds) (1994) *Skill and Occupational Change*, Oxford University Press, Oxford.

Perlow L (1997) *Finding Time: How Corporations, Individuals and Families Can Benefit From New Work Practices*, Cornell University Press, New York.

Piore M J (1986) 'Perspectives on Labour Market Flexibility', *Industrial Relations*, Vol. 25, No. 2 pp. 146-166.

Plan IT Plus – Career Zone
http://www.planitplus.net/careerzone/jobprofiles/default.aspx?letter=C

Polanyi M (1967) *The Tacit Dimension*, Routledge and Kegan Paul, London.

Poster M (1990) *The Mode of Information: Poststructuralism and Social Context*, Polity, Cambridge.

Prahalad C and Hamel G (1990) 'The Core Competence of the Corporation', *Harvard Business Review*, May/June pp. 79-91.

Pritchard C (2000) 'Know, Learn and Share: The Knowledge Phenomena and the Construction of a Consumptive-Communicative Body', chpt 11. In Pritchard C, Hull R, Churner M and Willmott H (Eds) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire pp.176-198.

Pritchard C, Hull R, Churner M and Willmott H (2000) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire.

Prospects AGCAS Occupational Profile
http://www.prospects.ac.uk/p/types_of_job/information_technology.jsp

Pyöriä P (2005) 'The Concept of Knowledge Work Revisited', *Journal of Knowledge Management*, Vol. 9, No. 3 pp. 116-127.

Quinn J B (1992) *Intelligent Enterprise: A Knowledge and Service Based Paradigm for Industry*, Free Press, New York.

Quintas P (1994) 'Programmed Innovation? Trajectories of Change in Software Development', *Information Technology and People,* Vol. 7, No. 1 pp. 25-47.

Ragharen S R and Chand D R (1989) 'Diffusing Software Engineering Methods', *IEEE Software*, July pp. 81-90.

Rasmussen B and Johansen B (2005) 'Trick or Treat? Autonomy as Control in Knowledge Work', chpt 5. In Barrett R (Ed.) *Management, Labour Process and Software Development: Reality Bytes*, Routledge, Oxon pp. 100-122.

Reed M I (2000) 'In Praise of Duality and Dualism: Rethinking Agency and Structure in Organisational Analysis', chpt 3. In Ackroyd S and Fleetwood S (Eds) *Realist Perspectives in Management and Organisations,* Routledge, London pp. 45-65.

Reed M (1992) *The Sociology of Organisations: Themes, Perspectives and Prospects*, Harvester Wheatsheaf, New York.

Reich R (1991) *The Work of Nations: Preparing Ourselves for 21st Century Capitalism*, Simon and Schuster, London.

Robertson M and Swan J (2004) 'Going Public: The Emergence and Effects of Soft Bureaucracy Within a Knowledge-Intensive Firm', *Organization*, Vol. 11, No. 1 pp. 123-148.

Rossman G B and Rallis S F (2003) *Learning in the Field: An Introduction to Qualitative Research*, Sage Publications, London.

Rousseau D M (1995) Psychological Contracts in Organisations, Sage, California.

Rowe and Thomson (1996) see Black A (2002) 'The Information Society: A Secular View', chpt 2. In Hornby S and Clarke (Eds) *Challenge and Change in the Information Society*, Facet Publishing, London pp. 19-41.

Rowlinson M and Hassard J (1994) 'Economics, Politics and Labour Process Theory', *Capital and Class*, Vol. 53 pp. 65-97.

Rueschemeyer D (1986) *Power and the Division of Labour*, Stanford University Press, Stanford.

Sathaye S (2008) 'The Scientific Imperative to be Positive: Self-Reliance and Success in the Modern Workplace', chpt 5. In Upadhya C and Vasavi A R (Eds) *In an Outpost of the Global Economy: Work and Workers in India's Information Technology Industry*, Routledge, New Delhi pp. 136 - 161.

Sayer A (2008) *Realism and Social Science*, Sage Publications Ltd, London.

Sayer A (2004) 'Why Critical Realism', Foreword. In Ackroyd S and Fleetwood S (Eds) *Critical Realist Applications in Organisation and Management Studies*, Routledge, London pp. 6-20.

Scarborough H (1999) 'Knowledge as Work: Conflicts in the Management of Knowledge Workers', *Technology Analysis and Management*, March, Vol. 11, No. 1 pp. 5-16.

Schofield J W (2002) 'Increasing the Generalisability of Qualitative Research', chpt 8. In Huberman A M (Eds) *The Qualitative Researcher's Companion*, Sage Publications Ltd, London pp. 171-203.

Scotland IS (2008) *Innovation and ICT Skills in Scotland*, July.

Scotland: Towards the Economy: The Report of the Knowledge Economy Taskforce, 12 April 1999.

Scottish Technology Industry Survey 2009.

Sennett R (1998) *The Corrosion of Character*, Norton, New York.

Sewell G (2005) 'Nice Work? Rethinking Managerial Control in an Era of Knowledge Work', *Organization*, Vol. 12, No. 5 pp. 685-704.

Sharone E (2004) 'Engineering Overwork: Bell-Curve Management at a High-Tech Firm', chpt 7. In Epstein C F and Kalleberg A L (Eds) *Fighting For Time: Shifting Boundaries of Work and Social Life*, Russell Sage Foundations, New York pp. 191-218.

Sharpe R (1998) 'Globalisation: The Next Tactic in the 50-Year Struggle of Labour and Capital in Software Production', *Proceedings of the Work Difference and Social Change Conference*, Binghamton University, 8th-10th May pp. 364-379.

Sichel D E (1997) *The Computer Revolution*, Brookings Institution Press, Washington DC.

Silverman D (2004) 'Introducing Qualitative Research', chpt 1. In Silverman D (Ed) *Qualitative Research: Theory, Method and Practice*, Sage, London pp. 1-8.

Silverman D (2000) *Doing Qualitative Research: A Practical Handbook*, Sage, London.

Smith C (2006) 'The Double Indeterminacy of Labour Power: Labour Effort and Labour Mobility', *Work, Employment and Society*, Vol. 20, No. 2 pp. 389-402.

Sonnetag S (1995) 'Excellent Software Professionals: Experience, Work Activities and Perception by Peers', *Behaviour and Information Technology*, Vol. 14, No. 5 pp. 289-299.

Spencer D A (2000) 'Braverman and the Contribution of Labour Process Analysis to the Critique of Capitalist Production – Twenty-Five Years On', *Work, Employment and Society*, Vol. 14, No. 2 pp. 223-243.

Spender J C (1998) 'Pluralist Epistemology and the Knowledge-Based Theory of the Firm', *Organization*, Vol. 5 pp. 233-256.

Spender J C and Grant R M (1996) 'Knowledge and the Firm: Overview', *Strategic Management Journal*, Winter, Vol. 17, Issue 5-9 pp. 5-24.

Springer R (1999) 'The End of Production Concepts? Rationalisation and Labour Policy in the German Auto Industry', *Economic and Industrial Democracy*, Vol. 20 pp. 117-145.

Stanworth C (1998) 'Telework and the Information Age', *New Technology, Work and Employment*, Vol. 13, No. 1 pp. 51-62.

Starbuck (1992) 'Learning by Knowledge-Intensive Firms', *Journal of Management Studies*, Vol. 29, No. 6 pp. 713-740.

Stehr N (2004) 'The Economic Structures of Knowledge Economies', chpt 15. In Webster F (2004) *The Information Society Reader*, Routledge, London pp. 212-236.

Storey J (1985) 'The Means of Management Control', *Sociology*, Vol. 19, No. 2 pp. 193-211.

Swart J, Kinnie N (2003) 'Sharing Knowledge in Knowledge-Intensive Firms', *Human Resource Management Journal*, Vol. 13, No. 2 pp. 60-75.

Tam Y M, Korczynski M and Frenkel S J (2002) 'Organisational and Occupational Commitment: Knowledge Workers in Large Corporations', *Journal of Management Studies*, Vol. 39, No. 6 pp. 775-801.

Tampoe M (1993) Motivating Knowledge Workers: The Challenge for the 1990s, *Long Range Planning*, Vol. 26, No. 3 pp. 49-55.

Target IT (2007)
http://targetjobs.co.uk/career-sectors/it-and-telecoms

Taylor P and Bain P (2007) 'Reflections on the Call Centre: A Reply to Glucksmann', *Work, Employment and Society*, Vol. 21, No. 2 pp. 349-362.

Thompson P (2003) 'Disconnected Capitalism: Or Why Employers Can't Keep Their Side of the Bargain', *Work, Employment and Society*, Vol. 17, No. 2 pp. 359-378.

Thompson P (1989) *The Nature of Work: An Introduction to Debates on the Labour Process*, The MacMillan Press, London.

Thompson P and McHugh D (2002) *Work Organisations*, Palgrave, Hampshire.

Thompson P and Newsome K (2004) ‘Labour Process Theory, Work and the Employment Relation’, chpt 5. In Kaufman B E (Ed) *Theoretical Perspectives on Work and Employment*, Cornell University Press, Madison pp. 133-162.

Thompson P and Smith C (2000) ‘Follow the Yellow Brick Road: Reflections on Pathways in and out of the Labour Process Debate’, *International Studies of Management and Organisations*, Winter, Vol. 30, No. 4 pp. 40-67.

Thompson P, Warhurst C and Callaghan G (2001) ‘Ignorant Theory and Knowledgeable Workers: Interrogating the Connections Between Knowledge, Skills and Services, *Journal of Management Studies*, Nov, Vol. 38, No. 7 pp. 923-942.

Tomaney J (1990) ‘The Reality of Workplace Flexibility’, *Capital and Class*, Spring, Vol. 40 pp. 29-60.

Totterdell P, Wood S and Wall T (2006) ‘An Intra-Individual Test of the Demands-Control Model: A Weekly Diary Study of Psychological Strain in Portfolio Workers’, *Journal of Occupational and Organisational Psychology*, March, Vol. 79 pp. 63-84.

Trauth E M, Farwell D W and Lee D (1993) ‘The IS Expectation Gap: Industry Expectations Versus Academic Preparation’, *MIS Quarterly*, Sept, Vol. 17, No. 3 pp. 293-307.

Tschang T and Xue (2005) ‘The Chinese Software Industry’, chpt 6. In Arora A and Gambardella A (2005) (Eds) *From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland and Israel, Oxford University Press, Oxford* pp. 131-167.

Tsoukas H (2005) *Complex Knowledge: Studies in Organisational Epistemology*, Oxford University Press, Oxford.

Tsoukas H (1996) ‘The Firm as a Distributed Knowledge System: A Constructionist Approach’, *Strategic Management Journal*, Winter, Vol. 17 pp. 11-25.

Tsoukas H (1989) ‘The Validity of Idiographic Research Explanations’, *Academy of Management: The Academy of Management Review*, Oct, Vol. 14, No. 4 pp. 551-561.

Upadhya C (2008) ‘Management of Culture and Management Through Culture in the Indian Software Outsourcing Industry’, chpt 4. In Upadhya C and Vasavi A R (Eds) *In an Outpost of the Global Economy: Work and Workers in India’s Information Technology Industry*, Routledge, New Delhi pp. 101-135.

Verbrugge L M (1980) ‘Health Diaries’, *Medical Care*, Vol. 18 pp. 73-95.

Voelcker J (1988) ‘Automating Software: Proceed With Caution’, *IEEE Spectrum*, July, Vol. 25, No. 7 pp. 25-27.

Voss-Dahm (2005) 'Coming and Going at Will? Working Time Organisation in German IT Companies', chpt 6. In Barrett (Ed.) *Management, Labour Process and Software Development: Reality Bytes*, Routledge, Oxon pp. 123-145.

Waite B M, Claffey R C and Hillbrand M (1998) 'Differences Between Volunteers and Non-Volunteers in a High Demand Self-Recording Study', *Psychological Reports*, Vol. 83 pp. 199-210.

Wallace J (1995) 'Organisational and Professional Commitment in Professional and Nonprofessional Organisations', *Administrative Science Quarterly*, Vol. 40 pp. 228-255.

Walz D B, Elam J J and Curtis B (1993) 'The Dual Role of Conflict in Group Software Requirements and Design Activities', *Communication of the ACM*, Vol. 36, No. 10 pp. 63-76.

Warhurst C and Thompson P (1998) 'Hands, Hearts and Minds: Changing Work and Workers at the End of the Century', chpt 1. In Thompson P and Warhurst C (Eds) *Workplaces of the Future*, Palgrave MacMillan, Hampshire pgs. 1-24.

Wheeler L and Nezlek J (1977) 'Sex Differences in Social Participation', *Journal of Personality and Social Psychology*, Vol. 35 pp. 742-754.

Wheeler L and Reis H T (1991) 'Self-Recording of Everyday Life Events: Origins, Types and Uses', *Journal of Personality*, Sept, Vol. 59, No. 3 pp. 339-354.

Wichert I (2002) 'Job Insecurity and Work Intensification: The Effects on Health and Well-Being', chpt 5. In Burchell B, Ladipo D and Wilkinson F (Eds.) *Job Insecurity and Work Intensification*, Routledge, London pp. 92-111.

Willmott H (2000) 'From Knowledge to Learning'. In Prichard C, Hull R, Churner M and Willmott H (Eds) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire pp.216-227.

Wilson F (1988) 'Computer Numerical Control and Constraints', chpt 4. In Knights D and Willmott H (Eds) *New Technology and the Labour Process*, MacMillan Press Ltd, Hampshire.

Wiseman V, Conteh L and Matovu F (2005) 'Using Diaries to Collect Data in Resource-Poor Settings: Questions on Design and Implementation', *Health Policy Unit* pp. 394-405.

Woodfield R (2000) *Women, Work and Computing*, Cambridge University Press, Cambridge.

Yakhlef A and Salzer-Mörling M (2000) 'Intellectual Capital: Managing by Numbers', chpt 2. In Pritchard C, Hull R, Churner M and Willmott H (Eds) *Managing Knowledge: Critical Investigations of Work and Learning*, MacMillan Press Ltd, Hampshire pp. 20-36.

Yardley D (2004) *The Times: Careers and Jobs in IT*, Kogan Page Limited, London.

Yin R K (2003) *Case Study Research: Design and Methods*, Sage Publications, Inc, London.

Zimbalist A (1979) *Case Studies on the Labour Process*, Monthly Review Press, New York.

APPENDICES

APPENDIX 1: SPECSOFT PARTICIPANT INFORMATION SHEET

Department of Human Resource Management
Graham Hills Building
University of Strathclyde
50 Richmond Street
Glasgow
G1 1XU
Tel. XXXX XXX XXXX
e-mail: XXXX

Dear XXXX employee,

INFORMATION SHEET

I am a second year PhD student at the University of Strathclyde conducting research into professional software work and will be carrying out case study research at XXXX in the Glasgow office. I am studying the nature of professional software work in terms of what jobs actually involve, levels of intensity associated with this type of work and factors contributing to intensity. This involves consideration of how work is designed, organised and managed, how individuals experience their work, pressures exerted and work-life balance issues. My research will also consider how employee experiences of work and pressure vary according to contextual variables such as organisational size, product market, opportunities for skills and knowledge development and leadership style. Another element to be studied is whether individual work experiences and perceptions vary according to software job roles (for example, whether an individual is an analyst, designer, developer or consultant).

Some research has been conducted within academia on professional software work in areas such as teamwork and identity. However, there is still a major research gap in terms of understanding what professional software workers actually do, software job roles and how contextual variables affect individual experience of work. My research aims to make a valuable theoretical and practical contribution through addressing these research gaps and also, providing an understanding of trends in the software industry in general.

I will be carrying out case study research at XXXX for between four and six months. During this time, I will be attached to different project teams within XXXX. This will enable me to generate an understanding of how the work context at XXXX, in terms of products produced, company size, development opportunities, leadership style, affects individual experiences of work.

My focal research group, as discussed earlier, is professional software workers, that is, those workers in XXXX engaged in the specification, design, development and testing of software. I will also require to engage project managers, human resources and directors, to further understand how professional software workers' experiences of work are affected by organisational and leadership approach.

The research methods I will be utilising are:

- general observation of work activities – to provide an understanding of what professional software work actually involves

- unstructured informal discussions – to discuss general areas relating to software work and uncover issues of interest

- semi-structured interviews – these will be structured yet exploratory to further uncover pertinent issues and also, to investigate associations between variables, in order to increase my understanding of work experiences according to contextual factors

- work diaries – participants will be asked to complete work diaries for one week, in order to increase understanding of day-to-day software work. This will be a simple process involving placing a code (which relates to an activity in a compiled activity list) in a time slot. This may also help participants to view how their working day is actually spent, identify potential pressures and how to manage these more effectively

- documentation analysis – studying archival information and documentation on terms and conditions, policies and practices to provide insight into organisational approach

Your participation in these research methods would be greatly appreciated. Individual participation in these research methods is, however, entirely voluntary. Participants are under no obligation to respond to all aspects of the research procedures. I am also sensitive to the fact that my research should not impact on the performance of day-to-day business activities at XXXX.

The information generated from my time at XXXX will be used for the purpose of my PhD and my thesis. Finally, you can be assured that all data gathering and writing up of my findings involves firm guarantees to confidentiality and anonymity.

If you have any questions at any point, please direct these to myself at the contact details given on this letter and I will be happy to help.

Yours faithfully,

Saira Reid

APPENDIX 2: INSOFT PARTICIPANT INFORMATION SHEET

Department of Human Resource Management
Graham Hills Building
University of Strathclyde
50 Richmond Street
Glasgow
G1 1XU
Tel. XXXX XXX XXXX
e-mail: XXXX

Dear XXXX employee,

INFORMATION SHEET

I am a third year PhD student at the University of Strathclyde conducting research into professional software work. I am studying the nature of professional software work in terms of what jobs actually involve, levels of intensity associated with this type of work and factors contributing to intensity. This involves consideration of how work is designed, organised and managed and how individuals experience their work and pressures exerted. My research will also consider how employee experiences of work and intensity varies according to contextual variables such as organisational size, product market, opportunities for skills and knowledge development and leadership style. Another element to be studied is whether individual work experiences and perceptions vary according to software job roles (for example, whether an individual is an analyst, designer, developer or consultant).

Some research has been conducted within academia on professional software work in areas such as teamwork and identity. However, there is still a major research gap in terms of understanding what professional software workers actually do, software job roles and how contextual variables affect individual experience of work. My research aims to make a valuable theoretical and practical contribution through addressing these research gaps and also, providing an understanding of trends in the software industry in general.

My focal research group, as discussed earlier, is professional software workers, that is, workers engaged in the specification, design, development and testing of software. I will also require to engage project managers, human resources and directors, to further understand how professional software workers' experiences of work are affected by organisational and leadership approach.

The research methods I will be utilising are:

- general observation of work activities – to provide an understanding of what professional software work actually involves

- unstructured informal discussions – to discuss general areas relating to software work and uncover issues of interest

- semi-structured interviews – these will be structured yet exploratory to further uncover pertinent issues and also, to investigate associations between variables, in order to increase my understanding of work experiences according to contextual factors

- work diaries – participants will be asked to complete work diaries for one week, in order to increase understanding of day-to-day software work. This will be a simple process involving placing a code (which relates to an activity in a compiled activity list) in a time slot. I am willing to provide individual feedback on this to participants; this feedback may help participants to view how their working day is actually spent, identify potential pressures and how to manage these more effectively

- documentation analysis – studying archival information and documentation on terms and conditions, policies and practices to provide insight into organisational approach

Individual participation in these research methods is entirely voluntary. Participants are under no obligation to respond to all aspects of the research procedures. I am also sensitive to the fact that my research should not impact on the performance of day-to-day business activities.

The information generated from my research will be used for the purpose of my PhD and my thesis. **Finally, you can be assured that all data gathering and writing up of my findings involves firm guarantees of confidentiality and anonymity.**

If you have any questions at any point, please direct these to myself at the contact details given on this letter and I will be happy to help.

Yours faithfully,

Saira Reid

APPENDIX 3: CONSENT FORM FOR WORK DIARY RESEARCH PARTICIPATION

Participation in this research is entirely voluntary.

Signatures indicate that participants are aware of what participation will involve and that any questions concerning the nature of this research have been answered to their satisfaction.

Participants understand that all information will be confidential and anonymity will be preserved.

Participants are under no obligation to respond to all aspects of the research procedures.

Participants reserve the right to terminate participation at any point and can also ask to have their data withdrawn from the study.

Signature

Date

APPENDIX 4: WORK DIARY INSTRUCTIONS

Dear Participant,

Thank you for participating in this diary study. The diary has been designed to be simple and easy to complete so it does not take up too much time. The aim of this diary is to enable me to establish a greater understanding of what professional software work involves and intensity of work, according to individual day-to-day working patterns and activities. Intensity is defined as ‘the amount of effort you have to put into your work to perform your tasks and activities’. This effort may lead to you experiencing work positively, through feelings of accomplishment, enjoyment, keeping your ‘train of thought’, earning some extra income or perhaps negatively, such as constantly thinking about work, feeling under pressure or experiencing stress, anxiety or sleeplessness.

This diary covers a week-long period with a 24 hour slot each day and should be filled in from **Monday _____________ until Sunday _____________**. Please place the completed diary, the ‘About You’ form and the consent form (signed) in the envelope provided; these will be collected by me on Monday _____________.

Instructions for Diary

Each of your work-associated day-to-day activities has been given a code on the ‘Activities and Codes’ page to make it as easy and quick as possible for you to carry out this exercise. On the diary, please enter the code(s) for the relevant activity/activities you are carrying out during each 1 hour slot. Please also include work-related activities such as travelling to/from work, travelling to client sites, lunch and tea breaks (the codes for these are on the Activities and Codes page). If you are on holiday or off sick during the working week, please write ‘Holiday’ or ‘Off Sick’ across the blank timeslots for the relevant day(s). If you do any work while on holiday, please enter the code(s) for the work-related activities you perform in the relevant time slot.

There are also two other boxes on the diary: ‘Intensity of Day’ and also, ‘Main Causes of Intensity’. For the ‘Intensity of Day’ box, using a scale of where 1=Not At All Intense, 2=A Little Intense, 3=Intense and 4=Very Intense, you should place either a 1, 2, 3 or 4 in this box to show how intense you found the day. For the ‘Main Causes of Intensity’, each day, if you experience a particular activity or activities as causing intensity (fitting the definition and descriptions provided above), you would place the code(s) for the activity/activities which made the day intense in the ‘Main Causes of Intensity’ box.

For Example

If you spend between 8am and 9am on Monday planning timescales and resources and composing work-related e-mails, you would look up those activities in the 'Activities and Codes' page, find the relevant codes (code 5 and code 51 for those activities) and place them in the 8-9 time slot. For the 'Intensity of Day' box, if on the Monday you experience work as a little intense, you would place a 2 in the 'Intensity of Day' box. For the 'Main Causes of Intensity' box, if you felt that composing work-related e-mails (code 51) was one of the main causes of intensity, you would place a code 51 in the 'Main Causes of Intensity' box.

TIME	7-8	8-9	9-10	10-11	INTENSITY OF DAY	MAIN CAUSES OF INTENSITY
DAY	AM	AM	AM	AM		
Mon		5, 51			2	51

Please complete the diary entries when you are carrying out the activity/activities **as soon as possible** rather than retrospectively. Completing the diary based on memory of activities which have already passed may bias the results.

I would appreciate if you could also complete some brief questions on the 'About You' page after the diary. This page will provide a context for the diaries and enable me to study how day-to-day working patterns, activities and intensity of work tie in with job titles, descriptions, length of service, gender and status of employment.

Completed diaries are the property of the researcher (Saira Reid) and will be treated with complete confidentiality.

I am happy to provide feedback to individuals on their individual diaries. Feedback will show participants the number of activities they have engaged in on a daily and overall week basis. This may help you to view how the working day/week has been spent, identify potential pressures and help you to manage these more effectively.

Again, thank you for your participation. If you have any queries regarding the diary or my research, please contact me on: XXXX or XXXX XXX XXXX.

Yours faithfully,

Saira Reid

APPENDIX 5: ACTIVITIES AND CODES LIST

REQUIREMENTS AND SYSTEMS ANALYSIS		**OTHER ACTIVITIES**	
Obtaining Understanding of Client's Current System	1	Write Progress Reports	25
Define Requirements and Specifications	2	Prepare, Review, Approve and Check Documentation	26
Produce Proposals for Systems	3	Database Administration	27
Develop and Budget Cost of New System	4	Configuration Management	28
Plan Timescales and Resources	5	Write User Manuals	29
Assemble Teams	6	Training for Users	30
		Provide Support	31
		Recommendations for Future Development	32
DESIGN		Line Management Duties	33
Research Possible Technical and Design Approaches	7	Project Management Duties	34
Create Project Plans	8	Training (for yourself)	35
Write Systems Specification of How System Works	9	Keeping Up-To-Date with Technology/Advances in the Field	36
Break Specifications into Basic Elements	10	Travelling To/From Work	37
		Travel to Client Site(s)	38
		Lunch	39
DEVELOPMENT		Tea Break	40
Allocate Sections for Development	11		
Develop Specific Section of Software	12		
Develop Large Section of Software	13	**MEETINGS**	
		Meeting with Colleague(s)	41
		Meeting with Manager(s)	42
TESTING/ INSTALLATION		Meeting with Line Manager	43
Create Testing Schedule	14	Meeting with Line Staff	44
Test Program Modules	15	Informal Discussion with Colleague(s)	45
System Test	16	Informal Discussion with Manager(s)	46
Reviewing, Analysing and Fixing Faults	17	Meeting with Clients	47
Make Changes if System Inadequate	18	Meeting with Users	48
Factory Acceptance Testing	19		
Site Acceptance Testing	20		
User Acceptance Testing	21	**TELEPHONE/ E-MAILS**	
Plan Systems Installation	22	Making Phone call(s)	49
Install Software/System	23	Receiving Phone call(s)	50
Maintain and Update System	24	Reading/Replying/Composing Work E-Mail	51
		Personal Phone call(s)	52
		Personal E-Mail(s)	53
		MISCELLANEOUS	
		Recording and Updating Observation Reports	54
		Planning for Meetings	55
		Sales	56
		Recruitment and Selection	57
		Re-Testing of Bugs	58
		Coaching Others	59
		Planning/Timetabling Work for Day	60
		Task Analysis	61
		Using Messenger	62
		Document and Code Reviews	63

As well as completing the boxes with the appropriate number(s), please asterisk (*) the diary if you are working at home.

APPENDIX 6: WORK DIARY

TIME DAY *	6-7 AM	7-8 AM	8-9 AM	9-10 AM	10-11 AM	11-12 AM	12-1 PM	1-2 PM	2-3 PM	3-4 PM	4-5 PM	5-6 PM	6-7 PM	7-8 PM	8-9 PM	9-10 PM	10-11 PM	11-12 PM	12-1 AM	1-2 AM	2-3 AM	3-4 AM	4-5 AM	5-6 AM	INTENSITY OF DAY **see below	MAIN CAUSES OF INTENSITY ***see below
Mon																										
Tues																										
Wed																										
Thurs																										
Fri																										
Sat																										
Sun																										

***For out-with normal working hours, only complete the diary for the hours in which you undertake work for XXXX.**

****Each day, using a scale of where 1=Not At All Intense, 2=A Little Intense, 3=Intense and 4=Very Intense, place a 1, 2, 3 or 4 into the 'Intensity of Day' box to show how intense you found the day.**

*****Each day, if you experience an activity or activities as causing intensity, place the relevant code(s) relating to the activity/activities in the 'Main Causes of Intensity' box.**

APPENDIX 7: ABOUT YOU FORM

1. What is your current job title?

____

2. Which (tick one box) of these descriptions best summarises your overall role working for XXXX:

   ☐ Work with employing company, project leaders and clients to discuss IT requirements, in order to design and produce IT specifications for software projects. Mixture of business, sales and technical areas.

   ☐ Take specifications for new systems and design them completely.

   ☐ Translate requirements and design specifications to make software programs.

   ☐ Develop software and IT systems solutions (either self-employed or company-employed) for clients

   ☐ Devise testing schedules, test programs, diagnose and fix faults and make final adjustments to ensure technical compatibility and user satisfaction.

3. How long have you worked for/been working for XXXX?

____

4. Are you (tick box): Male ☐ Female ☐

5. What is the status of your employment? (please tick the most appropriate box)

   ☐ Full Time Permanent Employee ☐ Part Time Employee

   ☐ Full Time Contractor ☐ Part Time Contractor

6. How typical is the week that you have described in the diary?

____
____
____

7. Are there any events that have occurred this week that have had an impact on your normal work/time allocations or that have changed the planned activities on which you were working? How has this affected you?

____
____
____

8. Please indicate whether you would prefer feedback to be given (tick box):
   Via E-Mail ☐ Via Post ☐

APPENDIX 8: SEMI-STRUCTURED INTERVIEW QUESTIONS FOR SOFTWARE PROFESSIONALS

Have consent form signed.
Start recording.
State date, who interviewing.
Firstly, state that interview is voluntary, confidential and anonymous.
Permission to record, can stop at any time.
Once finished interview, transcribe, e-mail a copy/post a copy to you, look through to make sure you are happy with what has been recorded, add things if you wish.

PROVIDE OVERVIEW
Studying professional software work in terms of what work actually involves and levels of intensity associated with this type of work, factors which may contribute to work intensity. Contextual element-comparing specialist firm/in-house IT dept in large firm/creative multimedia/computer games firm.

Define Intensity and Intensification

When asking you to consider
Work Intensity and Work Intensification, relates to the effort you put into your job during the time that you are working (Burchell, 2002)

Intensity/Intensification can be split into two aspects of your workload:
Qualitative, being the difficulty and complexity of your workload
Quantitative, being the amount of work you actually have to perform

To distinguish between the two:

Work Intensity, referring to the condition and experience of work at a moment or stage in time

Work Intensification, referring to the evolutionary nature of work over time i.e. asking you to consider how work has evolved over time

So, distinction relates to temporal factors.

BACKGROUND

- Could you start by giving me some background information on your education and qualifications?
- Could you outline your previous employment, experience and training?
- What computing languages do you know?
- Are you a member of any professional associations? If yes, what are the benefits/drawbacks of membership? *contractor/self-employed/other company*
- How did you get recruited into XXXX?
- How long have you been working at XXXX?

- What is (a) your job title and (b) could you briefly summarise your job description at XXXX?
- What are your contractual hours and the times of the day/week you tend to work? Overtime?
- When you started at XXXX, what sort of induction and training, if any, did you receive initially?

ORGANISATION OF WORK

- Do you work in formal teams? What is their location i.e. on-site, virtual?
- How many are in the teams and how are they composed i.e. how do you decide where to place people?
- Are your teams based around projects?
- What is the duration of these projects?
- What is the role of the Project Manager; Technical Leads?
- Do you have many meetings? What is the frequency and reason for these meetings? Where are they located?
- How is information provided in the workplace (e.g. noticeboards, newsletters, meetings, TUs, gossip)?
- What are the forms of employee representation at XXXX? E.g. unions, employee rep meetings, informal
- How much influence do you have at XXXX i.e. over areas such as staffing levels, people recruited, allocating people to work, changes to working practices? (Intensity)
- Has this influence changed over the years and how? (Intensification)

Tasks

- Could you give me a brief account of what a typical day at work might look like for you? How similar are days i.e. how much variation is there in what you do?
- Could you give me a brief overview of the project you are working on right now?
- What are you working on (overall picture and at the moment)?
- What do you like/dislike most about your work?
- Are there any tasks or additional duties to your normal work which you undertake which you feel impact on the intensity of your work? (e.g. documentation, line management duties, sales)
- Does your work deviate much from your normal tasks (e.g. to include testing, fixing faults, recording and updating ORs)? To what extent does this impact on the intensity of your work?
- What about the roles of your team members? Can you think of any individuals whose roles are more/less intensive and why you think this may be the case?

Breaks

- Do you tend to take a lunch break and do you take many breaks during the day? To what extent does taking/not taking breaks impact on the intensity of your work?

Carrying out Work/Setting Pace of Work

- How much control do you have over (a) how you carry out your work and (b) setting the pace of your work? (Intensity)
- Has the control you have over your work and setting the pace of your work changed over time? (Intensification)

Deadlines

- To what extent are you able to set your own work targets/deadlines?
- To what extent do deadlines and time pressures impact on the intensity of your work? (Intensity)
- Has the extent to which you are able to set your own targets and deadlines changed over time? (Intensification)

Clients

- What sort of role do clients have during projects? How much influence do they have? (Intensity)
- Has the role and influence of clients been different in the past compared to now? (Intensification)
- To what extent would you say clients have an impact on your work intensity of the work intensity of others? (probe for sales, planning for meetings)

Performance Metrics

- What sort of performance metrics (if any) are used which you need to follow (e.g. log sheets, status reporting, performance appraisal, performance-related pay)?
- Thinking back to when you started at XXXX, are these performance metrics a recent development or have they always been used?
- Are these performance metrics evaluated and how?
- To what extent do performance metrics impact on the intensity of your work? (Intensity)
- Can you see any evidence of performance metrics impacting on the intensity of the work of others? (Intensity)

CULTURE AND LEADERSHIP STYLE

Culture and Leadership Style

- How would you describe the culture at XXXX?
- How would you describe the leadership style at XXXX?
- Thinking of leadership styles at the various levels (director, head of division, project management, line management, Technical Lead) would you say they are similar or different in style? In what ways?
- To what extent do you think contractors, agency staff, women, men, older, younger, are treated equally and given the same opportunities at XXXX?
- Do you feel that factors such as your age, gender, qualifications, whether permanent or contractor have any impact on intensity of your work?

- Has the culture/leadership style at XXXX changed in any way and if so, how? (Intensification)
- How do you think the culture and styles of leadership impact on the intensity of your work and other people's work?

Role of HR

- How do you view the role of HR in XXXX? In what ways and how well does HR support employees when necessary? (Intensity)
- Has the role of HR at XXXX changed at all over the years and if so, how (Intensification)? How has this affected you?

Impact of Technical Leads and Supervisors

- What sort of role do Technical Leads and supervisors play? Is it facilitative/controlling?
- To what extent do supervisors/managers/Technical Leads impact on your work intensity?

Socialising

- Do you socialise with other colleagues outside of work (formal/informal events)? Do you think the amount of socialisation at work is enough?

TEAMWORK

Communication

- How do you communicate within teams? (Informal discussions/Informal meetings/Formal meetings/E-mails)
- To what extent do these methods of communication impact on intensity of your work? (interruptions, etc)?

Staffing Levels

- What are staffing levels like on your project? To what extent do staffing levels impact on intensity of work?

Interdependency Between Teams

- Would you say that specialist knowledge resides within particular individuals? Are there any mechanisms for dispersing this knowledge?
- Is there interdependency between team members (i.e. different skills, similar skills)?
- If yes, to what extent does this interdependency impact on the intensity of your work?
- What about other team members? Can you think about any ways in which interdependency impacts on the intensity of other people's work?

Meeting Personal Development Aspirations

- Do you think your education and experience has prepared you for what you do?

- What has helped you to learn and develop the kinds of skills and knowledge needed for your job? (education, training, help from colleagues, life, taught yourself)
- What skills do you think are important for performing your job effectively?
- Do you think the emphasis on particular skills has changed in any way over time and if yes, how?
- How dependent are you on XXXX to assist you in meeting your personal development needs?
- How easy is it for you to manage your own personal development aspirations along with those of the organisation?
- How easy do you think it is for you or others to move out of specialist niches into new areas, other projects, teams or divisions, fitting with aspirations?
- To what extent does managing and fulfilling these aspirations impact on the intensity of your work?
- What about other team members? Can you think about any ways in which managing/fulfilling aspirations impacts on the intensity of other people's work?

CONTEXTUAL FACTORS

Organisational Size

- Thinking of your own experience and more generally, do you think company size impacts on the intensity of people's work and, if yes, in what ways?

Organisational Type

- Do you think that organisational type (whether specialist software firm, in-house IT dept, computer games firm, multimedia firm) may have an influence on the levels of intensity experienced at work and, if yes, how?

Product Market

- Thinking of your own experience and more generally, do you think the product market you produce software for has any influence on the intensity of work? (e.g. work in different divisions producing different products for different markets) If so, how?
- Do you think market dynamics (explain) play any role in affecting intensity of work and if so, how?

IN GENERAL

- Overall, do you feel you experience intensity at work?
  If no, why do you think this is the case? What could potentially contribute to intensity for you?/ If yes, what causes it?
- Do you think the software industry has changed over time and if so, how?
- Do you think roles in the software industry have changed over time and if so, how?

APPENDIX 9: SEMI-STRUCTURED INTERVIEW QUESTIONNAIRE SCRIPT FOR OFFSHORE WORKERS

Dear Participant,

Thank you for taking the time to fill in this interview questionnaire-style script. The interview scripts will be utilised to provide me with a more detailed understanding of your experiences working as a software engineer. The questions require short answer responses; please type your responses on the lines provided. If you need more space for your answers, please feel free to add more lines as necessary. The questionnaire script should take roughly one hour to fill in, depending on responses. Please e-mail your completed script to: XXXX.

Below is some information on definitions of key terms used in the script. Please read this information before answering the questions.

It would be helpful if participants could type their names at the top of the interview script – this is just so that I know which forms belong to which people when I am analysing the data. **Completed interview questionnaire scripts are the property of the researcher (Saira Reid) and will be treated with complete confidentiality and anonymity.**

If you have any questions or queries, please contact me at: XXXX or XXXX XXX XXXX.

INFORMATION ON DEFINITIONS

When asking you to consider both work intensity and work intensification, both terms relate to the effort you put into your job during the time that you are working. Both intensity and intensification relate to two aspects of your workload, one being the difficulty and complexity of your work and the other relating to the actual volume of work that you have to perform.

Definitions of Work Intensity and Work Intensification

Questions asking you to consider 'work intensity' are asking you to think about your work at present, i.e. how you experience your work right now.

Questions asking you to consider 'work intensification' are asking you to think about how your work has evolved or changed over time (for example, for the length of time you have been working for your present company or for as long as you have been working in software engineering).

BACKGROUND

- Please briefly state your university education and qualifications

  ____

  ____

- Please briefly outline your previous employment, experience and training.

  ____

  ____

  ____

- Please list what computing languages you know.

  ____

- Are you a member of any professional associations? YES/NO If YES, please summarise the benefits/drawbacks of membership.

  ____

  ____

- How long have you been working for XXXX? ________________________

- What is (a) your job title and (b) could you briefly summarise your job description?

  ____

  ____

- Please state your working hours and the times of the day/week you tend to work, as well as if you work any overtime (if you do, please briefly state the reasons for this)

  ____

  ____

  ____

ORGANISATION OF WORK

- Do you work in formal teams? YES/NO (please apply italics to highlight answer) If YES, where are your team members located e.g. on-site, virtual, offshore?

  ____

  ____

- How many are in your team? ________________________________

- Please briefly outline the roles of team members and their locations (e.g. on-site, Mumbai, Pune, Glasgow, Newcastle, etc)

- To what extent does the physical proximity of team members impact on the intensity of your work?

- Who decides where to place people in teams and what are decisions based on?

- Are your teams based around projects? YES/NO If YES, what is the duration of these projects?

- How are projects structured (e.g. using 'waterfall cycle', 'Agile' methods, phased releases, overlapping stages in life cycle)?

- Please summarise how you view the role of the Project Manager; the role of Team/Technical Leads.

- Do you have many meetings? YES/NO If YES, What is the frequency and reason for these meetings? Where are they located (e.g. office floor, meeting room, etc)?

Tasks

- Please give a brief account of what a typical day at work might look like for you? How similar are days i.e. how much variation is there in what you do?

- Please give a brief overview of the project you are working on right now.

- Please summarise what you like/dislike most about your work.

- Are there any tasks or additional duties to your normal work which you undertake which you feel impact on the intensity of your work? (e.g. documentation, line management duties, sales, etc)?

- Can you think of any individuals whose roles are more/less intense? YES/NO. If YES, why you think this may be the case?

Breaks

- Please state if you tend to take a lunch break and if you take many breaks during the day.

- Please outline to what extent taking/not taking breaks impacts on the intensity of your work.

Carrying out Work/Setting Pace of Work

- Please outline how much control you have over (a) how you carry out your work and (b) setting the pace of your work.

- Has the control you have over your work and setting the pace of your work changed over time? YES/NO. If YES, in what way(s)?

____
____
____
____

Deadlines

- To what extent you are able to set your own work targets/deadlines?

____
____
____

- To what extent do deadlines and time pressures impact on the intensity of your work?

____
____
____

- Has the extent to which you are able to set your own targets and deadlines changed over time? YES/NO. Please briefly explain why this has changed or not changed.

____
____
____

Clients

- Who is/are your client(s) (e.g. XXXX only or also other groups)?

____

- What sort of role and influence do clients have during projects?

____
____
____

- Has the role and influence of clients been different in the past compared to now? YES/NO. Please briefly explain your answer.

____
____
____

- To what extent do clients have an impact on your work intensity or the work intensity of others?

____
____
____
____

Performance Metrics

- Please state what performance metrics (if any) are used which you need to follow (e.g. log sheets, status reporting, performance appraisal, performance-related pay).

  ____

  ____

- Thinking back to when you started at XXXX, are these performance metrics a recent development or have they always been used?

  ____

  ____

  ____

- Are these performance metrics evaluated? YES/NO. If YES, in what way(s)?

  ____

  ____

  ____

- Please outline to what extent performance metrics impact on the intensity of your work.

  ____

  ____

  ____

- Can you see any evidence of performance metrics impacting on the intensity of the work of others? YES/NO. If YES, in what ways?

  ____

  ____

CULTURE AND LEADERSHIP STYLE

Culture and Leadership Style

- How would you describe the culture at XXXX?

  ____

  ____

  ____

- How would you describe the leadership style at XXXX?

  ____

  ____

  ____

- Thinking of leadership styles at the various levels of XXXX (director, head of division, project management, line management, Technical Lead) would you say they are similar or different in style? In what ways?

  ____
  ____
  ____

- How would you describe the working style and approach to work of UK-based software workers?

  ____
  ____
  ____

- How would you describe the working style and approach to work of Indian software workers?

  ____
  ____
  ____

- To what extent do you think contractors, agency staff, women, men, older, younger, are treated equally and given the same opportunities at XXXX?

  ____
  ____
  ____

- Do you feel that factors such as your age, gender, qualifications, whether permanent or contractor have any impact on intensity of your work? YES/NO. If YES, please summarise in what way(s).

  ____
  ____
  ____
  ____

- Has the culture/leadership style at XXXX changed in any way? YES/NO If YES, in what way(s)?

  ____
  ____
  ____

- To what extent do you think the culture and styles of leadership impact on the intensity of your work and other people's work?

  ____
  ____
  ____

Impact of Technical Leads and Supervisors

- Please outline what sort of role Technical Leads and supervisors/project managers play (e.g. is it facilitative/controlling, mixture of both?).

- To what extent do supervisors/managers/Technical Leads impact on your work intensity?

TEAMWORK

Communication

- How do you communicate within teams? (e.g. Informal discussions/Informal meetings/Formal meetings/Phone calls/E-mails/Messenger, Other-please state).

- To what extent do these methods of communication impact on intensity of your work?

Staffing Levels

- What are staffing levels like on your project? (e.g. do you have enough team members, too few, too many)

- To what extent do staffing levels impact on intensity of work?

Interdependency Between Teams

- Would you say that specialist knowledge resides within particular individuals? YES/NO. If YES, please outline if there are there any mechanisms for dispersing this knowledge.

- Is there interdependency between team members (i.e. different skills, similar skills) YES/NO. Please briefly the nature of these interdependencies.

  ____
  ____
  ____

- If you answered YES to the previous question, to what extent does this interdependency impact on the intensity of your work?

  ____
  ____
  ____

- What about other team members? Can you think about any ways in which interdependency impacts on the intensity of other people's work?

  ____
  ____
  ____

Meeting Personal Development Aspirations

- Do you think your education and experience has prepared you for what you do? YES/NO. If YES, please summarise in what way(s)?

  ____
  ____

- What has helped you to learn and develop the kinds of skills and knowledge needed for your job? (e.g. education, training, help from colleagues, life, taught yourself)

  ____

- Please briefly summarise the skills you think are important for performing your job effectively.

  ____
  ____

- Do you think the emphasis on particular skills has changed in any way over time? YES/NO. If YES, please summarise in what way(s)?

  ____

- Please outline how easy you think it is for you or others to move out of specialist niches into new areas, other projects, teams or divisions, fitting with aspirations.

  ____
  ____
  ____

- To what extent does managing and fulfilling these aspirations impact on the intensity of your work?

  ____
  ____
  ____

- What about other team members? Can you think about any ways in which managing/fulfilling aspirations impacts on the intensity of other people's work?

  ____
  ____
  ____

CONTEXTUAL FACTORS

Organisational Size

- Thinking of your own experience and more generally, do you think company size impacts on the intensity of people's work? YES/NO. If YES, in what ways?

  ____
  ____
  ____
  ____

Organisational Type

- Do you think that organisational type (whether specialist software firm, in-house IT dept, computer games firm, multimedia firm) may have an influence on the levels of intensity experienced at work? YES/NO. If YES, in what ways?

  ____
  ____
  ____
  ____

Product Market

- Thinking of your own experience and more generally, do you think the product market you produce software for has any influence on the intensity of work (e.g. work in different divisions producing different products for different markets)? YES/NO. If YES, in what ways?

  ____
  ____
  ____
  ____

- Do you think market dynamics play any role in affecting intensity of work? YES/NO. If YES, in what ways?

  ____
  ____
  ____
  ____

IN GENERAL

- Overall, do you feel you experience intensity at work? YES/NO
- If NO, why do you think this is the case? What could potentially contribute to intensity for you? If YES, please summarise the main factor(s) which make you experience intensity at work.

  ____
  ____
  ____

- If you experience intensity at work, please briefly explain whether you feel intensity comes from (a) volume of work (b) difficulty and complexity of work or (c) a mixture of both the volume of work and also the difficulty and complexity of work

  ____
  ____
  ____

Thank you very much for taking the time to fill in this questionnaire script. Your help is very much appreciated.

APPENDIX 10: SEMI-STRUCTURED INTERVIEW QUESTIONS FOR PROJECT MANAGERS

Have consent form signed.
Start recording.
State date, who interviewing.
Firstly, state that interview is voluntary, confidential and anonymous.
Permission to record, can stop at any time.
Once finished interview, transcribe, e-mail a copy/post a copy to you, look through to make sure you are happy with what has been recorded, add things if you wish.

PROVIDE OVERVIEW
Studying professional software work in terms of what work actually involves and levels of intensity associated with this type of work, factors which may contribute to work intensity. Contextual element-comparing specialist firm/in-house IT dept in large firm/creative multimedia/computer games firm.

Define Intensity and Intensification

When asking you to consider
Work Intensity and Work Intensification, relates to the effort you put into your job during the time that you are working (Burchell, 2002)

Intensity/Intensification can be split into two aspects of your workload:
Qualitative, being the difficulty and complexity of your workload
Quantitative, being the amount of work you actually have to perform

To distinguish between the two:

Work Intensity, referring to the condition and experience of work at a moment or stage in time

Work Intensification, referring to the evolutionary nature of work over time i.e. asking you to consider how work has evolved over time

So, distinction relates to temporal factors.

BACKGROUND

- Could you start by giving me some background information on your education and qualifications?
- Could you outline your previous employment, experience and training?
- What computing languages do you know?
- Are you a member of any professional associations? If yes, what are the benefits/drawbacks of membership?
- How did you get recruited into XXXX?
- How long have you been working at XXXX?

- What is (a) your job title and (b) could you briefly summarise your job description at XXXX?
- What are your contractual hours and the times of the day/week you tend to work?
- When you started at XXXX, what sort of induction and training, if any, did you receive initially?

ORGANISATION OF WORK

- Do you work in formal teams? What is their location i.e. on-site, virtual?
- How many are in the teams and how are they composed i.e. how do you decide where to place people?
- Are your teams based around projects?
- What is the duration of these projects?
- What is the role of the Project Manager; Technical Leads?
- Do you have many meetings? What is the frequency and reason for these meetings? Where are they located?
- How is information provided in the workplace (e.g. noticeboards, newsletters, meetings, TUs, gossip)?
- What are the forms of employee representation at XXXX? E.g. unions, employee rep meetings, informal
- How much influence do you have at XXXX i.e. over areas such as staffing levels, people recruited, allocating people to work, changes to working practices? (Intensity)
- Has this influence changed over the years and how? (Intensification)

Tasks

- Could you give me a brief overview of the project you are working on right now?
- Could you give me a brief account of what a typical day at work might look like for someone in your team working on design; development; testing; support? How similar are days i.e. how much variation is there in what they do?
- What are you working on (overall picture and at the moment)?
- What do you think team members like/dislike most about their work?
- Are there any tasks or additional duties to your normal work which your team undertakes which may impact on the intensity of their work? (e.g. documentation, line management duties, sales)
- Does work for team members deviate much from normal tasks (e.g. to include testing, fixing faults, recording and updating ORs)? To what extent do you think this impacts on the intensity of their work?
- Can you think of any individuals whose roles are less/more intense and why this is the case?

Breaks

- Are team members encouraged to a lunch break and breaks during the day? Do you think they take lunch breaks and sufficient breaks during the day?
- To what extent does taking/not taking breaks impact on the intensity of their work?

Carrying out Work/Setting Pace of Work

- How much control do team members have over (a) how they carry out their work and (b) setting the pace of their work? (Intensity)
- Do you think the control team members exercise over their work and setting the pace of their work has changed over time? (Intensification)

Deadlines

- How much influence do you have in the setting of targets and deadlines?
- To what extent are team members able to set your own work targets/deadlines?
- Has the extent to which team members are able to set your own targets and deadlines changed over time? (Intensification)
- From your perspective, how do deadlines and time pressures impact on the intensity of their work? (Intensity)

Clients

- What sort of role do clients have during projects? How much influence do they have? (Intensity)
- Has the role and influence of clients been different in the past compared to now? (Intensification)
- To what extent would you say clients have an impact on work intensity of team members? (probe for sales, planning for meetings) Do team members have any way to offset this?

Performance Metrics

- What sort of performance metrics (if any) are used which team members need to follow (e.g. log sheets, status reporting, performance appraisal, performance-related pay)?
- Thinking back to when you started at XXXX, are these performance metrics a recent development or have they always been used?
- Are these performance metrics evaluated and how?
- To what extent do you think performance metrics impact on the intensity of work? (Intensity)

CULTURE AND LEADERSHIP STYLE

Culture and Leadership Style

- How would you describe the culture at XXXX?
- How would you describe the leadership style at XXXX?
- Thinking of leadership styles at the various levels (director, head of division, project management, line management, Technical Lead) would you say they are similar or different in style?
- To what extent do you think contractors, women, men, older, younger, agency staff are treated equally and given the same opportunities at XXXX?
- Do you feel that factors such as your age, gender, qualifications, whether permanent or contractor have any impact on intensity of work?
- Has the culture/leadership style at XXXX changed in any way and if so, how? (Intensification)

- How do you think the culture and styles of leadership impact on the intensity of work?

Role of HR
- How do you view the role of HR in XXXX? In what ways and how well does HR support employees when necessary? (Intensity)
- Has the role of HR at XXXX changed at all over the years and if so, how (Intensification)? How has this affected you and your team members?

Impact of Technical Leads and Supervisors
- What sort of support do team members receive from colleagues/line management/project management? Is support directional? Emotional? A bit of both?
- To what extent do you think Technical Leads and supervisors impact on work intensity of team members?

Socialising
- Does the team socialise outside of work (formal/informal events)? Do you think the amount of socialisation at work is enough?

TEAMWORK

Communication
- How do you communicate within teams? (Informal discussions/Informal meetings/Formal meetings/E-mails)
- To what extent do you think these methods of communication impact on intensity of work? (interruptions, etc)?

Staffing Levels
- What are staffing levels like on your project? To what extent do staffing levels impact on intensity of work?

Interdependency Between Teams
- Would you say that specialist knowledge resides within particular individuals? Are there any mechanisms for dispersing this knowledge?
- Is there interdependency between team members (i.e. different skills, similar skills)?
- If yes, to what extent does this interdependency impact on the intensity of work?

Meeting Personal Development Aspirations
- Do you think education and experience has prepared team members for what they do?
- What has helped individual team members to learn and develop the kinds of skills and knowledge needed for their jobs? (education, training, help from colleagues, life, taught yourself)
- What skills do you think are important for team members to perform their jobs effectively?

- Do you think the emphasis on particular skills has changed in any way over time and if yes, how?
- How dependent are team members on XXXX to assist you in meeting their personal development needs?
- How easy is it for team members to manage their own personal development aspirations along with those of the organisation?
- How easy is it for team members to move out of specialist niches into new areas, other projects, teams or divisions, fitting with aspirations?
- To what extent does managing and fulfilling these aspirations impact on the intensity of work?

CONTEXTUAL FACTORS

Organisational Size

- Thinking of your own experience and more generally, do you think company size impacts on the intensity of people's work and, if yes, in what ways?

Organisational Type

- Do you think that organisational type (whether specialist software firm, in-house IT dept, computer games firm, multimedia firm) may have an influence on the levels of intensity experienced at work and, if yes, how?

Product Market

- Thinking of your own experience and more generally, do you think the product market you produce software for has any influence on the intensity of work? (e.g. work in different divisions producing different products for different markets) If so, how?
- Do you think market dynamics (explain) play any role in affecting intensity of work and if so, how?

IN GENERAL

- Overall, do you feel you see evidence of intensity at work?
- If yes, what causes it?
- If no, why do you think this is the case? What could potentially contribute to intensity for you
- Do you think the software industry has changed over time and if so, how?
- Do you think roles in the software industry have changed over time and if so, how?

APPENDIX 11: SEMI-STRUCTURED INTERVIEW QUESTIONS FOR SENIOR MANAGERS/DIRECTORS

Have consent form signed.
Start recording.
State date, who interviewing.
Firstly, state that interview is voluntary, confidential and anonymous.
Permission to record, can stop at any time.
Once finished interview, transcribe, e-mail a copy/post a copy to you, look through to make sure you are happy with what has been recorded, add things if you wish.

BACKGROUND

- Could you give me some brief information on your role when XXXX was first established, how this role has evolved and your current position?

CONTEXT

Competitive market

- Who are XXXX's main competitors? (e.g. types of companies, industry types)
- Does market position affect XXXX and if so, how? (e.g. impacts on project times)
- Do you think the software industry has changed over time and if so, how?

Clients

- What sort of role do clients have during projects? How much influence do they have? (Intensity)
- Has the nature of contracts with clients changed at all? (Intensification)
- To what extent would you say clients have an impact on work intensity of employees?

Deadlines

- Have deadlines become tighter? Have project deadlines become more aggressive?
- From your perspective, how do deadlines and time pressures impact on the intensity of employees' work?

WORK ORGANISATION

Tasks and Roles

- Would you say that intensity varies at different career stages for a software engineer and if so, in what ways?
- Do you think roles in the software industry have changed over time and if so, how?

Staffing Levels

- Do you consider staffing levels to be appropriate?
- To what extent do you think these staffing levels impact on the intensity of work?

SOFTWARE INDUSTRY IN GENERAL

- Do you think company size impacts on the intensity of people's work and if yes, in what ways?
- Do you think that organisational type (specialist firm; in-house IT dept, creative firm) may have an influence on the levels of intensity experienced at work and if so, how?
- Do you think the product market you produce software for has any influence on intensity of work and if so, how?
- What do you think are the main sources of intensity experienced by employees?

APPENDIX 12: CONSENT FORM FOR SEMI-STRUCTURED INTERVIEW RESEARCH PARTICIPATION

Participation in this research is entirely voluntary.

Signatures indicate that participants are aware of what participation will involve and that any questions concerning the nature of this research have been answered to their satisfaction.

Participants understand that all information will be confidential and anonymity will be preserved.

Participants are under no obligation to respond to all aspects of the research procedures.

Participants reserve the right to terminate participation at any point and can also ask to have their data withdrawn from the study.

Signature

Date

APPENDIX 13: WORK INTENSITY INDEX

ORGANISATION OF WORK

Employee Representation
Influence
Meetings
Planning for Meetings
Physical Proximity

Tasks
Additional Responsibilities (e.g. line management duties, support)
Deviations From Normal Tasks
Project Management Duties
High Level Design Activities
Low Level Design Activities
Development Activities
Installing Software/System

Breaks
Breaks

Carrying out Work/Setting Pace of Work
Control Over Work
Setting Work Pace
Individual Discretion/Self-Set Targets

Deadlines
Targets and Deadlines
Time Pressures

Clients
Clients Or Customers
Sales
Planning For Meetings

Performance Metrics
Performance Metrics
Pay Incentives
Performance Appraisals
Log Sheets

CULTURE AND LEADERSHIP STYLE

Culture and Leadership Style
Culture
Leadership

Human Resources
Role Of Human Resources
Support From Human Resources

Impact of Technical Leads and Supervisors
Supervisors/Managers/Technical Leads
Support From Project Manager/Technical Leads

Socialising
Levels of Socialising

TEAMWORK

Teamwork

Colleagues

Interdependency Between Team Members

Communication
Informal Discussions/Meetings
Formal Discussions
Meetings
E-Mails
Phone Calls
Instant Messenger

Staffing Arrangements

INDIVIDUAL FACTORS
Gender
Age
Qualifications
Status (Permanent, Contractor, Agency Staff)

Meeting Personal Development Aspirations
Ability to Meet Personal Development Aspirations
Ability to Move on to New Things

CONTEXTUAL FACTORS

Company Size

Organisational Type

Product Market

Profit

APPENDIX 14: WORK DIARIES FOR SPECSOFT PT1 DESIGN TECHNICAL LEAD AND DESIGN TEAM MEMBER

ROLE LEVEL/TENURE	Design Technical Lead Technical Lead/Five Years	Design Team Member Junior/One Year
RATES OF INTENSITY	2, 3, 2, 2, 1	2, 3, 3, 3, 1
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	21	14
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	Informal Discussion With Colleague(s) (27%) Planning for Meetings (19%) Reading/Replying/Composing Work E-Mail(s) (9%) Travel to Client Site(s) (9%) Meeting With Clients (6%) Lunch (5%) Personal E-Mails (5%) Travelling To/From Work (5%) Database Administration (3%) Meeting With Colleague(s) (2%) Making Phone Call(s) (2%) Personal Phone Call(s) (2%) Sales (2%) System Test (2%) Develop Small Section of Software (1%) Fixing Faults (1%) Meeting With Line Staff (1%) Receiving Phone Call(s) (1%) Research Possible Technical and Design Approaches (1%) Tea Break (1%)	High Level Design (19%) Low Level Design (14%) Informal Discussion With Colleague(s) (12%) Tea Break (11%) Travel to Client Site(s) (8%) Meeting With Clients (7%) Travelling To/From Work (7%) Receiving Phone Call(s) (6%) Lunch (5%) Making Phone Call(s) (4%) Reading/Replying/Composing Work E-Mail(s) (3%) Prepare Documentation (2%) Personal E-Mail(s) (1%) Personal Phone Call(s) (1%)

APPENDIX 15: WORK DIARIES FOR SPECSOFT PT1 DEVELOPMENT TECHNICAL LEADS

ROLE LEVEL/TENURE	Development Technical Lead Technical Lead/Six and a Half Years	Development Technical Lead Technical Lead/Nine Years
RATES OF INTENSITY	1, 2→4, 1, 1 (on holiday for one day)	2, 4, 2, 2, 3, 2 (one extra day at weekend)
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	19	18
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	Fixing Faults (33%) Recording and Updating Observation Reports (15%) Travelling To/From Work (8%) Informal Discussion With Colleague(s) (7%) Personal E-Mail(s) (6%) Lunch (5%) Reading/Replying/Composing Work E-Mail(s) (4%) Define Requirements and Specifications (3%) Keeping Up-To-Date With Technology/Advances (3%) Low Level Design (3%) Training (for yourself) (3%) Tea Break (2%) Make Changes if System Inadequate (1%) Making Phone Call(s) (1%) Meeting With Colleague(s) (1%) Obtaining Understanding of Client's Current System (1%) Personal Phone Call(s) (1%) Receiving Phone Call(s) (1%) System Test (1%)	Fixing Faults (34%) Informal Discussion With Colleague(s) (11%) Travelling To/From Work (10%) Lunch (6%) Provide Support (6%) Recording and Updating Observation Reports (6%) Tea Break (5%) Recruitment and Selection (4%) Coaching Others (3%) Line Management Duties (3%) System Test (3%) Allocate Sections for Development (2%) Meeting With Colleague(s) (2%) Reading/Replying/Composing Work E-Mail(s) (2%) Write Progress Reports (2%) Personal E-Mail(s) (1%) Prepare Documentation (1%) Recommendations for Future Development (1%)

APPENDIX 16: WORK DIARIES FOR SPECSOFT PT1 DEVELOPMENT TEAM MEMBERS

ROLE LEVEL/TENURE	Development Team Member Junior/Four Months	Development Team Member Junior/Eight Months	Development Team Member Senior/One Year
RATES OF INTENSITY	No rates provided	1, 1, 2, 2, 1	2, 2 (on holiday for rest of week)
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	4	5	5
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	Fixing Faults (62%) Travelling To/From Work (18%) Develop Small Section of Software (11%) Lunch (9%)	Develop Small Section of Software (66%) Travelling To/From Work (17%) Lunch (8%) Tea Break (7%) Meeting With Line Manager (2%)	Fixing Faults (54%) Configuration Management (23%) Reading/Replying/Composing Work E-Mail(s) (9%) Recording and Updating Observation Reports (9%) Lunch (5%)

APPENDIX 17: WORK DIARIES FOR SPECSOFT PT1 DEVELOPMENT TEAM MEMBERS

ROLE LEVEL/TENURE	Development Team Member Average/One Year	Development Team Member Senior/Sixteen Years
RATES OF INTENSITY	3, 3, 3, 3, 3	1, 2, 1, 2, 2
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	10	10
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	System Test (39%) Informal Discussion With Colleague(s) (13%) Travelling To/From Work (13%) Fixing Faults (8%) Install Software/System (7%) Lunch (6%) Meeting With Colleague(s) (6%) Recording and Updating Observation Reports (4%) Allocate Systems For Development (3%) Reading/Replying/Composing Work E-Mail(s) (1%)	System Test (43%) Prepare Documentation (26%) Lunch (8%) Fixing Faults (5%) Provide Support (5%) Recruitment and Selection (5%) Informal Discussion With Colleague(s) (2%) Informal Discussion With Manager(s) (2%) Personal E-mail(s) (2%) Reading/Replying/Composing Work E-Mail(s) (2%)

APPENDIX 18: WORK DIARIES FOR SPECSOFT PT1 DEVELOPMENT TEAM MEMBERS

ROLE LEVEL/TENURE	Development Team Member Junior/Two Years Five Months	Development Team Member (Contractor) Average/Seven Months	Development Team Member/Support Manager Average/Eight and a Half Years
RATES OF INTENSITY	2, 1, 2, 2, 1	2, 2, 3, 3, 3	2, 3, 2, 2, 1, 1, 1 (two extra days at weekend)
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	13	15	23
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	Fixing Faults (37%) Unit Testing (15%) Lunch (11%) Informal Discussion With Colleague(s) (10%) Making Phone Call(s) (6%) Reading/Replying/Composing Work E-Mail(s) (6%) Coaching Others (3%) Meeting With Colleagues (3%) Prepare Documentation (3%) Re-Testing of Bugs (3%) Maintain and Update System (1%) Receiving Phone Call(s) (1%) Recording and Updating Observation Reports (1%)	System Test (21%) Recording and Updating Observation Reports (13%) Informal Discussion With Colleague(s) (10%) Informal Discussion With Manager(s) (10%) Travelling To/From Work (9%) Lunch (5%) Meeting With Colleague(s) (5%) Meeting With Manager(s) (5%) Re-Testing of Bugs (5%) Tea Break (4%) Fixing Faults (3%) Personal E-Mail(s) (3%) Personal Phone Call(s) (3%) Reading/Replying/Composing Work E-Mail(s) (3%) High Level Design (1%)	Develop Small Section of Software (13%) Reading/Replying/Composing Work E-Mail(s) (10%) Project Management Duties (9%) Travelling To/From Work (9%) Prepare Documentation (8%) Create Testing Schedule (7%) Informal Discussion With Colleague(s) (7%) System Test (7%) Provide Support (7%) Coaching Others (4%) Factory Acceptance Testing (4%) Receiving Phone Call(s) (4%) Lunch (2%) Personal Phone Call(s) (2%) Personal E-Mail(s) (2%) Unit Testing (2%) Informal Discussion With Manager(s) (1%) Install Software/System (1%) Meeting With Clients (1%) Meeting With Colleague(s) (1%) Meeting With Line Manager (1%) Recording and Updating Observation Reports (1%) Write Progress Reports (1%)

APPENDIX 19: WORK DIARIES FOR SPECSOFT PT1 TEST TECHNICAL LEAD AND TEST TEAM MEMBER

ROLE LEVEL/TENURE	Test Technical Lead Technical Lead/Two Years and Ten Months	Test Team Member Average/Nine Months
RATES OF INTENSITY	2, 2, 3, 3, 2, 1 (one extra day at weekend)	4, 4, 4, 4, 3
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	24	9
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	Informal Discussion With Colleague(s) (20%) Reading/Replying/Composing Work E-Mail(s) (13%) System Test (11%) Recording and Updating Observation Reports (10%) Receiving Phone Call(s) (6%) Travelling To/From Work (6%) Lunch (4%) Maintain and Update System (4%) Keeping Up-To-Date With Technology/Advances (3%) Make Changes if System Inadequate (3%) Meeting With Colleague(s) (3%) Personal Phone Call(s) (3%) Tea Break (3%) Informal Discussion with Manager(s) (2%) Making Phone Call(s) (2%) Coaching Others (1%) Create Testing Schedule (1%) Fixing Faults (1%) Install Software/System (1%) Meeting With Line Staff (1%) Plan Timescales and Resources (1%) Prepare Documentation (1%) Project Management Duties (1%) Write Progress Reports (1%)	System Test (37%) Recording and Updating Observation Reports (15%) Informal Discussion With Manager(s) (9%) Travelling To/From Work (9%) Meeting With Colleague(s) (8%) Re-Testing of Bugs (8%) Informal Discussion With Colleague(s) (6%) Lunch (6%) Reading/Replying/Composing Work E-Mail(s) (2%)

APPENDIX 20: WORK DIARIES FOR SPECSOFT PT2 TECHNICAL LEADS

ROLE LEVEL/TENURE	Software and Consulting Section Technical Lead/Ten Years	Software Section Technical Lead/Seven Years
RATES OF INTENSITY	3, 1→2, 1, 1, 1	3, 2, 3, 2, 3
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	18	25
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	Develop Small Section of Software (26%) Unit Testing (26%) Travelling To/From Work (11%) Provide Support (6%) Informal Discussion With Colleague(s) (5%) Line Management Duties (5%) Lunch (4%) Configuration Management (2%) Define Requirements and Specifications (2%) Informal Discussion With Manager(s) (2%) Meeting With Colleague(s) (2%) Obtaining Understanding of Client's Current System (2%) Reading/Replying/Composing Work E-Mail(s) (2%) Receiving Phone Call(s) (2%) Coaching Others (1%) Making Phone call(s) (1%) Personal E-Mail(s) (1%) Project Management Duties (1%)	Develop Small Section of Software (11%) Travelling To/From Work (11%) Reading/Replying/Composing Work E-Mail(s) (10%) Coaching Others (9%) Create Testing Schedule (9%) Provide Support (9%) Recruitment and Selection (6%) Meeting With Colleague(s) (5%) Informal Discussion With Colleague(s) (4%) Lunch (4%) Planning/Timetabling Work for the Day (4%) Research Possible Technical and Design Approaches (3%) Configuration Management (2%) Prepare, Review, Approve and Check Documentation (2%) Meeting With Line Manager (2%) Making Phone Call(s) (2%) Receiving Phone Call(s) (2%) Training for Users (2%) Informal Discussion With Manager(s) (1%) Line Management Duties (1%) Meeting With Client(s) (1%) Personal E-Mail(s) (1%) Personal Phone Call(s) (1%) Planning for Meetings (1%) Produce Proposals for Systems (1%)

APPENDIX 21: WORK DIARIES FOR SPECSOFT PT2 TEAM MEMBERS

ROLE LEVEL/TENURE	Software Section Junior/Fourteen Months	Software and Consulting Sections (Contractor) Junior/Two and a Half Years
RATES OF INTENSITY	2, 2, 2, 2, 2	2, 2, 3, 3 (off sick for one day)
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	11	5
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	Task Analysis (23%) Provide Support (18%) Develop Small Section of Software (15%) Travelling To/From Work (15%) Informal Discussion With Colleague(s) (7%) Lunch (7%) Database Administration (3%) Meeting With Colleague(s) (3%) Training for Users (1%) Training (for yourself) (1%)	Develop Large Section of Software (73%) Travelling To/From Work (14%) Lunch (7%) Planning/Timetabling Work for the Day (4%) Fixing Faults (2%)

APPENDIX 22: WORK DIARIES FOR SPECSOFT PT2 TEAM MEMBERS

ROLE LEVEL/TENURE	**Software and Consulting Sections Senior/Seven Years**	**Software and Consulting Sections Average/Three Years Four Months**
RATES OF INTENSITY	**1, 2, 1, 2 (on holiday for one day)**	**2, 2, 4, 3, 2**
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	**15**	**23**
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	**Provide Support (25%) Develop Small Section of Software (9%) Reading/Replying/Composing Work E-Mail(s) (8%) Informal Discussion With Colleague(s) (7%) Lunch (7%) Personal E-Mail(s) (7%) Travel to Client Site(s) (7%) Travelling To/From Work (7%) Meeting With Colleague(s) (5%) Task Analysis (5%) Unit Testing (5%) Obtaining Understanding of Client's Current System (2%) Planning/Timetabling Work for the Day (2%) Research Possible Technical and Design Approaches (2%) Tea Break (2%)**	**Reading/Replying/Composing Work E-Mail(s) (36%) Travelling To/From Work (9%) Provide Support (6%) Informal Discussion With Colleague(s) (5%) Maintain and Update System (5%) Travel to Client Site(s) (5%) Meeting With Line Manager (4%) Planning for Meetings (4%) Prepare, Review, Approve, Check Documentation (4%) Lunch (3%) Making Phone Call(s) (3%) Plan Systems Installation (3%) Recruitment and Selection (3%) Site Acceptance Testing (2%) Coaching Others (1%) Develop and Budget Cost of New System (1%) Install Software/Systems (1%) Meeting With Client(s) (1%) Meeting With Colleague(s) (1%) Personal E-Mail(s) (1%) Personal Phone Call(s) (1%) Produce Proposals for Systems (1%) User Acceptance Testing (1%)**

APPENDIX 23: WORK DIARIES FOR INSOFT BUSINESS ANALYSTS

ROLE **TENURE**	**Business Analyst** **Ten Years**	**Business Analyst (part-time)** **Thirteen Years**
RATES OF INTENSITY	**2, 2, 3, 3, 3**	**2, 1, 1, 1, 1**
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	**15**	**20**
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	**Reading/Replying/Composing Work E-Mail(s) (31%)** **Meeting With Colleague(s) (14%)** **Planning for Meetings (11%)** **Making Phone Call(s) (9%)** **Travelling To/From Work (8%)** **Lunch (6%)** **Using Messenger (6%)** **Receiving Phone Call(s) (5%)** **Informal Discussion With Colleague(s) (2%)** **Prepare Documentation (2%)** **Recommendations for Future Development (2%)** **Line Management Duties (1%)** **Personal Phone Call(s) (1%)** **Tea Break (1%)** **Training for Users (1%)**	**Reading/Replying/Composing Work E-Mail(s) (19%)** **Making Phone call(s) (8%)** **Meeting With Colleague(s) (8%)** **Obtaining Understanding of Client's Current System (7%)** **Using Messenger (7%)** **Meeting With Clients (6%)** **Planning for Meetings (6%)** **Tea Break (6%)** **Travelling To/From Work (6%)** **Informal Discussion With Colleague(s) (5%)** **Planning/Timetabling Work for the Day (5%)** **Prepare Documentation (5%)** **Meeting With Manager(s) (4%)** **Task Analysis ((3%)** **Define Requirements and Specifications (2%)** **Lunch (2%)** **Meeting With Line Manager (2%)** **Document and Code Reviews (1%)** **Meeting With Users (1%)** **Receiving Phone Call(s) (1%)**

APPENDIX 24: WORK DIARIES FOR INSOFT BUSINESS ANALYSTS

ROLE **TENURE**	**Business Analyst (Indian Offshore Contractor)** **Seven and a Half Years**
RATES OF INTENSITY	**2, 2, 2, 3, 2**
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	**20**
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	**Reading/Replying/Composing Work E-Mail(s) (30%)** **Using Messenger (17%)** **Receiving Phone Call(s) (16%)** **Travelling To/From Work (7%)** **Meeting With Colleague(s) (4%)** **Lunch (4%)** **System Test (4%)** **Meeting With Manager(s) (3%)** **Recommendations for Future Development (3%)** **Meeting With Line Staff (2%)** **Obtaining Understanding of Client's Current System (2%)** **Planning for Meetings (2%)** **Coaching Others (1%)** **Document and Code Reviews (1%)** **Maintain and Update System (1%)** **Meeting With Line Manager (1%)** **Personal Phone Call(s) (1%)** **Planning/Timetabling Work for the Day (1%)** **Recording and Updating Observation Reports (1%)** **Task Analysis (1%)**

APPENDIX 25: WORK DIARIES FOR INSOFT DEVELOPERS

ROLE TENURE	Developer (Indian Offshore Contractor) Two and a Half Years	Senior Developer Four Years
RATES OF INTENSITY	4, 3, 3, 2, 4	1, 1, 2, 1, 2
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	19	12
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	Provide Support (14%) Travelling To/From Work (14%) Develop Specific Section of Software (7%) Making Phone Call(s) (6%) Meeting With Colleague(s) (6%) Write Progress Reports (6%) Lunch (5%) Meeting With Client(s) (5%) Meeting With Manager(s) (5%) Project Management Duties (5%) Reading/Replying/Composing Work E-Mail(s) (5%) System Test (4%) Training (for yourself) (4%) Keeping Up-To-Date With Technology/Advances (3%) Configuration Management (2%) Recommendations for Future Development (2%) Test Program Modules (2%) Create Testing Schedule (1%) Recording and Updating Observation Reports (1%)	Fixing Faults (21%) Provide Support (21%) Reading/Replying/Composing Work E-Mail(s) (14%) Receiving Phone Call(s) (10%) Develop Specific Section of Software (8%) Lunch (7%) Re-Testing of Bugs (5%) Allocate Sections for Development (3%) Travelling To/From Work (3%) Meeting With Colleague(s) (3%) Write Progress Reports (3%) Meeting With Manager(s) (2%)

APPENDIX 26: WORK DIARIES FOR INSOFT SOFTWARE ENGINEERS

ROLE TENURE	Software Engineer Nineteen Years	Software Engineer Twelve Years
RATES OF INTENSITY	2, 2, 3, 3, 3	1, 2, 1, 2, 3
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	8	12
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	Develop Specific Section of Software (40%) Fixing Faults (20%) Tea Break (10%) Travelling To/From Work (10%) Lunch (8%) Keeping Up-To-Date With Technology/Advances (5%) Personal E-Mail(s) (5%) Informal Discussion With Colleague(s) (2%)	Fixing Faults (20%) Provide Support (19%) Travelling To/From Work (13%) Lunch (11%) System Test (11%) Reading/Replying/Composing Work E-Mail(s) (8%) Produce Proposals for Systems (5%) Develop Specific Section of Software (3%) Obtaining Understanding of Client's Current System (3%) Re-Testing of Bugs (3%) Personal E-Mail(s) (2%) Tea Break (2%)

APPENDIX 27: WORK DIARIES FOR INSOFT TESTER AND CONSULTANT

ROLE TENURE	Tester (part-time) Twenty-Three Years	Consultant (On-Site Indian Contractor) One Year
RATES OF INTENSITY	3, 3, 2	2, 1, 1, 1, 1
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	8	20
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	System Integration Testing (49%) Travelling To/From Work (18%) Reading/Replying/Composing Work E-Mail(s) (15%) Plan Timescales and Resources (6%) Informal Discussion With Manager(s) (3%) Lunch (3%) Meeting With Colleague(s) (3%) Receiving Phone Call(s) (3%)	Reading/Replying/Composing Work E-Mail(s) (19%) Develop Specific Section of Software (17%) Informal Discussion With Colleague(s) (10%) Meeting With Colleague(s) (7%) Document and Code Reviews (6%) Re-Testing of Bugs (6%) Using Messenger (6%) Fixing Faults (5%) Lunch (5%) Receiving Phone Call(s) (4%) Break (3%) Making Phone Call(s) (3%) Informal Discussion With Manager(s) (2%) Personal E-Mail(s) (2%) Research Possible Technical and Design Approaches (2%) Training (for yourself) (2%) Break Specifications Into Basic Elements (1%) Create Testing Schedule (1%) Personal Phone Call(s) (1%) Prepare Documentation (1%)

APPENDIX 28: WORK DIARIES FOR INSOFT TECHNICAL LEADS

ROLE TENURE	Architecture Technical Lead Twelve Years	Engineering Technical Lead Eight Years
RATES OF INTENSITY	2, 3, 3, 1, 2	2, 2, 2, 4, 4
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	22	22
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	Reading/Replying/Composing Work E-Mail(s) (16%) Research Possible Technical and Design Approaches (11%) Meeting With Colleague(s) (10%) Travelling To/From Work (10%) Making Phone Call(s) (8%) Develop Specific Section of Software (7%) Using Messenger (6%) Produce Proposals for Systems (5%) Informal Discussion With Colleague(s) (4%) Lunch (4%) Provide Support (4%) Tea Break (4%) Document and Code Reviews (2%) Plan Timescales and Resources (2%) Planning for Meetings (2%) Informal Discussion With Manager(s) (1%) Keeping Up-To-Date With Technology/Advances (1%) Receiving Phone Call(s) (1%) Recording and Updating Observation Reports (1%) Planning/Timetabling Work for the Day (1%) Task Analysis (1%) Write Progress Reports (1%)	Reading/Replying/Composing Work E-Mail(s) (24%) Using Messenger (14%) Receiving Phone call(s) (9%) Informal Discussion With Colleague(s) (8%) Meeting With Manager(s) (7%) Making Phone Call(s) (6%) Informal Discussion With Manager(s) (5%) Meeting With Colleague(s) (4%) Meeting With Line Staff (4%) Fixing Faults (3%) Lunch (3%) Research Possible Technical and Design Approaches (3%) Coaching Others (2%) Document and Code Reviews (2%) Prepare Documentation (2%) Define Requirements and Specifications (1%) Line Management Duties (1%) Personal E-Mail(s) (1%) Personal Phone Call(s) (1%) Planning for Meetings (1%) Planning/Timetabling Work for the Day (1%) Travelling To/From Work (1%)

APPENDIX 29: WORK DIARIES FOR INSOFT TECHNICAL LEADS

ROLE **TENURE**	**Support Technical Lead (On-Site Indian Contractor)** **Two Years**
RATES OF INTENSITY	**3, 4, 2, 3, 3**
TOTAL NUMBER OF ACTIVITIES FOR OVERALL WEEK	**22**
ACTIVITIES AND PERCENTAGES FOR OVERALL WEEK	**Making Phone Call(s) (19%)** **Reading/Replying/Composing Work E-Mail(s) (18%)** **Provide Support (15%)** **Receiving Phone Call(s) (8%)** **Travel to Client Site(s) (7%)** **Travelling To/From Work (5%)** **Lunch (4%)** **Meeting With Users (4%)** **Coaching Others (3%)** **Develop Specific Section of Software (3%)** **Meeting With Colleague(s) (3%)** **Keeping Up-To-Date With Technology/Advances (2%)** **Tea Break (2%)** **Database Administration (1%)** **Document and Code Reviews (1%)** **Fixing Faults (1%)** **Informal Discussion With Colleague(s) (1%)** **Meeting With Line Manager (1%)** **Prepare, Review, Approve and Check Documentation (1%)** **System Test (1%)** **Task Analysis (1%)** **Using Messenger (1%)**