

Controlling offshore knowledge workers: Power and agency in India's software outsourcing industry

Carol Upadhyia

The paper describes the modes of organisational control employed in the Indian software services outsourcing industry, highlighting the combination of subjective and panoptical managerial techniques. Drawing on ethnographic work in several software services companies in Bangalore, India, it explores the structures of power that operate in these organisations as well as the agency and subjectivity of software workers.

Introduction

The advent of the global information economy has spawned new forms of 'knowledge work' and a new global labour force of professional, technical and white-collar 'knowledge workers', altering modes of organisational control in the 'new workplace' (Castells, 1996; Gephart, 2002).¹ Most studies of managerial control in the knowledge economy have been carried out in the advanced post-industrial countries, especially the USA and the UK. But relations of capital and labour are no longer confined to specific geographical sites, and with the emergence of transnational networks of production and services linking together workers, managers and customers across multiple locations, the concept of *workspace* has become more appropriate than *workplace* (Halford, 2005). Although the phenomenon of outsourcing has been widely discussed, its effects on work, workers, management and labour relations in offshore knowledge industries have barely been studied. For instance, the advent of novel organisational forms such

□ Carol Upadhyia (carol.upadhyia@gmail.com), a social anthropologist, is a fellow at the School of Social Sciences, National Institute of Advanced Studies, Bangalore, India. Her recent research has focused on globalisation and contemporary social and cultural transformations in India, economic anthropology and the politics of anthropology. She has recently completed a large study (with A.R. Vasavi) on the Indian IT/ITES industry and workforce.

as the 'virtual team' has necessitated the fashioning of new technologies of control to govern dispersed workers and a fragmented labour process. How have the labour process and labour relations been transformed in this context, and with what consequences for global knowledge workers?

India's software services outsourcing industry is a prime example of the globalisation of knowledge work.² The industry has grown rapidly and made significant inroads into the global market because of liberalisation policies, the push towards technological modernisation, significant state support and cheaper labour costs in India (Heeks, 1996; Parthasarathy, 2005; Balakrishnan, 2006). Its success can be attributed especially to the availability of a large pool of skilled manpower—particularly engineering graduates—an outcome of Nehruvian state-led development policies and the country's extensive public and private higher education system. India's approximately one million 'IT professionals'—software engineers and other IT (information technology) workers—are engaged in a range of tasks, from software design and development, coding and testing to back office operations, working for clients located primarily in the advanced economies (especially the USA). This army of IT workers are the industry's most important resource, and controlling and coordinating their time, labour and knowledge is a critical task for managements.

The Indian software industry provides an ideal case through which to explore questions about work, workers and the workplace in the new economy, for several reasons. First, much of the theorisation on the 'new workplace' is based on studies in the West, and a wider base for comparative studies may be revealing. Second, software outsourcing is characterised by new forms of online labour, dispersed work teams, networked production and other such features regarded as central to the global information economy. Third, close examination of the 'cyberspace labour process' (Hakken, 2000) in offshore service providers may provide a new angle from which to understand the organisational forms that are emerging in this economy. Finally, it appears that a hegemonic model of 'best practices' for corporate management is being adopted by organisations across the world, especially in the high-tech and knowledge industries—yet the consequences of these management practices for knowledge workers in non-Western locations have barely been explored (but see Ong, 2006).

In this paper, I describe the labour process and forms of organisational control employed in the Indian software services industry and their implications for the 'subjectification' and agency of software workers, drawing on extensive and intensive fieldwork carried out in several software services companies located in Bangalore, India. By introducing comparative material from one of the more distant (yet highly visible) nodes of the global economy, I hope to contribute to current debates on power and subjectivity within the critical organisational studies literature as well as to the anthropology of work and globalisation.

Indian software labour and global control

The software services industry has become one of the major sites of India's rapid integration into the global economy. It emerged in the 1980s on the basis of 'body-shopping', a system of mobile contract labour in which Indian firms or consultants deploy software engineers at the customer's site (Xiang, 2007). Software outsourcing accelerated during the 1990s because of the increasing complexity of IT requirements, the trend towards customisation of software and the enhanced viability of offshoring of services. The global spread of the new information and communication technologies (ICTs) has allowed the software development cycle to be broken down into separate modules that can be carried out at distinct sites. In particular, the relatively low-skilled and labour-intensive processes such as coding and testing are outsourced to low-cost locations such as India. Taking advantage of these technological developments, the Indian industry has been shifting away from bodyshopping towards the offshore model, such that the major part of the work in a software project is now executed by engineers located in India. The industry has grown exponentially over the last two decades to generate total earnings of US\$39.6 billion in 2006–2007, of which

\$31.4 billion were from exports, and India now accounts for 65 per cent of the global market for offshore IT services (NASSCOM, 2005; 2007). Recent estimates place total direct employment generated by the industry at 1.6 million, including about 690,000 in the IT services and products export sector, 553,000 in the ITES-BPO (IT-enabled services and business process outsourcing) sector, and the rest (378,000) in the domestic sector (NASSCOM, 2007).

Outsourced software projects are typically executed by 'virtual teams' of engineers and managers spread across several geographical locations. With this dispersed form of organisation, or what the Indian industry has innovated as the 'global services delivery model', work can be carried out continuously ('24/7') as tasks are shifted from site to site, following the sun. Programmers located in India are often directly logged onto their clients' machines, such that the customer is able to monitor progress, check the quality of the work and communicate with programmers as if they were onsite. Thus, while the bodies of Indian software workers are becoming more *immobile* (in contrast to the physical mobility of the bodyshopping system), their mental labour is *mobilised* or 'liquefied' (Aneesh, 2006), flowing through computer and satellite links as they collaborate and communicate with colleagues, customers and other teams located on the other side of the globe.

Outsourced software services and software development are carried out by both captive offshore software development centres (ODCs) set up by foreign multinational companies (MNCs) and Indian-owned software companies that work on contract mainly for clients outside India. The major employers are the five or six major Indian companies with workforces of over 50,000 (two now have more than 100,000 employees each), but more than 200 multinational ODCs also have a significant presence. In addition, the software sector includes more than a thousand small- and medium-sized enterprises, ranging from innovative start-up products companies to small service providers and contractors. The industry is thus very diverse, encompassing a wide range of activities, types of work and skill levels, from 'high-end' research and development to 'low-end' services such as customisation and systems maintenance.

The distinction commonly used in the industry between 'low-' and 'high-end' work refers to the fact that as one moves through the steps of the software development cycle—from conceptualisation, design and analysis, to coding, testing, delivery/installation and maintenance—the level of skill and knowledge required becomes progressively lower. Indian software services companies in particular specialise in the provision of software services at the lower end of the cycle, which includes the more routine and labour-intensive tasks, while product development, engineering services and R&D constitute only 15 per cent of total software exports (NASSCOM, 2007). Although the industry frequently claims that it is 'moving up the value chain' towards providing end-to-end software development and consultancy, most companies still depend on low-end work and labour cost arbitrage for their survival (Balakrishnan, 2006). In contrast to this, most of the multinational ODCs operating in India work on software products for their parent companies, such as packaged software products and embedded software for non-computer products, and so carry out both low-end and higher-end work. In this paper, I focus mainly on the large Indian software services companies, although data from multinationals are also cited wherever relevant. Although both Indian and multinational companies engage in software outsourcing, the structure of the relationship between service provider and client differs—Indian software services companies work on contract with clients while MNCs set up captive ODCs in India to develop software for their own requirements. However, the functioning of virtual teams and management issues are broadly similar in the two cases.

Several features of the Indian software services industry are particularly relevant for understanding the modes of organisational control that are employed. First, the industry has concentrated on the 'low end' of the software development cycle, as a result of which profits depend primarily on the ability to marshal large numbers of well-qualified human resources, to deploy them on projects when and where needed, to maximise labour productivity and to keep costs down. Software services companies follow the 'human resources augmentation' model of project management, in which

revenue is directly related to the number of projects executed and the number of people working on a project (Tschang, 2001; Ilavarasan, 2008). In this context, control over the software labour process and over workers and their time allocation is crucial.

Second, the social and educational profile of the workforce creates specific problems of management and control. Software companies in India prefer to hire engineering graduates (who are available in plenty) because they are considered to have the educational background and broad skill sets needed for IT work, although their specific training is usually not directly relevant to the work they have to perform. This profile is due in part to the export-oriented and customer-driven nature of the industry—Indian companies attempt to ‘signal quality’ to customers by hiring mainly engineers (Athreye, 2005: 159). In line with the overall trend in the industry, three-fourths of the respondents in our employee survey had graduate or post-graduate engineering degrees, 13 per cent had post-graduate degrees in computer-related subjects, and the remainder were graduates in non-engineering subjects. As a consequence, many software engineers feel that they are overqualified for their jobs, and this frustration inflects labour relations and the self-identity of workers. Another salient feature of the workforce is the middle-class urban background of most software professionals (Upadhyaya, 2007), who place a premium on social status linked to employment and designation. These factors come into complex play as managements attempt to exert control over the software labour process even as software engineers pursue their own agendas through these new global workspaces.

Both Indian and multinational companies have experimented with a range of management techniques and organisational forms to find the optimum formula to handle their ‘resources’ and execute projects efficiently in an ever-changing market situation. However, there have been few studies of management practices in this industry, and even fewer workplace ethnographies.³ In what follows, I attempt to fill this gap through an account of the management techniques that are used to govern offshore software work and the responses of workers to these systems of control.

The paper draws on a study of the Indian software industry and its employees that was carried out in Bangalore and three European countries during 2004–2006. Research methods were primarily qualitative and ethnographic, consisting mainly of informal interviews and interactions as well as participant observation. Altogether we talked to nearly 600 people connected with the industry, including informal interviews with approximately 160 IT and ITES employees in Bangalore, semi-structured in-depth interviews with about 100 CEOs and managers, 50 interviews with other key persons and interviews with 80 Indian IT professionals working in Europe. About 80 days were spent in the observation of work activities in nine companies, such as training sessions, induction programmes, social events, team meetings and everyday work practices, while another 50 days of filming in three software companies added substantially to the archive of research material collected.⁴ In order to generate some amount of quantitative data, we also conducted semi-structured interviews with a purposively selected sample of IT employees. These interviews (totalling 132) and most of the workplace observations were carried out in a sample of software organisations consisting of two large Indian software services companies, two medium-sized Indian companies (one engaged in product development and the other in services), five multinational software development centres and six small firms, all located in Bangalore.⁵

A ‘New Age’ in Indian industry?

The advent of the software services outsourcing has ushered in a new era in business management in India—or so IT industry people often claim. Managers and employees alike assert that IT companies are different in many ways from ‘old economy’ companies—especially public sector industries with their bureaucratic and hierarchical organisational structures. Software organisations, in contrast, are said to have ‘flat’ organisational structures, flexible management practices, and open and informal work cultures. These claims clearly draw upon the currently dominant management ideology emanating from the West, where the ‘new workplace’ (Thompson and Warhurst, 1998)

and networked organisations are supposed to provide greater autonomy and fulfilment to employees and 'subjective' management techniques a more effective means of control over knowledge workers (McKinley and Starkey, 1998; Ray and Sayer, 1999; du Gay and Pryke, 2002). The frequency with which informants talked about the 'difference' of the IT industry in India suggests that the 'new managerialism' (Thrift, 1999)—the 'definitive corporate agenda of late capitalism' (McKinley and Taylor, 1998: 188)—is indeed becoming a hegemonic model of corporate management across the world. The new management ideology has been appropriated into the industry's image-building efforts, mounted in the context of stiff competition in the global market. Indian software companies are attempting to shed their dominant image as providers of cheap low-end IT services and represent themselves instead as global players who offer top-quality and value-for-money consultancy, products and services. One way in which they do so is by importing management models from the American IT industry and representing their employees as highly skilled knowledge professionals.

But image building is not the only reason for the adoption of 'New Age management' by the Indian software industry. 'Cultural' approaches to management are identified with high-technology and knowledge-based organisations globally (Deetz, 1998; Alvesson, 2000), where flexible organisational structures and normative control have (in theory) replaced top-down methods of bureaucratic control (Kunda, 1992). In knowledge-intensive organisations, where control over the knowledge of employees is a key objective, a 'cultural labour process' has emerged that emphasises communication, collaboration, teamwork and knowledge sharing through building strong social networks (Hakken, 2000). Under the new dispensation, workers are supposed to be motivated through identification with the organisation (Ogbor, 2001) and by their individual professional aspirations, and are transformed into self-managing or 'entrepreneurial' workers (Beck, 2000) through the promotion of a 'self-work ethic' (Heelas, 2002).

In contrast to much of the management literature, critical organisational theorists argue that the 'new managerialism' (Thrift, 1999), far from liberating employees, has introduced novel and insidious structures of power and inequality (Gephart, 2002). Willmott (1993), for instance, has drawn attention to the totalitarian consequences of 'corporate culturalism', while Thrift (1999) writes that 'soft capitalism', by requiring total commitment of the self, involves the 'super-exploitation' of both managers and workers (156). Although the relative absence of direct control gives a feeling of liberation to employees, they are engaged in continual 'self-surveillance' and 'strategised subordination' (Willmott, 1993; Deetz, 1998) in which a false sense of autonomy creates 'active consent' (cf. Burawoy, 1985). The employment relationship is no longer purely economic but invokes and invades the subjectivities of employees to an extent probably not seen in earlier phases of capitalism (Rose, 1989).

While Indian software companies have adopted 'New Age' management in part to project a professional and 'global' image to potential clients and to conform with customer expectations, these practices are also considered crucial for managing and retaining control over what is regarded, in the context of high rates of employee 'attrition', as a footloose and demanding workforce. The problem of generating and retaining employee loyalty is particularly salient in knowledge-based industries, especially in the context of favourable labour markets, because employees and the knowledge that they possess are their primary resource (Alvesson, 1993; 2000).

Corporate cultures and 'soft capitalism'

One of the key 'normative' techniques of control deployed by post-bureaucratic organisations is the manufacture of a strong corporate culture in order to incorporate employees into the organisation, thereby creating a self-motivated and committed workforce (Kunda, 1992). This strategy has been extensively adopted by software companies in India. In the case of the multinational ODCs, the task is to reproduce their already established corporate cultures in their Indian subsidiaries—a strategy that is

not without problems (Sathaye, 2008; Upadhya, 2008b)—while Indian software companies have developed distinctive cultures and identities that combine ‘global’ management ideas with what is understood as Indian ethos or work culture. Although Indian companies attempt to distinguish themselves from one another through their corporate cultures, their ‘values’ and ‘vision’ statements tend to be similar to one another, all featuring terms such as ‘transparency’, ‘integrity’, ‘commitment’, ‘teamwork’, ‘corporate best practices’, ‘customer service’ and the like. The key elements of ‘soft capitalism’ are central to these manufactured cultures, with their emphasis on teamwork, knowledge sharing, individual initiative and responsibility, and commitment. Induction programmes and other types of training sessions are organised to inculcate employees with the official corporate culture and create a sense of identity with the organisation. This process is also seen in the creation and dissemination of organisational identities such as ‘Infoscons’ and ‘Wiproites’ (in the software majors Infosys and Wipro, respectively).

Human resource (HR) policies are designed to represent the corporate culture and to attract and retain competent employees. Apart from the very high salaries offered by software companies, Indian software organisations boast of plush working environments and offer an array of facilities such as in-house gyms, child care centres and modern food courts. HR departments organise frequent social events such as team picnics and dinners in five-star hotels to promote employee satisfaction and team spirit, and ‘recognition and rewards’ programmes provide additional incentives for good performance and loyalty to the organisation. These HR strategies are aimed not only at stemming employee attrition in the context of a volatile job market (Upadhya, *n.d.*), but also at deflecting attention away from the sources of widespread employee dissatisfaction and frustration—the reasons for which are outlined in the following discussion.

Normative control strategies such as the deployment of a strong corporate culture are familiar management practices in the West, but in India they represent a departure from earlier organisational forms—hence the ubiquitous narratives about the IT industry’s ‘difference’. Although most managers we interviewed spoke in positive terms about this shift, many employees had a very different view, maintaining that the global corporate model exists only on paper while ‘traditional Indian’ organisational culture persists in the form of hierarchical structures, bureaucratic mentality and ‘feudal’ relationships. The reproduction of ‘traditional’ management practices was often explained by reference to the habits and cultural dispositions of managers and employees, who have yet to fully adapt to the culture of the ‘new workplace’. But, as I argue in the following discussion, there are other reasons why the perception of employees may be closer to the reality, and why top-down and direct techniques of management are in fact the norm in Indian software organisations, despite the obeisance that is paid to New Age management. These conflicting narratives also suggest that ‘soft management’ strategies aimed at aligning employees’ interests with those of the organisation are not always successful, and that employees can easily ‘see through’ the ideological pretensions of official corporate cultures.

English butlers and surrogate management

One of the most crucial elements of the corporate cultures of Indian services companies is what one organisation calls ‘customer-centricity’. The values of customer service and customer satisfaction are constantly driven home to employees through slogans on wall posters and computer mouse pads, messages imparted on company intranets and training programmes. In the induction programme of a medium-sized Indian software services company that we observed, for instance, the trainer explained ‘customer-centricity’ by invoking, without a trace of irony, the good ‘English butler who knows his master’s need even before the master has the need’. ‘Customer is god’ was the lesson for the day (shown in the film *The M-Way: Time + People = Money*; NIAS, 2006).

While customer satisfaction is a key objective in any service industry, in Indian software companies it may also be viewed as a mechanism of control over labour. Direct monitoring and constant communication from the customer side through the

computer system create a feeling that the client is always present and watching. 'Customer-centricity' thus creates a 'surrogate management situation' in which employees 'voluntarily' comply with client demands (Deetz, 1998: 162). The normative ideal of pleasing the customer at any cost is backed up by an implicit threat of loss of business and of jobs; this places the onus for retaining customers onto employees, thereby creating a higher level of commitment.

Teamwork and peer control

Teamwork is yet another normative management technique that is widely practised in Indian software companies. Software engineers are usually organised into teams of 5 to 10 members, each led by a team leader, which are in turn linked together into a larger group under a project manager. The logic of team-based organisation is to devolve decision making to the shop floor and foster cooperation and knowledge sharing. In the Indian software industry, however, teams appear to be organisational units more than work groups, because members of a single team may be involved in several different projects at once, working with people from other teams. Still, organisations place much emphasis on the ideology of teamwork and invest substantial resources in team-building activities and training programmes.

As has been noted in other contexts, team-based organisation functions as a subtle yet strong system of 'concertive control' through peer pressure (Barker, 1993) or peer surveillance (Ezzamel and Willmott, 1998; Sewell, 1998). For instance, in Indian software organisations, each team member's work is continually monitored by others, and those who are lagging behind are subjected to subtle or overt pressure from colleagues. Because the progress of an individual's work often depends on the completion of work by other team members, software engineers pressurise one another to finish tasks on time and to put in extra hours, if necessary.

A key function of team-based organisation is to extract the maximum 'commitment' out of employees in terms of time and effort. Team leaders and managers attempt to create a positive team spirit by organising team outings, social events and other informal activities, and when deadlines loom, the ideology of collectivity is invoked to convince team members to work late or over weekends, or to take on additional tasks. During team meetings, negotiations between engineers and their managers on issues such as extra working hours often take place. Although 'normative' control appears to dominate as team leaders appeal to members' sense of mutual responsibility, there lurks underneath the threat of more direct, coercive means of control that may be invoked if team members do not comply. The coercive effects of peer scrutiny and moral pressure from managers tend to undermine the 'rhetoric of mutuality' in team-based organisation (Sewell and Barker, 2006: 953).

The ideology of teamwork is also contradicted by processes of individualisation that are engendered by the 'engineered culture' of the workplace itself (Ezzy, 2001). For instance, the performance-appraisal system and performance-linked salary structure create competition among employees, which may lead engineers to withhold crucial information from one another. In this competitive atmosphere, 'visibility' is considered to be key to career mobility. Engineers employ various strategies to enhance their visibility with managers, such as continually updating and displaying their knowledge to their superiors and seeking recognition from the company in the form of awards and prizes for good work. Such strategies reinforce organisations' attempts to create self-managing 'entrepreneurial' employees. The management ideology of worker autonomy places the onus of responsibility for completing work onto individual engineers, even when they are faced with impossible deadlines or organisational obstacles (cf. Barrett, 2001). Soft skills training programmes are organised to make engineers more proactive and to induce them to 'take ownership' of their work and careers (Sathaye, 2008). But a close reading of the narratives of software engineers and managers points to a tension between the model of the self-starting global professional, to which they are expected to conform, and the reality of outsourced projects. As the following discussion will show, there are countervailing forces that tend to undermine

the ideology of worker empowerment, producing instead something more like 'cyborg' workers (Bain *et al.*, 2002; Poster, 2002).

Neo-Taylorism in the offshore software factory

Much of the literature on the 'new workplace' focuses on the subjective techniques of control previously discussed, but another set of studies—especially those on call centres—draw on Foucaultian theories of surveillance to understand the operation of technologically driven disciplinary mechanisms in these 'electronic panopticons' (Sewell and Wilkinson, 1992). Research on 'lean management' and Total Quality Management (TQM) has highlighted the centralised, ICT-enabled monitoring and information management systems that allow managements to extract ever more surplus value from workers (Ezzamel *et al.*, 2001). Sewell and Barker (2006) have termed these contrasting views of organisational control the 'coercion versus care' paradigms—while the first focuses on 'panoptical control', the second highlights the more insidious modes of subjective control through which the very selves of workers are colonised by the organisation. However, direct and indirect management strategies are not mutually exclusive, and organisations employ a range of techniques that vary in content and combination, depending on the nature of work and the industry (Barley and Kunda, 1992; Jermier, 1998). The most extreme forms of technology-enabled direct control are found in routinised low-level service jobs that require relatively little skill or education, such as call centres (Thompson *et al.*, 2001). In the case of more 'knowledge-intensive' work such as management consultancy, employees are given considerable autonomy and so managements depend on their consent and self-surveillance (Deetz, 1998). However, it is difficult to draw a clear distinction between 'real' knowledge work and other kinds of service jobs in terms of modes of organisational control. Computer programming is a case in point—while it appears to exemplify knowledge-intensive work in its dependence on the ability, creativity and skills of a highly educated workforce, in some contexts, it is closer to a routinised service occupation. In such cases, the rhetoric of employee empowerment may obscure a deeper reality of work intensification and enhanced managerial control through a combination of electronic surveillance and teamwork (Sewell, 1998).

This is precisely what we see in Indian software services organisations. Despite their espousal of the 'New Age' management ideology, they employ a range of computer-enabled techniques of direct management. There are several reasons for this. First, because of the increasing complexity of software products, projects and programmes, the software development process globally has moved away from individualised craft-like production towards a more routinised factory-like production process (Beirne *et al.*, 1998; Barrett, 2001; 2004; 2005). Software engineering often follows a modular system in which engineers simply assemble 'pre-fabricated' pieces of software. This 'assembly line' approach is employed by the major Indian software services companies, which excel at executing large projects quickly by dividing the work into small pieces, by putting many engineers to work simultaneously and by moving bits of work quickly across different sites. Second, because projects are 'customer-driven' and the client retains control over most processes, technology-based surveillance systems are employed to monitor offshore teams. Third, most Indian software organisations have received international quality certifications such as the International Standards Organisation (ISO) 9000 and Capability Maturity Model (CMM) Level 5—mainly to create legitimacy in the market—leading to standardisation and rationalisation.⁶ Software quality models prescribe systematic processes for the execution of tasks at every stage of the development cycle. Quality management is based on the idea that in order to maintain control over the work process, it must be quantified and measured, reducing work to quantifiable 'metrics' of time, 'effort', productivity and output. These processes include setting specific productivity goals for each worker and team on a daily and weekly basis and detailed monitoring, measuring and evaluation of work completed, creating a regime of endless reporting and recording of activities, which is one reason 'process' is widely despised by software engineers. Quality processes act as a system of

panoptical control through electronic surveillance, which ensures that engineers meet their production and quality goals (Prasad, 1998). As in classical Taylorism, their objectives are (1) to increase productivity and reduce costs by monitoring workers' use of time and performance, (2) to reduce the scope for individual initiative on the part of workers, and (3) to allow management to tighten control over the labour process as well as over the implicit knowledge of workers.

In accordance with the CMM model, software companies employ sophisticated centralised software tools and computer systems to track workflows, output and progress, similar to those found in call centres. In the time management system (TMS), for example, the central tool is the timesheet, on which engineers are required to record the time spent each day on a series of specified activities such as coding, attending meetings, and code review. This log is used for billing the client as well as for internal calculations of measures such as productivity and 'effort variance'. Productivity is defined as 'percentage utilisation of resources' (i.e. software engineers) and is calculated by dividing the actual hours worked by the standard eight hours. The recorded data are collated into status reports and reviewed by the quality assurance department, which then gives feedback to the teams and project managers on their performance. These systems allow managements to continually monitor workflows and performance by benchmarking tasks completed against the project timeline. The data thus generated are also used for other functions such as making estimates for project bids.

The TMS described here constitutes only a small part of the CMM Level 5 quality assurance process, which if followed in its entirety entails 38 different procedures and 212 forms to be filled out. This large amount of 'paperwork' is one reason why most engineers resent having to follow 'process'. Thus, far from 'empowering' employees as they claim, Indian software services companies have adopted a range of exacting neo-Taylorist management techniques in their quest to gain control over the software labour process. Although these organisations present an image of 'open' and 'flexible' workplaces, ultimately they must keep tight control over workflows in order to maintain their profit margins. Added to this are the strict project timelines that are stipulated in the contract and enforced by the client, under threat of withdrawing their business. Project managers are under extreme pressure from their own managers and from clients to ensure that projects are executed on time and commitments are fulfilled, and so ultimately they must revert to direct mechanisms of control. Employees are well aware of this contradiction between management ideology and practice. As an informant commented sarcastically: 'Project managers get paid well because they . . . drive the project to completion and drive the team members to work. We need slave drivers to do that kind of work'.

'Time slaves'

Software services companies bill clients on the basis of man-days and projects operate under strict timelines; hence, control over time is central to control over the work process. Indeed, extracting the maximum amount of time and 'effort' out of software engineers is key to profitability. The Indian software industry is known for the very long working hours that are put in by software engineers—typically 10 hours or more per day. A major reason for this pattern of overwork is that the man-days required for a project are routinely underestimated when making bids, to keep the cost estimate down. This forces engineers to work much longer than the stipulated eight hours per man-day in order to meet deadlines. Another reason is the time difference between India and the client site, which means that conference calls often take place late in the evening for the Indian team, when it is morning in the USA. Although in theory employees are allowed to come to the office later in the morning to compensate for staying late, they usually come in by 9:00 or 10:00 a.m. and still have to remain in office until the conference calls are over, until 8:00 or 9:00 p.m.

An employee of a large services company referred to this as a system of 'time slavery'. To motivate employees to work these long hours, subjective methods of control are crucial. For instance, an engineer who leaves 'early' (i.e. on time) is per-

ceived as not pulling his or her weight in the team and is subjected to peer pressure. This is a key issue for women engineers, who are often accused of putting in fewer hours because they may leave office earlier than their male colleagues to attend to domestic responsibilities (Upadhy, 2006a). Similarly, in order to enhance their 'visibility' in the organisation, most junior engineers will not leave the office before their managers do—staying late is taken as a sign of dedication and software engineers believe that this practice will enhance their chances of a good performance rating or promotion. Even in the multinational ODCs, which are under less pressure to turn a profit, there are similar mechanisms at work. An employee of the software subsidiary of a large German technology company said:

A team can't say no to a particular piece of work that they are asked to do, even if they are already overloaded. Our German counterparts insist that the work gets done by hook or crook. They say, 'I don't care if you have to work 24 hours, finish the work'.

The pattern of extended working hours is of course not confined to the Indian software industry. Shih (2004), writing about high-tech companies in Silicon Valley, argues that because work is organised through projects rather than by the clock, it is infinitely extendable. As one of our informants put it: 'Unlike in the manufacturing sector, the number of units produced is not definite. You can always do a little bit more'. However, it appears that this pattern has been taken to an extreme in the case of the Indian software industry. Many engineers who had worked abroad commented on the difference in working hours in India compared to the USA or Europe, and there is evidence that the Indian employees of multinationals put in longer hours than their European or American counterparts, even when they are 'onsite' (Upadhy, 2006b).

Thus, while most Indian software companies claim to have flexible management systems and to give their employees considerable freedom to manage their workloads, in practice, software development projects are heavily time-bound and require the close coordination of a number of people and activities across different locations. The social atmosphere in these workplaces encourages informal interaction and camaraderie between managers and engineers and among team members and the mantra of soft management gives scope for negotiation, but techniques of normative control are underwritten by more coercive, panoptical methods, producing a highly effective system of control over the labour process. For instance, team-based organisation is supposed to devolve responsibility for the assignment and completion of tasks onto individual teams and engineers, but in reality, targets are set and engineers are monitored by top-down CMM-based surveillance systems that work in tandem with the moral pressure exerted by team members and managers. How do employees then respond to these systems of managerial control? Does 'soft management' create active consent and self-subordination, or do panoptical techniques create resistance?

Consent, resistance and worker subjectivity

Questions about worker consent, resistance and subjectivity have become central to debates in critical organisational studies and labour sociology. In classical Marxist labour process theory, workers are seen as trapped in a system defined by antagonism between capital and labour but in which there is scope for collective action and resistance (Braverman, 1975; Burawoy, 1979). Subsequent scholars believed that the active subject of the worker was missing from much of this literature, and many turned to Foucault in their search for a theory of subjectivity in the workplace (Thompson and Ackroyd, 1995). However, Foucaultian organisational theorists have been criticised for emphasising the 'subjectification' of workers by power, neglecting their agency and forms of resistance (O'Doherty and Willmott, 2001). Although several such studies highlight the crafting of selves by workers (Thompson and Findlay, 1999: 169–170) or their struggle to construct a sense of identity at work, most of this literature is pre-occupied with surveillance and self-surveillance (Ezzamel *et al.*, 2001: 1059). In this 'technological fatalist' perspective, '...the interplay of peer pressure, corporate ideology and computerised workflow monitoring have rendered control ever more

perfect, ever more invisible' (McKinlay, 2002: 86). As an alternative, 'post-Braverman' labour process theorists propose that management systems be viewed as contested relations of power that generate resistance and opposition (Prasad and Prasad, 2000). Several studies have documented the range of strategies employed by workers to dilute and subvert both panoptical and 'empowering' management (Ezzamel *et al.*, 2001: 1073), or their cynical or ironical responses to 'cultural management' (Thompson and Findlay, 1999: 177). In this section, I consider briefly the question of consent and critique in the context of the Indian software industry, drawing on a few examples of workplace conflicts and negotiating strategies employed by software engineers.

Negotiation and subversion

The quality control processes employed by software organisations appear to provide little space for manoeuvre. However, these workspaces are characterised by constant negotiations between engineers and managers, overt and covert, over workloads, deadlines, allocation of time and the like. One source of this autonomy is provided by the ideology of 'soft management' itself—to the extent that managers have internalised the new leadership style, engineers are able to exploit it by using a strategy of 'resistance through negotiation' (Ezzamel *et al.*, 2004, quoted in Sewell and Barker, 2006: 949). Project managers and team leaders must devote considerable time and effort to persuading engineers to stay late, to work over the weekend or to take on undesirable tasks, and engineers do attempt to resist these impositions, but in the end they usually capitulate because of the threat of a poor performance appraisal or other sanctions that managers wield.

An example of this strategy is drawn from a role-play exercise that we observed during a leadership (management) training workshop in an American ODC, in which one trainee plays the role of a software engineer and another that of a manager. In this exercise, the manager's task was to persuade a reluctant engineer to take on extra work. The 'employee' attempted to negotiate with the manager by invoking the 'caring' management paradigm, but in response the 'manager' invoked an implicit threat, forcing the 'employee' to capitulate:

Manager: I would like you to work on the user interface, it's critical to get it finished fast.

Engineer: This kind of work won't do anything for my career growth.

Manager: Everyone wants to work on core elements, but this is critical to the customer. It will help you to build your skills.

Engineer: I cannot take on *ad hoc* work—I've done this work before.

Manager: No one else can do this job, so please just for this once do it and then we'll discuss your career plans.

While this is of course only a role play, it does indicate the kinds of strategies that may be used by engineers when faced with pressure from management, as well as those employed by managers—persuasion appealing to the employee's sense of personal responsibility, backed up by the threat of coercion. (A participant in this training session later confirmed that this role play represented a typical encounter.) In this case, the manager's control over the employee's career path lurks just under the surface of an apparently friendly negotiation. Ultimately, engineers are acutely aware that their managers are responsible for performance appraisals, which determine the 'variable' component of their salaries as well as increments and promotions; hence, they can refuse such requests only up to a point.

Software engineers employ negotiation on some fronts, while on others they find ways to subvert unpalatable systems of control. One of the major sites where this can be observed is around the issue of time (cf. Burawoy *et al.*, 2000). The TMS, with its complicated timesheets, reporting procedures and 'metrics', is easily undermined by engineers, who are responsible for keeping track of their own hours. While one might expect them to *over-report* their working hours in order to demonstrate that they are working hard, we found the reverse to be more common—in one case, all the members of a team were recording eight hours per day although they were actually working

much longer hours (NIAS, 2006). Probing revealed that they were attempting to shore up their productivity ratings, which would decline if they were to fill in the actual hours of work. Thus, quality processes and measurement mechanisms are vulnerable to worker manipulation on the 'shop floor'. In the final analysis, however, this space for manoeuvre is circumscribed by the project cycle and client demands, and engineers' work is constantly monitored by both managers and customers.

The under-reporting of working hours could also be interpreted as an example of 'self-subordination' rather than resistance—by reporting fewer hours of work, workers are complicit in the system of under-costing project bids which is the ultimate source of their exploitation.⁷ This highlights the complexity of understanding worker agency and consciousness—in such cases, are employees consciously attempting to subvert the system while at the same time complying with it? This example also suggests that management tools, such as the timesheet, operate as mechanisms of control in ways other than, and in addition to, their officially designated functions.

Consent, performance or self-subordination?

Although Indian software engineers often complain of overwork, stress or boredom, they rarely criticise management openly or directly resist managerial control. Does this mean that they manifest 'strong consent', actively producing themselves in the image formulated by the firm (Deetz, 1998: 168–169)? Deetz suggests that while employees are aware of the subjective techniques of control that are employed and may adopt an ironic stance to indicate their awareness, they nonetheless 'enact' the prescribed scripts, engaging in strategic self-manipulation to avail of the pay-off.

Many Indian software engineers also display a sceptical attitude towards their employers and managers, especially with regard to New Age management practices. For example, several informants debunked the official line about 'flat structures', maintaining that hierarchical relationships and bureaucratic procedures are the norm in their organisations. Similarly, many take an ironic stance towards efforts at 'cultural management', for instance, when they deride 'soft skills' training programmes as useless or even insulting (although they play along in order to escape routine work for a few days). Despite these critical attitudes, most employees reproduce the official scripts and adopt the sanctioned behaviour patterns in order to further their careers (cf. Kunda and van Maanen, 1999).

The long hours put in by software engineers may suggest that they engage in 'self-subordination'. Indeed, many defend this practice by reference to their desire to succeed and move up in the organisation, rather than complaining about the structural factors behind overwork (which they do recognise). But it is also a response to job insecurity and software companies' 'hire-and-fire' policies—engineers must pursue their careers in an uncertain industry that is buffeted by global forces, so they focus on 'making hay while the sun shines', engaging in self-subordination as a trade-off for future financial gains and sacrificing most of their waking hours to work in the hope that the money they earn will enable them to fulfil their other ambitions and desires. This strategy only reinforces their commitment to work, further whittling down the time available to pursue lives outside of work (cf. Hochschild, 1997; Deetz, 1998: 166).

Another element in their self-exploitation is the fact that most software engineers do not view themselves as 'workers', even though their structural position is in many ways no different from that of factory floor workers in manufacturing industries. This self-perception as knowledge professionals reinforces the individualised work culture of the industry, where 'resistance' too is individualised. For instance, salaries are always individually negotiated and there is little interest in collective bargaining. The culture of individualism, however, provides a potent source of worker agency, in that it encourages software engineers to pursue their own goals over those of their employers—in direct conflict with management efforts to foster employee loyalty. Although software engineers may engage in various strategies to negotiate their position within organisations, ultimately their primary and most effective weapon is the 'exit' option. Their main source of 'empowerment' comes from a buoyant labour market and the perception that

they can always find another job if the current one becomes too tough or monotonous or they feel they are stagnating. This is an important reason for the high rate of employee turnover, which is perceived to be a major problem for the industry. HR managers constantly complain about 'fickle' and 'demanding' software engineers who quit on the slightest pretext or for a higher offer, and one of their major tasks is to attract and retain competent engineers (Upadhyaya, *n.d.*).

This pattern of high labour mobility and turnover is typical of the 'new economy', especially in the USA (Benner, 2002), but represents a new trend in the white-collar/professional workforce in India. There are only a few cities in India where software jobs are easily available; hence, most software professionals must migrate from their hometowns in order to work. Software professionals are highly footloose, changing jobs and relocating, within India and globally, more often than other white-collar or professional workers—a pattern that is reinforced by the young age profile of the workforce (with an average age of 27). Major centres of software activity, such as Bangalore, host hundreds of organisations, and software engineers prefer to work in this city in part because it is relatively easy to find a job. 'Headhunters' abound in the centres of IT activity and their main function is to 'poach' engineers from other companies for their clients. Viewing the entire Indian software industry or even the global informational economy as the field in which they operate (rather than the individual organisation), software engineers can thus be seen as active subjects who are 'working' the system to their own advantage.

In the final analysis, however, the system can be manipulated only to a limited extent because software engineers ultimately must find and retain employment, at least for some period of time, during which they must conform to organisational norms and systems of control (which are much the same across the industry). An additional factor that reinforces the 'consent' and complicity of software engineers is the fact that most aim to become managers. They look to the 'managerial path' because of the absence of a viable 'technical path' of career development in the Indian industry, and so they tend to identify with management. These same software engineers, after a few years, also reproduce and enforce these systems of control when they in turn become team leaders or project managers.

Conclusion

This somewhat sketchy account of organisational control and worker response in the Indian software industry shows that contemporary 'soft management' theories are rapidly becoming globally hegemonic, yet often operate more as ideology than practice when exported to sites such as India. The five-star working environments, 'employee-friendly' HR policies and high salaries provided by these companies convey the impression of modern and 'professional' workplaces, glossing over the reality of very long working hours, intense work pressure driven by client-imposed deadlines and routinisation of labour. The top-down Taylorist systems of control that are employed by these organisations are quite different from what is presented as the official corporate culture, a fact that can be understood by reference to the specificities of this offshore service industry. Indian software companies engage in image management in order to sell themselves to potential clients as well as employees, and so they must not only pay obeisance to the contemporary ideology of 'soft capitalism', but they must also satisfy their customers and fulfil their contracts on time. As a result, exacting panoptical systems of control are deployed along with normative management strategies to maximise software labour productivity. Although these strategies may be much the same as those used all over the world in the software industry (Barrett, 2005), I argue that they are taken to an extreme in the Indian industry because of the client-driven organisational structure, the dependence on labour cost arbitrage and the strategy of acquiring quality certifications in order to be competitive in the global market.

What are the implications of these management systems for the subjectivity and agency of Indian software workers? Should they be understood as 'cyborgs' trapped in

'electronic panopticons'; as 'entrepreneurial' employees who are so 'entranced' by the idea of themselves as 'empowered' subjects (O'Doherty and Willmott, 2001: 469) that they buy into the dominant management ideology; or as strategising agents who pursue their own agendas by availing of the opportunities provided by this sunrise industry? Clearly, there is no simple answer. Like all workers, Indian software engineers are 'subjectified' by a range of conflicting discourses and experiences. On the one hand, many seem to have absorbed the dominant discourses of the global corporate workplace and accepted the industry's self-representations, retailing the official position in their narratives. In this sense, contemporary ideologies of self-management, individualism and ultimately of the market do gradually invade their lifeworlds. But while employees appear to engage in 'self-subordination', they often articulate critical counter-narratives and mount resistance through negotiation. Their 'subjectification' is not total—the frustration of HR managers dealing with a flighty and unpredictable workforce can only be explained if we recognise the multiple strategies through which employees in turn 'colonise' software organisations, seizing the opportunities provided by this new industry to further their own agendas.

The struggles that can be discerned in these global workspaces are evidence for this claim. At the level of day-to-day work practices, there are constant conflicts and negotiations over the way in which work is allocated and carried out, and time is apportioned and used. The balance of power between managements and employees is unstable, and while ultimately everyone in the organisation is constrained to 'deliver', employees attempt to alter the terms on which they do so. Here the dominant ideology of 'soft management' is a weapon that can be wielded by both sides, although not with equal force. Moreover, employees are primarily interested in pursuing their individualised career strategies, and their field of action is the industry as a whole rather than a particular company—an attitude that conflicts with organisations' demands for loyalty and commitment. As a result, software organisations are continually engaged in wars of attrition and position (to use Gramsci's term) with their employees, who submit to most forms of control as long as they perceive it to be in their own interest, but whose 'consent' is often half-hearted or ironical. These complexities and contradictions can be understood by viewing software engineers as embedded in power relations both as subjects and objects, for whom work is not simply only a source of subjectification, but also a resource that can be tapped by them as conscious and active agents.

Notes

1. This paper is based on a sociological study of the Indian IT/ITES (information technology and IT-enabled services) workforces that was carried out by A.R. Vasavi and me, along with a research team, at the National Institute of Advanced Studies, Bangalore, between November 2003 and March 2006. The research project was funded by the Indo-Dutch Programme on Alternatives in Development (IDPAD), the Netherlands, and was conducted in collaboration with Peter van der Veer of the University of Utrecht. For a comprehensive report of the study's findings, see Upadhyaya and Vasavi (2006). I thank Michael Burawoy, Sonali Sathaye, A.R. Vasavi and the two anonymous reviewers for their comments on earlier drafts; the usual caveats apply.
2. The software industry is commonly referred to generically in India as the 'IT industry', and the terms 'IT professional' and 'software professional' are used interchangeably.
3. The extant management literature on global virtual teams focuses primarily on problems of communication, sharing of information and establishment of trust, to the neglect of questions of power and control that dominate the literature on organisational control. See, for example, *Organisation Science* 10, 6, Special Issue: Communication Processes for Virtual Organisations (November–December 1999).
4. The work cultures and modes of organisational control employed in three software companies in Bangalore are depicted in the NIAS-IDPAD film series 'Coding Culture: Bangalore's Software Industry' (<http://www.codingculture.com>).
5. See Upadhyaya and Vasavi (2006) and Upadhyaya (2008a) for further details on research methods and sampling.

6. The SEI-CMM model was developed by the Software Engineering Institute (SEI) of Carnegie Mellon University and is now called Capability Maturity Model Integration (CMMI). Most of the CMM Level 5 companies in the world are in India.
7. I thank one of the anonymous reviewers of this article for this insight.

References

- Alvesson, M. (1993), 'Organisation as Rhetoric. Knowledge-Intensive Companies and the Struggle with Ambiguity', *Journal of Management Studies* **30**, 6, 997–1015.
- Alvesson, M. (2000), 'Social Identity and the Problem of Loyalty in Knowledge-Intensive Companies', *Journal of Management Studies* **37**, 8, 1101–1124.
- Aneesh, A. (2006), *Virtual Migration; The Programming of Globalisation* (Durham, NC: Duke University Press).
- Athreye, S.S. (2005), 'Human Capital, Labour Scarcity and Development of the Software Services Sector', in A. Saith and M. Vijayabaskar (eds), *ICTS and Indian Economic Development; Economy, Work, Regulation* (New Delhi: Sage Publications), pp. 154–174.
- Bain, P., A. Watson, G. Mulvey, P. Taylor and G. Gall (2002), 'Taylorism, Targets and the Pursuit of Quantity and Quality by Call Centre Management', *New Technology, Work and Employment* **17**, 3, 170–185.
- Balakrishnan, P. (2006), 'Benign Neglect or Strategic Intent? Contested Lineage of Indian Software Industry', *Economic and Political Weekly* **41**, 3865–3872.
- Barker, J.R. (1993), 'Tightening the Iron Cage: Concertive Control and Self-Managing Teams', *Administrative Science Quarterly* **38**, 408–437.
- Barley, S.R. and G. Kunda (1992), 'Design and Devotion: Surges of Rational and Normative Ideologies of Control in Management Discourse', *Administrative Science Quarterly* **37**, 3, 363–399.
- Barrett, R. (2001), 'Labouring under an Illusion? The Labour Process of Software Development in the Australian Information Industry', *New Technology Work and Employment* **16**, 18–34.
- Barrett, R. (2004), 'Working at Webboyz: An Analysis of Control over the Software Development Labour Process', *Sociology* **38**, 4, 777–794.
- Barrett, R. (ed.) (2005), *Management, Labour Process and Software Development: Reality Bytes*. (London: Routledge).
- Beck, U. (2000), *The Brave New World of Work* (Cambridge: Polity Press).
- Beirne, M., H. Ramsay and A. Panteli (1998), 'Developments in Computing Work: Control and Contradiction in the Software Labour Process', in P. Thompson and C. Warhurst (eds), *Workplaces of the Future* (Hampshire: Macmillan), pp. 142–161.
- Benner, C. (2002), *Work in the New Economy; Flexible Labour Markets in Silicon Valley* (Oxford: Blackwell).
- Braverman, H. (1975), *Labour and Monopoly Capital: The Degradation of Work in the Twentieth Century* (New York: Monthly Review Press).
- Burawoy, M. (1979), *Manufacturing Consent: Changes in the Labour Process under Capitalism* (Chicago, IL: University of Chicago Press).
- Burawoy, M. (1985), *The Politics of Production: Factory Regimes under Capitalism and Socialism* (London: Verso).
- Burawoy, M., J.A. Blum, S. George, Z. Gille, T. Gowan, L. Haney, M. Klawiten, S.H., Lopez, S. O'Riain and M. Thayer (2000), 'Net-Working for a Living; Irish Software Developers in the Global Workplace', *Global Ethnography: Forces, Connections and Imaginations in the Postmodern World* (Berkeley, CA: University of California Press), pp. 175–202.
- Castells, M. (1996), *The Information Age: Economy, Society and Culture. Vol. I: The Rise of the Network Society* (Oxford: Blackwell).
- Deetz, S. (1998), 'Discursive Formations, Strategised Subordination, and Self-Surveillance', in A. McKinley and K. Starkey (eds), *Foucault, Management and Organisation Theory* (London: Sage), pp. 151–172.
- Ezzamel, M. and H. Willmott (1998), 'Accounting for Teamwork: A Critical Study of Group-Based Systems of Organisational Control', *Administrative Science Quarterly* **43**, 2, 358–396.
- Ezzamel, M., H. Willmott and F. Worthington (2001), 'Power, Control and Resistance in the "Factory that Time Forgot"', *Journal of Management Studies* **38**, 8, 1053–1079.
- Ezzamel, M., H. Willmott and F. Worthington (2004), 'Accounting and Management-Labour Relations: The Politics of Production in the "Factory with a Problem"', *Accounting, Organisations, and Society* **29**, 269–302.
- Ezzy, D. (2001), 'A Simulacrum of Workplace Community: Individualism and Engineered Culture', *Sociology* **35**, 3, 631–650.

- du Gay, P. and M. Pryke (eds) (2002), *Cultural Economy* (London: Sage).
- Gephart, R.P. Jr (2002), 'Introduction to the Brave New Workplace: Organisational Behaviour in the Electronic Age', *Journal of Organisational Behaviour* **23**, 327–344.
- Hakken, D. (2000), 'Resocialising Work? The Future of the Labour Process', *Anthropology of Work Review* **21**, 1, 8–10.
- Halford, S. (2005), 'Hybrid Workspace: Re-Spatialisations of Work, Organisation and Management', *New Technology Work and Employment* **20**, 1, 19–33.
- Heeks, R.B. (1996), *India's Software Industry: State Policy, Liberalisation and Industrial Development* (New Delhi: Sage Publications).
- Heelas, P. (2002), 'Work Ethics, Soft Capitalism and the "Turn to Life"', in P. du Gay and M. Pryke (eds), *Cultural Economy* (London: Sage), pp. 78–96.
- Hochschild, A.R. (1997), *The Time Bind: When Work Becomes Home and Home Becomes Work* (New York: Metropolitan Books).
- Ilavarasan, P.V. (2008), 'Software Work in India: A Labour Process View', in C. Upadhy and A.R. Vasavi (eds), *In an Outpost of the Global Economy: Work and Workers in India's Information Technology Industry* (New Delhi: Routledge), pp. 162–189.
- Jermier, J.M. (1998), 'Introduction: Critical Perspective on Organisational Control', *Administrative Science Quarterly* **43**, 2, 235–256.
- Kunda, G. (1992), *Engineering Culture: Control and Commitment in a High-Tech Corporation* (Philadelphia, PA: Temple University Press).
- Kunda, G. and J. van Maanen (1999), 'Changing Scripts at Work: Managers and Professionals', *Annals, AAPSS* **561**, 64–80.
- McKinlay, A. (2002), 'The Limits of Knowledge Management', *New Technology Work and Employment* **17**, 2, 76–88.
- McKinley, A. and K. Starkey (eds) (1998), *Foucault, Management and Organisation Theory* (London: Sage).
- McKinley, A. and P. Taylor (1998), 'Through the Looking Glass: Foucault and the Politics of Production', in A. McKinley and K. Starkey (eds), *Foucault, Management and Organisation Theory* (London: Sage), pp. 173–190.
- NASSCOM (2005), *Summary of NASSCOM-McKinsey Report 2005*, NASSCOM Newline No. 50, December, <http://www.nasscom.org> (accessed 22 May 2006).
- NASSCOM (2007), *IT Industry Factsheet*, <http://www.nasscom.org> (accessed 4 April 2007), October.
- National Institute of Advanced Studies (NIAS) (2006), 'Coding Culture: Bangalore's Software Industry', film series by Gautam Sonti in collaboration with Carol Upadhy. Bangalore.
- O'Doherty, D. and H. Willmott (2001), 'Debating Labour Process Theory: The Issue of Subjectivity and the Relevance of Post-Structuralism', *Sociology* **35**, 2, 457–476.
- Ogbor, J.O. (2001), 'Critical Theory and the Hegemony of Corporate Culture', *Journal of Organisational Change Management* **14**, 6, 590–608.
- Ong, A. (2006), 'Corporate Players, New Cosmopolitans, and Guanxi in Shanghai', in M.S. Fisher and G. Downey (eds), *Frontiers of Capital: Ethnographic Reflections on the New Economy* (Durham, NC: Duke University Press), pp. 163–190.
- Parthasarathy, B. (2005), 'The Political Economy of the Computer Software Industry in Bangalore, India', in A. Saith and M. Vijayabaskar (eds), *ICTs and Indian Economic Development; Economy, Work, Regulation* (New Delhi: Sage Publications), pp. 199–230.
- Poster, M. (2002), 'Workers as Cyborgs: Labour and Networked Computers', *Journal of Labour Research* **23**, 339–354.
- Prasad, M. (1998), 'International Capital on "Silicon Plateau": Work and Control in India's Computer Industry', *Social Forces* **77**, 2, 429–452.
- Prasad, P. and A. Prasad (2000), 'Stretching the Iron Cage: The Constitution and Implications of Routine Workplace Resistance', *Organisation Science* **11**, 4, 387–403.
- Ray, L. and A. Sayer (eds) (1999), *Culture and Economy after the Cultural Turn* (London: Sage).
- Rose, N. (1989), *Governing the Soul: The Shaping of the Private Self* (London: Routledge).
- Sathaye, S. (2008), 'The Scientific Imperative to be Positive: Self-Reliance and Success in the Modern Workplace', in C. Upadhy and A.R. Vasavi (eds), *In an Outpost of the Global Economy: Work and Workers in India's Information Technology Industry* (New Delhi: Routledge), pp. 136–161.
- Sewell, G. (1998), 'The Discipline of Teams: The Control of Team-Based Industrial Work through Electronic and Peer Surveillance', *Administrative Science Quarterly* **43**, 2, 397–428.
- Sewell, G. and B. Wilkinson (1992), 'Someone to Watch over Me: Surveillance, Discipline and the Just-in-Time Process', *Sociology* **26**, 2, 271–289.

- Sewell, G. and J.R. Barker (2006), 'Coercion versus Care: Using Irony to Make Sense of Organisational Surveillance', *Academy of Management Review* **31**, 4, 934–961.
- Shih, J. (2004), 'Project Time in Silicon Valley', *Qualitative Sociology* **27**, 2, 223–245.
- Thompson, P. and C. Warhurst (eds) (1998), *Workplaces of the Future* (Hampshire: Macmillan).
- Thompson, P., C. Warhurst and G. Callaghan (2001), 'Ignorant Theory and Knowledgeable Workers: Interrogating the Connections between Knowledge, Skills and Services', *Journal of Management Studies* **38**, 7, 923–942.
- Thompson, P. and P. Findlay (1999), 'Changing the People: Social Engineering in the Contemporary Workplace', in L. Ray and A. Sayer (eds), *Culture and Economy after the Cultural Turn* (London: Sage), pp. 162–188.
- Thompson, P. and S. Ackroyd (1995), 'All Quiet on the Workplace Front? A Critique of Recent Trends in British Industrial Sociology', *Sociology* **29**, 615–633.
- Thrift, N. (1999), 'Capitalism's Cultural Turn', in L. Ray and A. Sayer (eds), *Culture and Economy after the Cultural Turn* (London: Sage), pp. 135–161.
- Tschang, T. (2001), 'The Basic Characteristics of Skills and Organisational Capabilities in the Indian Software Industry', ADB Working Paper 13, Tokyo: Asian Development Bank Institute.
- Upadhy, C. (2006a), 'Gender Issues in the Indian Software Industry', in A. Gurumurthy, P.J. Singh, A. Mundkur and M. Swamy (eds), *Gender in the Information Society; Emerging Issues* (Bangkok: UNDP-APDIP), pp. 74–84.
- Upadhy, C. (2006b), 'The Global Indian Software Labour Force: IT Professionals in Europe', IDPAD Working Paper Series 2006, No. 1. The Hague: Indo-Dutch Programme on Alternatives in Development.
- Upadhy, C. (2007), 'Employment, Exclusion and "Merit" in the Indian IT Industry', *Economic and Political Weekly* **42**, 20, 1863–1868.
- Upadhy, C. (2008a), 'Ethnographies of the Global Information Economy: Research Strategies and Methods', *Economic and Political Weekly* **43**, 17, 64–72.
- Upadhy, C. (2008b), 'Management of Culture and Management through Culture in the Indian Software Outsourcing Industry', in C. Upadhy and A.R. Vasavi (eds), *In an Outpost of the Global Economy: Work and Workers in India's Information Technology Industry* (New Delhi: Routledge), pp. 101–135.
- Upadhy, C. (n.d.), 'Taking the High Road? Labour in the Indian Software Outsourcing India', forthcoming in edited volume from IILS-ICSSR-IHD International Workshop on Global Production Networks and Decent Work: Recent Experience in India and Global Trends (Bangalore, November 18–20, 2007).
- Upadhy, C. and A.R. Vasavi (2006), 'Work, Culture, and Sociality in the Indian IT Industry: A Sociological Study'. Final Report submitted to IDPAD, Bangalore: NIAS.
- Willmott, H. (1993), 'Strength Is Ignorance, Slavery Is Freedom; Managing Culture in Modern Organisations', *Journal of Management Studies* **30**, 4, 515–552.
- Xiang, B. (2007), *Global 'Body Shopping': An Indian Labour System in the Information Technology Industry* (Princeton, NJ: Princeton University Press).q1