Revisiting Technical Workers: Professional and Organizational Identities in the Software Industry

Abigail Marks
School of Management and Languages
Heriot-Watt University, Edinburgh

Dora Scholarios
Department of Human Resource Management
University of Strathclyde

Submitted to OLKC 2006 Conference at the University of Warwick
Coventry on 20th - 22nd March 2006

Introduction

There is a growing body of work which examines the nature of contemporary or 'new' professions and their relationship with knowledge work (e.g. Scarborough, 1999). For the most part, however, this work assumes homogeneity within occupational groups. Perceiving employees in this way not only ignores the complexity of many occupational groups, but also fails to acknowledge insights from the examination of traditional work. Despite many writers recognising that knowledge work is a rather loose categorisation (e.g. Alvesson, 2001), there have been few frameworks employed to disentangle its meaning and the divergent experiences of knowledge workers.

In this paper we take a group of knowledge workers - software professionals - and analyse their identity with their organization and profession. We aim to assess the extent to which software workers, divided based on skill level and entry qualifications are also divided in their perceptions of professional and organizational identity. Instead of seeing them as a uniform employment category, we follow Perrolle (1986) and assume that there is a division of labour in software work. We view this division as being analogous to established distinctions found between the elite design engineering role and the test engineers, draughter and technical authors in the Research and Development worksites of traditional electronics and aerospace companies (Barely & Orr, 1997; Smith, 1987).

Hence, we identify a divergent group of workers that can be distinguished by a number of inter-related variables including labour market position, education, skill level and work role. This parallels Meiksins and Smith's (1996) account of traditional engineering work. In software, at the one extreme, we have a group that possess the skills and characteristics usually associated with knowledge workers, with high-levels of educational attainment, interesting design projects, significant autonomy and high labour market value. In contrast, we have what may be described as low skilled employees who have fewer conventional qualifications, and who generally work with older programming languages or in support or testing roles.

The experience of these groups of software workers is explored in this paper from the perspective of the employee's own identification with the software profession and the organizations by which they are employed. Little work has been undertaken on the nature of professional identification for new occupations with most research tending to focus on traditional professions (Wallace, 1985). Although writers

concerned with new professions and knowledge work, have tended to propose the organization as the core agent for identity formation (e.g. Alvesson, 1985; Alvesson, 2001; Kunda, 1992), more recent work has highlighted the limited ability of organizations to influence any strong degree of organizational identification (Beirne, et al, 1998; Marks & Lockyer, 2005).

We begin the discussion by developing the debate on the employment status and categorisation of software workers in relation to traditional technical workers. This is followed with a consideration of the different salience of organizational and professional identity for the two groups of employee. We locate this discussion in a social identity framework, but acknowledge the limitation of social identity theory (e.g. Tajfel, 1972) in its weak consideration of individual interests. The social identity of software workers is explored via the examination of three research propositions which are applied to an empirical study which focuses on the experiences and perceptions of software developers, software testers, software analysts and software programmers[1]. The data are drawn from employee questionnaires and semi-structured interviews in five research sites in Scotland.

Software professionals – knowledge workers or craft workers?

New occupations such as software work are often associated with the increasing demand from western economies for knowledge work and knowledge-based organizations (Drucker, 1993; Scarborough, 1996). Indeed, software developers are frequently seen as typical of knowledge workers (Reich, 1991; Scarborough 1999). As Barrett (2005; p.3) notes, 'these workers are among those predicted to be the future aristocrats of the labour market given the centrality of knowledge as a commodity and characteristics of the contemporary economy'. However, as much as writers such as Ackroyd, et al. (2000) argue that software workers are the key occupation to examine in the future studies of knowledge work and knowledge workers, the reality of software work is often very different.

Many software development organizations survive only as minor players within a dynamic and rapidly evolving industry (Barrett, 2005). Software workers appear to occupy a more precarious position than once seemed. The perception of

[1] For ease of discussion, unless specifically required to separate occupational groups, the terms software workers, software employees and software professionals are used throughout the paper as generic desciptors for all participants.

affluent, upwardly mobile professionals earning high wages and working in modern and enlightened workplaces (Barrett, 2001) is somewhat questionable. On the one hand, software employees have more in common with skilled service workers – teachers, lawyers or accountants – than manual workers or engineers, as they work with their minds rather than their hands. However, software workers produce commodities and share characteristics of both industrial and service work.

It is this dual character of software work that has produced some debate as to the true nature of the work and occupational location. Most recent work maintains the view of software workers as symbolic analysts and knowledge workers (Barrett, 2005); Andrews, Lair and Landry (2005), for instance, describe software work as involving considerable discretion, creative freedom and technical autonomy. Others suggest that software work has more in common with Taylorised production (Perrolle, 1986). Kraft (1992), looking at 'Tech'- a large American high-tech corporation – found a significant division in software work. Whereas some employees were involved in highly creative development work, other employees were involved in lower-status support activities. This work included maintenance, testing and quality control and service work – functions often covered in traditional manufacturing work. Kunda (1992) viewed this division in software work as reflecting disparities in qualifications and entry routes. While most of Tech engineers were university graduates, there were some - frequently computer programmers and software engineers - who had no formal training in either engineering or computer science. This latter group of employees were either self-taught or the recipients of company-sponsored training. The university graduates, for the most part, were involved in the more creative 'glamorous' work, whereas the self-taught tended to be engaged in the lower level support work. As such, the title of engineer can be both a professional title and an organizationally defined employment category and this differentiation reflects both entry route and work content.

Despite, being viewed as the vanguards of the new economy, employment patterns in software work are by no means straightforward. As Perrolle (1986) notes, the expansion of software development work in the 1960s and 1970s led to a division of labour. Highly skilled systems analysts produced flowcharts, whilst clerically trained high school graduates produced code. What was emerging during this period was a hierarchical partition of work, similar to that in blue collar industries. Whilst early software work required individuals to undertake all aspects of their trade

(system design, programme design, writing code, testing and debugging), from the 1960s onwards, these processes were broken up into distinct forms of work (Kraft & Dubnoff, 1986).

Although Barrett (2005) argues that many software development firms exist as marginal players in a dynamic and rapidly evolving industry, we argue – equally importantly - that many software workers are also marginal players in the process of software production. We propose that there is a differentiation between software employees based on qualifications and entry routes. As with traditional engineering and technical workers, software workers at the extreme, follow what Meiksins and Smith (1996) call *craft* or *managerial* career routes. That is, *craft* forms are closer to the traditional apprenticeship structure, where training occurs either through short courses, mentoring by senior staff and by self-directed learning. At the craft end, we suggest that employees are likely to have no higher level qualification or possess a higher level qualification that is unrelated to software work. On the other hand, *managerial* forms are associated with a university based route for the acquisition of expertise and qualifications which are institutionalized through specialist associations. Although we view this as a simplification of the range of skills and backgrounds in software work, Meiksins and Smith's types, broadly fit the extremes of the histories of software workers.

Those employees that follow the traditional or craft-based skills and training route are more likely to possess organizationally specific skills and therefore be less employable in the open labour market than those with occupationally specific skills. This is because those with higher qualification in an IT based subject have more generic and externally recognisable skills and credentials. Moreover, if we base our assumption on evidence from traditional technical work (e.g. Smith, 1987) it follows that those software workers with the occupationally specific skills are also more likely to partake in more elite project and higher skilled work. These ideas form the basis of our first proposition:

> **Proposition One: A division exists in software work which is determined by the entry level qualifications of the worker. Those employees that have higher level, IT based qualifications will undertake more complex, cutting edge work.**

Software workers and loci for identification

In Britain, regardless of skill level or experience, software engineers tend to be clustered as one 'catch all' group, based on similarities in terms of unionization and weak representation in management hierarchies (Lee & Smith, 1992). As Perrolle (1986) argued, job titles such as 'systems manager', 'software designer' and 'analyst' tend to be used as generic titles applied to different skill levels and skill sets which often do not correspond to wages, working conditions and work content.

Collins (1979) interprets this as a move in the middle classes towards a 'consciousness of formalism'; that is, the material realities of the work experience are less important than the relative values of cultural currency. In a society that is increasingly concerned with self and status, the meaning, and projected meaning of work has changed. What one actually does is being exchanged by what one exhibits in terms of symbolic status. Even if employees are performing Taylorised, low skilled maintenance and testing work, the fact that the work at least appears intellectual, rather than manual, will sufficiently enhance their symbolic status (Perrolle, 1979). Ironically, whilst there may be a de-skilling of software work (at least for some), these software workers are still viewed as knowledge workers and demand the associated material rewards.

Hence, we would expect both the division of labour in software work and its symbolic status to have an impact on the professional and organizational identity of software employees. To examine this we broadly apply a social identity framework. Indeed, much of the traditional work explaining identity in organizations and to occupations employs social identity theory as an explanatory theoretical framework (e.g. Turner, 1982; Ashforth & Mael, 1989). At one level, social identity theory (SIT) provides a basic structure for the understanding of identification with a collective; it views identity as part of a person's sense of who they are, which is associated with an internalised group membership, and can be logically separate from personal identity. Social identity theory (e.g. Ashforth & Mael, 1989; Turner, 1982, 1984) considers how individuals' feelings of self-worth are reflected in the status of the collective and suggests that the key function served by membership of a collective is the provision of members with information that aids in their efforts to develop and maintain a favourable self-concept (Tyler & Blader, 2001). The theory also holds that individuals self-categorise in order to reduce uncertainty, as uncertainty reduction, specifically about matters of value that are self-conceptually relevant, is a core human motivation

(Hogg & Terry, 2000). Certainty provides confidence in how to behave and what to expect from a particular social situation as can be seen in the importance afforded to impression management in career strategy through presentation skills and techniques (Feldman & Klich, 1991) and efforts to convey an image consistent with organizational or societal expectations of a professional role (Goffman, l959).

'The most developed conceptualisation of organizational identification, sees it as a form of social identity whereby a person comes to view him or herself as a member of a particular social entity, the organization' (Bergami & Bergozi, 2000: 557). This is described as occurring through a cognitive process of self-categorization where the individual forms self-categories with organizational membership and similarities with others in the organization, along with dissimilarities with others in different organizations. By self-categorising, individuals undertake a cognitive process which highlights similarities and differences in a hierarchical classification of increasing levels of abstraction and inclusiveness (Turner, 1984). This leads to a cognitive connection between the definition of an organization and the definition that a person applies to him or herself (Dutton et al., 1994). This self-awareness of belonging to an organization is viewed as one way in which individuals achieve social identity (Tajfel, 1978).

Categories are assumed to be both salient to an individual's sense of self and the sector in which they are working (Alvesson, Robertson & Swan, 2001). Hence, in the software sector, a self-categorization of 'cutting-edge' or 'innovative' would be appropriate. However one of the limitations of SIT is that it makes assumptions about the natural fit between describable social categories and logical progression to identification. SIT ignores the role of individual actors and individual motivations, specifically that of self-interest. As much as an organization may provide objective basis for identification - specific categories that are of importance to employees' self-definitions and self-esteem - there is still a possibility that they may not fully identify with the organization. There may, in reality, be little need to identify.

Alvesson, et al. (2001) developed the idea of an elite social identity, which they found pervasive in knowledge-intensive companies, including software organizations. This idea of an elite social identity was based on the exclusivity of organizational membership and was viewed as being constructed and reinforced by a number of criteria including size, recognition, profit, successful products and expert

rankings. Elite identity was also created based on more esoteric dimensions which were less publicly recognised, and included specialisation and professionalism.

For the software organizations examined within the present study (all well thought of and successful in their own ways), we made no assumptions regarding the existence of an elite organizational identity. We do however suggest that the division of labour and in entry qualifications for software workers will be reflected in differences in organizational identity. As we touched upon earlier in this paper, the division of labour is likely to impact upon the relationship with and attachment to the employing organization. The more *traditionally qualified workers* (university education, IT based degrees) are less likely to possess a strong organizational identity as they are less reliant on the organization for their occupational identity or feelings of self-worth. There is no self-interest in having a strong identity with their organization. Their increased employability will result in a more transactional relationship with their organization than for employees that have less traditional backgrounds. Those with less specific or no IT qualifications we believe, will have a greater attachment to their organization as their 'elite' status as a software professional is inextricably tied to their employing organization. Those employees without a strong educational background in software or IT need to remain in the organization as it provides access to work which is, at least perceived to be, high status (Haslam, 2001). For employees who have backgrounds and experiences which are more similar to traditional craft workers, there is unlikely to be any access to an 'elite' professional identity without an association with their employing organization. These ideas form the basis of our second proposition.

> **Proposition Two: Those employees without an IT-based qualification will have stronger organizational identity than employees who have a higher-level qualification in computer science, software engineering or electronic engineering.**

The potential for organizational identity to be based on entry qualifications takes account of the idea of employee self-interest. As previously stated, this concept is frequently absent in traditional discussion of identity. Associated theorists such as Goffman (1959) and Jenkins (2004) accept the role that interests play in determining behaviour. This is demonstrated in Jenkins' (2004) work on the interaction order, where individuals reflect on their external image and continually regulate their self-image and public image to remake their social identity. Image is amended in the best

interest of the individual. The omission of self-interest in SIT is most clearly illustrated in its lack of clarity in the relationship between social identity and personal identity. Although self-categorization theory attempted to remedy this, the social identity approach still suggests a boundary between personal and social identity. Yet, as Brown (1997) argues, the two identities must be interrelated. He believes that one's social identity will affect someone's individual character. Moreover, personal identity must contribute to one's choice of social identities (again, membership of an organization or profession) or the way in which the social identity is acted out.

Hence, the social identity approach, specifically in terms of identity salience, views identity in part as a conscious decision making process. It is the self-conscious decision making process that is fundamental to the idea of interests that has been omitted from traditional social identity theory. Whereas we believe it is principally in the interests of the non-technically educated software workers to strongly identify with the organization, we consider that it is in the interests of all software workers to identify with the profession. Identity with software work as a profession provides what Perrolle (1986) describes as cultural currency. For this group, professional identity is a process of impression management and self-interest, in that software employees 'have joined the ranks of the young, upwardly mobile, white-collar professionals' (Barrett, 2001, p.2). Based on Perrolle's (1979) assumptions, the importance of symbolic representation and association with software work will produce a high degree of professional identity for all those involved in software work

Our third and final proposition, therefore, is concerned with the relationship between skills and qualifications and professional identity. Whereas Alvesson, et al. (2001) recognize organizations as a source of elite identity, we view this notion of 'eliteness' as being constructed from the profession. Alvesson, et al. (p.7) argue that being elite calls for some element of confirmation (even though this may well be weak, incoherent and contested). We believe that this confirmation comes from media depictions of highly creative and intellectual work. The dot.com generation and the internet boom had made identifying with this occupational group highly desirable. Whether or not employees are engaged with this high end work is irrelevant. There is still a high degree of self-interest in being associated with hi-tech work.

> **Proposition Three: All software workers will possess a high degree of professional identity regardless of their skill type, skill level and entry qualification.**

Methodology

The aim of our empirical study was to understand how differential work roles and technical qualifications affect professional and occupational identity. Although survey methods are appropriate for ascertaining formalized work role and academic attainment, they are not ideal for understanding the nature of identity. Identity may not always be explicit, even for the individual involved (Tyler & Blader, 2001) and its socially constructed nature makes it difficult to reduce to simple measurement. Thus, in order to understand the complexities of identity we augmented survey research with exploratory and semi-structured interviews.

Research sites

Five Scottish case study organizations were chosen - one independent software house employing 248 people (Omega), one software division of 275 people which was part of a large national telecommunications company (Beta), and three medium-sized to smaller independent firms (Gamma, Pi, and Lambda) employing 150, 50 and 20, respectively. These were selected to be representative of the range of employment contexts in the Scottish software sector, from applications development of bespoke systems and core products, to development of robotic tools, database integration, resourcing, testing, and client support (e.g. training, maintenance). Within each case study employee questionnaires and semi-structured interviews were conducted. Background data on company history, operating procedures, employment policies and staff characteristics also was gathered as part of an intensive process of case study analysis and observation involving four researchers in each of the five companies for approximately four months.

Employee questionnaire

Sample and respondent characteristics

The questionnaire was distributed to all employees in the five workplaces as part of the case study process with the researchers directly involved in distribution and collection. This process resulted in a high response rate to the questionnaire (mean = 69%). The single company where direct contact with employees was not possible because they worked a significant proportion of time on clients' premises (Gamma) returned a substantially lower response rate (25%) compared to all the other companies, where the rate ranged from 72% to 88%.

From the surveys returned, 25 were not used in the present analysis as they represented employees in a support or non-technical role. This provided a total of 303 useable questionnaires (see Table 1 for respondent characteristics). The organizations were representative of the sector in that they employed a combination of contractors (16%) and permanent employees (84% of sample); a majority of males (72% overall) although Omega, Pi and Lambda employed between 32 % and 41 % females; and a generally young workforce, with 73 % under 40. Employees in Beta, Omega and Gamma were located on more than one site, and those in Gamma and Omega frequently worked on client sites either permanently or for long periods of time. Employees working 'onsite' comprised 8 % of the total sample.

Measures

Organizational and employee characteristics. The questionnaire provided measures of gender; age (represented by an ordinal scale where 1=16-20, 2=21-30, 3=31-40, 4=41-50, 5=over 50); location of employee (working within employing organization or onsite at clients' premises); temporary staff/contractors (coded 0/1); part-time status (coded 0/1); agency employee (19 participants within two of the case study organizations were employed by Indian software organizations and located on a semi-permanent basis within the UK). We decided to categorize qualifications into three groups. After assessing the range of educational backgrounds of our participants it became clear that the nature of contemporary software work is more complex than traditional technical work. Hence, qualifications were coded as 'IT-specific' where the degree was in an occupationally relevant discipline e.g. software, electronic engineering or IT; 'IT-related' where degree qualification was in an alternative technical qualification e.g. physics, mechanical engineering; and 'non-IT qualification' where there was no qualification or a non-related qualification. Microsoft™ or Sun™ certification was treated as product knowledge rather than professional related[2].

Job characteristics: Five items measuring job control on a scale of 1 'not at all' to 5 'a great deal' were drawn from the Perceived Intrinsic Job Characteristics Scale (Warr, Cook and Wall, 1979), which included items on control over method of working (we used this to represent autonomy), opportunity to use own skills, and

[2] A post-graduate degree/diploma was coded IT-specific or IT-related depending on the title of the degree course and its relationship with the content of the participant's first degree.

amount of job variety. Also added were three questions on cognitive demand (Jackson et al., 1993), which included monitoring demand (i.e. the requirement to pay close and constant attention) and problem-solving demand (i.e. the requirement to diagnose and solve problems). On the basis of a principal components factor analysis with varimax rotation, these items were combined to form two mean composite measures: method control (α=.74) and cognitive demands (α=.73).

Professional identification: There has been limited evidence of empirical distinctiveness between commitment and identification (van Knippenberg & van Schie, 2000), hence three items measuring different dimensions of commitment were applied to identification with a profession, each using a scale of 1 'strongly disagree' to 5 'strongly agree'. Firstly, affective professional identification was measured using a single item from the Occupational Facet of Commitment Scale (Blau, 1985) ('If I could, I would go into a different occupation'). Secondly, perceptions of behavioural identification were measured using a single item ('I take an interest in current developments in the software sector') based on questions from The Use of Profession as Major Referent Scale (Hall, 1968; Snizek, 1972). Finally, the normative dimension of professional identification was examined using a single item ('I am proud to tell others that I am employed in the software sector') from Vandenberg and Scarpello's (1994) modification of the Occupational Commitment Questionnaire (Mowday, Steers & Porter, 1979).

Organizational identification: Organizational identification was measured using Allen and Meyer's (1990) affective commitment dimension which is an identification based component of organizational commitment. Six items, drawn from Allen and Meyer's original scale, were used (e.g. 'I feel a strong sense of belonging to my company' and 'I would turn down a job with a bit more pay in order to stay with this company') each using a scale of 1 'strongly disagree' to 5 'strongly agree'. Responses to these items were averaged to form a single score (α =.77).

Importance of IT skills and knowledge in present job: Due to the variety of software work, six items were used to gauge the importance of different skills and tasks in respondents' jobs. These were listed as: software programming, systems analysis, business analysis, testing, software design and user/application support, and were measured on a four point scale from 1 'Not too important' to 4 'Absolutely essential'.

Employee interviews and qualitative analysis

Exploratory interviews took the form of open discussions with key groups of informants (management, project leaders, employees) on the work process and issues of identity lasting an average of one hour each. A total of 73 semi-structured interviews/observations also were conducted. These interviews, which lasted from one to two hours, selected a representative group of employees in each company according to gender, age, job type and job/organizational level, and explored three themes in greater depth: (a) work and educational history (b) experiences of working in the present company (content and control of work, identity, identification to company/peers/job, social relations) and (c) work-life linkages and the future (perceptions of job risk/uncertainty, relative importance of work, perceptions of society/class/status). Between 14 and 20 semi-structured employee interviews conducted in the workplace were obtained for each company, except for the small start-up (Lambda) where three employees were thought to adequately represent the company. Each interview was either transcribed verbatim or produced as research field notes. Using this data of software workers' own accounts of their work, we categorised each of the interviewees according to the level of skills they applied in their work and their formal qualifications. The resulting two-by-two matrix shown in Figure 1 presents a summary of the findings from each of the five case study companies.

Results

Proposition One: A division exists in software work which is determined by the entry level qualifications of the worker. Those employees that have higher level, IT based qualifications will undertake more complex, cutting edge work.

As can be seen in Table 1, 42% of the sample had a degree or higher qualification related specifically to IT (e.g., Computing Science, Electrical/Electronic Engineering). An almost equal proportion, 41%, had no IT based or interrelated qualifications at all. The remaining 17% possessed higher qualifications in IT-related technical or scientific fields (e.g., physics, maths). For those groups without formal technical qualifications, IT expertise was developed though years of work experience and through certification courses in specific software. This pattern varied across the

companies slightly, with Beta and Gamma having a higher proportion of IT-specific qualifications and Pi a higher proportion of non-IT qualifications. This parallels the greater concentration of new development work in the former and single-product refinement (Pi's legal software) in the latter.

Insert Table 1 about here

In order to assess whether there were any work location or status factors that would affect work type and identity, we analysed the distribution of qualifications in the management body as well as by contractual status, and onsite/offsite location. The distribution across IT-specific, IT-related and non-IT qualifications was the same for those with and without managerial roles in their respective companies; that is, management were no more likely to possess an IT-specific qualification ($\chi^2(2)=.79$, p=.674). Examining only the two larger companies (Beta and Omega), where there was a sufficient number of observations, there was no evidence either of a difference in qualifications by contractual status, with contractors and permanent employees in both companies showing the same variation in the pattern of qualifications ($\chi^2(2)=1.27$, p=.530). Similarly, looking only at Omega's onsite versus offsite workers, the same distribution of qualifications again was evident for both groups ($\chi^2(2)=2.98$, p=.225).

However, moving to Table 2, it is clear that there are substantial variations in the skill types and levels required between the three 'qualifications' categories. Across all the companies, staff with IT-specific qualifications reported greater importance for the more creative and skilled dimensions of software work, such as software programming, systems analysis and software design. In addition, for each of these job areas, there was a clear and statistically significant gradation in importance, with the higher IT-specific qualified group providing the highest ratings, followed by the IT-related group and finally the non-IT group. There was no difference in the ratings across qualification groups for the less specialized dimensions of work (user support and business analysis), but the non-IT qualified group did rate user support, testing and systems analysis slightly more important than the more technical areas (programming and design). We also found that younger workers (under 30 years of age) tended to be focused on programming and testing work, regardless of qualifications, and that the importance of business analysis increased with age. Age (which here we take as representative of career stage) was not a factor in determining

the importance of design work or systems analysis, which was divided only in terms of qualifications.

Analysis of the more generic job characteristics variables – method control and cognitive demands - showed no difference between scores for the degree of cognitive demands required, suggesting that regardless of qualifications, all these software workers occupied roles requiring a high degree of cognitive complexity. Intriguingly, though, those with non-IT qualifications indicated significantly higher degrees of method control than the other two 'more IT-qualified' groups ($F(2,281)=5.11$, $p<.01$). This means they rated their jobs as providing them with more control over how they went about their work, the amount of job variety and opportunity to use their skills, all of which may indicate the less technically specialised nature of the projects they were working on.

Thus, the survey data suggests that software workers can be differentiated by the types of work they perform and formal IT-related qualifications, rather than managerial or contractual status, may explain this distinction. Workers with degree level IT-specific qualifications tend to perform the high-end work roles involving development and design in addition to less technical functions, such as user support and business analysis.

Insert Table 2 about here

Due to a much smaller sample, we simply divided the qualitative group into IT and non-IT employees, based on their entry qualifications. The two-by-two matrix shown in Figure 1 presents the numbers of interviewees in each of the five case study companies who were in line with Proposition One (those in quadrant 1 and 3) and those who were not (those in quadrant 2 and 4). The qualitative data mirrors the survey findings. Our group of interviewees were evenly divided in terms of formal qualifications (39 with degree level IT qualifications and 34 with no IT or IT -related degree background). Those with formal technical qualifications were more likely to be employed in higher-end, technical roles (quadrant 1) while those without an IT qualification were more likely to be employed in non-technical functions (quadrant 3). Only 19 employees (14 higher qualified employees in lower level work; 5 lower skilled employees in higher level work) did not fully support Proposition One.

Insert Figure 1 about here

The job titles given in parentheses in each quadrant illustrate the nature of these different roles. For those we classified in the high-end skill quadrants (1 and 4), there

was a strong sense of transferable, core skills which would possess high value across the occupation and in different businesses.

> 'I've used the same skills for every project I've worked on. So while you've maybe got a bit of a learning curve on a different programming language or a different technology to me it is all one in the same …I know I'm capable enough to learn [a new programming language or technology] quickly so it doesn't bother me at all' (Beta, project manager, interview 6).

As well as programming, though, many of those employed in higher level work were involved in the entire life-cycle of a project and had developed project management, design, and commercial skills. A graphics applications developer described his work as requiring 'problem-solving and precision' which he gained from a grounding in Maths: 'we start with a set of requirements…doing the coding first to develop the application' then 'working with two designers to develop forms and screens' for the user interface (Gamma, interview 14). Team leaders and project managers may still be involved in development, but also described their roles as involving 'estimating, scheduling and more interaction' (Beta, interview 13) or solving problems like 'volume and bottlenecks' (Beta, interview 15). The three quotations below describe the work of high-level employees in different companies: the first is an Omega employee, who was one of the few high-level technical people the company were prepared to keep on without progressing to a non-technical role; the second is how one Beta manager described the need for a business orientation as well as programming skills; and the third describes how one project manager in Gamma 'treaded the divide' between development and deployment.

> 'John is a systems architect at the moment and he will be looking very generally at technology. He won't run a team of programmers…. He will be a steering influence on which technologies we concentrate and what's going to be the next big thing and that will filter down to how people are trained and the kind of sales things we pursue ultimately.' (Omega, database consultant, interview 2).

> 'A lot of the skills that people pay us for aren't just the technology skills, you can go out and get you know 1000 jammer programmers but if you want to actually bring them onto a project in say a bank the best jammer programmers are ones who have worked at other banks because they have been through the terminologies they know the best ways of doing certain

programmes that are common to every bank and they might know some nice inside info about their competitors who they used to work (Beta, manager, interview 2).

'Developers [can] lose sight of what's really required in the business side. You can get kind of blinkered doing your little piece …I see the ground breaking stuff they are doing and I get a good feel for what's happening out there' (Gamma interview 11).

The interviewees we classified in quadrants 2 and 3 were in purely non-technical roles, often customer facing or involving interaction between developers and users.

'I have to make something as simple as possible [for the developer] or actually debug the customer's product …a lot of it [time] is actually spent on talking with …managers and other developers about exactly who's doing what and where' (Gamma, support engineer, interview 10).

'The technical people in our deployment team.. if you like they are the gurus of what we sell... their technical knowledge is higher but the business knowledge isn't as high so that's where we come in' (Gamma, deployment project manager, interview 13).

Despite many of the non-IT staff working in more front line roles, Figure 1 demonstrates that it is possible for developers to progress to high-level development and design work without a technical degree. This position is illustrated by the 14 interviewees classified in quadrant 4 (and not conforming to Proposition One). As described below, there appear to be three main criteria for such advancement. The first involved, again, a formal educational route by undertaking either a technical Masters during employment or other forms of certification. The second concerned substantial investment in training by the employing organization, and the final criterion centred on an individual's motivation to network and seek advancement.

A number of employees either had taken career breaks or part-time Master's or diploma courses in IT software whilst working in the industry. Advancement was dependent on the other two criteria – access to training or experienced colleagues, and networking. A Beta developer with a degree in Librarianship and postgraduate conversion course described her experience with in-house training and development.

'You get all the technical training that you want … Also just experience wise it's fantastic because you get to work with so many people, I mean when I first started the person I sat beside had been in Beta for 32 years. He started as an

apprentice when he was 16. You can't help learning from somebody like that' (Beta, engineer, interview 3).

This development route appeared to be more likely in the case of the larger companies with more diverse product ranges and the need for new products, and especially in Beta, where it was acknowledged that the infrastructure existed for progression and specialist training. A Gamma engineer who had progressed through Beta's graduate training (himself a Computing Science graduate) described the contrast between the specialisation in Beta compared to the informality and generalist needs of the smaller Gamma.

> 'When I first came here I was basically a software engineer or programmer if you like but very quickly, mostly I think due to the very small size of the company, at the time anyway, with very few people around I ended up doing lots of things which partly I wasn't really qualified for and certainly weren't part of my job description such as I understood it when joining' (Gamma, male manager, interview 11).

Further examples of Beta's approach are provided by the following interviewees: 'they honed the skills that I'd got from University to make them applicable in the workplace (Beta, applications support analyst, interview 10); 'when I started in the Centre, they tried to move you around projects to give you a kind of wider focus' (Beta, engineer, interview 4); 'one of the very good things they do is rotate people through roles and projects' (Beta, team leader, interview 8); 'they treat them [the younger guys] quite well in terms of training courses…. for the first year or so, they work around them….on their weaknesses and try to turn them into like more rounded employees …with the pace of change in technology Beta needs to keep their own staff up to date so it's in their own interests to train them' (Beta, project manager and contractor, interview 2).

In the other companies, valued skills and training in new technologies and languages was tailored around business need or expected to be self-taught. An implementations team leader in Pi commented on his preference for non-IT specialists who were needed if the business was to survive, and how 'training courses have to be tailored to individual clients' needs' (Pi, unstructured interview). The same was true in Omega - 'It's hard to get training at the expense of chargeable work, so if you are busy, you'll just work' (Omega, team manager, interview 2); 'they train you when there's an opportunity for us to use those skills (Omega, engineer, interview 1). This

was recognised as an increasingly short-termist strategy: 'there will come a point where those skills are just played out and nobody wants that kind of system developed in that language any more or whatever' (Omega, interview 2). Management, also, did not receive technical training in Omega: 'they think you don't need to be technical but we do..because we are managing technical people' (Omega, interview 9). Another Omega employee suggested that the company increasingly employed people with knowledge in specific languages rather than invest in training, especially for older people.

> 'They never took me on to train me there was no question of that, certainly not at my age and even probably somebody younger would have difficulty getting in and getting trained for the job they are going to do ….When I left university … people were prepared to take you on the strength of your academic success or whatever and you could take sort of 3-6 months to get up to speed with the specifics of the language' (Omega, engineer, interview 3).

Even in Beta, though, it was individual pro-activity and networking which came to the fore in order to progress into higher-end work. This is illustrated by an engineer who was able to overcome the initial disadvantages of having only a Business degree.

> When I first started …..they were under a lot of pressure, the S team, and I didn't really get the sort of …..encouragement is maybe not the right word….they just didn't have enough time basically to develop me and I think a lot of prior knowledge was assumed. ..I don't have the programming background…. so I was sort of thrown in at the deep end at the start and I really began to flounder …I just began to stagnate and I just felt myself on a downward spiral for about 3 or 4 months. For the first couple of years, I was not getting any decent work to do, probably because they could tell I wasn't happy. A never ending circle type thing. So I kicked myself up the bum, started really being a bit more pro-active myself, I accept responsibility for it as well. It is probably partly my fault (Beta, interview 7).

The developer described earlier with the Librarianship degree was assertive in identifying high prestige projects to which she wished to be assigned: 'part of it is just chance you speak to the right people at the right time. My Personal Development Manager was driving the external ventures project so I was able to go right to him and say ok I'm interested so … "either move me or I'll move myself"' (Beta, interview 3).

Access to training and good work in Omega seemed to be confined to the 25% or the 'elite' (Omega, placement student, interview 16) and clearly was felt to be unnecessary for those working on standard products and at the later stages of their career, as expressed by one senior systems builder: 'We don't get much [training]. At this stage in my career, unless it's something new that I am going to be doing… I don't really think I need it' (Omega interview 13). And in Gamma, one senior manager acknowledged that much of personal development in software had to be self-motivated: 'you can do the same job for twenty years without learning a new trick, but good staff are extremely good at taking on new skills and avoiding losing interest' (Gamma, Head of Development). Gamma's business development manager, with a non-traditional background in banking, commented on the catch-22 of this situation: 'the organization expects people to be self sufficient and look out for work and create their own work and there is a certain amount of freedom within the organization to actually develop your own skills which I think is good – but … if people are not on a project or not visible they tend to get left behind'.

The sum of this evidence shows how those with formal, degree-level IT qualifications tend to be hired for and progress into the development and design areas of software work, sometimes combined with project management responsibility. As a profession, though, software is represented by an equal number of non-standard entrants without such qualifications. Those who have progressed from this position to the higher-level roles were more likely to have benefited from the structured systems in place for hiring and developing skills for the diverse product range of a company like Beta, rather than being constrained by the business-driven approach to training typical in the smaller companies. In all companies, access to training or progression through work on high-level projects also seemed to be contingent on individuals' own pro-activity with respect to learning new technologies and self-promotion.

Proposition Two: Those employees without an IT-based qualification will have stronger organizational identity than employees who have a higher-level qualification in computer science, software engineering or electronic engineering.
The ratings of questionnaire responses in Table 3 show a moderate level of organizational identification for the overall sample, with a mean rating around the midpoint of the scale. Table 2 also shows that organizational identification for the

non-IT qualified group was significantly higher than for the other two groups ($F(2,215)=5.02$, $p<.01$). This provides preliminary support for Proposition Two.

However, when we controlled for other potentially influential factors which may explain organizational identification (age, gender, and management level) in a four-way analysis of variance, the significant effect of qualifications was replaced by the effects of age ($F(2,33)=3.85$, $p<.05$) and being in a management role ($F(1,33)=4.63$, $p<.01$). Qualifications still played a role in interaction with age ($F(4,33)=2.61$, $p<.05$). This interaction can be illustrated in terms of the ratings of organizational identification broken down by age in Table 3. This shows that levels of identification with the organization were highest for the older workers, especially those over 40, and qualifications had a significant effect only in the case of the 31-40 age group. Further analysis also showed that the effects of age were not significant for the non-IT qualified group – for these workers, age did not increase identification. However, for those with IT-specific or IT-related qualifications, the increase in identification between age 31-40 and over 40 was statistically significant ($t(37)=3.76$, $p<.01$ and $t(19)=3.57$, $p<.01$, respectively).

Insert Table 3 about here

Our analysis of the qualitative data expanded on this interplay between qualifications, age and organizational role. First, for those without formal IT qualifications, the company's investment in them through training or internal advancement seemed to be reciprocated with feelings of commitment, which may explain the higher overall organizational identification for this group. The following two non-IT qualified employees commented on their organizational commitment relative to the prospects of training and a senior management position:

> 'I do feel committed here. I feel they have invested a lot of training, time and development in me, so I do feel that I owe that back to the company' (Lambda, support assistant, interview 1).

> 'Yes, I feel committed. I do actually. I would like to be settled in a company but I would like to know that I was going somewhere with Omega as well, you know, that if I did like to become a business analyst that there was an opportunity there' (Omega, manager, interview 10).

Younger employees (those below 40) without an IT background described the relationship in similarly transactional, but more material terms.

> ‘They pay my wages, they pay my mortgage that’s important, people should realise that. I’m very keen to make sure this company moves from being a £20 million turnover company to £100 million to £150 million company it has to be done’ (Gamma, business development manager, interview 3).
>
> ‘I’m quite loyal to the company but we work hard for them and in return they do treat us very well. Last year we went to Barcelona for the weekend for the company conference, we’ve been to Gleneagles - I guess it’s all sort of material rewards …. I like working in a company where you can see that they appreciate the job you do’ (Gamma, business analyst, interview 5).

For older workers without an IT background, the organization took on greater importance as a source skill and formal certification. An Omega engineer who was a non-standard entrant to the profession (left school at 15, entered the software industry at 40) described the barriers to progression faced without formal qualifications and, as result of this perhaps, his own commitment to the organization.

> ‘They [companies] tend to offer you a cheaper rate than anybody else even though you might be able to do the same type of work - you never get the same rate as someone who’s been to university or has been doing it for most of their lives… they wouldn’t offer you the same salary as someone with an academic background. Maybe that’s why I’m still here I don’t know’ (Omega, software developer, interview 1).

Moving, secondly, to those with IT qualifications who expressed high organizational identification, this seemed to be based on a two-way investment and their reluctance to jeopardise this by leaving for an unknown employer, illustrating, in part, the emergence of a relational, rather than transactional, psychological contract. This was expressed most often by the ‘older’ employees, a finding which is in the same direction as the findings shown in Table 3 that older qualified employees were more likely to identify with the organization. Typical of this group was one Gamma engineer, who distinguished himself from the younger computer science graduates. Aged 31-40, married with children, with a science PhD (IT-related qualification), and with the company for five years, he described this relationship and his sense of identification in these relational terms.

> ‘I have invested quite a bit of my time and effort in the company. I know that whenever you change companies that it provides new opportunities but you are essentially throwing away a lot of your investment. So I do have that

commitment, I hope it's a two way thing. The company has invested in me and I've invested in the company and therefore the commitment's I think got to be there (Gamma, development engineer, interview 1).

In Beta, there were several senior managers (with IT qualifications) who had been with the company since the early days of the software centre and who identified strongly with its ethos, despite its recent voluntary redundancies and restructuring, like this 31-40 year old. This organizational loyalty also extended to older workers with non-graduate backgrounds who had been developed by Beta and who, in line with the findings for Proposition One, tended to perform the lower-level support functions.

'I would probably say I'm more affiliated with Beta, than a project because I have more than one project, and in the past it was very much about a Centre……that was recognised throughout the company. It had sort of a culture and a reputation.... a lot of other companies and even other sub-companies within Beta have a very macho style of management – "just do it, I don't care if it's impossible, just do it and I want it by next week" - that happens. So I think you don't get that in here, people are willing to consider people issues and try to develop people (Beta, project leader, interview 18).

'People on support work had been developed in-house compared to the graduates who belong to a more professional community. They were time served and had breathed Beta since the age of 16' (Beta, project manager, unstructured interview).

The final group of interest with respect to Proposition Two are employees who reported the lowest levels of organizational identification in Table 3 - the IT-specific qualified, 31-40 age group. Here, identification with the organization provided no value gain in terms of skill profile or career advancement, in contrast to the experience-hungry under 30s or the mature careers of the managerially-orientated over 40s. As such, many corresponded to the image of the mobile 'knowledge worker', highlighting the importance of frequent job shifting for enhancing skills and employability.

'I feel there is a certain advantage in moving from company to company because it keeps your skills present - you don't end up fulfilling your role but then become sort of old technology and you are the only person that knows it or you are one of the few people that know it …. So that is quite a good reason

for moving from place to place. I don't think you have got to stay too long at a company these days. Two years is an acceptable time..You move on'

(Omega, software engineer, interview 5)

Others with relatively well-developed careers sought good working conditions, as well as management trust and fair treatment. In Beta especially, once the benefits from the well-regarded training schemes had been reaped, mid-career employees seemed to become dissatisfied, and identify less, with the goals of the organization, particularly if they felt their skills were being abused, unrewarded or undeveloped. One example of lack of challenge for this group of workers is reflected in the following quotation.

'The biggest change I've had is this year, doing the web enabling call centre that has been new and quite exciting but even that, I've now looked at four or five products, they are just variations on a theme. So it's not stretching' (Beta, technical architect, interview 19).

Thus, we find broad support for Proposition Two in that organizational identification was highest amongst those without formal IT qualifications. In addition, organizational identification increased for employees over 40, even those with IT-specific qualifications, and was lowest for those qualified workers who could be categorised as in their mid-career stage (31-40). Explanations of these findings from the qualitative data suggest that for older and non-IT qualified employees the organization becomes an important source of personal identity.

Proposition Three: All software workers will possess a high degree of professional identity regardless of their skill type, skill level and entry qualification.

Table 3 shows the responses to three survey questions which asked about professional identification in software. Interest in the software sector, pride in the occupation and a desire to stay in the sector were all rated higher than organizational identification (all paired t-test comparisons were significant at the 95% level). This was the case across all categories of employees, but consistent with Proposition Three, these mean ratings did not vary significantly by qualifications.

There were a number of examples from the interviews that further illustrated identification to the profession over and above that to the organization. As we showed in the findings relating to Proposition Two, for the high skilled developers, the work

itself rather than the organization was the focus of their commitment. To take just one example from this group:

> 'I think in development most of us are committed more to the job than to the company because we are all in it because we enjoy programming and that's the first thing, the second thing is what company you work for and what sort of work you do' (Pi, software programmer, interview 10).

There were frequently close associations made between professional identity and professionalism. Not surprisingly, Beta provided the greatest number of references to professionalism as the term resonates throughout the formal structure of the organization. It is used in the grading structure of software workers – the managerial and professional grade - which all graduates receive whether or not they are managers. One project manager acknowledged that this separates them as belonging to a 'professional community' compared to those people on support work who had been developed in-house (Beta, unstructured interview). Despite this though, the 'traditional' electrical/electronic engineering technology of the company is still reflected in a sense of professional identity, even for those without formal qualifications. For example, this Beta 'long lifer' who had entered the company without any qualifications but progressed to a high-level role, expressed commitment more in professional terms, rather than organizational, embodied by attachment to the job and his flexibility to move with required change: 'the commitment's to Beta inasmuch as they are paying me to do what I do and I like to do the job to the best of my ability…. I think I'm flexible enough and have been in the past that when changes come in, I've moved with them' (Beta, applications project analyst, interview 11).

However, many interviewees, irrespective of qualifications status, made frequent reference to a sense of professionalism. Some consistent themes which emerged were pride in their work, ensuring client satisfaction, dealing with other IT professionals, and being judged by peers on the quality of their work. For some workers, there was also an awareness of wider career opportunities and professional salaries in IT tied closely to their skills and market value; as we argue in our analysis of Proposition Two, for the less qualified, the organization becomes a vehicle for adding to their skills set and employability within the wider market.

Taking, first, some examples of a general professional identity stemming from the job, software engineers were defined as 'your average professional person' (Beta, engineer, interview 4) – 'very professional … there's not a great deal of need to push

people to do their job. [People] take quite a lot of pride in it and are very keen to do a good job' (Beta, team leader, interview 5). Beta's senior manager talked of 'professional delivery' (Beta unstructured interview, Head of Software Centre) and another manager described what she liked most about the company in these terms: 'we are all fairly professional and you know when if you have a deadline coming you have to put in a few extra hours just to meet your deadline' (Beta, engineer, interview 14). Summing this up for Beta,

> 'There's a great deal of professional pride. …We try to recruit people who are driven, doing well at school/university, and want to continue doing a good job. These people will want to put things right even if it's not their job' (Beta, unstructured interview, Personal Development Manager).

This identification with a professional job was evident in all companies. The managing director of Omega stated 'we are IT professionals selling to IT professionals' (Omega, unstructured interview, management team), a sentiment echoed by the head of development in Gamma 'we are professionals with judgement (appraisal) by professionals' (Gamma, project development manager, interview 1). There were frequent references to 'doing a good job' and 'customer satisfaction':

> 'personal pride in it [the job] certainly matters to me and that means testing a system properly and …. you can say with a certain degree of confidence to the client, yes I believe this will work' (Omega, tester, interview 7).

The second area of interest regarding professional identity was the awareness of a professional community. Those in the high status roles, and, as we have already seen, those more likely to possess formal qualifications, were acutely aware of their market potential and talked in terms of the professional community to which they belong. The following account was typical of this group.

> 'I try to keep in good touch with my peer group from university days, in fact we are still all in pretty regular contact and we have a fairly good knowledge of the positions that everybody else is in. …it's good for me because it means that I can keep in touch with what is happening in the industry and where I sit in terms of what the industry average is' (Beta, applications support analyst, interview 10).

More interesting for our examination of Proposition Three, were the software workers who had transitioned from non-IT backgrounds and expressed an identity deriving the software profession. The following individual had a clear picture of which

professional skills he needed to develop while with Gamma and how these linked in with his wider software career path.

> 'If you are good at what you do then you should be able to take these skills elsewhere…. in two years time I will be 37 or 38 and I'll be going to another company and saying well I'm just a developer I haven't got any project leadership experience, so I'd have to get back on the lower rung….I need experience of project leadership (Gamma interview 7 science background).

Particularly, many of the non-IT background participants associated their own professional identity and professionalism with their current employer, rather than the more independent position on professional identity held by the IT-specific employees. However, for those non-IT or IT-related staff who had developed towards higher-level roles the relationship between professional identity and organizational identity started to mirror that of the IT-specific participants. Indeed, the following Beta manager, who had no IT background, illustrates a strong sense of professional identity and his own marketability which he acquired through his own self-development and experiences with the company.

> 'The technical side of things I'm doing just now is quite interesting stuff it is pure webwork and it's primarily been sort of self taught and acquired through work experience…...If it keeps going the way it is just now, I will probably leave. …..I am getting more inclined to leave, partly because there is no point in doing this and staying in the one place all the time and partly because if the technology keeps going the way it's going just now the skills we'll have in six months, other companies will pay more than Beta for them' (Beta, manager, interview 2).

Discussion and Conclusions

There are a number of key themes and ideas that have emerged from the current work. Perhaps most significantly, we have established that there is a division of labour in software work which is predominantly based upon entry qualifications. This division has an impact on organizational identity but not on professional identity. This partition in software work also leads us to question assumptions about knowledge workers and whether software work is, in reality, any different to traditional technical work. The following discussion reflects on these issues.

The qualification histories of our sample clearly parallel those patterns that occur in traditional technical or engineering work. Although 17% of our sample (the IT-related group) did not comply with the precise extremes of what Meiksins and Smith (1996) label *craft* and *managerial* groups, the remainder of the sample, fit almost equally into two groups (non-IT qualified and IT-specific) that in terms of qualifications at least, suit these categorisations. Moreover, there was evidence that those with IT-specific qualifications were undertaking higher skilled work and those with non-IT qualification were - for the most part - undertaking routinised less specialised work. There was a clear gradation in terms of 'elitism' of work from those with job-specific higher levels qualifications to those with unrelated or low level qualifications (Proposition One). This would suggest, with the exclusion of a few specific cases (described in the results section), that new forms of work are really only reproducing traditional patterns. Contemporary division of labour, although superficially integrating 'head' and 'hand' tasks, is really built on a traditional notion of divided work (Greenbaum, 1994, p. 64).

This supports other writers' concerns that the description of types of employment as 'knowledge work' or groups of employees as 'knowledge workers' is problematic, both analytically and empirically (Warhurst & Thompson, 1998). Even discounting the implausibly broad definitions of knowledge workers by writers such as Drucker (1993), we are left with the question as to whether software workers are 'knowledge workers'. Those with technical degrees predominantly undertake 'creative systematic activities in order to increase the stock of knowledge….and the use of this knowledge to devise new applications' (Despres & Hiltrop, 1995, p. 17). More specifically, a hierarchical system based on education and qualifications can be identified within the range of software work. Indeed, those without technical degrees that successfully progress in software work tend to have undertaken a formal IT/software qualification at some point during their career. Hence, many management theorists have found it problematic to position technical workers within existing categories (Darr & Scarselletta, 2002). Yet, it could be argued that this is due to the 'red herring' of the knowledge work allegory. The depiction of our sample of employees as 'knowledge workers', 'symbolic analysts', 'white-collar workers', or 'new professionals' (Barrett, 2001; Barley, 1996) conceals the reality, that they are little different from traditional technical workers.

However, as we proposed at the start of this paper, the point at which our group of software developers do actually differ from traditional engineering workers is by way of the image that they present. Despite the clear separation in work according to qualification, all our employees fit in the category of software worker, and are viewed by others - whether they are in reality or not – as knowledge workers and elite professionals. The consistently high-levels of professional identity in terms of desire to remain in the occupation, pride in the occupation and interest in the sector demonstrates the importance to all our participants, of being seen to be associated with this 'privileged' category (Proposition Two). In this instance, professional identity is a process of impression management and self-interest. As Perrolle (1986) describes, identity with software work as a profession affords some degree of cultural currency. It could be argued that the creation of the dot.com generation and the internet boom has made identifying with this particular profession highly desirable. Whether they are knowledge workers or not the type of work and its association is clearly relevant to people's self-perception.

Yet, for this group of workers, there is some evidence that professional identity is more complex than traditional explanations allow. Established conceptions of identity for technical workers (e.g. Darr & Scarselletta, 2002; Alvesson, et al., 2001) promote those with high professional identity as being instrumental about their career, frequently changing jobs and being fully aware of their strong position within the labour market. However, we found a group of employees (predominantly IT-related or non-IT qualified), who are responding to a changing profession. These employees, it may be argued, are compensating for their less traditional qualifications by focussing on 'professionalism' and customer orientation, and in part, construct their professional identity around this. As one of the Beta's senior managers argued, it is this type of skill that will determine employability in the future, rather than a more narrow focus on high level technical skills. With a number of theorists (e.g. Barrett, 2001; Ramsay, 1999) arguing that software work is progressively being de-skilled as a result of technological advances, logically, other skills must start to come into play. Whilst software work holds many parallels with traditional engineering work, this de-skilling process may lead it, in the future, to be more readily comparable with service sector employment. Hence, interpersonal skills, essential for 'good' customer service may become more and more important.

Tam, et al. (2002) observed that for IT workers' professional commitment was more salient than organizational commitment, a broad picture paralleled in the current research; but this still fails to present the full story. We proposed earlier, that there would be a division between our three groups in terms of organizational identity (Proposition Two). This was reflected in both questionnaire and interview responses but only for the 31-40 age group. For this group, there was a statistically significant progression in level of organizational identification, where those with IT-specific qualifications had the lowest organizational identity and the non-IT qualified group possessed the highest. It could be argued that this group of employees are still building their career, in contrast to the over 40s. We found that they also tend to take on more complex work, in more elite projects involving a range of skills, rather than solely programming or testing. This may be either by self-design or because organizations use this in exchange for some organizational loyalty, but the result is, in essence, that those in their mid-career stage have the motivation to move around as well as labour market advantage. Importantly, a number of other researchers (e.g. Barrett, 2001) have commented upon the ageist nature of software work. As we have found in our research, when software workers get beyond the age of forty, they find it increasingly difficult to find new employment. Regardless of skill sets, they are more dependent on their organization to maintain employment in the industry.

However, the less technically qualified, and arguably less marketable, employees were more dependent on the organization for gaining access to work. These employees had more organizationally specific skills. For example, one group of lower-skilled Omega employees were working with outdated programming languages and with technology that was peculiar to their employing organization (see Marks & Lockyer, 2004 for similar findings). Across the sample there were other employees that were primarily involved with technical authorship and testing code which are again more organizationally specific skills. This is likely to have led the non-IT qualified group to reflect these higher levels of organizational identification. Referring to an earlier argument, these lower-skilled, less-qualified, employees were frequently engaged in more customer facing work. A requirement to present a positive image of the organization may well have re-enforced their identification with the organization.

For all groups of employees, there was however, a substantial degree of self-interest in a strong professional identity. Whilst social identity theory ignores the concept of interests, the strength of professional identity across categories in this

research, indicates how important this concept is for understanding the relationship between personal and social identity. As Jenkins (2004) explained, external image is frequently monitored to ensure that it is compatible with both personal and social identity. The software workers in this study, regardless of qualifications or skill level, were clearly connecting their external image with the benefits of a strong professional identity. Association with a highly regarded occupation is likely to make these employees feel positive about themselves, irrespective of the reality of their day-to-day work. Hence, they would achieve a positive personal identity based on a positive social identity. Furthermore, this self-conscious decision making process that is fundamental to the concept of interests, leads the non-IT qualified group in particular, to acknowledge the importance of their organizational membership. Despite all our participant organizations being viewed as 'front runners' in their particular fields, the exclusivity of the elite (organizational) social identity recognized by Alvesson, et al. (2001) was not pervasive across all employees. Elite identity for our employees was predominantly based on professional status and perceived exclusivity of their profession, not on the status of their employing organization. Identification with the organization had a transactional basis as it was based on the self-interest of the individual with regard to the benefits of organizational membership.

The current paper has presented an overview of software work which contrasts with, and contradicts other accounts of the software profession as either de-skilled (Barrett, 2001, 2005; Beirne, et al., 1998) or highly skilled (Ackroyd, et al., 2000). As it stands, software work is, to a large extent, polarised work – between low and high skill – and this is reflected in the organizational and professional identification of members of the occupation. However, there is some indication that for some, the world of software work is changing. Whilst it was traditionally, and still is to a large extent, an occupation that values high level intellectual skills, aspects of customer service work are creeping into the profession. Ability to 'interface' with clients and manage projects rather than a sole focus on technical skills is becoming more and more valued and becoming more closely associated with professionalism as well as professional identity. Although they were not a large population, at least from the qualitative data, those non-IT specific and IT-related employees appear to represent a body of employees who are managing to transcend what appears to be traditionally divided work. As one Beta manager stated,

> ‘Software was a craft industry, you built something with your hands, you breathed life into your system. I used to describe it like choreography you know, you had something in your mind, you then communicated it to a team of people who worked together to make it happen and you could sit back and you could see it all happening and that was a wonderfully satisfying sort of thing and most of the people who have been through that only leave it with great regret as they move into management and all that kind of thing’ (Beta, Centre manager, interview 13).

Whilst entry into this high level work is still by academic attainment, we have identified this small group that could be viewed to be knowledge workers in the most optimistic sense. In the spirit of the knowledge economy, these employees have gained access to elite work without a traditional educational background. Yet, despite this, our study shows that they still represented a minority position. For the majority, our initial premise stands. The software sector is a divided industry between high level intellectual work, undertaken by those with traditional qualifications in the field, and lower level more mundane work performed by those without specific technical skills.

References

Ackroyd S, Currie W and Glover I, (2000) 'The triumph of hierarchies over markets: information system specialists in the current context', in I. A Glover and M.D. Hughes (eds.) *Professions at Bay.* Control and Encouragement of Ingenuity in British Management, Ashgate Publishing, Aldershot, pp. 267-305.

Allen. N.J. and Meyer, J.P. (1990) ‘The measurement and antecedents of affective, continuance and normative commitment to the organization,’ *Journal of Occupational Psychology*, 63, 1-18.

Alvesson, M., Robertson, M., and Swan, M. (2001) *The Best and the Brightest. The role of elite identity in knowledge intensive companies.* Paper presented at the Critical Management Studies Conference. July 11-12.

Alvesson, M. (2001) ‘Knowledge work: ambiguity, image and identity’. *Human Relations*. 54 (7): 863-86.

Alvesson, M. (1995) *Management of Knowledge Intensive Companies*. Berlin/NewYork: de Gruyter.

Andrews, C.K., Lair, C.D., and Landry, B (2005) 'The labor process in software startups: production on a virtual assembly line'. In R. Barrett (ed.) *Management, Labour Process and Software Development: Reality Bytes*, London: Routledge.

Ashforth, B. and Mael, F. (1989) 'Social identity theory and the organization,' *Academy of Management Review*, 14, 20-39.

Barely, S. R. and J. E. Orr (1997) 'Introduction: The neglected workforce', in S. R. Barely and J. E. Orr, *Between Craft and Science: Technical Work in US Settings*, Ithaca, NY, 1-19.

Barrett, R. (2001) 'Labouring under an illusion? The labour process of software development in the Australian information industry'. *New Technology, Work and Employment* 16 (1)

Barrett, R (2005) 'The reality of software developing'. In R. Barrett (ed.) *Management, Labour Process and Software Development: Reality Bytes*, London: Routledge.

Blau, G.J. (1985) 'The measurement and prediction of career commitment'. *Journal of Occupational Psychology*, 58, 277-288.

Beirne, M., Ramsay, H., & Panteli, A. (1998) 'Developments in computing work: control and contradiction in the software labour process'. In Thompson, P., & Warhurst, C. (Eds.), *Workplaces of the Future*. London: Macmillan.

Bergami, M., & Bagozzi, R. P. (2000) 'Self-categorization, affective commitment and group self-esteem as distinct aspects of social identity in the organization'. *British Journal of Social Psychology*, 39, p. 555-577.

Brown, A. D. (1997) 'Narcissism, identity and legitimacy'. *Academy of Management Review*, *22*(3), 643-686.

Collins, P. H. (1991) *Black Feminist Thought: Knowledge, Consciousness, and the Politics of Empowerment*. New York: Routledge.

Darr, A. and Scarselletta, M. (2002) 'Technicians, clients and professional authority: structured interactions and identity formation in technical work.' *New Technology, Work and Employment,* 17, 1, 61-73.

Depres, C. and Hiltrop, J. (1995) 'Human resource management in the knowledge age: current practice and perspectives on the future.' *Employee Relations*, 17,1, 9-23.

Drucker, P. (1993) *Post-capitalist society* .New York: Harper.

Dutton, J. E., Dukerich, J.M., and Harquail, C.V. (1994) 'Organizational images and member identification'. *Administrative Science Quarterly*, 39, p. 239-263.

Feldman, D.C. and Klich, N.R. (1991) 'Impression management and career strategies'. In R.A. Giacalone and P. Rosenfeld (eds) *Applied Impression Management*, Newbury Park, CA: Sage, pp. 67-80.

Goffman, E (1959) *The Presentation of Self in Everyday Life*. Garden City, N.Y.: Doubleday.

Hall, R.H. (1968) 'Professionalization and bureaucratization'. *American Sociological Review,* 33, 92-104.

Haslam, SA (2001) *Psychology in Organizations: The Social Identity Aapproach,* 1st Edition. Thousand Oaks, CA: Sage.

Hogg, M. A., and Terry, D. J. (2000) 'Social identity and self-categorization processes in organizational contexts'. *Academy of Management Review*, 25, p.121-140.

Jackson, P.R., Wall, T.D., Martin, R., and Davids, K. (1993) 'New measures of job control, cognitive demand, and production responsibility', *Journal of Applied Psychology*, 78, 753-762.

Jenkins R (2004) *Social identity*, 2nd Edition. London: Routledge

Kraft, P., (1977) *Programmers and Managers: The Routinization of Computer Programming in the United States*, Springer-Verlag, New York

Kraft, P., (1979) 'The routinizing of computer programming'. *Sociology of Work and Occupations* 6 (2), 139-155

Kraft, P. and Dubnoff, S. (1996) 'Job content, fragmentation, and control in computer software work'. *Industrial Relations* 25, 2, 184-196

Kunda, G. (1992) *Engineering culture, control and commitment in a high-tech corporation*. Philadelphia: Temple University Press.

Marks A. and Lockyer, C (2004) 'Self-interest and knowledge work: the bugs in the programme for teamwork?' *Economic and Industrial Democracy*, 25 (2), p. 213-238.

Meiksins, P. and Smith, C. (1996) *Engineering Labour: Technical Workers in Comparative Perspective*. London: Verso

Mowday, R.T., Steers, R.M. and Porter, L.W. (1979) 'The measurement of organizational commitment', *Journal of Vocational Behaviour*, 4, 224-247.

Perrolle (1986): 'Intellectual assembly lines: The rationalization of managerial, professional, and technical Work,' In Dunlop, Charles, and Rob Kling (eds.) (1991), *Computerization and Controversy* , Academic Press. pp. 221-235.

Ramsay, H. (1999) *Close encounters of the nerd kind*. Paper presented to the Worklife 2000 Programme, Sweden.

Reich, R. (1991) *The work of nations: Preparing ourselves for 21st century* capitalism. New York: Knopf.

Scarborough, H. (1996) 'Understanding and managing expertise'. In H. Scarborough, (ed.), *The Management of Expertise*. London: St Martin's Press, pp.23-47.

Scarborough, H. (1999) 'Knowledge as work: Conflicts in the management of knowledge workers'. *Technology Analysis and Strategic Management*, 11(1), 5-16.

Smith, C. (1987) *Technical workers: Class, Labour and Trade Unionism*. MacMillan.

Snizek, W.E. (1972) 'Hall's Professionalism Scale: an empirical reassessment'. *American Sociological Review*, 37, 109-114.

Software Echo (2004) 'The Magazine of the Scottish Software and Internet Community'. Issue 36. *Scottish Enterprise*

Tajfel, H. (1972) 'Some developments in European social psychology'. *European Journal of Social Psychology*, 2, p. 307-322.

Tajfel, H. (1978) 'Social categorization, social identity and social comparison'. In ed. H. Tajfel (ed.) *Differentiation between Social Groups*, London: Academic Press, pp. 61-76.

Turner, J.C. (1982) 'Toward a cognitive redefinition of the social group'. In H. Tajfel (ed), *Social Identity and Intergroup behaviour*. Cambridge: Cambridge University Press.

Turner, J. C. (1984) 'Social identification and psychological group formation'. In *The social dimension: European developments in social psychology*, in H.Tajfel (ed) Cambridge: Cambridge University Press.

Tyler, T.R., and Blader, S.L. (2001) 'Identity and co-operative behaviour in groups'. *Group Processes and Intergroup Relations*, 4 (3), p 207-226.

van Knippenberg, D. and van Schie E. C. M. (2000) 'Foci and correlates of organizational identification'. *Journal of Occupational and Organizational Psychology*, 73, 137-147.

Vandenberg, R J. and Scarpello, V. (1994) 'A longitudinal assessment of the determinant relationship between employee commitments to the occupation and the organization.' *Journal of Organizational Behaviour* 15, 535-547.

Wallace, J. E (1995) 'Organizational and professional commitment in professional and non-professional organizations'. *Administrative Science Quarterly*, 40(2).

Warhurst, C and Thompson, P. (1998) 'Hands, hearts and minds: Changing work and workers at the end of the century'. In P.Thompson & C.Warhurst. (eds.) *Workplaces of the Future*. Houndmills Basingstoke and London: MacMillan Business.

Warr, P.B., Cook, J. and Wall, T. D (1979) 'Scales for the measurement of some work attitudes and aspects of psychological well being'. *Journal of Occupational Psychology*. 52, 129-148.

Table 1. Survey employee characteristics

	Contractual status				***Qualifications***						***Main location***			
	Permanent employees		**Temporary contractors**		**IT-specific**		**IT-related**		**Non-IT**		**Onsite**		**Offsite**	
	N	*%*	*N*	*%*	*N*	*%*	*N*	*%*	*N*	*%*	*N*	*%*	*N*	*%*
Omega	104	87	16	13	40	35	20	18	54	48	98	81	23	19
Beta	94	84	18	16	58	54	16	15	33	31	112	100	0	0
Pi	38	100	0	0	9	24	6	16	22	60	38	100	0	0
Lambda	14	100	0	0	5	36	2	14	7	50	14	100	0	0
Gamma	18	100	0	0	10	59	5	29	3	12	18	0	0	0
Total	*268*	*89*	*34*	*11*	*122*	*42*	*49*	*17*	*132*	*41*	*280*	*92*	*23*	*8*

Table 2. Job characteristics by 'qualifications'

	IT-specific qualification (N=122)		IT-related qualification (N=49)		Non-IT qualification (N=118)			
	Mean	*SD*	*Mean*	*SD*	*Mean*	*SD*	*F*	
Importance of:								
software programming	3.33	(.97)	2.77	(1.19)	2.19	(1.11)	33.47	***
systems analysis	3.05	(.83)	2.58	(1.03)	2.33	(.98)	18.05	***
software design	3.16	(.85)	2.81	(1.07)	2.13	(1.03)	34.25	***
testing	2.91	(.88)	2.88	(.94)	2.39	(1.09)	9.19	***
user support	2.64	(.92)	2.61	(1.08)	2.66	(1.07)	.04	ns
business analysis	2.62	(.86)	2.45	(1.00)	2.29	(.91)	3.86	ns
Method control	3.79	(.76)	3.65	(.76)	4.01	(.68)	5.11	**
Cognitive demands	3.98	(.61)	3.92	(.78)	4.04	(.61)	.60	ns

Notes. * p<.05 ** p<.01 *** p<.001 ns not significant

All job characteristics variables were measured on a scale of 1 'not too important' to 4 'absolutely essential' except for method control and cognitive demands which were measured on a scale of 1 'not at all' to 5 ' a great deal'. Identification variables were measured on a scale of 1 'strongly disagree' to 5 'strongly agree'. Patterns of socialising were measured on a scale of 1 'never' to 5 'daily'.

Table 3. Identification with organization and profession by 'qualifications'

	IT-specific qualification (N=122)		**IT-related qualification (N=49)**		**Non-IT qualification (N=118)**			
	Mean	*SD*	*Mean*	*SD*	*Mean*	*SD*	*F*	
Organizational identification	2.99	(.69)	3.16	(.92)	3.29	(.70)	5.02	**
Age: <31	2.94	(.70)	2.96	(.69)	3.16	(.62)	.94	ns
Age: 31-40	2.87	(.67)	2.93	(.97)	3.30	(.74)	3.77	*
Age: >40	3.49	(.51)	4.06	(.66)	3.36	(.73)	.03	ns
Desire to stay in occupation	3.59	(1.16)	3.58	(1.30)	3.56	(1.12)	.02	ns
Pride in occupation	3.69	(.88)	3.50	(1.05)	3.67	(.86)	.82	ns
Interest in sector	3.94	(.87)	3.69	(.90)	3.77	(.78)	1.15	ns

Notes. * p<.05 ** p<.01 *** p<.001 ns not significant
Identification variables were measured on a scale of 1 'strongly disagree' to 5 'strongly agree'. Patterns of socialising were measured on a scale of 1 'never' to 5 'daily'.

Figure 1. Classification of qualitative data by skill and qualifications

Qualifications	Company	Higher-end skill (including project management)	Lower-end skill (testing, business analysis, support)
		QUAD 1	**QUAD 2**
Degree level, IT-specific (e.g., computing science) Total = 39	*Omega (9)*	7 (team leader, developer, project managers)	2 (student placement, business manager)
	Beta (14)	14 (engineer, technical architect, managers)	0
	Pi (5)	3 (programmer, team leader)	2 (support manager, technical writer)
	Lambda (1)	1 (programmer)	0
	Gamma (10)	9 (developer, engineer, graphics applications)	1 (product consultant)
		QUAD 4	**QUAD 3**
Non-degree and non-IT/IT-related degrees Total = 34	*Omega (10)*	5 (systems analyst, systems builder, tester)	5 (testers, business managers)
	Beta (6)	6 (engineer, applications analyst/manager, team leader)	0
	Pi (9)	1 (programmer)	8 (pre-sales support, services manager, training)
	Lambda (2)	0	2 (customer service, support/training)
	Gamma (7)	2 (senior developers/deployment analysts)	5 (business developer, systems analyst, graphic designer)

Notes. 1. The numbers of interviews in each company were: Omega (19); Beta (20); Pi (14); Lambda (3); Gamma (17). The total number of interviews was 73.

2. Classification of employees into IT-specific, IT-related, and non-IT was based on the same categories as for the quantitive analysis.