

STANDARDIZING SOFTWARE PROCESSES—AN OBSTACLE FOR INNOVATION?

Ivan Aaen¹ and Jan Pries-Heje²

¹Aalborg University, Denmark; ²The IT-University of Copenhagen, Denmark

Abstract: Over the last 10 years CMM has achieved widespread use as a model for improving software organisations. Often CMM is used to standardise software processes across projects. In this paper we discuss this standardisation of SPI in relation to innovation, organisational size and company growth. Our discussion is empirically based on years work and experience working with companies on SPI. In the concrete our discussion is enhanced by vignette stories taken from our experience. As a result we find that standardisation focussing on process, metrics, and controls may jeopardize innovative capabilities and company growth. We conclude by identifying and describing four mechanisms that can be used to avoid disaster.

Key words: Innovation, Standardization, Software Process Improvement, organizational change, company growth

1. INTRODUCTION

A standard is “something established and generally accepted as a model, example, or test of excellence ...” or “something established as a measure of reference of weight, extent, quantity, quality, or value” (Websters 1997). The effort that software producing organizations are making to become better, more efficient, faster, or improved in other ways have come to be known as Software Process Improvement (SPI). Some SPI efforts are carried out in an ad-hoc based way but quite many has used the Capability Maturity Model as a map to improve (Aaen *et al.* 2001).

The Capability Maturity Model - or just CMM - is a framework characterizing a 5-step path for software process improvement (Paulk et al. 1995). The path describes key processes at each of five levels. The description includes a number of goals at each level. An organization has to

meet the goals at one level to reach the next. I.e. to go from the basic level 1 where behaviour is characterized by being ad-hoc and intuitive to level 2 you need to achieve the goals incorporated in six key process areas: requirements management, sub-contractor management, project planning, project tracking, quality assurance, and configuration management.

In fact CMM has become a generally accepted model for software process improvement. Several thousand organizations have undertaken SPI using CMM. In September 2003 the Software Engineering Institute reported (SEMA 2003) that 11823 projects in 2150 organizations had been appraised using CMM since 1987, and 1342 of them within the last four years. Thus it is fair to call CMM a standard for SPI.

But is it a good idea to standardize SPI using CMM? A focus on process, on metrics, and on controls may jeopardize innovative capabilities. 10 years ago CMM proponents such as Bill Curtis (cf. 1994) argued that the CMM enables innovation by removing some of the process-based impediments for exploring-designers, giving them time to think, rather than inhibiting them with a chaotic environment that limits their time and opportunities to exchange ideas. On the other hand critics such as James Bach (1994) claimed that “CMM is dangerous to any company founded upon innovation. Because the CMM is distrustful of personal contributions, ignorant of the conditions needed to nurture non-linear ideas, and content to bury them beneath a constraining superstructure, achieving level 2 on the CMM scale may very well stamp out the only flame that lit the company to begin with.”

So Bach claimed that a standard such as CMM was a major obstacle to innovation. He based this argument on (among other things) personal observations in the two companies Apple and Borland; both producing IT products in Silicon Valley. We set out to research whether that is (still) a true statement in practice – in a Danish context? Or whether it is possible to standardize without sacrificing innovative capability in software process improving organizations?

2. RESEARCH METHOD

The first step in answering the research question was to survey what others had written on innovation, standards and SPI. We found quite many references to each of these three areas but very few to the combination of 2 or 3 of them. In the sections below we give an account of what we found.

The second part of our research was a qualitative study. Over the last seven years since 1997 we have been involved as both neutral observers and action researchers in three companies. Furthermore we carried out an interview study in a smaller company. Together this gave us a large amount

of data that we could use to analyze the impact of CMM as a standard in practice. Details about the companies are given in **Table 1**. We have chosen not to reveal the true identity of the companies.

Table 1. Facts about the case studies drawn on in this paper

Case	Size of organization at the time of the interviews	Industry and what offered	Where situated	Employees interviewed	Organizational roles represented in interviews
Mini	20-25 employees.	E-commerce, systems integration and web applications	Denmark, Copenhagen	3 in 2001, and 3 in 2003	CEO; software architect and developer
Midi	More than 500 employees – approximately 1/3 develops software	IT applications for the financial sector	World wide	More than 10 in each year 2003	CEO; SPI-manager, project manager; process user / developer
Maxi	More than 1000 IT-related employees	IT applications for use in the financial sector	Scandinavia	More than 10 each year in 1997-99 and again 2001-03	CEO; department managers, project managers, software developers
Mega	1980s; More than 10000 IT-related employees	Tele-communications	World wide	More than 10 in each year 1997-1999	CEO; project manager; software architect/ developer

It would be a paper in itself to give an account of each of the companies. Instead we have chosen to include so-called “vignettes” where it is relevant in our analysis. In that way our empirical data is used to illustrate and substantiate important points.

3. INNOVATION

Innovation is both a term of importance but also a quite ambiguous term that many researchers have tried to understand and give an overview of (cf. Nelson & Winter 1977, Dosi 1988). A core distinction seems to be between *innovative* product development and *routine* product development (Christensen 2002). Innovative product development is based on an idea that is perceived as new by experts or buyers/users (the latter is equal to Rogers’ (1995) definition of innovation). The development process is associated with uncertainty, either technical (can it be done?) or use-related (will they accept

it?), and the process itself is often carried out as a distinctive and targeted searching process. Whereas the result of routine product development to experts or users often is obvious, there is no substantial uncertainty, and the process can often be characterised as just ordinary “seek and find”.

Vignette #1 illustrating routine product development

Mini is a software house that among other things develops database intensive web applications. Every application is new, the customer is new, and to the user it probably appears that the final application is new. But Mini has a standardized development process that they use every time. It included making 2-3 prototypes, showing them to the user, getting feedback, and incorporating the feedback in the next prototype or the final application. By having this process in place that among other things reduce ambiguity and uncertainty Mini has reduced an innovative development process to a routine development process.

Our question is whether both or only one type can be standardized? Certainly the process of coping with uncertainty can be standardised. One can also standardise the seek-and-find process involved in routine product development. But the good idea – the “new” thing – cannot be standardised.

Another important distinction seems to be between:

- Product innovation where the results of the innovation is sold on a market and thereby used outside the company
- Process innovation where the results (better processes) are used internally

Vignette #2 illustrating the schism between standardisation and innovation

In the mid-90s Mega Headquarter decided to require that all the subsidiaries in different countries should use CMM. After a few years all subsidiaries were certified at level 2 or higher. Then, however, Mega ran into financial troubles. At this point Mega was heavily criticized in the press for developing products that were not innovative enough but over-engineered.

CMM is clearly aimed at process innovation, but with a belief that good processes will lead to good products (we will return to that belief in the next section). In practice it can also be done – by some at least. For example the Danish Software House Systematic, that are certified at CMM level 3 was appointed the most entrepreneurial company in Denmark in 2003. And the CEO pointed to CMM and good processes as a major reason for that achievement.

4. INNOVATION VS. STANDARDIZATION – HOW VS. WHAT & WHY

In this section we shall return to our main question: How can we standardize without sacrificing innovative capability in software process improving organizations? How can we avoid throwing out the baby with the bath water when doing SPI?

In *Managing the Software Process* (Humphrey 1989) Watts Humphrey defines the fundamental ideas embedded in the CMM. Here he defines a standard as a “rule or basis for comparison that is used to assess size, content, or value”, i.e. as a product standard. He combines this with a process-standard by defining a procedure as a “defined way to do something, generally embodied in a procedures manual” and by way of illustration he likens this definition with the procedure concept used in programming (Humphrey 1989). Based on IEEE-STD-610 the CMM combines Humphrey’s two notions into one and defines standard as (Paulk, Curtis et al. 1993): “mandatory requirements employed and enforced to prescribe a disciplined uniform approach to software development”.

CMM based SPI focus on improvement via standardization and formalization as exemplified by the following two process concepts key to CMM based SPI (Paulk, Weber et al. 1993):

- *Organization’s standard software process* - The operational definition of the basic process that guides the establishment of a common software process across the software projects in an organization. It describes the fundamental software process elements that each software project is expected to incorporate into its defined software process. It also describes the relationships (e.g., ordering and interfaces) between these software process elements.
- *Project’s defined software process* - The operational definition of the software process used by a project. The project’s defined software process is a well-characterized and understood software process, described in terms of software standards, procedures, tools, and methods. It is developed by tailoring the organization’s standard software process to fit the specific characteristics of the project. (See also *organization’s standard software process*, *effective process*, and *well-defined process*.)

To Watts Humphrey SPI targets a process that can be controlled, measured, and improved. Improvements to the process will only stick if they are reinforced by careful introduction followed by periodical monitoring (Humphrey 1989): “The actions of even the best-intentioned professionals must be tracked, reviewed, and checked, or process deviations will occur. If there is no system to identify deviations and to reinforce the defined process, small digressions will accumulate and degrade it beyond recognition.”

Here and elsewhere focus is on establishing and following a defined software process. Innovation and innovative capability is not a foremost concern in CMM based SPI. As defined and practiced in many organizations the CMM tends to emphasize operating norms and focus on building an ability to detect and correct errors in relation to such norms. Despite these efforts many organizations experience that such operating norms are often not adhered to, even after having invested heavily in developing standards and educating the work force in them.

CMM presents itself as TQM applied to Software developing organizations. Citing prominent TQM authors such as Deming, Juran, and others Watts Humphrey acknowledges the heritage from TQM, and Mark Paulk of the Software Engineering Institute states this relation directly: “CMM-based improvement is an application of TQM principles to software” (Paulk 1996).

In light of CMM being TQM applied we would expect a similarity between the two in the perceptions of standardization but this is not the case. There is a marked difference, and the difference seems not to be due simply to dissimilarities between the software industry and the industries traditionally using TQM, i.e. the car industry and similar manufacturing industries.

Kondo – a prominent writer on TQM – describes work standards broadly as consisting of three items: (1) aim of the work, (2) constraints on carrying out the work, and (3) means and methods to be employed in carrying out the work (Kondo 2000). In a major book Creech – another prominent writer on TQM – not even mentions let alone defines the term (Creech 1994).

Applying TQM to software teams Zultner also use a relatively lax definition of standards. To him a standard (Zultner 1993) “... may be a set of instructions, macros, drawings, and so forth – anything that allows the team to do something in a consistent way”.

The traditional CMM-SPI approach is centralist and top-down: (1) understand the current status of the process relative to CMM, (2) create a vision of the desired process relative to CMM, (3) establish a list of required actions, (4) produce a plan for accomplishing these actions, (5) commit resources required to execute, and (6) start over at step 1 (based on (Humphrey 1989)). Each loop is expected to contribute to the set of standards enforced in the organization. Due to the time frame involved in one loop this tends to create a gulf between SPI efforts and everyday practical processes.

In line with traditional TQM Zultner focus on teams in his description as opposed to CMM’s *centralist* perspective on function. Creech describes centralism as one major difference between TQM and mainly process-oriented type of improvement approaches such as CMM.

Another difference between CMM and TQM is scope. According to Paulk et al. the CMM essentially is a subset of TQM: “The CMM focuses on the process aspects of a Total Quality Management effort” (Paulk, Curtis et al. 1993).

Commenting on the “quality movement” Gareth Morgan describes how many TQM programs have got caught in old bureaucratic patterns and cultural norms, leading to disastrous failure rates (Morgan 1986). According to Morgan the philosophy of promoting continuous improvement has done much to institutionalize the practice of challenging taken-for-granted norms and practices at an *operational* level. Morgan describes the power of the Japanese concept of Kaizen like this (Morgan 1986):

- Employees are asked to dig beneath the surface of recurring problems and uncover the forces that are producing them.
- They are encouraged to examine existing modes of practice and find better ones.
- They are encouraged to create “languages,” mind-sets, and values that make learning and change a major priority.

Challenging operating norms and assumptions in this way, Morgan argues that quality approaches create information, insights, and capacities through which a system can evolve to new levels of development.

CMM-based SPI builds on formalization by process specialists and thus on separating thinkers from doers. This is at odds with Kaizen insisting that ongoing improvement be the province of every member of the organization and that alignment be ensured throughout the organization as well.

Even if we accept to focus only on the process aspects of TQM there are serious problems in CMM-based SPI. Zultner describes ongoing improvement (Kaizen) in TQM as a product of variation – all real-world processes vary and this variation forms the very locus of improvement. Thus ongoing improvement requires a combination of

- Standardization – standardize-do-study-act – to reduce variability and lock in later gains,
- Improvement – plan-do-study-act – at the team level.
- Innovation – mainly breaking through bottlenecks for dramatic process improvement.

CMM being a subset of TQM – the process part - leads to a focus on universal goals rather than goals derived from the market situation, and this again tends to separate CMM-based SPI from practice and to a focus on process perfection for the sake of the process itself, and aiming to build the ideal software process for the organization.

Separated from everyday concerns such process perfection will likely be relatively static failing to seize opportunities for improvement in practical work. In fact we can expect the process standard to become an obstacle to

change. As Kondo puts it: “since standardized working means and methods have been formulated after careful consideration of all the angles, they must be the most productive and efficient means and methods possible, regardless of who uses them – at least the people who drew up the standards think so” (Kondo 2000).

TQM is much more than Kaizen, and Kaizen is more than the CMM equivalent of SPC. Zultner on Quality function deployment (QFD) and Policy deployment (PD): These are completely absent in CMM – leading to only fickle focus on horizontal integration, on customers, on vertical alignment, and on advancing the organization for long-term survival.

4.1 Threats to innovation in the CMM approach

The above discussion has identified a number of important characteristic traits in the CMM approach to TQM:

1. *A strict process orientation.* According to Creech builds on five pillars: Product, Process, Organization, Leadership, and Commitment, but CMM basically only address the process-pillar (Creech 1994). Following Creech this means that SPI is not a holistic approach to quality.
2. *A centralist approach.* SPI is often practiced as a committee-oriented approach, alternatively as something being done by a specialist staff function. Either way this means a separation of thinkers from doers with a main focus on the organization-wide process. A focus on conformance rules, inputs, and procedures.
3. *A focus on function.* SPI projects are predominantly internal to the software organization. The aim is to improve internal work procedures based on the assumption that good products come from good processes.
4. *A focus on standard adherence.* The goal is to minimize variation and thereby improve process reusability and learning from previous projects.

What then are the threats to innovative capability coming from these traits?

A strict process orientation: The lack of customer orientation may lead to goal deflection in the improvement effort. Such goal deflection might result in employees seeing process adherence as a goal rather than a means. Process development might overshadow customer satisfaction leading to a waste of creative strengths on problems of less importance to the customer. A strict process orientation might also lead to process gold plating building ever more comprehensive and universal processes. Finally and perhaps worst: employees might “misuse” creativity on building process workarounds when organizational processes stand in their way.

Vignette #3 illustrating a very rigid and centralised process orientation

In Midi it was decided to use CMM to improve the organisation. An intensive effort led to the achievement of CMM level 2 in 2001. This effort was led by a centralised SPI-group. However, developers were complaining that following CMM added too much bureaucracy and that innovation was hurt. To cope with this Midi intensified the focus on a “bottom-up” SPI approach and focused improvements initiatives on the individuals in the organisation. A new de-centralised improvement effort was implemented to supplement the effort in the centralised SPI-group. The core idea was that responsibility for the core of both development and improvement needed to be decentralised to work.

A centralist approach: Centralism traditionally favours the deployment of solitary methods – i.e. that only one method is provided for a particular type of task. Kondo describes how solitary methods might lead people to feel that they are not responsible for non-conformance of product quality. Restrictive centralism thus might lead to employee alienation. Separating thinkers from doers means a split between insiders and outsiders in software projects. How can opportunities be seized and how can process innovations come to life? The split in itself might lead to stifled motivation at both ends. Turf guarding is also a likely outcome at both ends if careers accidentally collide in a clash between specific interests in the project and general interests in the staff function or committee. Lastly organization-wide process standardization might lead project members to turn on the autopilot or drop into apathy. None of these problems will benefit creativity or innovative capabilities neither at the level of software project nor at the organizational level.

Vignette #4 illustrating the danger of a centralised development method

In Maxi there was a development method in place describing phases and milestones, and making available templates and examples. An interview study in the organisation uncovered that the method wasn't being used due to three things: (1) Difficult to find things. (2) Build on one-size-fits all concept. (3) No help to tailor method for specific project. A project was established focusing on process tailoring (a CMM level 3 key process by the way). But before the project could report their recommendations top management had decided on a new centralized project model that every project should follow. In the new model there was a core and a number of options. So the tailoring project was stopped and the new project model was implemented. Everyone followed the new

project model because it was enforced – but in interviews it was found that the three problems originally identified were still there.

A focus on function. Despite the undisputed assumption that good products come from good processes the focus on function implies not focusing on the end product of the software project. This lack of a product focal point might lead to conflicts in installing a sense of purpose and achievement to project members. The focus on function is one step away from the machine bureaucracy where routines may be routinized for the sake of routinization alone. Such focus on formalization and routinization might easily stand in the way for creativity.

Vignette #5 illustrating too much focus on function

At Mega the Headquarter had had enforced everyone to use CMM. So a lot of effort was put into implementing processes and following them strictly. In hindsight it is clear that the effort should have been directed towards the product instead. Because competitors took over several important markets while Mega was honing their processes.

A focus on standard adherence. Where adherence to norms becomes and end in itself fundamentalism lurks in the shadows. Fundamentalism might take the form of dogmatism, of mind control, or of goal deflection. This is when rules and regulations come to stand in the way for creativity and achievement. Another and less conspicuous threat coming from standard adherence is a blocking of variation. If successful SPI efforts lead to software projects becoming ever more similar in process the end result might be, that the very variation that fosters creativity and innovation is obliterated.

Vignette #6 illustrating too much standard adherence

This is a story that was part of the culture at Maxi. About 10 years ago there was a development model that was quite lengthy and very detailed. Everybody was supposed to follow it from page 1 to page 2000. The saying was that you didn't have to bring your brain to work; you could just start on page 1 and follow the process. Furthermore it was controlled whether you followed it, and warnings were given if you didn't. But as it always happens in organisations there came a catastrophic project. But the project leader claimed that he had followed the process – a good example of too much standard adherence. And so it was decided to give up the very detailed development model.

5. STANDARDISATION OF SPI IN RELATION TO COMPANY GROWTH

Innovation literature talks about two “engines” that lies behind technological innovation in organisations. One is the entrepreneur model the other is the big company model. Traditionally it has been argued that bigger is better. If you are big you can invest more, you are less vulnerable to failure, and you have the resources to attract the best. An entrepreneurial organisation on the other hand is a small independent company that can react fast. Acs & Audretsch (1988) and Rothwell & Zegweld (1982) argue that the entrepreneurial organisation has four advantages compared to the big company:

1. Management and decision structures are lighter and faster in small companies
2. Small companies are often headed by dynamic innovative people involved in the innovation process themselves
3. Innovation often involves communication across functions. In small companies that is simple, whereas in larger companies that type of communication can involve crossing departments, borders, countries, culture etceteras
4. Larger companies rewards their best and most innovative people by lifting them up to a management position – and thereby effectively taking them away from innovation

So can one standardize the advantages of a small entrepreneurial company? Let us take the advantages one by one. Small and light decision structures are at odds with CMM. In fact CMM specify a number of key processes that will add structure and will enforce more decisions to be made. Standardisation will not influence the involvement of dynamic managers in the innovation process. Whom you promote is neither influenced. But how you communicate is at the core of CMM. In fact CMM enforces a specific repeatable and measurable way of communicating (at least if you move up the levels). However, there is nothing in CMM that enforces slow or cumbersome communication. Probably the continuous improvement idea built into CMM will surmount some of communication disadvantages of enterprise growth.

Vignette #7 illustrating how rewarding may counterbalance innovation

At Maxi a local project manager was known all over the company for having saved the organisation from an embarrassing failure. An off-shore outsourcing project was again and again delivering code that could not be

used. Finally the local project manager stepped in, broke all the rules (also a kind of process innovation), and managed to deliver the new innovative product on time as promised to the customers. She was soon promoted to second-in-command in that development unit, and thereby effectively out of the loop when talking about the day-to-day work on innovations – but probably with more influence on innovations from a management viewpoint.

Vignette #8 illustrating communication barriers

In Mega three set of stakeholders were trying to set the agenda. Headquarter were pushing CMM. Another department at Headquarter had helped the Danish subsidiary identify a number of so-called Vital Few Actions. And locally a number of improvement areas had been decided. Communication between the three functions was lacking. The result was paralysis. Nothing happened until the local management cut through and made a decision on which one of the three agendas should be the one to be followed.

Companies start as small and possibly entrepreneurial, and if they are successful they may expand. Larry Greiner (1972, 1988) has developed an evolutionary model of company growth.

In his model the organization starts its life in the Creative Phase shown in the lower left corner. In this phase, the founders of the company are typically entrepreneurs, communication among the people in the organization is informal, long work hours are normal, and the feedback from the market is immediate, as well as the reaction from management.

As the company grows in size and matures, it reaches its first crisis. Dubbed the Crisis of Leadership this is when informal communication no longer will suffice. The dedication, long hours and small salaries of the first hired “pioneers” are no longer adequate motivation. Furthermore new procedures are needed to exploit economies of scale and to provide better financial control. To solve the leadership crisis a strong manager is required. Often the owner/founders will not have the necessary skills and knowledge, and so will “hate to step aside even though they are ... unsuited to be managers” (Greiner 1972, 1998).

Having survived the leadership crisis the organization will “embark on a period of sustained growth under able and directive leadership” (Greiner 1972, 1998). In this Direction Phase hierarchies develop within the organization, the upper levels take responsibility for the direction of the organization, and communication becomes more formal. Also in this phase, formalized systems for accounting, incentives, work practice, and job specialization will arise.

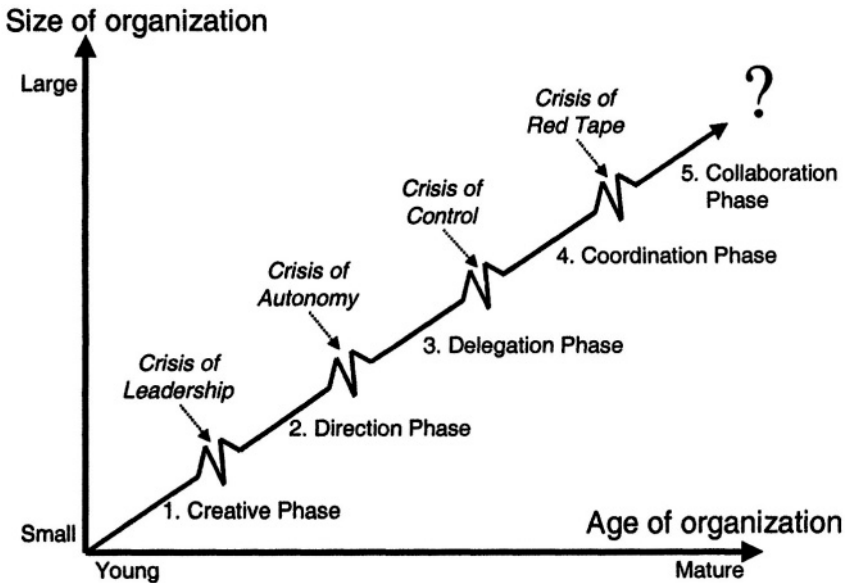


Figure 1.

In this second phase of organisational growth a standard can be useful to formalise systems and communication.

The second crisis then is the Crisis of Autonomy. Middle-level managers come to see the centralized decision structure of the second phase organization as a burden, and some autonomous field unit middle managers will start running their own shows. Often the reaction by top management is an attempt to restore centralized management. At this point some organisations stick to standardisation but face a number of the disadvantages discussed in section 4 such as a too centralist approach or an overly strong focus on standard adherence. To solve the autonomy crisis a more decentralized organization structure is needed offering greater responsibility and autonomy to middle management. This structure marks the start of the Delegation Phase.

Vignette #9 illustrating the usefulness of CMM

In Mini the owner and CEO wanted to appear and actually work more professional to be able to survive in the marketplace (with many big competitors) and maybe even grow from the 20 employees they had had for along time. They decided to use CMM as inspiration. CMM was presented to the employees. They discussed it and voted on what they

found most useful. The outcome were three new processes that were then adapted to the organisations and implemented as the local “standard” – this is how we do things here. After the implementation the CEO described CMM as highly useful in the professionalisation of Mini.

The Delegation Phase ends in the Crisis of Control where top management realizes that they have lost “control over a highly diversified field operation.” Again using standardisation to centralise and to focus on standard adherence is not the answer. This third crisis is overcome by using coordination techniques such as formal planning, by organising product groups as investment centres, and by initiating staff functions that controls and reviews for line managers. This way the company enters the Coordination Phase.

The next crisis then is the Crisis of Red Tape where the line is looking at staff functions with more suspiciousness, and distrustfulness evolves between headquarters and the field. Overcoming this fourth crisis then leads the organization into what Greiner calls “the last observable phase in previous studies.” In this last phase strong interpersonal collaborations are established to overcome the crisis. A more flexible and behavioural management approach is implemented through the use of teams. Staff functions are reduced in number, and the motivational structure is geared more to team performance than to individual achievements.

6. DISCUSSION

We have now analysed the standardisation of SPI in relation to innovation, organisational size and company growth. We have found that:

- a) Standardisation can mainly be used for routine product development – and coping with uncertainty in innovative product development
- b) Standardisation using CMM is clearly aimed at process innovation
- c) Using CMM for standardisation easily leads to process orientation that is too strict
- d) Using CMM for standardisation can easily be too centralized
- e) Using CMM for standardisation can lead to much focus on function
- f) Using CMM for standardisation can lead to too much focus on standard adherence
- g) CMM can be useful in the first phases of a company’s growth life-cycle, especially in the direction phase
- h) CMM can be dangerous to use in the later phases of a company’s growth life-cycle

So what can be done? We have four suggestions:

1. A holistic quality approach
2. A decentralist approach building on teams at the operational level
3. A project-oriented approach (as opposed to a focus on function)
4. A focus on situational discretion (as opposed to a focus on standard adherence)

But let us take the discussion one at a time.

A holistic quality approach (as opposed to a strict process orientation). This is probably the most controversial of our propositions. To our mind CMM-based SPI seems fundamentally flawed by its focus on process, metrics, and controls. TQM is a systemic or holistic approach involving process but also product, organization, leadership, and commitment. Perhaps software process improvement (SPI) really should be software total quality management (STQM). We believe that innovative capabilities may be strengthened by a focus on customers and on products. For project members this leads to an emphasis on goals and achievements relative to the project.

A decentralist approach building on teams at the operational level (as opposed to a centralist approach). A decentralist approach could mean methodological pluralism where projects experience degrees of freedom towards means combined with clarity with respect to ends. When organizational level means and methods are required they could be split in two groups: one for novices and one for experienced workers. Kondo argues in favour of such a split (Kondo 2000). Novices can use manual describing essentials of a basic process that should be known by every employee. Experienced workers can share special tips and tricks in the other group.

A decentralist approach would build on personal involvement and responsibility. The variations involved in any real-world software project would provide good opportunities to seize opportunities for process innovations.

A project-oriented approach (as opposed to a focus on function). Such an approach would entail a focus on the team rather than the function. Teams and individuals situated in practice allows for innovative solutions to processes and products. The project team is the ideal unit for handling improvisations possibly leading to process and not least product innovations.

A focus on situational discretion (as opposed to a focus on standard adherence). Situational discretion requires that authority and accountability be localized in the same team. Kondo argues in favour on less emphasis on process standardization and more on product standardization (Kondo 2000). This implies more focus on the aim of the work and on providing freedom in the use of means and methods, and less focus on process enforcement to allow for creativity. Following Kondo our proposition is to limit process standards to a minimum and thereby to focus less on “dos” and more on

“don’ts” when procedural regulations are needed. The will result in fewer and leaner instructions replaced when needed with constraints. An ideal incubator for innovation.

7. CONCLUSION

In this paper we have discussed the standardisation of SPI in relation to innovation, organisational size and company growth. Our discussion is empirically based on years work and experience working with companies trying to improve their software development. In the concrete our discussion was substantiated by vignette stories from our experience. As a result of the conclusion we find that standardisation with a main focus on process, on metrics, and on control may jeopardize innovative capabilities and company growth. However, we suggest to use the following four mechanisms thoughtfully to allow for a culture of innovation: (1) A holistic quality approach, (2) A decentralist approach building on teams at the operational level, (3) A project-oriented approach, and (4) A focus on situational discretion

ACKNOWLEDGEMENTS

The authors contributed equally to this article; the order of their names reflects alphabetical order.

REFERENCES

- Aaen, I., Arent, J., Mathiassen, L. and Ngwenyama, O. (2001). A Conceptual MAP of Software Process Improvement. *Scandinavian Journal of Information Systems*, Special Issue on Trends in the Research on Software Process Improvement in Scandinavia, Vol. 13, pp. 123-146.
- Acs, Zoltan J. & David B. Audretsch (1988). Innovation in large and small firms: An empirical analysis. *American Economic Review*, vol 78, no. 4, 678-690.
- Bach, James (1994). The immaturity of the CMM. *American Programmer*, 7 (9), 13-18.
- Christensen, J.F. (2002). Produktinnovation – process og strategi (“Product Innovation – process and strategy”). 2nd Edition. Handelshøjskolens Forlag.
- Creech, B. (1994). The five pillars of TQM: how to make total quality management work for you. New York, Truman Talley Books/Dutton.
- Curtis, Bill (1994). A mature view of the CMM. *American Programmer*, 7 (9), 19-28.
- Dosi, Giovanni, Christopher Freeman, Richard Nelson, Gerald Silverberg & Luc Soete (Eds.) (1988). *Technical Change and Economic Theory*. Pinter Publishers.

- Greiner, Larry (1972). Evolution and revolution as Organizations grow. *Harvard Business Review*, (50), 37-46.
- Greiner, Larry (1998). Evolution and revolution as organizations grow. *Harvard Business Review*, May-June, pp. 55-64.
- Humphrey, W. S. (1989). *Managing the Software Process*. Reading, Massachusetts, Addison-Wesley Publishing Company.
- Huy, Quy Nguyen (2001). Time, Temporal Capability, and Planned Change. *Academy of Management Review*, 2001, Vol. 26, No. 4, pp. 601-623
- Kondo, Y. (2000). "Innovation Versus Standardization." *TQM Magazine* 12(1): 6-10.
- Morgan, G. (1986). *Images of Organization*. Newbury Park, Cal., Sage Publications.
- Nelson, Richard & Sidney Winter (1977). In search of useful theory of innovation. *Research Policy* (6) 1, pp. 36-76.
- Paulk, M. C., Weber, C., Curtis, B., and Chrissis, M. B. (1993a). *Capability Maturity Model for Software, Version 1.1*, Software Engineering Institute.
- Paulk, M. C., Weber, C., Curtis, B., and Chrissis, M. B. (1993b). *Key Practices of the Capability Maturity Model, Version 1.1*, Software Engineering Institute.
- Paulk, M. C., Weber, C., Curtis, B., and Chrissis, M. B. (1995) *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, Reading, Mass.
- Paulk, M. C. (1996). *Effective CMM-Based Process Improvement*. 6th International Conference on Software Quality, Ottawa, Canada, SEI.
- Rogers, Everett M. (1995). *Diffusion of Innovations*. 4th Edition. Free Press, New York.
- Rothwell, Roy & Walter Zegveld (1982). *Innovation and the small and medium sized firm*. Frances Pinter, London.
- SEMA (2003). *Software Engineering Measurement and Analysis Initiative. Process Maturity Profile, Software CMM, CBA IPI and SPA Appraisal Results, 2003 Mid-Year Update*, September 2003. Downloaded November 2003 from www.sei.cmu.edu.
- Websters (1997). *The new International Webster's Concise Dictionary of the English language*. International Encyclopedic Edition. Trident Press International.
- Zultner, R. E. (1993). "TQM for technical teams." *Communications of the ACM* 36(10): 79-91.