

# De-briefing and motivation of knowledge workers

Mette Mønsted & Nicoline Jacoby Petersen

**Copenhagen Business School**  
**Dept. of Management, Politics & Philosophy**  
**Blaagaardsgade 23 B**  
**DK- 2200 Copenhagen**  
**Denmark**  
**Tel. +45 3815 2655**  
**Fax: + 45 3815 3635**  
[monsted@cbs.dk](mailto:monsted@cbs.dk) [np.lpf@cbs.dk](mailto:np.lpf@cbs.dk)

## *Abstract:*

*On the basis of interviews and video-analysis of multimedia firms, the experience of management as a process is illustrated by different examples and reflections of how demands to project management have changed in this kind of condition, especially in relation to staff working with advanced technology or software. Two cases are presented to illustrate the de-briefing of developers. The asymmetries of knowledge, where managers know much very little about what developers in IT firms do and know. Still decisions on division of labour with customers and other staff has to be taken, and the need to know as to create meaning is a constant challenge.*

*The asymmetry of knowledge in advanced IT firms, and the problems of identifying the knowledge that makes a difference among the knowledge workers opens questions on management and communication, and the problems of finding boundary objects and translations of the insights in technology (with examples in programming). Different forms of communication and debriefing of personnel arise as needs to get an insight of knowledge to organise and manage. Managers seem to have to create management as a social construction, and create legitimacy and relevance in the communication with self-managed knowledge workers.*

*The relationship between managers and expert knowledge staff raise issues of how to understand management of knowledge workers, as management is dependent on translators and may be totally cut off from the information flows. It is not easy to perceive loosely coupled managers as very powerful, or beneficial to the organisations. New forms of communication and organising are demanded in these kinds of organisations, or at least the desperate need for more communication is more obvious than in many other organisations, as the mutual dependency is high.*

*Paper for the OKLC conference in Barcelona April 13<sup>th</sup> -14<sup>th</sup> 2003*

# De-briefing and motivation of knowledge workers

Mette Mønsted & Nicoline Jacoby Petersen

## 1. INTRODUCTION

Management of knowledge workers is a delicate balance between motivation and coordination, between playfulness and efficiency. It is also a perspective of leadership, where the manager is responsible for division of labour, but where he is the one of the persons who knows the least about the professional issues. The asymmetric knowledge and the dilemmas between the motivating knowledge workers involved in innovation, R&D and the need to produce efficiently is illustrated by a de-briefing case from a small multimedia firm, which has been studied longitudinally for 2½ years,

The relationship between management and the knowledge workers represents a structure, where the manager's dependency on the knowledge worker is very high as replacement is very difficult in these skills. If the value for the developers is tied to play, fun development tasks, proof of competence to develop beautiful codes, then too strict rules to make processes efficient may chase away the really good development people.

The nerds and stars who are the highly valued programmers, and the many young unskilled who are employed and are trained on the job are developing a culture of play, which is probably very efficient to solve certain tasks and to create commitment, but hard to control to make it economic feasible, when projects have to meet deadlines and be sold.

Motivation and coordination is a balance, and the article outline some of the dilemmas, and in what kind of situations it becomes obvious. Such illustrations of different values indicate not only different communities of practice, but also the need for good management to develop only the necessary, rather than the "Rolls Royce model", which may only be sold for the price of the "Volkswagen". In the process the legitimacy of the manager is at stake.

Management of deadlines and deliveries to be sold has been fundamental for all the IT firms, though some have overlooked this. In the dot.com period the experience has socialised a number of employees to a relaxed attitude to economic management as value of sales and profit did not reflect the value of the firm.

The article is focusing on the management implications of asymmetric knowledge, and how to deal with knowledge sharing for constructing a management platform. The case of de-briefing IT nerds is a point of departure for folding out the arguments, and the special features of how turbulence and rapidity constitutes itself in IT-projects illustrate some of the management problems relative to motivated knowledge workers.

## 2. DE-BRIEFING ON WHERE WE ARE AND WHAT WE KNOW.

The whole discussion of making tacit knowledge explicit could be related to the communication of knowledge in IT projects. The notion of de-briefing of spy's comes very close, observing how managers at different levels try to find out what the single programmers actually do, and how this could be exploited in the joint project. The importance is that division of labour, presupposes that the work task is known, and can be delimited and specified for all involved. Both in innovation and in most IT projects in the early stages, this is not the case, which makes management a very demanding task of coordination on the boundary of knowledge.

The asymmetric knowledge vis-à-vis managers demands some learning-spaces to try to communicate on the content of the work, as to be able to coordinate the consequences for other staff. The process of de-briefing of existing staff takes a lot of time, but is the foundation for transfer not only of information, but also of the level of knowledge as to act and work along the other programmers, and in a research project( Jensen & Monsted 2001b) we have used video registrations of some of these meetings.

Even at a stage without immediate crisis, the project manager has to make an effort at regular intervals to de-brief the advanced developers:

*A video-scene where the project manager is trying to find out, what the programmer has been doing. They start talking leaning over a piece of paper. Then the project manager goes to the whiteboard, and start explaining how he interprets, what the programmer is saying. The project manager questions and interprets the answers. After a while they begin to "think together" on possible solutions and interpretations.(Jensen 2001)*

The scene illustrates some of the problems of communication and translating codes by the programmer, who cannot formulate in words, what he is doing, or the reflections behind the choices made. The whiteboard is a boundary object in Susan Leigh Starr's (1989) version as a means to communicate and translate the part of tacit knowledge, which is routines and self-evident knowledge to the programmer being de-briefed. The project manager uses the drawings and codes to create a dialogue for understanding the codes developed and the reasoning behind. The knowledge of knowledge workers is thus not in a form that can easily be explained as also seen in other scientific work ( Collins 1974).

The insight of what is necessary to communicate in order to understand enough as to act, and to make explicit what is evident for the programmer is a complex de-briefing necessary to get an overview in innovative projects. The project manager need the information, not only to give feedback to the programmer, but also to coordinate with other staff, working on other parts of the programme. Division of labour demands that either the manager or the team know the task and can create overlapping knowledge to handle boundaries between the different parts of the programme. The complexity of the tasks makes simple knowledge management impossible and may take a lot of valuable time in an efficiency production perspective. Knowledge is tied to the people and the actions involved, and to communication (see also Stacey 2001).

Another de-briefing scene is described in an video-clip:

*One of the "talented" young programmers has a lot of difficulties in telling what he does, and he is de-briefed not only by the project manager in R&D, but together with another strong programmer, who is one of the owners.*

The problem in this scene is that the primadonna cannot formulate in proper words what he is doing in a way which is understandable even for the competent project manager, who has programming skills. The other “star” has to participate in the de-briefing, as to help translate, and follow up with supplementary questions to get into an interpretation of codes and consequences, and of the choices made in the sections of the programme he works on.

The ability to put in words what the codes mean, and how they link to other programmers, and application work is rather limited. In this case they are not stupid, but it is a totally different language and culture, and translation to explain interpretations becomes important as a part of the joint development and interpretation of the project as organisational knowledge. Beautiful codes are not the purpose of the firm. Codes have to be used, and the young talent often has to be reminded about users. He puts a “post-it” on the screen frame “*Remember user*”. The application is essential for a firm, which is not a research lab, but has to sell the products, and their applications. The dilemma between the need for programmers to play on the forefront and the need to do the necessary only to make price-competitive products is evident.

The scenes illustrate some of the fundamental problems of what firms know, and who knows what in the firms. The illustrating cases are from very small firms, but the boundaries between the groups of people are very similar to the boundaries between communities of practice or in tight networks. The problem is tied to individual learning, and has to be a social or organisation based knowledge. It is tempting to go into the discussion of transforming tacit to explicit knowledge (Nonaka & Takeuchi 1995), or translating one form of explicit knowledge in the form of codes to another form of explicit knowledge in the form of communication in words of what is achieved and the purpose of the codes. The translation is used in order to overcome communities of practice, and communicate across professional groups. Such translation is not really the same as the perception of tacit knowledge in Nonaka & Takeuchi (1995), but much more communication and learning across communities of practice or across structural holes in networks for complementary knowledge (Burt 1992, Moensted 2003).

The de-briefing in this case was seen as a communication of socially bound knowledge across cultural and professional barriers. The perception of knowledge is not just a translation, but is closer to Stacey’s (2001) perception of creating knowledge, not only as a mental individual map, but also as a social process constituted by communication. The de-briefing is not only an effort to get knowledge as a ready formulated ‘thing’ out of the head of the programmers, but an effort to create a learning space to get the codes into a perspective of creating knowledge about codes, purpose and consequences for other parts of the system. Managers have to constitute meaning in the fragments of information, and this is a social communication process (Weick 1995).

The asymmetric knowledge is a management problem. The researchers or developers do not feel the same need. As individuals with a specific responsibility, they can “see” where they are going. The need for complex coordination in a larger perspective is not necessarily the perspective of experts or developers. The autonomy and self-management, does not in this case lead to a feeling of responsibility for the organisational perspective.

Management and leadership under these circumstances demand a high level of communication and awareness of interpretations in other communities of practice. The way the de-briefing is functioning is providing a platform for the manager’s credibility for coordinating and getting the economic framework accepted.. If the manager cannot get through to the understanding of what is

going on, the personal credibility is at stake, and it is even more difficult the next times, where debriefing is necessary. The communication across professional boundaries is based on trust in the other person's competence. Respect for other epistemological capabilities could be seen as a negotiated process of creating legitimacy, trust and meaning (Newell & Swan 2000). If the manager does not maintain his respect both at a general management level, and in relation to the concrete scenes, then the framework for a joint sense-making cannot be created, and misunderstanding and even efforts of cheating are tempting. The perspectives illustrate some of the problems of creating management as a social construction in firms. If a joint framework is not established, the play and art culture may go undisturbed from the limits of the economically motivated deadlines leading to economic ruin.

### 3. PROJECT MANAGEMENT IN TIME DILEMMAS.

At the level of project management in IT-projects, the turbulence and time pressure is extreme, and often changing and tightening during a project. Uncertainty on technology has to be dealt with. When crisis management is becoming "normal", a number of regular project management tools become obsolete, and a number of good management reflections and coordination are lost. The case presented may be extreme, as everything is crisis management, but it raises a few questions on how knowledge workers handle and have to handle responsibilities in coordination at many levels. It raises the question on what kind of management can cope with impossible situations? In many ways it is "guerrilla-war" tactics, and maybe it is the reason for Department of the Navy in the US to adopt new ways of knowledge management in their training programme.<sup>1</sup> In the dot.com bubble, a number of firms managed to create projects with a very high and unrealistic level of ambition. Such ambitions and ideas actually inspired authors on project management (Kidder 1981, Christensen & Kreiner 1991), and created a special motivation of doing the impossible. An example of this role:

*A newly recruited project manager manages a large project of making a PC-based learning system. The firm had never before had such a large project. The sales manager has promised too much, and they decided in the firm and with the developer that they can do it, even if they have to develop the tool, and are forced not to use the existing tools normally used. It appears as a totally impossible task. The new project manager start building up a unit of production team making text and drawings. Most of these employees do not have much education and have to be trained on the job. The balance is to grow and get the right staff, but also to balance out the communication with the programmer responsible for the tool "producer". Her tools from informatics and the business school, and a large GANTT scheme tried to get an overview of the complexity, but only for the application side of the project, as the tool development was dependent on the developer. The effort to mobilise motivation and energy among staff was one major aspect of the job, which managed to finish, but not on time.*

The awareness of the problems of management on the boundary of the xml tool "producer", and the application, demanded a few more recruited for the tool development and adjustment to the application "Learning". The clash between the autodidact artist developer and the educated and the recruited structured developers revealed the basic problems in the project. The two new recruits demanded a proper briefing, if not they were going to leave, as they could not get into the role they were supposed to do. A time-out for debriefing of the developer had to be accepted in order to allow

---

<sup>1</sup> Alex Bennet responsible for this programme, presented the knowledge management programme at an OECD conference in Copenhagen March 2000,

new developers to work in the programme, as they could not read the mind of the original developer.

The complexity of the knowledge in IT makes several practitioners to state that recruiting more people increase the time used on the programme (Brooks 1995). The complexity of sharing knowledge within IT projects makes it difficult to make explicit what the firm knows. In complex tasks a lot of time is used on learning and exploring. The costs of introducing new staff are very high, and need a long period of training and adaptation, making it extremely difficult when new staff is recruited to speed up the process, as it may demand more time to train than the new staff can create of value added (Brogren 2000). The development based on curiosity takes long time and in an efficiency perspective of production, this is highly time consuming and maybe inefficient for production, though it depends of whether the focus is on creativity and innovation or on efficient production. The knowledge workers and managers do not always agree on the level of “new ideas” and rethinking needed. In relation to knowledge workers the innovative part is motivating, and thus difficult to eliminate if the commitment should be maintained.

The complexity of the project is increased by the parallel development of tools and applications.

Figure I: The flow should in a linear process be:

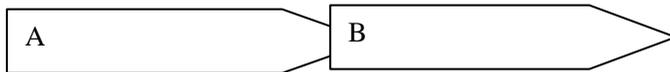
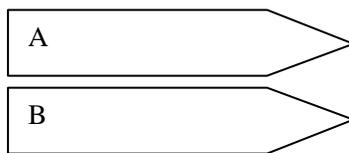


Figure II: There is no time to wait for A to finish and B is cut and placed parallel to A



Based on drawing in Jensen & Monsted 2001b.

The project manager is “the helmet man” balancing between Scylla and Charybdis (Hampden-Turner 1990). She is managing a project dependent on a tool, which is not ready yet (A). Usually the ‘producer tool’, would be first, and the application for Learning would follow (B) as shown in Figure I. Management should try to coordinate this process. As the timeline is “folded” to a parallel process, there is a need for communication and coordination between the tool (A) and the application (B). The need for communication and coordination increases both at project management level, but also on other levels of management, but the means of control for project managers as well as top managers are meagre.

The work in such parallels, where the project manager is managing and getting an overview in an enormous Gantt scheme, and recruiting and training new staff, without much education, while the tool development for a long time is only “a-one-man-army”, with talent and ability to correct and understand the application side, but very unstructured or rather un-explicit on the development process itself correcting at night and developing at daytime. Applications are developed as a separate process on the basis of the “black box”. Strangely enough the firm managed to deliver the product, not in time, but still to the satisfaction of the customer.

The coordination and management of “black boxes of the unknown” is appearing in other cases of these firms as well. The barriers of communication and learning over communities of practice are a problem, when knowledge is asymmetric, and the mutual dependency is still a dominant feature in the organisation. The challenges and thrill of an R&D project are motivating factors, which may even motivate to elaborate and change standard tasks to R&D as to get a motivation for these jobs. Even if the parallel development is not efficient, and is creating both errors and stress, it is also creating a “gut-feeling” and motivation for the project. Such motivation is harder to feel, if jobs are more standardised, especially with an emphasis on timing and deadlines.

Within IT however, turbulence and rapid change in technology is a dominating feature, and creates a very special perception of time. Evaluating time in a non-routine work is risky business that most innovative projects know. Strangely enough it is seldom an exceptional short time, which is the big surprise. When time-pressure is so high and compressed as is normally the case, many tasks have to be done parallel, which would otherwise be linear, as the second part depend on the first. The lack of time for linearity is a new feature, which is not only putting pressure on production but also on both R&D and especially on management coordination (Jensen & Mønsted 2002a). In research management, one of the top research managers said *“I did not expect earlier that I would go around to R&D engineers and talk them out of linear thinking. There is no time for linear thinking.”* The research manager is an engineer himself.

In IT the time pressure creates cases of parallel development of tools and applications, not only make coordination extremely difficult, but stresses new management roles as coordination is not only between two separate tasks that have to be combined afterwards, but communication is an embedded part of avoiding redoing many aspects of the two tasks. It is interpreted in a systemic innovation perspective, then the constant coordination of the boundaries is necessary, a translation between relatively independent productions, and with very different skills and competence in the two projects. The learning experience as to allow for more time, but then the price is too high for the offer.

The turbulence of IT projects provides new perspectives on time. The rapid development and turbulence does not change everything. It is not a radical change every 3<sup>rd</sup> months, then programmers talk of new programmes and changes. Turbulence and renewal imply that certain aspects change rapidly, but a lot of features remain. The time perspective could maybe more be seen in a biological way of perceiving change, as a seasonal and age change, where the main structure is still the same, but many features change and have to adapt to seasons or ageing (Adams 1990). Such perspectives on time could be helpful when considering IT projects, living with the intense and drastic turbulence and pressure from new technology and new methods. New methods may change a lot in a project, but changes are mostly recognisable patterns for the community of practice working within the field. The turbulence and “panic solutions” put pressure on managers and on the communication between developers and managers. Both parties are stressed and take chances, which are sometime good decisions, but not necessarily reflecting efficiency.

The problem of time is the engagement of developers as experts, who put too much time into the work, and a lot of time in the creative part of the solution. It is hard to manage, as the profitability is tied to the capacity to meet deadlines. The overtime used, should be “free research time at night”. This is also often the case.

#### 4. KNOWLEDGE INTENSIVE FIRMS AND KNOWLEDGE WORKERS.

Knowledge intensive firms are to be interpreted as a “system of persuasion” rather than being production units based on expert knowledge (Alvesson 2001). Expertise is not an objective characteristic, but a function.

In IT firms the ability to persuade customers about expertise, to create credibility and selling ideas is fundamental for the survival of advanced technology and service firms. The value of the solution is seen only if it works afterwards in the application. This is a dominant feature of all service and immaterial production ( Normann 2000).

Knowledge workers could be defined as highly educated professionals, which is a quite normal way of defining knowledge workers (Alvesson 2001 p.863, Davenport & Prusak 1998, Newell et al. 2002). The highly educated who work in non-standard settings and have to innovate or analyse and reflect independently form a special group, and demand autonomy and authority of their expert skills. Some professionals have more routinized work, and may act on the basis of existing knowledge, as would be the case with lawyers, some engineers and architects. But professionals also in these professions work on the basis of their expertise, the insight and skills needed, which demands professional minds rather than databases or robots. The group of knowledge workers who independently exploit and explore new knowledge is maybe essential for the understanding of new conditions for management. The innovativeness and unpredictability of knowledge workers in IT development demand a capacity to create meaning in dialogues, and thus create knowledge as learning in dialogue (Stacey 2001).

The skills of knowledge workers for such projects include the capacity to evaluate relevance and potential of knowledge. The ambitions of managers and knowledge workers may not be the same, as R&D projects or consultancy projects may “slide away” and expand to “interesting related issues”, and thus take much more time.

Skills in many knowledge firms are hard to define in terms of educational competence, as with consultancy and IT-programmers, experienced-based learning play an important role. The autonomy of knowledge workers to define what is relevant, interesting and necessary is hard to control and judge for people outside the profession or community of practice.

In IT firms, the skills of developers are hard to define, as mostly it is a description of practice and knowledge of existing programmes, but the ability to solve new problems within a certain sphere of programming is essential. Within the community of practice of the highly skilled programmers, stories and metaphors of work and codes are closer to art than to science. Methods and solutions are described by “the beauty” and elegance. Between programmers they know that the elegance has something to do with efficiency, shortcuts and new possibilities, as one of the managers in a small IT firm said:

*“ The difference in efficiency between a good programmer and the excellent talent is 1:20, and this is not found in many other types of skills or work”.*

This difference adds to the perception of IT appearing more like art, than science. The ability to imagine possibilities and to start creating them is very close to artists work, and maybe managed in the same way. If they are efficient there is no conflict to management. But a conflict arise from

problems of time-limits. What solution is possible within the deadline? This is a question of the necessary solution, rather than the best solution. Such compromise does not create much motivation or commitment.

The problem seems to be to exploit the talent of the super-expert, and then the stimulation of talent and communication of consequences for the dependent people becomes essential for management.

Skills in programming may be acquired as engineers, mathematicians, physics, or in informatics, but some of the talented programmers are autodidacts. This is not seen in other professional knowledge firms or disciplines. The IT field in multimedia production also include a large group of “unskilled young production workers”. Some from such groups have talents for programming, and a high commitment to learn in practice. Generally however there is more often a need to get very well defined jobs and instructions. The way of organising in the two groups is very different, and in many ways only the developer types are considered knowledge workers with decentralised responsibility. The unskilled group needs other types of instruction and management, some even in a very Tayloristic and task-specific manner.

The organisation of work in knowledge work responds to the demand for dialogue, self-management and distributed knowledge. The organisation as a system of distributed knowledge as developed by Tsoukas (1996) illustrates the way tacit knowledge and explicit knowledge is intertwined and mutually dependent. The tacit knowledge is hidden in routines and presumptions about the work, as illustrated by Latour & Wolgar (1979) and Gourlay (2002). This perspective stresses the importance of a community perspective on knowledge (Newell et al. 2002, p.107), but also the complexity of sharing knowledge, which is dynamic and fluid, and socially embedded.

Management of knowledge in IT demands a certain understanding of the field, but not necessarily with know-how for actions. Understanding the principles and knowing how to act is not the same thing (Garud 1997). The essence for management is an understanding of implications of codes, as to create conditions for a division of labour with other staff. Decisions have to be taken before knowledge is certain, often at a stage of ideas and ascribed potential meaning of new ideas (Garud 1997). IT systems are very interdependent as system development, and with a high level of complexity and system dependence, the competence of the team and the management is to have knowledge of more than the part, they are working with just now (Kogut & Zander 1996). The mutually dependency in systems, demand much more coordination for integration and managing consequences for other people, than the multidisciplinary aggregation of solutions (Newell & Swan 2000).

## 5. MANAGEMENT OF MOTIVATED KNOWLEDGE WORKERS.

The self-management of knowledge workers increases the need for coordination and knowledge sharing. The recent high focus on knowledge sharing reflects the needs for knowing what the organisation knows, and for creating occasions to share knowledge. The flat organisation and high level of decentralised management has an embedded problem of coordination and overview, which is easier in hierarchical organisations. Loose couplings in organisations create a high level of flexibility, but at the same time lose knowledge and overview. Newell et al. (2002) argue that managers have a more narrow span of control, and that subordinates control their own activities, and continue: "... middle managers are no longer able to act as the communication conduits within

an organization – they do not know in much detail what is happening within their particular sphere of responsibility nor do they necessarily have time to engage in such activities” ( Newell et al. 2002, p. 101). This implies, that “ the very same organizational forms that help to nurture knowledge creation also provide more opportunity for knowledge loss” (ibid.).

Management in knowledge intensive firms is less a position than action, stressing the ability to facilitate and create the context and motivations to create knowledge intensive products. It becomes more like the network manager, who is only the manager by acting and by being accepted by the partners as a manager ( Moensted 2003). The position and control aspects of the manager role are loosened considerably. The profile of knowledge workers and their wish for autonomy creates a knowledge-sharing context, where leadership is tolerated, as long as the manager is credible both in his acting and as a person and does not set too many obstacles to the part of work which is interesting and fun.

The problem however is more fundamental, as the cases also reflect some of the dilemmas of coordinating between different communities of practice (Wenger 1998), and between disciplines. The manager coming from outside represent a different community of practice, and the IT developers usually form their own understanding of relevance, competence and organising. When self-managed knowledge workers refuse to accept management, they continue as they had done before, and then even less is coordinated, as authority is not accepted. It is a very dangerous situation, and emphasises the need for capacity to communicate and create a human resource environment for knowledge sharing and mutual need for coordination. In many ways such traits may be found in other institutions as well, in research institutions, but the difference to the IT-world is the “self-identification” as artists, and not accepting a role as highly educated who understand the need to communicate and organise the knowledge. The arrogance of the IT elite-developers reveals a culture of autodidacts, and a culture, where the dependency on people with these special talents, create acceptance – maybe also for too long live with this kind of anarchy.

The reaction of the developers in many ways illustrates some of the problems of the “playfulness” and ways of working in the dot-com bubble. It is a very irresponsible way of working in a firm, as it is not “only a playground”. The lack of responsibility of the decentralised authority cannot continue. In this case the bubble bursted. The problems of decentralisation and the need for deadlines and creating economy in the projects is fundamental, and methods to control are partly the same as always, as recruitment and laying off people is still an option. More than ever, it is important to create a joint platform of understanding, from which the knowledge workers can decide on methods and more incremental issues, where other decisions involving other people, and other groups in the firm, have to be communicated and decided at management level as well.

## 6. MOTIVATION AND CONTROL.

The problem of credibility and trust to management are serious in a knowledge intensive firm. The ability to persuade externally is based on internal knowledge, and internally the insight depends on credibility. When managers are recruited to complement skills, and supply insight in economic issues and contracts, their knowledge can only be used in action, if they can relate to the subject matter and the real deadlines of both R&D and production. Credibility is easier accessible within disciplines and communities of practice, but also the demand for other skills in management create diversity and boundaries. When communicating across disciplines, as many professional managers

do, it is even more important to question and de-brief to get a platform for decision making, which may be recognised as relevant by the knowledge workers. If credibility is lost, the hidden more than tacit knowledge is kept hidden for decision makers in the firm, and the credibility of the efficiency measures may be at stake.

A small IT firm may have more than one owner, with different management functions, and some projects leaders who also participate in most decisions. Such structures are part of the knowledge work, and its organisation. The hidden knowledge is seen even in large corporation, where R&D workers may get a green light to start a programme, but deliberately keep top management in the dark in the stages, where the project is very 'experimental'. Only when a new project begin to look less chaotic and more organised it may be "sold" internally in a political legitimisation process (see also Staudt 1997).

The project manager has to create legitimacy as manager, both internal in the group and outside the group. The project managers are more like the developers and may easier get internal credibility, but the conditions within the IT market, often imply that the time pressure and resources are under constant changes, and project managers are also part of the stressful game of efficiency to limit the beautiful solutions for the necessary solutions. They have to adapt to new and stricter conditions at regular intervals, and have to negotiate and motivate in their teams to keep up motivation and energy in the critical moments.

The role of management in knowledge intensive firms could be seen as a negotiation and social construction of power in the communication with the people subordinates. Power is given by the employees, and constituted in the community. Motivation has to be constituted and motivations are not linked to "efficient work on the necessary", but the chance to elaborate on the best solutions to explore and learn. Interviews with Christensen indicate the need to remain playful on the challenges of development. "if not, they could just as well have a 40 hour job with one of the large firms"( Christensen & Monsted 1999)

The knowledge workers are doing more than they should do, but in this way they overdo solutions, and keep on the track of artisan work rather than industrialise and recycle solutions.

## 7. CONCLUSION.

The empirical cases in the article are used to raise questions on roles of management in knowledge intensive firms as well as try to use and illustrate a method of videointerviewing. The knowledge intensive firm is based on mobilising knowledge workers. Being loyal to the perception of this kind of firm as a 'system of persuasion', managers have to follow these rules as well as the other members of staff. In a development process, where new knowledge is created, the lack of shared knowledge is detrimental to the organising of the firm, and new roles of managers have to be found.

The project manager's de-briefing of developers in IT firms in order to get organisational knowledge for managing is used as the illustrating case. The de-briefing is not only a translation of codes, but also an effort to understand the context of codes for managing division of labour. The explicitation of knowledge in these interviews is a very complex task, and demands a high level of insight and ability to communicate across disciplines and communities of practice. The sharing of knowledge is specially difficult in some high tech communities, where only the experts understand

what is going on, and the manager belongs to another professional discipline and community of practice, but (s)he has to coordinate and decide. The communication within the communities of practice may not depend on such a level of explicit knowledge, but the communication to managers appears to be across communities or disciplines, and the understanding of the complexity is difficult. When other disciplines and managers have to get into the substance, the emphasis is on explication, even if it may not provide sufficient understanding within a foreign discipline.

The understanding of the necessity of management is very different in the various groups, and reflects not only difference in conditions of work, but also education, interests and the level of playfulness of the dot-com generation. The IT-field is not dominated by highly educated as other knowledge intensive firms (Alvesson 2001, p. 863), but have a lot of the same features and other types of experienced expertise. If the knowledge workers are motivated by factors, which are related to "artisan" production, rather than industrial standardisation, then managers will have to balance out motivation and efficiency. The over-doing work could be an interesting issue to study in other knowledge intensive work as well.

A main research question remains: They work too much. Is this a problem?  
Maybe motivation is working on good solutions, but economy is in the necessary solution.

#### LITERATURE:

Adams, Barbara. (1990) *Time and Social Theory*. Cambridge: Polity Press.

Alvesson, M. (2001) Knowledge Work: Ambiguity, image and identity. *Human Relations*, 2001, 863-886

Brogren, J. Waras. (2001) Om materialiseringer af ideer. Et casestudie af samspil, videnprocesser og ledelse i den midlertidige IT-organisation. cand.merc.int. hovedopgave . Copenhagen Business school

Brooks, F. B. (1995) *The Mythical Man-Month*. Boston: Addison-Wesley.(first edition 1975)

Burt, R. (1992) *Structural Holes. The social Structure of Competition*. Mass. Harvard University Press.

Christensen, Søren and Kristian Kreiner. (1991) *Projektledelse i løst koblede systemer - Ledelse og læring i en ufuldkommen verden*. København: Jurist- og Økonomforbundets Forlag

Christensen, P. Holdt & Mette Mønsted: (1999) Uncertainty as Motivation or When Nerds Rush In - a counterculture to managing routines, in *During & W., Oakey, R.P. and Mukhtar S.M. 'New Technology Based Firms in the 1990s Vol.6'*, Elsevier Science Ltd, Oxford

Collins, H. M. (1974), "The TEA Set: Tacit Knowledge and Scientific Networks," *Science Studies*, 4, 165-86.

- Davenport, Thomas H. and Laurence Prusak. (1998) *Working knowledge. How organizations manage what they know*. Boston: Harvard Business School Press
- Garud, R. (1997) On the Distinction between know-how, know-what and know-why. *les Cahiers du Management Technologique*, 1997, 5-30
- Gourlay, S. N. (2002), "Tacit Knowledge, Tacit Knowing, or Behaving?" *The Third European Conference on Organizational Knowledge, Learning, and Capabilities* (Athens, Greece, 5 Apr-6 Apr), forthcoming in H. Tsoukas and N. Mylonopoulos (eds) (2003) [a selection of papers from the conference]
- Hampden-Turner. (1990) *Charting the Corporate Mind. - From Dilemma to Strategy*. Oxford: Blackwell Publishers
- Jensen, Sisse Siggaard (2001) De digitale delegater – tekst og tanke i netuddannelse. Multivers Kbh.
- Jensen, Sisse Siggaard and Mette Mønsted. (2001a). Tidsparadokser og ledelsedilemmaer i IT projekter. In *Det er et spørgsmål om tid*, edited by Poulfelt, F. and M. Mønsted København: Samfundslitteratur CD-ROM publication.
- Jensen, Sisse Siggaard and Mette Mønsted. (2001b). Managing uncertainties in R&D processes in small IT firms. EGOS Conference 2001, Lyon. subtheme 26, The Scientific & Technical Odyssey within & between Organization.
- Kidder, Tracy. (1981). *The Soul of a New Machine*. New York: Avon
- Kogut, Bruce and U. Zander. 1996. What Firms Do? Coordination, Identity, and Learning. *Organization Science*, 1996, 502-518
- Latour, B. & S. Wolgar (1979) *Laboratory Life. The Social Construction of Scientific Facts*. Sage. Los Angeles
- Newell, Sue, M. Robertson, H. Scarbrough, and Jacky Swan. (2002). *Managing Knowledge Management*. N.Y.: Palgrave
- Newell, S. & J. Swan (2000) Trust and inter-organizational networking. *Human Relations*. Vol. 53 n.10 pp. 1287-1328
- Nonaka, I. and H. Takeuchi. (1995). *The Knowledge Creating Company. How Japanese companies create the dynamics of innovation*. Oxford: Oxford University press
- Normann, R. (2000) *Service management. Strategy and leadership in service business*. John Wiley Chichester.
- Stacey, Ralph D. (2001) *Complex Responsive processes in organizations. Learning and knowledge creation*. London: Routledge

Starr, Susan L. (1989) The structure of Ill-Structured Solutions: Boundary Objects and Heterogeneous distributed problem solving. In *Distributed Artificial Intelligence vol. II*, edited by Gasser, L. and M. N. Huhns (London: Pitman)

Staudt, Hubert. (1997) Initiating Corporate renewal: A Study of the process of Fundamental Innovation Development. Initiating Corporate renewal: A Study of the process of Fundamental Innovation Development, Copenhagen Business School, Ph.D. Dissertation

Weick, Karl E. (1995) *Sensemaking in Organizations*. Sage, Thousand Oaks

Wenger, ( 1998) *Communities of Practice. Learning in Doing. Social, Cognitive, and Computational Perspectives*. Cambridge University Press.