FROM LABOR PROCESS TO ACTIVITY THEORY

Paul S. Adler
Marshall School of Business
University of Southern California
Los Angeles CA 90089-0808
Email: padler@usc.edu

Version: May 26, 2004

Forthcoming in P. Sawchuk, M. Elhammoumi, abd N. Duarte, eds., **Critical Perspectives on Activity Theory, Education and Work: An International Collection**, Cambridge University Press

FROM LABOR PROCESS TO ACTIVITY THEORY

ABSTRACT

Starting with Braverman (1976), labor process theory has been an influential school of thought in the critical analysis of work. However, it has recently lost momentum. One reason for this loss of momentum is its inability to deal with evidence of a long-term upgrading trend in skill requirements. This inability results from the truncated reading of Marx that underpins labor process theory. It has ignored the fundamental contradiction Marx saw between the progressive socialization of the labor process and the persistence of capitalist profitability constraints. Activity theory provides a framework in which we can trace the various locii of the contradiction between socialization and profitability. I use this framework to analyze the rationalization of software development, and show how socialization has been simultaneously stimulated, retarded, and distorted by profitability pressures.

FROM LABOR PROCESS TO ACTIVITY THEORY

As new technologies and new organizational forms proliferate, work is moving back to the center of organizational research (Barley, 1996; Barley and Kunda, 2001). And as interest in work grows, so too does the corresponding theoretical challenge: how best can we apprehend the nature of work itself and its links to both the organizational structure and to the lived experience of work?

This essay returns to the roots of one of the more prominent theories of work organization, Labor Process Theory (LPT) (see Thompson, 1989; Wardell et al., 1999; Jermier, 1998). Inaugurated by Braverman's "classic" work (1976; classic according to Burawoy, 1996) and inspired by Marx, LPT has been a key influence on critical research on work around the world. Recently, however, LPT has been losing momentum, in favor of post-modernist strands of theorizing, in particular those building on the work of Michel Foucault, where the inspiration comes from Nietzsche rather than Marx.

I argue that one reason for LPT's loss of momentum has been its inability to deal with an increasingly discomforting anomaly. LPT is fundamentally incompatible with the growing consensus that capitalist development has been associated not with the "deskilling" trend predicted by Braverman, but with a trend towards the upgrading of worker skills and responsibilities. I argue that LPT's inability to accommodate upgrading is due to the one-sided of its reading of Marx. This essay proposes an alternative, more dialectical reading of Marx that may help us better understand the changing nature of work.

The paper begins with a brief recapitulation of labor process theory. I then outline the empirical case for skill upgrading and show the futility of LPT's efforts to deal with this "anomaly." I then propose an alternative reading of Marx and explicate that reading using Activity Theory (as developed by Engestrom, 1987, 1990; and discussed by Blacker, 1993; Holt and Morris, 1993). The body of the paper illustrates the potential fruitfulness of this alternative framework through a brief analysis of the case of the rationalization of software development. A discussion section suggests a number of implications of this activity-theoretic perspective for number of other streams of work research. A conclusion looks towards the future.

LABOR PROCESS THEORY, BRIEFLY

Inspired by Marx, labor process theory proposes that the key to understanding work organization lies in the structure of the broader society within which it is embedded, rather than in human psychology or in the dynamics of dyadic exchange. Social structure, in turn, is seen as fundamentally determined by the prevailing relations of production — the nature of property rights over productive resources. The relations of production characteristic of capitalist societies derive from the nature of the commodity (the "germ," or core, of capitalist production: Marx, 1977: 163). The commodity is something produced for sale, and as such has two contradictory aspects: its use-value – its value as something useful to the purchaser — and its exchange-value — its power to command money in exchange.

As a system of commodity production, capitalist relations of production have two main dimensions. First, ownership of productive resources is dispersed among firms which confront each other as commodity producers in market competition: call this the "capital relation." Second, alongside those who enjoy such ownership is a class a non-owners who, lacking access to means of production, must sell their capacity to work ("labor power") as if it were a commodity on the labor market: call this the "wage relation." These features of the capitalist social structure have strong implications for the organization of work. Under the capital relation, and notwithstanding important variations across industries, regions, periods, and strategies, firms must reduce costs of production.

Management responses include downward wage pressure, work intensification, work reorganization, technological change, relocation, as well as efforts to align owners' and workers' financial interests. These responses in turn elicit reactions from workers. Under the wage relation, an incomplete contract, workers sell their capacity to work (labor power) at the going rate, in exchange for the promise to do whatever they are asked within some zone of indifference; but it is management's challenge to extract effective work effort. Piece-rate systems do not fundamentally change this, but merely change the form of the conflict at the heart of the wage-effort bargain (as elucidated by both Marx and Taylor). Management responses to this pressure include closer supervision, financial incentives, work reorganization, technological innovations, and normative control. Firms might attempt to derive competitive advantage from collaboration with other firms and with their own employees; but the fundamentally competitive nature of capitalist, commodity-based relations of production — both among firms and between management and workers within firms — threatens constantly to undermine any such collaboration.

Whereas more traditional readings of Marx — as indeed many non-Marxist theories — give a key role to technological change as a driver of social change and determinant of work organization, labor process theorists — along with other social constructionists — have been adamantly opposed to "technological determinism." Marking it as distinctively neo rather than traditional in its reading of Marx, LPT argues that attributing a basic causal role to technology would be to naturalize the socially-constructed, historically-specific, capitalist relations of production (eg Burawoy, 1979: 14ff, 220). Technology is itself shaped by these relations of production (Noble, 1979).

The net result for work organization, according to labor process theory, is that:

> "Control and cost reduction structure the division of labour, involving the design of work and the division of tasks and people to give the most effective control and profitability. This is sustained by hierarchical structures and the shaping of appropriate forms of science and technology." (Thompson and McHugh, 2002: 367)

Alongside these broad structural features of capitalist work organization, LPT acknowledges that empirically observed situations will also reflect a host of local factors specific to firms, markets, institutional contexts, the ideologies of the various actors, and the history of their interrelations. Labor process theory does not deny the importance of these local factors, but argues that this variation is shaped by the deep structure of capitalist relations of production, and that our theories therefore should acknowledge the layered causality involved.

THE UPGRADING CHALLENGE TO LABOR PROCESS THEORY

From its inception, LPT has been critical of those who claim to see upgrading trends in work and the emergence of "new paradigms" in work organization (e.g.: Bell, 1973; Touraine, 1969; Piore and Sabel, 1984; Kern and Schumann, 1984; Mathews, 1994). The first wave of LPT, based on the reading of Marx just summarized, argued that capitalist imperatives of control and cost reduction led inexorably to "deskilling" – fragmenting jobs, reducing skill requirements, and replacing worker autonomy with management control. Taylorism was thus presented as paradigmatic of capitalist work organization. Numerous studies compellingly described cases of deskilling in various occupations (e.g. Zimbalist, 1979).

Over the years, and confronted with a barrage of counter-examples and counter-arguments, LPT proponents have nuanced their positions — retreating to an increasingly narrow "programmatic core" of theory (Lakatos, 1974). Backing away from deskilling, some labor process theorists embrace a polarization view, according to which some occupations might be upgraded but others, presumably representing the larger number of workers, are deskilled; and others have gone further, to a contingency view according to which outcomes are dependent on the state of the class struggle. Most recently, LPT has been challenged by an ascendant foucauldian post-modernism that has abandoned both references to Marx and efforts to identify long-term trends in skill (e.g.: Covaleski, Dirsmith, Heian and Samuel 1998; Ezzamel and Willmott, 1998; Knights and Willmott, 1989; see neo-Marxist critiques by Thompson and Smith, 2001, and Tinker, 2002). This new approach has shifted focus from the broader structural features of capitalism to a richer portrait of subjective experience.

One reason, I submit, for the loss of centrality of LPT and its Marxist premises is the *prima facia* implausibility of its central claim of deskilling. On the neo-Marxist view outlined above, it is inconceivable that over the longer term and in the aggregate, job characteristics could have trended upward. Yet not only are upgrading counter-examples common in the literature, but the overall evolution of work organization under capitalism in the last century is increasingly presented in empirical studies as one of upgrading (Form, 1987; Attewell, 1987).

Consider, first, the evolution of the occupational distribution of the workforce. Exhibit 1 shows data on the case of the U.S. over the 20th century. There are, of course, many difficulties in interpreting these data, but it is hard not to see in this mutation of the occupational structure the reflection of an upgrading trend, notably in the massive contraction of the unskilled farm and non-farm laborer category, the more recent contraction of the operative category, and the growth of the professional and technical category. (We might note too that many people classified in the growing category of managers and administrators have very little managerial authority and arguably belong to the working-class broadly construed.)

[put Exhibit 1 about here]

How do Marxist-inspired labor process theorists respond to data such as these? Braverman (1974, Ch. 20) anticipated the most common responses. He suggested we simply ignore such occupational data, because (a) they do not recognize the experience-based skills of farmers and farm laborers; (b) commentators often inflate the skills of manufacturing operatives, classifying them as semi-skilled merely because they work with machinery, while classifying laborers as unskilled merely because they do not; (c) the data ignore the class difference between middle-class professional/technical categories and the working-class narrowly construed; and (d) they mask the dilution over time of skills in the craft category.

While there is arguably some truth to all these objections, it nevertheless takes a huge effort of imagination to see the shift registered in these statistics as compatible with an aggregate deskilling story. Deskilling within each of the occupational categories in Exhibit 1 would have had to be pervasive to outweigh the upgrading effect of the compositional change traced in the Exhibit – where in fact the evidence points towards upgrading within most occupations too. Where scholars have been able to use independent measures of skill such as the Dictionary of Occupational Titles, none has found evidence of aggregate deskilling a modest upgrading trend is the almost universal conclusion (see the comprehensive review of U.S. studies by Spenner, 1988.

Second, consider the average education level of the work force — arguably an important indicator of skill requirements: it, too, has increased dramatically. The fraction of U.S. 17 year-olds who had completed high-school grew from 6% in 1900, to 57% in 1950, to over 80% by the end of the century. Braverman (1974) suggests we ignore this evidence too, since (a) it reflects the demands of urbanization rather than industry; (b) it is biased by the inclusion of non-working-class categories; (c) school is a way to keep unemployed youth off the streets; and (d) many workers' education is under-utilized.

Again, these points all have some validity; however, despite this huge increase in the supply of more-educated labor, high-school and college education has continued to yield a sizable positive economic return in the labor market (Goldin and Katz, 1999), and this result is difficult to understand unless at least some of this increase in education levels reflected increasing skill requirements rather than pure screening and credentialism (Abramowitz and David, 1996). As Goldin and Katz write, the most plausible explanation for this pattern is that "technological change and capital deepening have both served to increase the demand for more-skilled labor over the long run" (1999: 25-26). A considerable body of economic research has consistently found that capital equipment and worker skills are complements rather than substitutes (Goldin and Katz, 1998).

Faced with evidence such as this, it is not surprising that neo-Maxist LPT has shifted away from the strong version of the deskilling thesis advanced by Braverman. A somewhat weaker version is expressed in the polarization thesis. Braverman (1974: 425) invoked polarization to show that his deskilling thesis was compatible with an increase in average skill levels, by arguing that the categories that were upgraded were not truly working-class: "The mass of workers gain nothing from the fact that the decline in their command over the labor process is more than compensated for by the increasing command on the part of managers and engineers" (1974: 425). Later writers allowed that some working-class categories too might experience upgrading, but still wanted to argue that the mass at the bottom of the skill distribution was growing in size and was further degraded in skills. However the data cited by Spenner (1988) do not support any such polarization stories. To restrict our vision to manufacturing for a moment, polarization is difficult to reconcile with Steiger's (1999) finding that the proportion of skilled workers among production workers increased between 1950 and 1990 in 16 of 19 manufacturing industries. If there has been polarization, it is not because one part of the work force experienced deskilling while another part experienced upgrading, but rather because those at the high end of the skill distribution experienced faster upgrading than those at the low end.

Faced with evidence such as this, it is not surprising that neo-Maxist LPT has shifted away from broad trend generalizations towards a contingency view. Summarizing the contingency view, Smith and Thompson write:

> "LPT is not dependent on deskilling or Taylorism as the characteristic form of the capitalist labor process. Its core theory merely recognizes that competitive relations compel capital to constantly revolutionize the labor process and that within that framework, capital and labor will contest the character and consequences of such changes" (Smith and Thompson, 1999: 211).

Compared to the deskilling argument, such a contingency view is easier to reconcile with the data just summarized — but it is harder to reconcile with LPT's ostensible Marxist grounding. It is one thing to argue that workers sometimes succeed in forcing management to upgrade jobs and sometimes succeed in forcing government to provide greater access to education. But the idea that the balance of class power should be so favorable to workers over such large aggregates and over such a long period is difficult to reconcile with any theory that would characterize contemporary society as basically capitalist. If the data do show a long-term, aggregate upgrading trend, any theory claiming a Marxist lineage must attribute primary causality to capitalist industry's need for skilled labor.

To date, those who have seen in the data evidence of broad upgrading trends have usually distanced themselves from Marx. Exceptions to this generalization are rare: Hirschhorn (1984), Kenney and Florida (1993), and Engestrom (1987; 1990) are exemplary. Much of the upgrading literature has often simply ignored the scandalously large mass of low-skilled workers that still anchors the bottom of the occupational skill distribution. Many write about the long-term trends they claim to discern as if this mass were about to disappear overnight. Some recent champions of the "knowledge society," for example, write as if we will all shortly be "symbolic analysts," while, in reality, low-skilled, routine jobs continue to proliferate (R. Reich, 1991; US Bureau of Census Statistical Abstract, 2000: 419).

We thus have on the one side a utopian optimism that ends up masking a scandal, and on the other side a polemical denunciation of this scandal that seems unable to acknowledge some basic facts. There is, however, a version of Marx – one I call "paleo-Marxist" – that is easy to reconcile with *both* a broad pattern of upgrading *and* a multitude of counter-examples of deskilling. I call this view *paleo* not to signal any pejorative connotation — on the contrary, I will argue that it is the more fruitful interpretation — but simply to signal the fact that it was common prior to World War I but was subsequently eclipsed by neo-Marxism. This essay explicates that paleo point of view, arguing that capitalism progressively upgrades work as part of the process that Marx called the socialization of the forces of production.

READING MARX

LPT, like other theories that take inspiration from Marx, takes as its starting point the proposition that:

> "the development of the contradictions of a given historical form of production is the only historical way in which it can be dissolved and then reconstructed on a new basis" (Marx, 1977: 619)

The term "contradiction" is here used in a Hegelian sense, to designate incompatibilities between real forces rather than merely between logical propositions. For LPT and other neo-Marxists, class struggle is the motor of history, and the development of the contradictions of capitalism consists of intensified worker struggles in reaction to exacerbated exploitation and misery. The more traditional reading of Marx — the paleo-Marxist view — sees the basic contradiction as being between the forces and the relations of production. The forces of production are composed of technology in the form of instruments and materials, and workers' productive faculties; the relations of production, as discussed above, are the relations of ownership and control over the productive forces.

The paleo-Marxist reading accepts all of the elements of labor process theory laid out in the earlier exposition — with the exception of the dismissal of technology as an important causal factor, and as a result, comes to a different conclusion regarding the vector of change in work organization. (My exposition of the paleo reading is based on G.A. Cohen's (1978) presentation of Marx's theory of history. Cohen's version of Marx has been criticized by, amongst others, Levine and Wright (1980), and J. Cohen (1982); see G. A. Cohen's (1988) reply, also Wright, Levine and Sober (1992). This essay takes G. A. Cohen's interpretation from the general societal plane into the production process.)

On the paleo view, the long-term path of development of the class struggle is determined by the evolution of the underlying contradiction between the "socialization" tendency of the forces of production and the persistence of private-property based relations of production (Engels, 1978). Socialization is commonly construed as the process whereby people new to a culture internalize its norms: Marx's use is broader. Marx's discussion of the socialization of the forces of production (e.g.: Marx, 1973: 705; 1977: 1024) (as distinct from his arguments in favor of the socialization of property relations through nationalizations) suggests that this psychological internalization is just one form of a more general phenomenon: the forces of production are socialized insofar as they come to embody the capabilities and constraints developed in the larger society rather than only those that emerge from isolated, local contexts.

The "objective" socialization of the forces of production is visible at the societal level in the complexification of the social division of labor — the specialization of industries and regions, and their increasing global interdependence (see also van der Pijl, 1998, Sohn-Rethel, 1978; Engels, 1978). At the enterprise level — where society's forces of production are instantiated as specific labor processes — objective socialization was characterized by Engels (1978: 702) in these terms:

> "Before capitalist production. i.e. in the Middle Ages. […] the instruments of labor – land, agricultural implements, the workshop, the tool – were the instruments of labor of single individuals, adapted for the use of one worker [… The bourgeoisie transformed these productive forces] from means of production of the individual into *social* means of production, workable only by a collectivity of men. The spinning-wheel, the hand-loom, the blacksmith's hammer were replaced by the spinning-machine, the power-loom, the steam-hammer; the individual workshop, by the factory, implying the cooperation of hundreds and thousands of workmen. In like manner, production itself changed from a series of individual into a series of social acts."

To these "objective" dimensions of socialization corresponds a subjective dimension -- to reprise the conventional meaning of socialization, and indeed, to recover an important intersection between paleo-Marxism and the foucauldian arguments about identity and subjectivity. When the effective subject of production is no longer an individual worker but the "collective worker," workers' identities change – workers are re-socialized. (Recall that in Marx's view human nature is nothing but "the ensemble of the social relations" -- Marx, 1975: 423.) Socialization in this subjective sense can be understood as the emergence of more "interdependent self-construals" (Markus and Kitayama, 1991). The civilizing mission of capitalism is not only to stimulate enormously the quantitative development of the objective components of the forces of production, but also to take a decisive step in the realization of humankind's fundamentally social nature:

> "When the worker cooperates in a planned way with others, he strips off the fetters of his individuality, and develop the capabilities of his species" (Marx, 1977: 447)

The *Communist Manifesto* is eloquent on this interweaving of objective and subjective aspects of socialization:

> "The bourgeoisie, historically, has played a most revolutionary part. The bourgeoisie, wherever it has got the upper hand, has put an end to all feudal, patriarchal, idyllic relations. …In place of the old local and national seclusion and self-sufficiency, we have intercourse in every direction, universal interdependence. … And as in material, so in intellectual production. The intellectual creations of individual nations become common property. National one-sidedness and narrow-mindedness become more and more impossible… The bourgeoisie … has rescued a considerable part of the population from the idiocy of rural life. … The bourgeoisie cannot exist without constantly revolutionizing the instruments of production. … [W]ith the development of industry, the proletariat not only increases in numbers; it becomes more concentrated in greater masses, its strength grows, and it feels that strength more….Thereupon the workers begin to form combinations (trade unions)…. Now and then the workers are victorious, but only for a time. The real fruit of their battles lies not in the immediate result, but in the ever-expanding union of the workers. The union is helped on by the improved means of communication that are created by modern industry and that place the workers of different localities in contact with one another…. The advance of industry, whose involuntary promoter is the bourgeoisie, replaces the isolation of laborers, due to competition, by their revolutionary combination, due to association…. What the bourgeoisie, therefore, produces, above all, is its gravediggers" (Marx and Engels, 1959)

The development of the forces of production pulls workers out of what Marx and Engels call "rural idiocy." In the *Poverty of Philosophy*, Marx similarly celebrates the end of "craft idiocy." Marx's use of the term idiocy preserves both its colloquial sense and the meaning from the Greek *idiotes*, denoting an asocial individual isolated from the polis. At the opposite end of the spectrum from the *idiotes* – in the form of the unskilled worker or the craftsman -- is the "social individual" described by the *Grundrisse* – in the form of the technician who accesses and deploys society's accumulated scientific and technological knowledge:

> "to the degree that large industry develops...it is neither the direct human labor he himself performs, nor the time during which he works, but rather the appropriation of his own general productive power, his understanding of nature and his mastery over it by virtue of his presence as a social body -- it is, in a word, the development of the social individual which appears as the great foundation-stone of production and of wealth." (Marx, 1973: 704-706)

Under capitalism, this socialization tendency is simultaneously stimulated, retarded, and distorted by the prevailing relations of production. Competitive pressures force firms to break down parochialisms and to stimulate technological progress; but instead of a broadening association of producers progressively mastering their collective future, capitalism imposes the coercion of quasi-natural laws of the market over firms and the despotism of corporate bureaucracy over workers. The limitations on collective mastery that result from the dominance of the market over firms are visible in capitalism's inability to manage public goods and externalities. The limitations resulting from the despotic authority of managers over workers within firms is visible in the Sisyphean nature of corporate human resource management strategies — condemned to futility by the capitalist firm's need for workers who are simultaneously dependable and disposable (Hyman, 1987). These handicaps become increasingly intolerable fetters on social development with the increasing complexity of technology and the growing knowledge-intensity of the economy (Adler, 2001).

However, in the overall dynamics of capitalism, these various constraints must and do slowly cede to the overall progress of socialization. In modern industry, competitive advantage often flows from skill upgrading and from greater collaborative interdependence within and between firms. The pursuit of those sources of competitive advantage makes capitalists the "involuntary promoters" of socialization – to use the phrase quoted above from the Communist Manifesto (see Cohen, 1978; Levine and Wright, 1980). Amongst workers, the overall, long-term, aggregate effect – Marx's logic encourages us to conjecture – is to foster upgrading: greater intellectual sophistication and broader worldviews.

As a result of the persistence of capitalist relations of production, the path of socialization, both objective and subjective, is halting and uneven. Globalization integrates markets, but by whipsawing regions against each other. Management mobilizes the collective worker, but then finds profits dependent on savage downsizing and outsourcing. There is a long-term upgrading trend, but firms often find the low road of deskilling and super-exploitation too tempting. We should also note that under capitalist conditions, even progressive change has social costs: as technologies advance and markets integrate, workers often bear the burdens of structural unemployment; old union craftsmen are often pitted against young nonunion technicians; contracting-out and globalization often undermine old solidarities.

In analyzing the evolution of skill and work organization, LPT has truncated this dialectic. According to Marx, the production process within capitalist firms has two aspects (reflecting the two aspects of the commodity): the *labor process,* in which workers, tools, and materials are combined to create new use-values, and the *valorization process,* in which these use-values appear in the form of exchange-values, and in which the operative considerations are not technical but monetary -- wages, capital, and profit (Marx, 1977: Appendix; Thompson, 1989; Bottomore, 1983: 267-270). How then should we understand the relations between these two aspects of production? Marx writes:

> "If capitalist direction [of work] is thus twofold in content, owing to the twofold nature of the process of production which has to be directed -- on the one hand a social labor process for the creation of a product, and on the other hand capital's process of valorization -- in form it is purely despotic" (Marx, 1977: 450)

Neo-Marxist LPT interprets this passage to mean that the historical development of capitalist work organization reflects above all the balance of class forces -- despotism versus resistance. Paleo-Marxists, by contrast, recall that in Marx's Hegelian discourse, content and form can be in contradiction with each other. The paleo reading of this passage thus highlights the growing contradiction between an increasingly socialized labor process (the content) and the barriers posed to further socialization by the persistence of valorization constraints (the form).

The following sections develop and deploy an Activity Theory framework for understanding the socialization of the labor process and its dialectical contradiction with capitalist property relations. My argument is that these paleo-Marxist Activity Theory lenses are useful for studying work and its history because they enable us to grasp the significance of a real but uneven trend of upgrading as a reflection of the deepest contradictions of capitalism. This viewpoint allows us to grasp the contradictions that beset capitalist management, and to grasp them in a surprisingly intuitive way: on the one hand, management needs and cultivates the productive power of the collective worker, and on the other, management limits this development due to pressures of corporate profitability.

AN ACTIVITY THEORY FRAMEWORK

If our task is to understand work and its relation to organization structure and to lived experience, Marx's model of the labor process provides a fruitful starting point. The main elements of the labor process are, according to Marx, "(1) purposeful activity, that is work itself, (2) the object on which that work is performed, and (3) the instruments of that work" (Marx, 1977: 284). Like Cohen (1978: Ch. 3), I take the first element to refer more specifically to the worker's productive capabilities. I also make one further amendment by differentiating a fourth element: the community within which the worker works and which shares the object of that work. The resulting model is summarized in Exhibit 2. (It can be read as a stripped down version of the model proposed by Engestrom, 1987, 1990, and discussed by Blackler, 1993; Holt and Morris, 1993).

[put Exhibit 2 about here]

Some comments will help situate this model. Taking as a baseline the more common schema in psychology, in which object and subject appear alone, as stimulus and response, note, first, that for Marx, human activity is tool-mediated activity -- where tools include both material tools such as hammers and symbolic ones such as language and concepts (Vygotsky, 1962, 1978). Second, the appropriate unit of analysis for the study of work is not the quasi-automatic reflex *operation*, nor the discrete, goal-oriented, individual *action*, but rather the *activity* understood as a motivated collective endeavor. And in such collective activity, the subject's relation to the object is mediated not only by tools but also by community (Leont'ev, 1978). Third, in Marx's analysis, the "object" of work (in German: *Gegenstand*) includes both of the dictionary senses: the material on which the work is performed (in German: *Objekt*) and the intended goal of the activity. With this notion, Marx attempts to overcome both simplistic materialism – which accords insufficient place to intentionality in shaping the object – and classical idealism – which accords insufficient recognition to the obdurate nature of the object's materiality (Marx, 1959: 243; Engestrom, 1987; Foot, 2002).

Exhibit 2 is a model of what Marx calls "production in general" (Marx, 1973: 85): it is trans-historical insofar as it does not acknowledge any more historically-specific, "concrete" determinations. Since we are here trying to develop a theory of capitalist work organization, the most basic of these relevant to our present task are the determinations characteristic of the capitalist form of society. Of these latter, most fundamental is the contradiction between use-value and exchange-value of which the commodity is the germ. This contradiction is reflected in each of the elements of the labor process. The object of work is both the creation of useful things (use-values) and the generation of profit (exchange-value). Tools and community are both means of technical accomplishment and means of extracting useful labor from potentially recalcitrant labor-power. The worker is simultaneously a creative member of the collective worker and a disposable, variable-cost, budget item under another's control. More generally, viewed as use-value, each element participates in the socialization process; viewed as exchange-value, each is subordinate to the valorization process's profit imperative. In the neo-Marxist view, these contradictions are only virtual, since the use-value content disappears behind its exchange-value form; but in the paleo view, these contradictions are real, driving the evolution of work organization. The following section illustrates the potential fruitfulness of theorizing work organization though such lenses.

A CASE: SOFTWARE DEVELOPMENT

As software has grown more complex over the past few decades, the software development process has slid into chaos (Gibbs, 1994; Lieberman and Fry, 2001). One 1994 survey of 8,330 projects in 365 firms in banking, manufacturing, retail, wholesale, healthcare, insurance, and government found that only 16% of projects were on time, within budget, and met originally specified requirements — only 9% in large companies. Some 31% of projects were "impaired" and eventually cancelled. Approximately 53% of projects were "challenged," and the average challenged project met only 61% of its requirements. The average impaired or challenged project was 189% over budget and 222% over schedule (Standish Group, 1994).

It is therefore not surprising that over this same period, the software field has been the object of numerous rationalization efforts (Cusumano, 1991; Swanson et al, 1991; Griss, 1993: Weber, 1997; Friedman and Cornford, 1989). Examples include structured programming, project planning models, information engineering, and object-oriented programming. Currently, one of the most influential of these efforts is that based on the "Capability Maturity Model" (CMM) (see Software Engineering Institute, 2002). (I leave for another occasion discussion of other lines of evolution in the software field.)

The CMM owes its birth to the U.S. Department of Defense's increasing frustration with chaos in defense systems software development (Humphrey, 2002). The Department of Defense (DoD) funded the Software Engineering Institute (SEI), based at Carnegie-Mellon University, to develop a model of a more reliable development process. With the assistance of the MITRE Corporation and with input from nearly 1000 industry people, SEI released the CMM in 1991. The model – summarized in Exhibit 3 -- distinguishes five successively more "mature" levels of process capability, each characterized by mastery of a number of Key Process Areas (KPAs). Level 1 represents an entirely ad hoc approach. Level 2 represents the rationalization of the management of individual projects. Level 3 characterizes the systematic management of its portfolio of projects. Levels 4 addresses the quantification of the development process. Level 5 addresses the continuous improvement of that process. The underlying philosophy of this hierarchy was inspired by Crosby's (1979) TQM approach to quality in manufacturing (Humphrey, 2002).

[put Exhibit 3 about here]

The CMM has become the basis for numerous software service organizations' improvement efforts in both the government and commercial sectors. Its diffusion has been driven in considerable measure by its use in sourcing decisions by the DoD and other government and commercial-sector organizations. The first sourcing evaluations pressed suppliers to reach Level 2, but by the late 1990s, the bar had been raised to Level 3.

Accumulating evidence suggests that moving up the CMM hierarchy leads to improvements in product cost, quality, and timeliness (Clark, 1999; Harter, Krishnan, Slaughter, 2000; Krishnan, Kriebel, Kekre, Mukhopadhyay, 2000; Herbsleb, Zubrow, Goldenson, Hayes, Paulk, 1997). But many skeptics remain unconvinced (e.g. Crocca, 1992; Bach, 1994, 1995; Conradi and Fuggetta, 2002; Lynn, 1991; Ngwenyama and Nielson, 2003). Gains may be specific to the sampled organizations. They may be earned at the expense of developer morale and commitment, and given the importance of developers' attitudes to performance, any performance gains may therefore be ephemeral. Typical of opposition to such standardized and formalized methodologies is this assessment by two well-respected software management experts:

> "Of course, if your people aren't smart enough to think their way through their work, the work will fail. No Methodology will help. Worse still, Methodologies can do grievous damage to efforts in which people are fully competent. They do this by trying to force the work into a fixed mold that guarantees a morass of paperwork, a paucity of methods, an absence of responsibility, and a general loss of motivation." (DeMarco and Lister, 1987, p. 116)

Most LPT research on these kinds of efforts to rationalize software development has interpreted them as mechanisms of exploitative control (Kraft, 1977; Greenbaum, 1979; Friedman and Cornford, 1989; Prasad, 1998). Kraft (1977: 61) summarizes the analysis this way:

> "Canned programs, structured programming, and modularization are designed to make the supervision of software workers by managers easier and more like the supervision of other workers…Such managerial techniques have made possible the use of relatively less skilled programmers for what were formerly the most complex software tasks."

Some of the more recent LPT research on software has nuanced this analysis, moving closer to a contingency view (Greenbaum, 1998; Beirne et al., 1998).

To explore the impact of the CMM, I studied a large software consulting firm I will call GCC, conducting interviews with developers and managers in four programs (a program is an organizational unit devoted to a series of projects with a single large customer) (see Adler, 2003). These programs all developed and maintained relatively large-scale systems for government clients. Two programs were at CMM Level 5: Program A, which had recently downsized due to changing client needs from 1600 to 450 people, and Program C, with 450 people. Two sister programs were at Level 3: Program B with 215 people, and Program D with 470. In late 1999, I interviewed between 15 and 22 people at various hierarchical levels and in various functions in each of these four programs. (Note: for simplicity, I will refer to all employees directly involved with software development –

specifying customer requirements, programming, testing – as “developers,” to distinguish them from support staff in Quality Assurance and other such functions, and from supervisors and managers.) Interviews lasted approximately one hour. They were tape-recorded and interviewees were assured anonymity. The recordings were transcribed, and edited versions were sent back to interviewees for review and correction. I also consulted voluminous internal documentation from each of these programs as well as documents from corporate entities supporting them.

Socialization of the development process

Traditionally, at the lowest levels of process maturity, developers enjoyed considerable autonomy, task variety, and task identity. Greenbaum (1979: 64-4) quotes a veteran programmer thus:

> “I remember that in the fifties and early sixties I was a ‘jack of all trades.’ As a programmer I got to deal with the whole process. I would think through a problem, talk to the clients, write my own code, and operate the machine. I loved it — particularly the chance to see something through from beginning to end.”

The labor processes within which developers worked were largely local. Kraft (1977: 56) writes of this period: “Programmers (and analysts) followed a logic and procedures which were largely of their own making,” Being tacit rather than codified, tools were difficult to communicate across locales and skills were not easily transferable. Working knowledge was in these senses private rather than social. As one of Greenbaum’s interviewees put it:

> “No one knew what was going on — certainly not the managers. But even the programmers and systems analysts were confused. There were no standards for doing anything — coding, testing, documenting — they were all done the way each person felt like it, or in fact, they were not done at all. [...] Programmers never documented what it was their program was to do. It was the same with setting up testing procedures and test data. When the whole system was put together, we never knew if it really worked because nothing got written down.” (Greenbaum, 1979: 73-7)

At higher levels of process maturity, developers were embedded in larger social aggregates, and encountered pre-specified methods that were the fruit of a complex, organized, large-scale process development effort. Tools, materials, community, and skills were no longer naturally emergent phenomena grounded in local experience. They were formalized and standardized. Developers were aware that their effectiveness was the not only the result of their own individual effort and skill and of informally shared tricks of the trade, but also and increasingly the result of this social, rather than private, accumulation of working knowledge:

> “I came from a background in industrial process computers and the organization I worked for was much less structured in how they handled all this. The process was basically just define the requirements, write the code, then do a final test. Apart from that, you were basically on your own. Here the processes tell you a lot more about how to do the work. [...] Previously, it was more like a ‘hand-me-down’ — you learned how to do your work with some help from other people on the job, or just by yourself.” (B: development)

At first, this socialization took a form many developers experienced as alienating, coercive, bureaucratic authority. Discussing the Military Standards for software quality control that come into force in the mid-1980s, one veteran noted that: “[Military Standard] 2167A was supposed to make coding a no-brainer” (D: development manager). In the civilian Program A too, the initial experience with process was top-down, oriented to conformance, and “most managers felt that it was just a matter of ensuring that people were implementing it” (A: program manager).

By the time of my study a decade or more later, the Level 5 Programs had pushed the socialization of the production process considerably further, both in its extent and its form. The term “software process” was now used to refer to a whole hierarchy of standard operating procedures, from “Policies” defining broad, corporate requirements down to “Instructions” defining individual tasks. The “granularity” of process at its finest levels can be gauged by the Instructions at one of the Level 5 programs, Program C. There were separate Instructions that covered high-level design, two types of low-level design, two types of code reviews, one for testing, as well as Instructions for filling out

Change Request Implementation forms and Root Cause Analysis forms. Each Instruction was several pages in length. They often included the specific forms to be completed as well as flow-charts detailing the sequence of associated tasks. Overall, the process documentation summed to some eight linear-feet of shelf space. In recent years, almost all of this documentation had been put on-line, along with a host of other management information and communication tools. Prescribed work-flows were being built into automated document routing systems.

If the documentation that developers were required to read was voluminous, so too was the documentation that they were required to write. In the words of one interviewee (perhaps exaggerating for dramatic effect):

> "I can write the code in two hours, but then I have spend two days documenting it! It can be very frustrating. We have to document check-in and check-out, a detail design plan, a development plan. We have to print out all the differences between the old and the new code. There's documentation for inspection and certification. There's an internal software delivery form. A test plan. And all these need to be signed. […] I used to be an independent developer for about three years. I never even created a flow-chart of my work! The only documentation I needed was a 'to do' list. So I had to change of lot of habits when I got here." (B: development)

Over the previous decade, the socialization of the development process had also taken a progressively more enabling form. Interdependence was no longer a function of bureaucratically imposed authority but experienced as collaboration within the collective worker:

> "Where I used to work before I came to GCC, the development process was entirely up to me and my manager. What I did, when I did it, what it was going to look like when it was done, and so forth, was all up to me. It was very informal. Here everything is very different. It's much more rigid. It's much more formal. A lot of people lay out the schedule, the entire functionality, and what I'm going to be accountable for — before I even get involved. […]
>
> When I got here I was kind of shocked. Right off, it was 'Here are your Instructions.' 'So what does this tell me?' 'It tells you how to do your job.' I thought I was bringing the know-how I'd need to do my job. But sure enough, you open up the Instructions, and they tell you how to do your job: how to lay the code out, where on the form to write a change request number, and so on. I was shocked.
>
> But I can see the need now. Now I'm just one of 30 or 40 other people who may need to work on this code, so we need a change request number that everyone can use to identify it. It certainly feels restrictive at first. They explained the Instructions and the whole Program C process to us in our orientation seminar, but it's hard to see the value of it until you've been around a while. Now I can see that it makes things much easier in the long run.
>
> I hate to say it. As a developer, I'm pretty allergic to all this paperwork. It's so time-consuming. But it does help. You've got to keep in mind, too, that by the time we see the Instructions, they've been through a lot of revision and refinement. So they're pretty much on target." (C: development)

Socialization was visible in all four elements of the labor process, as discussed in the following subsections.

Tools. Formalized process created a common vocabulary – a key tool for collaboration:

> "In a Level 1 organization, one without a common process, even one where there was a lot of goodwill between the functions, they wouldn't have the common vocabulary, or common definitions of key tasks, and everything would be subject to conflicting interpretation, so people would be fumbling in the dark. A common process greatly simplifies things." (C: project manager)

Memory had become objectified and collective, augmenting the stock of available tools:

> "Process gives people access to assets from prior work — for estimation, for standards and procedures, and for lessons learned. In our asset library, we keep the standards and procedures of all our projects, and project managers refer back to these to use as templates. We encourage people to share and borrow." (A: quality assurance)

> "Take for example our internal software delivery procedure. At first, developers thought that this was just more burdensome paperwork. But soon they found it was a great memory system." (B: quality assurance)

The formalized process was seen as an enabling tool to the extent it leveraged similarities across projects for more effective work:

> "[E]ven when tasks are more innovative, you can still get a lot of advantage from process. You need to sit down and list the features you want in the system, and then work out which are similar and which are different from your previous work. And you need to make sure the differences are real ones. You'll discover that even in very innovative projects, most of the tasks are ones you've done many times before. Then, for the tasks that are truly novel, you can still leverage your prior experience by identifying somewhat related tasks and defining appropriate guidelines based on those similarities. They won't be precise instructions of the kind you'll have for the truly repetitive work: but these guidelines can be a very useful way to bring your experience to bear on the creative parts of the work." (B: testing, formerly with A).

Statistical process control tools were used to systematically improve the process:

> "We used to be a group of hackers. If we'd have had to rebuild a system, we simply wouldn't have been able to do it because we wouldn't have had the documents. We've come a long way from that! Now we function according to a defined process and we collect data on ourselves so we can do defect causal analysis to drive continuous improvement." (A: quality assurance)

All four types of tools were distinctive in being the fruit of a socialized development process rather than locally emergent.

The CMM also functioned as a kind of tool: in Engestrom's terminology, the CMM functioned as a "more advanced model" of software development that was used to guide improvement efforts. As such, the CMM too represented a highly socialized tool, insofar as it was seen as an "industry-validated approach" rather than merely a local initiative:

> "The CMM is helping us move ahead. Just to take an example, even if the CMM didn't exist we would need a technology change management process [a Level 5 KPA]. Of our 450 people, we have about 50 people in CM, QA, and data management. To move them from one process to another is sometimes like herding cats! The CMM helps us in that by providing an industry-validated approach." (C: program manager)

Object. The object of work was brought into clearer focus and stabilized by the collective discipline of the process:

> "Our policies and procedures mean that I have better information on what we're trying to do because we have better requirements documents and better information on how to do it with Instructions etc. At Level 5 versus Level 1, I'm more confident we're all playing to the same sheet of music. Looking across the organization, process also means that managers understand better the way the whole system works, so they are all playing the same game." (C: development manager)

The object was expanded socially and temporally to include other people who would work on the software:

> "I think that our process — and even the paperwork part of it — is basically a good thing. My documentation is going to help the next person working on this code, either for testing or maintenance. And vice versa when I'm on the receiving end." (C: development)

The object was also expanded technically: the process itself became an object of developers' work:

> "Perhaps the biggest change as we've become more process mature is that it makes everyone more interested in process improvement. Take an example: now I'm working on a new software utility. Top management asked us to evaluate it, to see if we should all use it. So I've been facilitating a series of meetings with all the managers, where everyone is talking about the utilities they are using and the problems they're having. It's been great to see this kind of

> problem-solving work going on. That's the effect of having a defined technology change management process [a CMM Level 5 KPA]. CMM got this process going for us." (D: logistics)

Community. Process maturity also brought greater rationality, both formal and substantive, to the structuring of the collective worker, in three sets of relations: staff/line, horizontal, and vertical.

With process maturity, new staff functions such as Configuration Management and Process Engineering emerged, and new line/staff relations were created. Quality Assurance (QA) illustrates the new relations. In the past, QA was often remote from the daily work of developers, arriving on the scene at the end of the work cycle to inspect the output. QA's role evolved with process maturity to (a) a greater focus on process quality rather than only product quality, (b) greater responsibility for infusing process rather than only auditing it, and (c) a closer and more collaborative relation with the line departments:

> "QA is not a policeman! QA is there to help the project identify the processes you need, tailor existing ones to your needs, learn that process, and do a check to see if you're using it. If I find a problem, it's my job to help the project work out how to address it and how I can help." (B: quality assurance)

In the community's horizontal relations, greater specialization went hand in hand with more systematic coordination and integration:

> "Process means that people play more specialized, defined roles, but also that these specialists get involved earlier and longer as contributors to other people's tasks. If we analyzed the way a coder uses their time, and compared it with comparable data from, say, 15 years ago, we'd find the coder doing less coding because of more automated tools. They'd be spending more time documenting their code, both as it was being built and afterwards in users' guides. They'd be spending more time in peer reviews. And they'd be spending more time in design meetings and test plan meetings. As for testers […] now the testers are more involved in system concept definition and requirement definition activities." (A: quality assurance)

Organization-wide processes of coordination were made more visible:

> "A well-defined process gives you a kind of map of the whole enterprise." (B: quality assurance)

> "The overall process is more intelligible now. All the organization charts, the people, the processes and documents, and the minutes of various groups are on the website." (C: program manager)

The collective worker expanded to encompass the client organization:

> "There's a great focus now on 'accountability' all through the system. We are expected to be more aggressive in pushing back when things are inconsistent with our processes. And that goes down to our project managers. Instead of simply supporting our customer management counterparts, the project managers have to be willing to push back. That's changed the tone of some of our monthly unit review meetings with the customer. This culture change goes right down to the staff. In general, we try to buffer the staff from these issues, but if they get instructions that violate our processes, they have to push back too." (C: program manager)

In the community's vertical structuring, formalized process meant that the parochial concerns of subgroups and individuals and the resulting conflicts were drawn into the open. These concerns became the objects of collective scrutiny and thus less covert:

> "We say it's important to document software errors, but that's hard to sell. Developers are used to just doing the corrections, and the testers hate the documentation too. But we try to sell the testers on this by explaining that this way they can get credit for the problems they find. And we try to explain to them that if we document the errors, we can track them, and if errors recur, we can find root causes. That will help us convince the developers for example, that a given module has too many problems. When it's documented, it's less personal, and it helps the dialogue with the developers. But you also have to ensure that managers won't use the data punitively." (B: testing, formerly with A)

> "I think formalized process and metrics can give autocratic managers a club. But it also gives subordinates training and understanding, so it makes the organization less dependent on that

> manager: he can be replaced more easily. Before I came to GCC, I worked for one of the most autocratic managers you can find. It was always, 'And I want that report on my desk by 5 p.m. today,' with no explanation or rationale. Compared to that kind of situation, an organization with a more mature process leaves a lot less room for a manager to arbitrarily dictate how you should work and when work is due. And a more mature process also means that there are more formal review points, so any arbitrary autocratic behavior by a manager will become visible pretty quickly." (D: quality assurance)

At higher levels of process maturity, formal procedures were more numerous, but developers had more opportunity to participate in defining and refining them. Through a formalized "Tailoring Cycle," software development standards and procedures ("S&Ps," of which there were over 100 at Program A) were modified for each project with the participation of the developers themselves:

> "People have to be a part of defining the process. We always say that 'People support what they help create.' That's why the Tailoring Cycle is so important. As a project manager, you're too far away from the technical work to define the S&Ps yourself, so you have to involve the experts. You don't need everyone involved, but you do need your key people. It's only by involving them that you can be confident you have good S&Ps that have credibility in the eyes of their peers." (A: project manager)

> "When S&Ps are chosen for a project, the rule is that they have to be sent out to everyone affected for review. And sometimes we give some pretty negative feedback! I remember I wrote on one draft, 'Hey, you've forgotten to tell us how to get out of bed in the morning and how to brush our teeth!' It was way too detailed and rigid. Those kinds of things get shot down pretty quickly. Over a period of years, people learned how to write procedures that were reasonable for our work environment. [...]
>
> When I managed software development on one of our bigger projects, I asked all our software developers to help me tailor our S&Ps. The GCC people knew the drill, but we also had some other contractors working on this with us [...] and they would say, 'No, just tell me how you want us to do this.' About a year into the project, I remember one of the contractors who had complained the most about this extra work coming to me to thank me, saying, 'If you'd have written these, I would have just ignored them. But since I helped write them, I've felt duty bound to follow them.'" (A: development)

The Tailoring Cycle was not the only vehicle for participation in process definition. In Programs C and D, Software Engineering Process Groups (SEPGs) also served this purpose. In recent years, the SEPGs had put increasing weight on encouraging suggestions for process improvement from lower-level staff. Moreover, many departments in all four programs had process improvement teams. Whereas these teams were sparse and temporary in the two less mature programs, they were ubiquitous and on-going in the two more mature programs.

Workers. Process encouraged a shift from a traditional form of training — apprenticeship — towards something more systematic. Apprenticeship is a mode of learning that is appropriate and necessary when knowledge is the local, tacit, private property of the artisan-craftsman (see for example Sacks, 1994; Lave, 1988). A more socialized production process relies on forms of knowledge that are more codified and on forms of training that can thus be more rationalized. Going back a couple of decades, this transformation began with the shift to formal university training requirements for development jobs; more recently, under the pressure of CMM, the transformation continued with the further rationalization of the acquisition of firm-specific skills:

> "We've developed a formal mentoring program. There's a checklist of the key processes everyone needs understand, and every new person is assigned a mentor whose job it is to explain each of these in turn. The checklist is audited by QA." (A: testing)

> "We had an informal training and mentoring program, and when we got serious about the CMM, we wrote it down. Writing the process down has had some great benefits. It's made us think about how we work, and that's led to improvements. For example, formalizing the training program has helped bring some outliers into conformance." (C: training)

Through its multiple effects on the other elements of the labor process, process maturity led to a changed subjective identity among developers — towards a more interdependent self-construal (Markus and Kitayama, 1995). The collective worker was not longer merely objective, but now also a lived reality. What mattered to these professionals' self-esteem and identity was now not so much their individual efficacy as their collective efficacy (Bandura, 1997; Gibson and Earley, n.d.). In my interviews, "we" tended to replace "I" as the subject of work, because people increasingly saw themselves as part of a collective effort. The ratio of mentions of "we" to mentions of "I" in my interview notes was 1.83 in Program A and 1.95 in Program C (the two Level 5 programs), and 1.29 in Program B and 1.44 in Program D (the two Level 3 programs). Interviewees expressed their experience of the shift:

> "Here, I'm just a small part of a bigger project team. So you don't do anything on your own. It's a collaborative effort. So there has to be a lot of communication between us. And the process is there to ensure that this communication takes place and to structure it. The process helps keep us all in sync. In a small organization doing small projects, you have a lot of flexibility, but there's not much sharing. You're kind of on your own. Here, I'm just a small part of a bigger project team. So you don't do anything on your own. It's a collaborative effort." (C: development)

> "Developers want above all to deliver a great product, and the process helps us do that. What I've learned coming here is the value of a well thought-out process, rigorously implemented, and continuously improved. It will really improve the quality of the product. In this business, you've got to be exact, and the process ensures that we are. You have to get out of hacker mode!" (A-14)

> "Some programmers here used to be very isolated. We had one fellow who just sat in his cube all day from six in the morning till two in the afternoon. Many of us didn't even know his name! But the process here drew him into team meetings and into new conversations. Eventually we even got him helping with training." (B: development)

More concretely, this new self-construal emerged through a mix of adult socialization (e.g., Kohn and Schooler, 1983) and "attraction-selection-attrition" (Schneider, 1987). On the effects of the former, we have the testimony quoted earlier of "But I can see the need now" (C-13). In Program D, one interviewee described his experience in these terms:

> "I was not originally a believer in this process stuff. I remember seeing coding guidelines when I joined the Program D. I just threw them into a corner. But a year later, I found that my code didn't make it through the code checker, and that got me to reconsider. So I went to some CMM training a few years ago — and I've been converted! Most of the developers and leads are being dragged into process kicking and screaming. Any coder would rather just hack." (D: process engineering)

(See Conn, 2002, for discussion of the process of developer socialization in another software factory.) On the importance of attraction-selection-attrition, two excerpts are illustrative:

> "You won't fit in well here if you don't like structure, you prefer working by yourself, you don't like getting suggestions from other people, or you don't like taking responsibility for your work and for making it better." (A: testing, formerly with A)

> "We still have to deal with the 'free spirits' who don't believe in process. [...] Most of them adapt, although some don't and they leave." (C: process engineering)

A new professionalism

This more interdependent self implied a corresponding mutation in the nature of developers' notion of professionalism. Some aspects of professionalism were preserved, while some were significantly transformed – in a socialized direction.

On the one hand, process leveraged traditional values of professionalism, including the appeal to individual pride in the results of one's own work:

> "We appeal to people's sense of professionalism, saying something like, 'We're all professionals. And as professionals, we're both pretty mobile *and* committed to high quality work. Since I may

> leave here at some point, even soon, it's my duty as a professional to give the organization the documentation it needs to continue serving the customer.'" (B: quality assurance)
>
> "Our process makes for better testing, which means earlier detection of problems, which in turn makes the life of the programmer a lot easier and avoids a lot of embarrassment." (B: department manager, formerly with A)

On the other hand, however, process seemed to encourage a broadening of professionalism, its socialization beyond the closed, "guild" form it has often taken. Whereas traditional conceptions of professionalism give great weight to the individual practitioner's judgment and thus to their autonomy — if not economic autonomy, at least technical decision-making autonomy (Freidson, 2001) — process maturity encouraged the emergence of a more collective professional subject. This mutation is particularly significant because it appeared to moderate the traditional tension between professional autonomy and bureaucratic authority:

> "Usually people run away from audits. But amazingly, recently we've seen several projects volunteering — they want to show off their accomplishments and process capabilities." (A: process engineering)
>
> "The Improvement Team's work [...] made everyone realize that there are real business benefits to sharing information — instead of just worrying about your own rice bowl. I'm your [internal] customer, so I need you to understand my requirements. And the effect has been to make people interested in improving their own operations on their own, even without management being involved or pushing them." (D: logistics)

Professionals tend to have strong ties to and identify with their occupational community (Gouldner, 1957; Van Maanen and Barley, 1984) Interviews suggested that the greater process maturity strengthened both developers' professional-cosmopolitan orientations and their bureaucratic-local orientations. Insofar as the object of their work expanded to include process, at least some developers read more industry journals and attended more conferences, in particular those focused on process issues. Simultaneously, they were drawn into more discussions around these process issues with hierarchical superiors and with staff from other GCC units.

Socialization versus valorization

The progressive socialization of the labor process in software development was simultaneously stimulated, retarded, and distorted by the valorization process. On the one hand, process improvement and CMM certification efforts were stimulated by senior management's conviction that these would help profitability; but on the other hand, progress was limited by (a) contradictions between the pursuit of technical performance (use-value) and profit (exchange-value), (b) the competitive rivalry between firms that undermined their collaboration, (c) the tension between corporate interests and the collective interests of its employees, and (d) the tension between the collective nature of work and the individualizing effects of the wage relation. In aggregate, the centrifugal effects of valorization appear to have been weaker than the centripetal effects of socialization, but the former were strong enough to make progress of socialization halting and uneven.

Technical performance versus profit. Clearly, part of the CMM effort was "for show," responding to symbolic legitimacy pressures rather than technical performance pressures. As such, it sometimes led to a decoupling between formal process and daily practice (as described by Meyer and Rowan, 1977). Comments such as these were common in the two Level 3 Programs but rare in the two Level 5 Programs:

> "We do have written processes, but some are not always used consistently. They are not always being used by the developers. They are not always used by the program managers in their regular reviews." (B: process engineering, formerly with A)
>
> "The evaluation and CMM SCE forced us to update our documents. We didn't really change anything in how we work though." (D: developement)

In part, this symbolic/technical tension reflected a deeper contradiction between use-value and exchange-value. Interviewees were often aware that their labor process improvement efforts were at risk of being overridden by a higher, valorization imperative:

> "One key challenge is maintaining buy-in at the top. Our top corporate management is under constant pressure from the stock market. The market is constantly looking at margins, but Government business has slim margins. That doesn't leave much room for expenditures associated with process improvement — especially when these take two or three years to show any payoff." (C: process engineering)
>
> "As I see it, GCC is a corporation, and that means it's run for the benefit of the major stockholders. So top management is incentivized to maximize dollar profits. Quality is only a means to that end, and in practice, quality sometimes gets compromised. I used to be a technical person, so I know about quality. But now I'm a manager, and I'm under pressure to get the product out — come what may. I just don't have time to worry about the quality of the product. I have a manager of software development under me who's supposed to worry about that." (D: development manager)
>
> "It's hard to convince people that improving the process will help us get or keep business. [Referring to the recent downsizing of Program A:] We had a world-class process, and look what happened to us! Jobs in an organization like this depend a lot more on the vagaries of contracting than on our process excellence." (A: department manager)

The contradiction between socialization and valorization was particularly visible to the interviewees in the form of missed opportunities for process improvement:

> "We could do better at capturing and using lessons learned. We have all the vehicles for doing it — presentations, newsletters, databases. But it takes time. And there are so many competing priorities. In the end, it's all about profit and meeting schedules!" (laughs) (A: project manager)
>
> "We do ask project teams to do a Lessons Learned report at the end of the project. We post the results on the database. But there's no staff support for the process." (A: quality assurance)

This contradiction was also visible in the gap between the expanded object of work as a use-value and limited tools available to employees for tackling this new object:

> "All these forms have a valid purpose, but it takes so long to fill them out that it just doesn't seem very efficient. We really need a lot more automation in doing all this." (B: development)
>
> "There's no doubt that more process maturity means more paperwork. Some of it is good, and some of it is an impediment, especially from a productivity point of view. Unless we have the tools to automate this documentation, it has to slow us down. We still don't have the right tools." (C: project manager)
>
> "The key issue moving forward, I think, is that we still don't have the resources we need to devote to process. A program of this size should have a full-time staff dedicated to our internal process maintenance." (C: quality assurance)

Collaboration versus competitive rivalry. The weight of the valorization imperatives was also visible in tensions that disrupted collaboration with clients:

> "The biggest problem here has been the customer and getting their buy-in. At Program A, our customer grew towards process maturity with us. Here [at Program B], we started with a less mature client. Some of the customer management even told us that they didn't want to hear about QA or our quality management system — they saw it as wasteful overhead. When you bid a project, you specify a budget for QA and so forth, but if they don't want to pay, you have a resource problem. And once you get the contract, then you start dealing with specific project managers within the customer organization, and these managers don't necessarily all believe in QA or simply don't want to pay for it out of their part of the budget. On the Y2K project, the customer kept changing standards and deadlines. Basically, we were dealing with a pretty process-immature customer, and that made it difficult for us to build our process maturity. Things have improved considerably since then." (B: process engineering, formerly with Program A)

Employees' interests versus corporate interests. GCC managers understood that process maturity required a high level of employee participation. But the vertical authority structure that expresses the wage relation created a constant risk that managers would veer off into coercion:

> "We didn't initially have any questions on the employee survey about your boss. Frankly, people were worried that managers might retaliate. But now we do, and we find the data very useful in surfacing management problems. The earlier rounds of the survey did show some big communications problems in some groups. Counseling often helped, and in some cases, we moved people out to other positions." (A: program manager)

Collective nature of work versus individual wage relation. The contradiction between the collective and collaborative requirements of effective process (the use-value aspect) and the individual and competitive nature of the employment contract (the exchange-value aspect) was visible in concerns voiced by some interviewees about job security:

> "If you have a good process, then people become like widgets you can stick into it, and everyone knows what their job is. Obviously that's a big advantage for the organization. [...] On the other hand, it also brings some fear for job security. It does make my job as a programmer easier to fill." (B: department manager, formerly with A)

This contradiction also helps explain a certain disinterest and passivity on the part of some developers:

> "I follow the rules because they are there." (B: development)
>
> "By and large, people just accept the Instructions pretty passively." (C: development manager)
>
> "It's hard to scare up much process improvement effort from the troops. Almost all the process improvement activity comes from people assigned to that task." (C: training)

Overall. The overall result of the socialization tendencies and valorization counter-tendencies was an uneven process of socialization – an unevenness visible in variation across individuals:

> "We still have to deal with the 'free spirits' who don't believe in process. These are typically people who have worked mainly in small teams. It's true that a small group working by itself doesn't need all this process. But we rarely work in truly independent small teams: almost all our work has to be integrated into larger systems, and will have to be maintained by people who didn't write the code themselves. These free spirits, though, are probably only between 2% and 4% of our staff. We find some of them in our advanced technology groups. We have some in the core of our business too, because they are real gurus in some complex technical area and we can't afford to lose them. And there are some among the new kids coming in too: many of them need convincing on this score. Most of them adapt, although some don't and they leave." (C: process engineering)

DISCUSSION AND IMPLICATIONS

This paleo-Marxist version of activity theory also has implications for a number of related themes in current research on work and organization:

Tools. The paleo view highlights the tension between socialization expressed in the progressive differentiation and integration of tool makers and tool users and the persistent valorization pressures that encourage firms to design and use tools to coerce more effort from recalcitrant users. This perspective suggests that there is something terribly one-sided about the current fascination with tacit knowledge (see also Adler, 1996; see also Hedlund, 1994: 76; Zollo and Winter, 2002). In the most common construal (stemming from Polanyi), tacit knowledge is individual and private; in an alternative construal, tacit knowledge can be collective, the property of a community of practice; but in either case, tacit knowledge is essentially local, the antithesis of universal, socialized knowledge. Tacit knowledge is often illustrated by reference to the difficulty of articulating our knowledge of how to ride a bicycle. As I write this, the Tour de France has just concluded, and Lance Armstrong won for the fifth consecutive time. The amount of formal, articulated, engineering and scientific knowledge that has been invested in perfecting his riding technique, his bicycle, his training program, his team's organization, and so forth is massive. Clearly, there is a tacit dimension to Armstrong's performance; but just as clearly, theory should acknowledge the emergence of the whole field of sports science, and the deep transformation of the structure of knowledge that is its corollary.

Object. The basic object contradiction is between the progressive expansion and enrichment of the object viewed as a technical challenge and the narrowness and poverty of profit as an object. This tension is visible in numerous studies of Total Quality Management: management encourages workers to invest themselves in process improvement, but profit imperatives deter management from acting on the resulting suggestions. In part, the contradiction is captured in institutionalization theory (Scott, 1995; DiMaggio and Powell, 1991), as a tension between technical and legitimacy constraints; but institutionalization theory has not always been explicit about the deep ambiguity of "technical" constraints under capitalist conditions: it has often conflated the use-value and exchange-value aspects of the technical domain. As a result, even though institutionalization theory has identified the role of social construction in the symbolic legitimacy domain, it has tended to "naturalize" the technical-task domain.

Community. The case sketches highlighted the persistent, contradictory coexistence of enabling and coercive features of modern organizational forms. This analysis sits uneasily with the broad architecture of current organization theory. Organization theory has long been split between "rational" and "natural" systems views, and within the latter, split between consensus and conflict approaches (Scott, 2003). Perhaps organizational research would advance more fruitfully if, instead of playing these perspectives off against each other as if they were incommensurable paradigms, we acknowledged that each reflects part of the whole picture, and focused our research more systematically on understanding their interrelations. To take an example, research on teams in organizations might advance more fruitfully if teams were seen both as a high-performance organization design and as a form of normative and concertive control, and if the tension between the two aspects brought into focus as an object of study.

Workers. The preceding cases suggested that firms' attempts to strengthen collaboration and coordination have profound effects on workers' self-construals. Marx's linkage of objective and subjective socialization is developed and extended in Elias's (2000) discussion of "figuration," and this perspective is emerging as a fruitful avenue for organizational research (Van Iterson et al., 2002). Even though valorization pressures limit the trend, we see knowledge-intensive firms such as GCC encouraging the emergence of new forms of collectivism, ones that we might interpret as differentiated from traditional collectivism by their lower power-distance (Triandis and Gelfand, 1998) and/or by their coexistence with high individualism (Kagitcibasi, 1997). This suggests that our theories of motivation may also need expansion. When firms mobilize the collective worker to ensure a more effective labor process, and when workers respond by internalizing this community, then autonomy – whether of the individual or of the self-directed team – becomes less salient as a source of motivation in job design: collaborative interdependence may activate more collectivistic sources of motivation.

SOCIALIZATION AND SKILL IN HISTORICAL PERSPECTIVE

In conclusion, let us return to the starting point: the future of critical research on work organization. Others have criticized Braverman and the earliest labor process theory research for ignoring factors such as class consciousness, the role of workers' power in shaping work and skill, the limited diffusion of scientific management techniques, and gender and other social and discursive forces in shaping the social construction of skill categories (for recent reviews of criticism and debate, see Wardell, 1999; Grugulis, Willmott and Knights, 2001). By contrast, the preceding discussion of the Census occupation data and the case sketches suggest a very different critique, one that demands a theory that accounts for both a long-run increase in average skill levels and a profound change in the form of skill.

The key to an effective response to this challenge lies, I have argued, in restoring the causal weight of technology -- the forces of production -- in our account. Whereas neo-Marxist labor process theory focuses on the role of capitalist relations of production in shaping work organization, the paleo-Marxist approach I have proposed situates skill and work organization at the intersection of the forces and the relations of production, influenced by both.

On the one hand, under the impact of the progressive socialization of the forces of production, tasks become on average more complex, and there is a progressive differentiation of roles and increasing collaborative interdependence at various levels: between workers (team work), work units (process management), hierarchical levels (employee involvement), specialized functions (cross-functional teams), and firms (supplier partnerships). Yes, the autonomy and nobility of traditional crafts are trampled underfoot in this process; but the larger mass of workers often find the complexity of their tasks increased and their work relations broadened. Increasing proportions of men and women are drawn into mixed-gender, interdependent work relations. Workers are drawn from local isolation into the web of globalization, which (*pace* neo-Marxist theories of dependency) also tends to increase work complexity and broaden relations in developing countries (as in the paleo-Marxist account offered by Warren, 1980).

On the other hand, and in contrast, the persistence of capitalist relations of production has a fundamentally ambiguous effect. The capital relation of competition sometimes stimulate the advance of the forces of production, but sometimes oblige firms to sacrifice long-term for short-term gains, and systematically oblige them to privilege owners' private benefits over broader social benefits. Tentative moves towards inter-firm collaboration are both stimulated and undermined by competitive rivalry. The wage relation privileges owners' interests over workers'. As owners' agents, managers sometimes find it profitable to upgrade workers' skills, but sometimes find it more profitable, if only in the short-term, to deskill work, manipulate teamwork to create peer pressure, let horizontal specialization degenerate into adversarial rivalry, and use hierarchy for command and control. These effects should not be ignored as mere "noise" in the data: they reflect the deep structure of property relations under capitalism. It is therefore appropriate that scholars should highlight and criticize them as reflecting an important, imminent tendency of capitalism.

What can we say about the relationship between these two sets of forces in the overall evolution of work and skill? The neo-Marxist interpretation offered by LPT gives little causal efficacy to the forces of production (e.g. Burawoy, 1979: 14ff, 220), and argues that capitalist development, insofar as workers' struggles have not been able to counteract capitalist pressures, leads to increasing misery, including the deskilling and degradation of work. When Marx and Engels write in the *Communist Manifesto* that capitalism develops its own "gravediggers," neo-Marxist LPT takes this to mean that capitalism creates a class with so little left to lose that it has no alternative but to revolt – workers will have nothing to lose but their chains.

On the paleo-Marxist view, the development of capitalism is profoundly shaped by the progressive socialization of the forces of production. Over the long run, the overall effect is to create a working-class that is increasingly capable of taking on successfully the task of radically transforming society and of assuming the leading role in a new form of society. This task is made progressively easier by the gradual socialization of relations of production even within the womb of capitalism itself, notably in the form of the increasing concentration and centralization of capital and the growing role of government in the economy. (Whether workers are *motivated* to undertake this historic, revolutionary mission – as distinct from being *capable* of undertaking it -- depends on distinct, socio-political, super-structural factors.)

Marx's writings themselves are ambiguous on the relationship of the two sets of forces. Elsewhere (Adler, 1990), I have argued that this is because these writings, even *Capital*, mixed the analysis of long-term and shorter-term trends, and combined objective analysis with polemical advocacy. The paleo interpretation was nevertheless dominant until World War I. Ideologically, it went hand in hand with a sense of the historical inevitability of socialism and a great self-confidence on the part of the major working-class parties.

Since around the First World War, the more radical parts of the left have argued that the paleo-Marxist view concedes too much continuing legitimacy to capitalism. The objection would appear to be that if capitalism continued to foster the development of the forces of production and the working class's capabilities, it would be difficult to justify radical hostility to it. But on the paleo view, there are plenty of fundamental, and increasingly compelling, reasons to doubt that capitalism is the "end of history" (*pace* Fukuyama, 1992). Even if the aggregate, long-term trend in work organization is towards upgrading, the unevenness of this process is a scandal that is increasingly resented. More generally, capitalism seems unable to eliminate its "savage inequalities" (in Kozol's, 1991, phrase), its persistent un- and under-employment, its recurrent economic crises and wars, and its ecological irresponsibility. The paleo view allows critical scholars to advance this critique while acknowledging the progressive aspects of capitalist development. The neo view turns the critique into shrill polemic.

We need a robust theory of work. A paleo-Marxist version of activity theory provides a promising starting-point. Empirically, it provides us with a surprisingly intuitive way of grasping the everyday contradictions of the capitalist firm. Theoretically, it allows us to characterize the fundamental limitations of capitalism and how these limitations conflict with a long-term socialization trend. And politically? In the short term, prospects for radical change due to this escalating conflict may seem dim, but the socialization thesis puts history on the side of radical change.

REFERENCES

Abromowitz, Moses and Paul A. David (1996) “Technological change and the rise of intangible investments: The U.S. economy’s growth part in the Twentieth century,” in Employment and Growth in the Knowledge-Based Economy: 35-60. Paris: Organization for Economic Cooperation and Development

Adler, P.S, “Market, Hierarchy, and Trust: The Knowledge Economy and the Future of Capitalism,” Organization Science, March-April 2001: 214-234.

Adler, P.S. (2003) “Practice and Process: The socialization of software development,” unpublished ms., University of Southern California.

Adler, P.S. and B. Borys, 1996, “Two types of bureaucracy: Enabling and coercive,” Administrative Science Quarterly, 41, 1: 61-89.

Adler, P.S., (1993) “The Learning Bureaucracy: New United Motors Manufacturing, Inc.” in Barry M. Staw and Larry L. Cummings (eds.) Research in Organizational Behavior, vol. 15: 111-194, Greenwich, CT: JAI Press.

Adler, P.S., “Marx, Machines and Skill,” Technology and Culture, 31, 4, Oct. 1990, pp. 780-812.

Adler, P.S., 1996, “The dynamic relationship between tacit and codified knowledge: Comments on Nonaka’s ‘Managing innovation as a knowledge creation process,” in G. Pogorel and J. Allouche (eds.), International Handbook of Technology Management: 110-124, Amsterdam: North-Holland.

Adler, P.S., B. Goldoftas and D. I. Levine, 1999, “Flexibility versus Efficiency? A Case Study of Model Changeovers in the Toyota Production System,” Organization Science, 10, 1: 43-68.

Adler, P.S., B. Goldoftas and D. Levine (1997) “Ergonomics, Employee Involvement, and the Toyota Production System: A Case Study of NUMMI's 1993 Model Introduction,” Industrial and Labor Relations Review, 50, 3: 416-437.

Adler, P.S., Barbara Goldoftas, and David I. Levine, “Stability and change at NUMMI,” in Robert Boyer, Elsie Charron, Ulrich Jürgens, and Steven Tolliday (eds.), Between Imitation and Innovation: The Transfer and Hybridization of Productive Models in the International Automobile Industry, pp. 128-160, New York: Oxford University Press, 1998.

Attwell, P., 1987, “The deskilling controversy,” Work and Occupations, 14, 3: 323-346.

Babson, S., ed., (1995) Lean Work: Empowerment and Exploitation in the Global Auto Industry, Detroit: Wayne State University Press.

Bach, J., 1994, “The immaturity of CMM,” American Programmer, 7, 9, on-line at http://www.satisfice.com/articles/cmm.htm.

Bach, J., 1995, “Enough About Process: What We Need Are Heroes,” IEEE Software, 12, 2: 96-98.

Bandura, A. 1997, Self-efficacy: The exercise of control, New York: W.H. Freeman.

Barley, S.R., 1996, “Technicians in the workplace: Ethnographic evidence for bringing work into organization studies,” Administrative Science Quarterly, 41, 3: 404-441

Barley, S.R., and G. Kunda, 2001, “Bringing work back in,” Organization Science, 12, 1: 75-94.

Beirne, M., H. Ramsay and A. Panteli, “ Developments in computing work: Control and contradiction in the software labour process,” in P. Thompson and C. Warhusrt, eds., Workplaces of the Future: 142-162, Houndmills: Macmillan Business.

Bell, D. (1973) The Coming of Post-Industrial Society, New York: Basic Books.

Blackler, F., 1993, “Knowledge and the theory of organizations: Organizations as activity systems and the reframing of management,” Journal of Management Studies, 30, 6: 863-884.

Bottomore, T. (ed.), 1983, A Dictionary of Marxist Thought, Cambridge, MA: Harvard University Press.

Braverman, H. (1974) Labour and Monopoly Capital: The Degradation of Work in the Twentieth Century New York and London: Monthly Review Press

Burawoy, M., 1979, Manufacturing Consent, Chicago: University of Chicago Press.

Burawoy, M., 1996, "A classic in its time," Contemporary Sociology, 25, 3: 296-299.

Clark, B. (1999) "Effects of process maturity on development effort," unpublished paper, at http://sunset.usc.edu/~bkclark/Research

Cohen, G.A. (1978) Karl Marx's Theory of History: A Defense. Princeton University Press

Cohen, G.A. (1988) "Human nature and social change in the Marxist conception of history," The Journal of Philosophy, 85, 4: 171-191

Cohen, Joshua (1982) Review of G.A. Cohen, Karl Marx's Theory of History, The Journal of Philosophy, 79, 5: 253-273.

Conn, R., 2002, "Developing software engineers at the C-130J software factory," IEEE Software, September/October: 25-29.

Conradi, R., and A. Fuggetta, 2002, "Improving software process improvement," IEEE Software, July-August; 92-99.

Covaleski, M.A., M.W. Dirsmith, J.B. Heian, and S. Samuel, 1998, "The calculated and the avowed: Techniques of discipline and struggles over identity in Big Six public accounting firms," Administrative Science Quarterly, 43, 2: 293-327

Crocca, W. T., 1992, "Review of "Japan's Software Factories: A Challenge to U.S. Management," Administrative Science Quarterly, 37, 4: 670-674.

Crosby, P. B., 1979, Quality is Free, New York: McGraw-Hill.

Cusumano, M.A., 1991. Japan's Software Factories: A Challenge to U.S. Management, New York: Oxford University Press.

De Geer, H. (1982) Job Studies and Industrial Relations: Ideas about efficiency and relations between the parties of the labour market in Sweden, 1920-1950, Stockholm: Almqvist & Wiksell International

DeMarco, T., and T. Lister, 1987, Peopleware: Productive Projects and Teams, New York: Dorset.

Elias, N., (2000) The Civilizing Process, Malden, MA: Blackwell.

Engels, F. 1978, "Socialism: Utopian and Scientific," in R, C. Tucker (ed.), The Marx-Engels Reader, 2nd edition: 683-717, New York: Norton.

Engestrom, Y. (1987) Learning by Expanding: An Activity-theoretical Approach to Developmental Research, Helsinki: Orienta-Konsultit

Engeström, Y., 1990, Learning, Working and Imagining: Twelve Studies in Activity Theory. Helsinki: Orienta-Konsultit.

Ezzamel, M., and H. Willmott, 1998, "Accounting for teamwork: A critical study of group-based systems of organizational control," Administrative Science Quarterly, 43, 2: 358-396.

Foot, K.A., 2002, "Pursuing an Evolving Object: Object Formation and Identification in a Conflict Monitoring Network," Mind, Culture and Activity, 9, 2: 132-149

Form, W., 1987, "On the degradation of skills," Annual Review of Sociology, 13: 29-47.

Fraser, S. (1991) Labor Will Rule: Sidney Hillman and the Rise of American Labor, New YorK: Free Press.

Freidson, E., 2001. Professionalism: The Third Logic. Cambridge, UK: Polity.

Friedman, A.L., and D. S. Cornford, 1989, Computer Systems Development: History, Organization and Implementation, Chichester: John Wiley& Sons.

Fukuyama, F., 1992, The End of History and the Last Man, New York: Free Press.

Gibbs, G.G., 1994, “Software’s chronic crisis,” Scientific American Sept.: 86-92.

Gibson, C.D. and P. C. Earley, n.d., “Work-team performance motivated by collective thought: The structure and function of group efficacy,” unpublished, University of Southern California.

Goldin, C. and L.F. Katz, (1999) “The returns to skill in the United States across the Twentieth century,” National Bureau of Economic Research Working Paper 7126

Goldin, C. and L.F. Katz, “The origins of technology-skill complementarity,” Quarterly Journal of Economics, 113, 3, August 1998, 693-732.

Gouldner, A. W., 1957. “Cosmopolitans and locals: Toward an analysis of latent social roles,” Administrative Science Quarterly, 2, 3: 281-306.

Greenbaum, J. M., 1979. In the Name of Efficiency, Philadelphia: Temple University Press.

Greenbaum, J.M. (1998) “The times they are a’changing: Dividing and recombining labour through computer systems,” in P. Thompson and C. Warhusrt, eds., Workplaces of the Future: 124-141, Houndmills: Macmillan Business.

Griss, M.L., 1993, “Software reuse: From library to factory,” IBM Systems Journal, 32, 4: 548-566

Grugulis, I. H. Willmott. And D. Knights (2001). Special issue on the Labor Process Debate, of International Studies of Management and Organization, 30, 4.

Harter, D.E., M.S. Krishnan, S.A. Slaughter, 2000, “Effects of process maturity on quality, cycle time, and effort in software development,” Management Science, 46, 4: 451-466

Hedlund, G. 1994, “A model of knowledge management and the N-form organization,” Strategic Management Journal, 15: 73-90

Herbsleb, J., D. Zubrow, D. Goldenson, W. Hayes, and M. Paulk, 1997, “Software quality and the Capability Maturity Model,” Communication of the ACM, 40, 6: 30-40.

Hirschhorn, L. (1984) Beyond Mechanization. Cambridge, MA: M.I.T. Press

Holt, G.R., and A. W. Morris, 1993, “Activity theory and the analysis of organizations,” Human Organization, 52, 1: 97-109

Humphrey, W. S. (2002) “Three process perspectives: Organizations, Teams, and People,” Annals of Software Engineering, 14: 39-72

Hyman, R., 1987, “Strategy or structure? Capital, Labour and Control,” Work, Employment and Society, 1, 1: 25-55.

Jacoby, S. M., 1983, “Union-management cooperation in the United States: Lessons from the 1920s,” Industrial and Labor Relations Review, 37: 18-33.

Jermier, J.M., 1998, “Critical perspectives on organizational control,” Administrative Science Quarterly, 43, 2: 235-256

Kagitcibasi, C., 1997, “Individualism and collectivism,” in J. W. Berry, M. H. Segall, and C. Kagitcibasi (eds.) Handbook of Cross-Cultural Psychology: 1-49, Needham Heights, MA: Allyn & Bacon.

Kanigel, R., 1997, The One Best Way: Frederick Winslow Taylor and the Enigma of Efficiency, New York: Viking.

Kelly, J.E., 1982. Scientific Management, Job Redesign and Work Performance, London: Academic Press.

Kenney, M., and R. Florida (1993) Beyond Mass Production: The Japanese System and its Transfer to the U.S., New York: Oxford University Press.

Kern, M., and M. Schumann (1984) Das Ende der Arbeitesteilung? [The End of the Division of Labour?] Munich: C.H. Beck

Knights, D. and H. Willmott, 1989. “Power and subjectivity at work: From degradation to subjugation in social relations,” Sociology, 23, 1989: 535-558

Kohn, M. L., and C. Schooler, 1983, Work and Personality. Norwood, NJ: Ablex.

Kozol, J. (1991) Savage Inequalities: Children in America's schools, New York: Crown

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

Krishnan, M.S., C.H. Kriebel, S. Kekre, and T. Mukhopadhyay, 2000, “Productivity and quality in software products,” Management Science, 46, 6: 745-759

Lave, J. 1988, Cognition in Practice, New York: Cambridge University Press.

Leont’ev, A.N., 1978. Activity, Consciousness, and Personality. Englewood Cliffs: Prentice-Hall.

Levine, A. and E. Wright (1980) “Rationality and class struggle,” New Left Review, 123: 47-68

Lieberman, H. and C. Fry, 2001. “Will software ever work?” Communications of the ACM, 44, 3: 122-124.

Lynn, L. H., 1991, “Assembly line software development,” Sloan Management Review, 88:

Markus, H.R., and S. Kitayama, 1991, “Culture and the self: implications for cognition, emotion, and motivation,” Psychological Review, 98, 224-253

Marx, K. (1973) Grundrisse, Harmondsworth: Penguin Books.

Marx, K. (n.d.) The Poverty of Philosophy, Moscow: Progress

Marx, K. 1959, “Theses on Feuerbach,” in L.S. Feuer, ed., Basic Writings on Politics and Philosophy, Karl Marx and Friedrich Engels: 243-245, Garden City, NY: Anchor Books.

Marx, K. 1975. Karl Marx: Early Writings. New York: Vintage

Marx, K., and F. Engels, 1959. “The Communist Manifesto” in L. S. Feuer, ed., Marx and Engels: Basic Writings on Politics and Philosophy: 1-41, New York: Anchor.

Marx. K. 1977. Capital. Vol. 1. New York: Vintage.

Mathews, J.A. (1994) Catching the Wave: Workplace Reform in Australia, Ithaca, N.Y. : ILR Press

Meyer, J. W., and B. Rowan, 1977, “Institutionalized organizations: Formal structure as myth and ceremony,” American Journal of Sociology, 83: 340-363

Nelson, D. 1980, Frederick W. Taylor and the Rise of Scientific Management, Madison: University of Wisconsin Press,

Nelson, D., ed., 1992. A Mental Revolution: Scientific Management since Taylor, Columbus: Ohio State University Press.,

Ngwenyam, O. and P. A. Nielson, 2003, “Competing values in software process improvement: An assumption analysis of CMM from an organizational culture perspective,” IEEE Transactions on Engineering Management 50, 1: 100-112.

Nyland, C. (1998) “Taylorism and the Mutual Gains Strategy,” Industrial Relations, 37, 4: 519-542.

Piore, M.J., and C.J. Sabel (1984) The Second Industrial Divide: Possibilities for Prosperity, New York: Basic Books.

Powell. W., and P. DiMaggio, (eds.) The New Institutionalism in Organizational Analysis, Chicago: University of Chicago Press, 1991

Prasad, M. 1998, “International capital on ‘Silicon Plateau’: Work and control in India’s computer industry,” Social Forces, 77, 2: 429-452.

Reich, R., 1991. The Work of Nations, New York: Addison-Wesley.

Rinehart, J., C. Huxley, and D. Robertson, 1997, Just Another Car Company? Lean Production and its Discontents, Ithaca: ILR Press.

Sacks, M., 1994, On-The-Job Learning in the Software Industry, Westport, CT: Quorum.

Schachter, H.L. (1989) Frederick Taylor and the Public Administration Community, Albany NY: State University of New York Press.

Schneider, B. 1987. "The people make the place," Personnel Psychology, 40, 437-454.

Scott, R.W., 2003, Organizations: Rational, Natural and Open Systems, 5th edition, Upper Saddle River: Prentice-Hall.

Scott, W. Richard, Institutions and Organizations, Sage 1995

Scott, W. Richard, Organization: Rational, Natural and Open Systems, 5th ed., New York: Prentice Hall, 2003

Smith, C., and P. Thompson (1999) "Reevaluating the Labor Process Debate," in M. Wardell, T.L. Steiger, P. Meiskins, eds. Rethinking the Labor Process: 205-232, Albany: State University of New York Press.

Software Engineering Institute, 2003. "Process Maturity Profile, 2002 year end update," download from http://www.sei.cmu.edu/sema/profile_SW-CMM.html

Sohn-Rethel, A. (1978) Intellectual and Manual labour : A Critique of Epistemology. Atlantic Highlands, N.J.: Humanities Press.

Spenner, K.I. (1988) "Technological change, skill requirements, and Education: The case for uncertainty," in R. M. Cyert and D.C. Mowery, eds., The Impact of Technological Change on Employment and Economic Growth: 131-184, Cambridge MA: Ballinger.

Standish Group (1994) "Chaos study report," at www.standishgroup.com.

Steiger, T. L. (1999) "Forms of labor process and labor's share of value," in M. Wardell, T. L. Steiger, and P. Meiskins, eds., Rethinking the Labor Process, pp. 189-204, Albany: State University of New York Press.

Swanson, K., D. McComb, J. Smith, and D. McCubbrey, 1991, "The application software factory: Applying Total Quality Techniques to systems development," MIS Quarterly, Dec.: 567-579

Thompson, P. (1989) The Nature of Work, 2nd edition. London: Macmillan

Thompson, P. and D. McHugh, 2002, Work Organizations: A Critical Introduction, 3rd edition, Houndmills: Palgrave.

Thompson, P., and C. Smith, 2001, "Follow the redbrick road: Reflections on pathways in and out of the labor process debate," International Studies of Management and Organization, 30, 4: 40-67.

Tinker T., 2002, "Spectres of Marx and Braverman in the twilight of postmodernist labour process research," Work, Employment & Society, 16, 2: 251-279

Touraine, A. (1969) La Société Post-Industrielle [Post-Industrial Society], Paris: Editions Minuit

Triandis, H.C., and M. J. Gelfand, 1998, "Converging measurement of horizontal and vertical individualism and collectivism," Journal of Personality and Social Psychology, 74, 1: 118-128.

U.S. Bureau of the Census (1975) Historical Statistics of the United States, Washington DC: U.S. Government Printing Office.

U.S. Bureau of the Census (2000) Statistical Abstract of the United States, Texas: Hoover's Business Press

Van der Pijl, K. 1998, Transnational Classes and International Relations, London: Routledge.

Van Iterson, A., W. Mastenbroek, T. Newton, and D. Smith (eds.), 2002, The Civilized Organization: Norbert Elias and the future of organization studies, Philadelphia: John Benjamins.

Van Maanen, J., and S.R. Barley, 1984, "Occupational communities: Culture and control in organizations," Research in Organizational Behavior, 6, 287-365.

Vygostky, L.S., 1962, Thought and Language, Cambridge MA: MIT Press.

Vygotsky, L.S., 1978, Mind in Society, Cambridge MA: Harvard University Press.

Wardell, M, T.L. Steiger, P. Meiskins, eds. (1999) Rethinking the Labor Process, Albany: State University of New York Press.

Wardell, M., 1999, "Labor processes: Moving beyond Braverman and the deskilling debate," in M. Wardell, T.L. Steiger, P. Meiskins, eds. (1999) Rethinking the Labor Process: 1-16, Albany: State University of New York Press.

Warren, B. (1980) Imperialism: Pioneer of Capitalism. London: Verso

Weber, H. (ed.), 1997, The Software Factory Challenge, Amsterdam: IOS Press.

Wertsch, J. V. (ed.), 1979, The Concept of Activity in Soviet Psychology, Armonk, New York: M.E. Sharp.

Wright, E.O., A. Levine and E. Sober (1992) Reconstructing Marxism: Essays on explanation and the theory of history, London and New York: Verso

Zimbalist, A., ed., 1979, Case studies on the labor process, New York: Monthly Review Press.

Zollo, M. and S.G. Winter, 2002, "Deliberate learning and the evolution of dynamic capabilities," Organization Science, 13, 3: 339-351

EXHIBIT 1: EVOLUTION OF THE U.S. OCCUPATIONAL STRUCTURE

Year	**1900**	**1970**	**2000**
Clerical	**0.03**	**0.18**	**0.16**
Professional, technical	**0.04**	**0.14**	**0.16**
Service workers, excl. private household	0.04	0.11	
Private household workers	0.05	0.02	
Total service, incl. private household	**0.10**	**0.13**	**0.14**
Salesworkers	**0.05**	**0.07**	**0.12**
Operative and kindred	0.13	0.18	
Laborers, excl. farm and mine	0.13	0.05	
Total operatives plus laborers (excl. farm)	**0.26**	**0.23**	**0.12**
Managers, administrative, proprietors	**0.06**	**0.08**	**0.11**
Craftsmen, foremen	**0.11**	**0.14**	**0.12**
Farmers	0.20	0.02	
Farm laborers and foremen	0.18	0.01	
Total farmers plus farm laborers	**0.38**	**0.03**	**0.04**

Source: U.S. Bureau of the Census. 1900, 1970: Historical Statistics; 2000: author's imputation based on Statistical Abstract. Census data after 1970 combine operatives and laborers, do not distinguish private household workers, and do not distinguish farm laborers from farmers and farm managers

EXHIBIT 2

THE STRUCTURE OF THE LABOR PROCESS

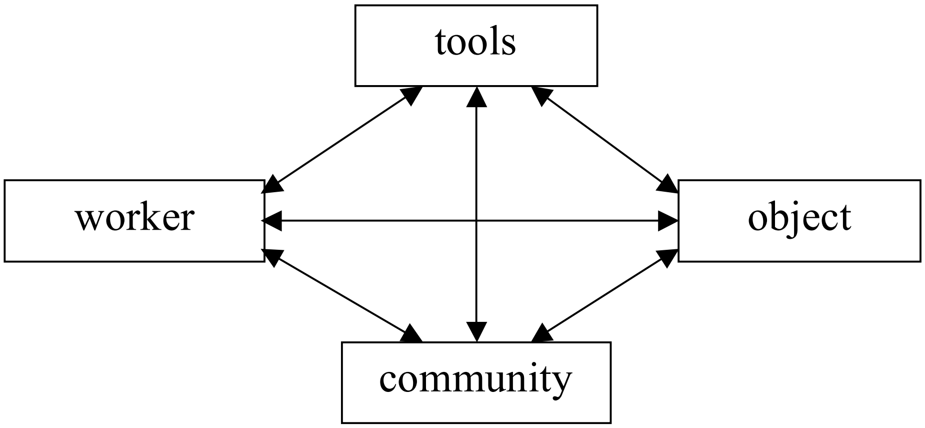

EXHIBIT 3: THE CAPABILITY MATURITY MODEL

Level	**Focus and description**	**Key Process Areas**	**Proportion of appraised organizations 1998-2002**
Level 1: Initial	**Competent people and heroics:** The software process is ad hoc, occasionally even chaotic. Few processes are defined, and success depends on individual effort and heroics.		16.9%
Level 2: Repeatable	**Project management processes:** Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.	* software configuration management * software quality assurance * software subcontract management * software project tracking and oversight * software project planning * requirements management	43.2%
Level 3: Defined	**Engineering processes and organizational support:** The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.	* peer reviews * intergroup coordination * software product engineering * integrated software management * training program * organization process definition * organization process focus	24.6%
Level 4: Managed	**Product and process quality:** Detailed measures of the software process and product quality are collected. Both the software process and products are 7.3%quantitatively understood and controlled.	* software quality management * quantitative process management	8.0%
Level 5: Optimizing	**Continuous process improvement:** Improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.	* process change management * technology change management * defect prevention	7.3%

Source: Software Engineering Institute, 2003