Quality Management for Small Enterprises

Jürgen Börstler, Olof Johansson, Astrid Laryd, Terttu Orci, Hans Segerberg, Ola Ågren

UMINF-02.20
ISSN-0348-0542

UMEÅ UNIVERSITY
Department of Computing Science
SE-901 87 UMEÅ
SWEDEN

Quality Management for Small Enterprises

Jürgen Börstler, Olof Johansson, Astrid Laryd, Terttu Orci, Hans Segerberg, Ola Ågren

Umeå University
Department of Computer Science
Umeå, Sweden

Technical Report UMINF 02.20

ABSTRACT

Software Process Improvement has gained increasing interest and attention in the software industry. Frameworks such as SW-CMM and ISO/IEC 15504 (SPICE) have been applied widely. More recently they have also been validated in case studies and surveys. The validation has so far concerned use of CMM in large organisations. Most frameworks are not directly applicable to small organisations. A legitimate question to ask is whether the small organisations need software process improvement in the same sense as understood by SW-CMM, or not. We argue that they do. It is preferable to grow in an orderly and predictable way instead of an uncontrolled chaotic way. Furthermore it is easier to do process improvement in good time, so that certain processes are in place when they are really needed.

In order to avoid the organisational overhead inherent in the SW-CMM we have downscaled and adapted this model to the needs of small (growing) organisations.[*] Frameworks and models like this should ultimately be validated in long-term industrial use. Such a validation has not yet been possible for our model, constrained by the time window and frequency of its application. We present a validation by reasoning along the lines of Paulk's framework of "common sense".

1. Introduction

Software Process Improvement (SPI) has gained increasing interest and attention in the software industry. Well-known frameworks, for example SW-CMM (Paulk et al, 1994), ISO/IEC 15504 (ISO/IEC, 1998) and ISO 9000 (Kahoe & Jarvis, 1995), have been applied extensively (Herbsleb et al, 1997). It is often argued that SW-CMM is too heavy a framework to be usable for small organisations to initiate and conduct improvement work. The SW-CMM for example, proposes more than 25 organisational roles, with various tasks and responsibilities. In small organizations many of those roles are obsolete, but even for the remaining ones, there are not enough people to assign to them. The documentation of the models is furthermore quite extensive and not suitably organised. The documents, roles and responsibilities are scattered all over the text, without an easily accessible oversight.

Another often discussed argument is that small organisations do not need software process improvement, because there is easy communication and a simple management hierarchy. Contrary, we argue that this would be an ideal situation to start software process improvement for two main reasons; (1) it is easier to implement for exactly the same reasons, and (2) it will be easier for the organisation to grow in an orderly and predictable way. However, to make the SW-CMM applicable to small organisations it needs to be scaled down.

We propose the usage of Dynamic CMM for Small Organisations, which is compatible with the original SW-CMM, but adapted to the needs of small (growing) organisations. An organisation that grows in size and/or maturity beyond the limits of our proposed model can seamlessly change to the original SW-CMM building on the work done within the context of our framework. All documents produced will remain valid.

[*] This work was partially supported by VINNOVA, the Swedish Agency for Innovation Systems, by grants 2001/01908 and 2001/02020.

So far, our model has been introduced to four software organisations. However, not all organisations made use of all of the components provided by our model. A second formal assessment to validate improvement efforts and the suitability of our model could be made in one case only. The validity of our model can ultimately only be proven by means of long-term industrial use. In the context of this project this is unfeasible. However, we will validate the model including its extensive supplements by reasoning, according to Paulk's framework for "common sense", drawn from industrial knowledge and experience (Paulk, 1998).

The remainder of this paper is organised as follows. In Section 2 we discuss related work. Section 3 gives an overview over the Dynamic CMM for Small Organisations, and section 4 presents the framework for validation. Section 5 concludes with some initial results using our model.

2. Related Work

Brodman and Johnson (1995) describe the issues encountered by small businesses, organisations, and projects, when applying the CMM for SPI, like documentation overload, layered management, scope of reviews, limited resources, high training costs, and unrelated practices. A set of tailoring guidelines for each of the issues is presented in (Brodman & Johnson, 1995), and a tailored CMM, based on the original CMM with explicitly marked additions and deletions. The same authors have also presented more detailed tailoring guidelines for different issues, like documentation tailoring and review tailoring.

It is important to emphasise the scalability issues of Software CMM for small organisations. We believe that not only should the model be downscaled, but also the representation should be restructured for simplicity and ease of use. We have included these criteria in our work. Kautz (1998) investigated critical success factors for SPI work in small organisations and presented the following list; a flexible, tailored assessment and improvement approach, a functioning network of small enterprises and their environment, external technical help by mentors for change, and external financial support coupled to some conditions for performance. Our model is intended to support the first factor. Although external technical help may be needed to take the first steps in the SPI work, we believe that only organisations with an internal reasonably strong competence in SPI have the chance to conduct successful long-term SPI work. External services should only be needed for assessments to obtain impartiality.

3. The Dynamic CMM for Small Organisations

Small organisations need models with the following characteristics to be able to undertake SPI work in a cost-effective way.

Downscaled to small organisations' needs. The roles, activities, guidelines, documents etc. comprising the model must be adapted to suit organisations with few employees. The bureaucratic overhead necessary to implement and sustain the model must be minimised.

Easy to use and comprehend. The documentation must be organised in an accessible way. Information must not be scattered over several documents.

Provide guidance for assessment and improvement work. The model should enable organisations to implement and evaluate SPI activities locally, reducing the need to hire external experts.

Support organisational growth. The model must be flexible and adaptable to the needs of a growing/shrinking organisation. An organisation that grows in size and/or maturity beyond the limits of our model should be able to seamlessly change to the original SW-CMM.

3.1 The Model

The most obvious downscaling concerns the number of employees. We have focused on 1-50 people in the development organisation. The smallest size was chosen to suit even the smallest start-up. For more than 50 people we would recommend to start with the original SW-CMM model directly. In order to offer realistic possibilities to implement the model, the number of roles must be brought down. We believe that special conditions apply to an organisation with one or two people only, because it is fairly easy to be continuously informed of what the colleague is doing. We call such organisations eXtra eXtra Small organisations (XXS). As soon as the

organisation grows to include three persons, the loss of oversight and easy communication are at risk. Yet, an organisation with three employees is very much smaller and easier to manage than an organisation with nearly 50 employees with several products, several versions of each product or customers at a time. It seems adequate to further classify the organisations to eXtra Small (XS) with 3-15 employees and Small (S) with 16-50 employees. The models are intended to be applicable for organisations developing software products to open market, as well as for organisations developing customer specific solutions. In case there is a difference in roles or otherwise between these organisations, it is indicated in the text. The employees adding to the count are assumed to be either development staff, management, or marketing and sales, excluding administrative services and human resources staff.

The submodels differ mainly in the number of roles. The model is intended to be applied mainly by management and an organisation's internal staff, with some experience of SPI. The basic idea is that the model users can easily get an overview of the model, of the magnitude of the effort required and of the roles, responsibilities, tasks, and documents. With those intentions in mind, the original CMM has been interpreted, reduced, reorganised, and partially represented in a graphical notation.

The current model is restricted to maturity Level 2. When an organisation either grows beyond the limit of a small organisation, or maturity Level 2, the original SW-CMM should be a natural basis for the following SPI activities. To ensure this compatibility, the original (SW-)CMM intentions are kept intact as much as possible.

The model has been briefly presented in (Laryd & Orci, 2000) and (Orci & Laryd, 2000b) and in detail in (Orci & Laryd, 2000a). The supplements are described in detail in (Orci et al, 2001). The model is based on SW-CMM for pragmatic reasons. SW-CMM is the most well-known and most commonly used model, and the model most familiar to us. Other models, ISO15504 (SPICE), could equally well be used as the basis.

3.2 Role Models

The SW-CMM proposes more than 25 roles on maturity Level 2, which is too high for small organisations, and must be reduced in some way. One possibility is that one and the same person takes on different roles. We have studied the roles in CMM, their interdependencies, responsibilities and activities, to find conflicts that would prevent a person from taking on certain combinations of roles. There is only one role, Software Quality Assurance group (SQA), that should not involve any person from the management. Neither should it include persons involved in the software development. The reasons are obvious: quality assurance without impartiality can not fulfil its intended purpose.

The reduction of the number of roles can also be based on the need. We have studied the roles in the context of small organisations and classified them to three types. First, a role may be important enough to be kept as a formal role and assigned to a person with associated responsibilities, regardless of the size of the organisation. Every organisation must for example have a Senior Management (SM), with overall responsibility for all projects, including the long term strategy for SPI. Second, a role may be important in terms of activities and responsibilities associated with it, but without need for a formal role. In such a case, the activities and responsibilities can be included in another existing role. An example of such a role is Documentation Support Group (DSG). Since documentation still is an important activity, we assume that DSG's activities and responsibilities will be taken over by software engineers (SE). Third, some roles may be irrelevant. An organisation without subcontractors for example does not need Software Subcontract Management (SSM).

Table 1 below gives a summary of all roles as defined in the SW-CMM (Paulk et al, 1994) and their relationships to the roles in our submodels.

Table 1: Roles in SW-CMM compared to Dynamic CMM for Small Organisations.

Role in SW-CMM	**short name**	**XXS**	**XS**	**S**
Contract management group	n/a	SM	MS	MS
Customer SQA representative	CSQA	CSQA	CSQA	CSQA
Document Support Group	DSG	SE	SE	DSG
First line software managers	n/a	n/a	n/a	n/a
Group responsible for analysing and allocating system requirements	n/a	SG	SG	SG
Hardware engineering group	n/a	SG	SG	SG
Individuals who have experience and expertise in the application domain and in software engineering	n/a	other	other	other
Manager	n/a	SM	SM	SM

Role in SW-CMM	short name	XXS	XS	S
Managers in SQA reporting chain	n/a	SM	SM	SM
Marketing and Sales	MS	SM	MS	MS
Other individuals	n/a	other	other	other
Other affected groups	other	other	other	other
Prime contractor's management	n/a	n/a	n/a	SSM
Project Manager	PM	PM	PM	PM
Project software manager	n/a	PM	PM	PM
Senior Manager	SM	SM	SM	SM
Software Configuration Control Board	SCCB	SCM	SCM	SCCB
Software Configuration Management group	SCM	SCM	SCM	SCM
Software configuration management manager	n/a	n/a	n/a	n/a
Software Engineering group	SE	SE	SE	SE
Software estimating group	n/a	PM	PM	PM
SoftWare Manager	SWM	SWM	SWM	SWM
Software Quality Assurance group	SQA	SQA	SQA	SQA
Software quality assurance manager	n/a	n/a	n/a	n/a
Software task leaders	n/a	PM	PM	PM
Software Subcontract Manager	SSM	n/a	n/a	SSM
System engineering Group	SG	SG	SG	SG
System Test Group	STG	STG	STG	STG

3.3 XXS Organisations

An XXS organisation is assumed to have one or two employees and one product under development. This is the starting scenario for many software development organisations. If there is only one employee, this person is both the manager in the role Senior Manager (SM) and a Software Engineer (SE). If there are two employees, one is both SM and SE, the other SE only. Still, two persons have a good chance to get insight into each others work, and we see no reason to have formally assigned project management. By that, we do not claim that project management is abandoned, project planning and tracking are important activities. The implication is only that project management activities can be carried out informally by SM. The SoftWare Manager (SWM) is responsible for the software development environment, but also for the operative software and hardware in the organisation. In companies developing customer specific solutions, the SWM becomes the expert on computers and adjustments at the customer site. Still, a person possessing one of those roles may very well work in another role as well.

We make the assumption that subcontracting is not an issue in a two-person organisation, implying that the Key Process Area (KPA) Software Subcontract Management and the roles associated with it are irrelevant. The System Test Group (STG) is supposed to be responsible for system tests, while SE has the responsibility of tests during the development. Validation and verification is of primary importance to obtain a quality product. However, an internal STG role is hardly realistic in an XXS organisation. For organisations with customer specific development, the customer is the most adequate group for acceptance tests. In product development organisations in the beginning of the business, an external test service can be bought.

SQA should not be shared with persons involved in development activities or in the management. In an XXS organisation, it is therefore impossible to appoint an internal SQA, without violating the originally intended roles and responsibilities. In an organisation with specific customers, Customer SQA (CSQA) can be used to conduct SQA activities, if SQA is available at the customer site. In product development organisations, an external SQA service can be bought.

The System Engineering Group (SG) is responsible for systems with both software and hardware. The roles in an XXS organisation are presented in figure 1.

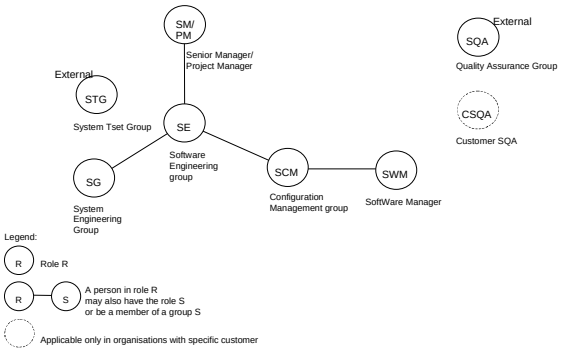

Figure 1: The Roles in an XXS Organisation.

In the model, a circle denotes a role. A dotted line around a circle means that the role is relevant only in organisation developing customer specific solutions. Roles (directly or indirectly) connected by a line can be taken by the same individual(s). However, the more links there are between two roles the less suitable is it to combine these roles. For example, the activities and responsibilities of

SE may be shared by SG. From figure 1 it is obvious that the model is applicable in an organisation with only one person. This person takes the roles SM/PM, SE, SG, SCM, and SWM, and buys external services for STG and SQA, or uses CSQA, if applicable.

3.4 XS Organisations

An XS organisation has 3-15 employees and several products/product versions or projects. If the organisation is developing products for the open market, the time of entering the XS model is probably the time of the first release, i.e. the time when the hard work with successive releases and change management begins. Moreover, the life after the first product is of strategic importance. In order to keep the first product successful and on the market, the organisation needs to grow. Most probably, new products are on the way, also implying need to recruit more staff. Transitioning from XXS to XS requires new roles. Correct project management requires a formal role of Project Manager (PM). Most probably, marketing and sales (MS) is a role with increasing importance at the time an organisation grows out from XXS. System testing (STG) should become an internal role as the number of products/product versions and/or projects grows.

Figure 2 presents the model for XS organisations. The shading of a circle indicates that the role is an additional one compared to the XXS organisation. For example, when growing out of XXS, PM and MS are roles to be added. The frames depict projects. The roles within a frame denote roles, which may be specific for each project, while the roles outside the frame have organisation-wide responsibility. For example, there is a SCM role in each project. SWM is general for the organisation, but the person with the SWM role may very well be the SCM in one or more projects. SM and MS are general for the organisation, possibly one and the same person. SM can also work as SE, at least when the organisation still is close to the lower end of the XS size scale.

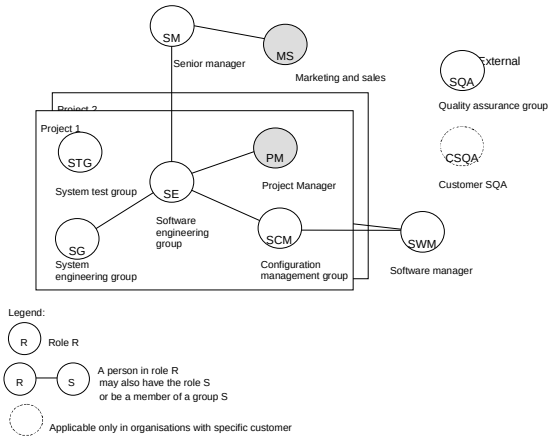

Figure 2: The Roles in an XS Organisation.

3.5 S Organisations

In an S organisation, the number of employees is 16-50 and there are several products/versions or projects running in parallel. In particular for organisations with product development, documentation support is of great importance implying the introduction of a Documentation Support Group (DSG). Software subcontracting becomes more and more relevant. This implies the introduction of the KPA Software Subcontract Management with the role Software Subcontract Manager (SSM). In some organisations, SM may not be involved in MS activities, but in some organisations, sharing SM and MS by one person is considered important. Therefore, we leave this possibility open by retaining the link between MS and SM. The links imply possibility, not a mandatory sharing of responsibilities. MS is related to DSG, and the same person can share DSG and MS roles. Further, until subcontracting is widely practised, the person in role SM may also have the role SSM. Quality assurance in organisations of this size motivates an independent internal SQA, which is also important for obtaining continuity in the software process improvement activities. SCM is a more demanding task in an organisation with many projects and products/versions, and therefore it is assumed that the person in the SCM role does not work as SE in addition to that. The model for an S organisation is presented in figure 3.

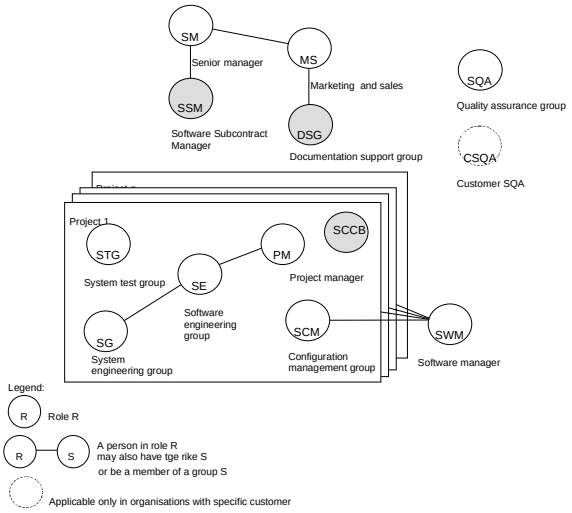

Figure 3: The Roles in an S Organisation.

3.6 Responsibilities and Activities

For each role, there are responsibilities and activities. In our model, the responsibilities and activities are described by role and by type, giving a good overview over the role requirements. Table 2 shoes an example for our representation of the responsibilities and activities for roles.

Table 2: Responsibilities and activities of SWM.

SWM	**Object(s) of concern**
Is responsible for:	- the HW/SW development environment
Reviews:	- statement of work - project plan - resource estimates - quality assurance plan - configuration management plan - the progress against the project plan - specification for subcontract - subcontract

3.7 Documents

The (SW-)CMM requires a large number of documents, required as the basis for activities or as input to or output from activities. The phrase "according to a documented procedure" appears frequently in CMM texts. The required documents can be classified as follows:

- *policy documents* - one for each KPA
- *documented procedures*, describing how to carry out certain activities, as for example requirements management, project planning, estimation
- *plans*, for example configuration management plan, project plan, quality assurance plan
- *status reports*, for example quality assurance report

In the SW-CMM, the documents are described in text only and scattered around the different KPAs. It is therefore hard to get a good overview over the number and the types of documents required. In our model, documents are classified by type along the above document types. The documents are not described here for space reasons, but can be found in (Orci & Laryd, 2000a) and (Orci et al, 2001).

3.8 Activity Diagrams

One of the intentions with our model is to provide a good overview. Graphics support both overview and structure. We therefore present activities graphically, in a simplified form. A graphical overview of the KPA Software project Planning (SPP) is given in Figure 4. The activities take allocated requirements as input and produce statement of work, estimates, project administrative data, and project plan as output. Activities are reviewed periodically by the SM and periodically as well as event-driven by the PM. SM is responsible for a written organisational policy for SPP, and a set of documented procedures for SPP. The PM is responsible for measurement data concerning the status of SPP activities. SQA reviews the activities, documents and work products of the KPA, indicated by SQA outside the KPA frame. The activities are presented more in detail in figure 5.

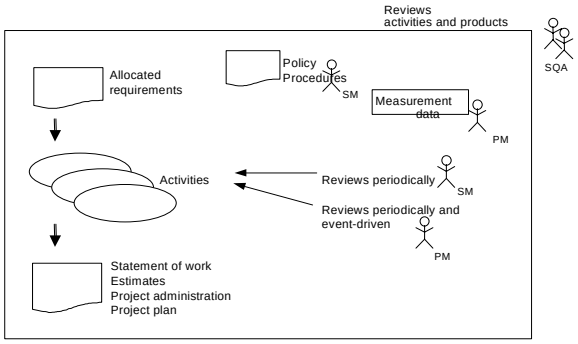

Figure 4: Software Project Planning – Overview.

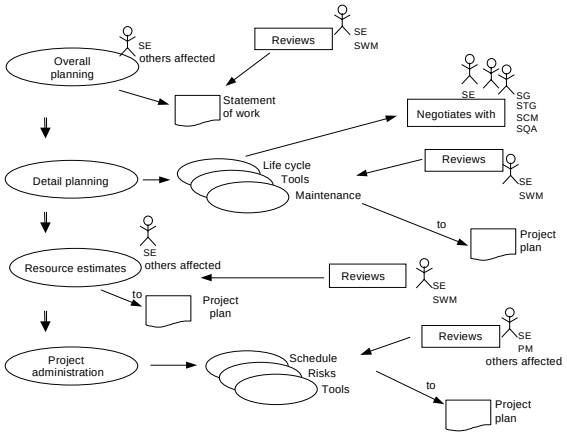

Figure 5: Software Project Planning – Activities.

4. The Validation Framework

The SW-CMM represents a "common sense engineering" approach to software process improvement (Paulk, 1998). It has been extensively discussed and reviewed within the software community, and applied over and over again. Hayes and Zubrow (1995) have performed long-term studies of (SW-)CMM application. Taking all that into account, it is fair to say that SW-CMM has been validated in practice. It represents a broad consensus of the software community, and is therefore a useful tool for guiding improvement efforts. The validation of Dynamic CMM for Small Organisations by the same means is not possible at the time of writing. This requires first of all application, but also dissemination, discussions, and feedback from the user groups. Therefore, only a theoretical validation is possible at the time. We have selected the "common sense" approach used by Paulk (1998), which evaluates the usage of the SW-CMM in small organisations. This approach uses the following aspects of SPI as a framework for validation.

- Architectural properties in terms of evaluation
- Defining Small Projects
- Process assessment
- Interpreting the CMM
- Mapping of CMM terminology and the terminology/language used in the organisation
- Invariants and context dependence
- Senior management commitment
- Measurement
- Software Engineering Process Group (SEPG)
- Priorities
- Documentation
- Process tailoring and formality
- Deployment of a buy-in
- Risk management
- Planning
- Peer reviews
- Other issues

4.1 Architectural Properties

SW-CMM architecture consists of maturity levels, Key Process Areas (KPAs), Goals and Key Practices. Evaluation is done in terms of the goals defined for each KPA, not in terms of the key practices themselves. The key practices are not intended as prescriptive, but they are known to work towards fulfilment of the corresponding goals. Different implementations/approaches to achieve the goals are possible.

4.2 Defining Small Projects

Paulk (1998) uses the project size as the dividing principle according to table 2.

Table 2: Project sizes.

Size	# people	Schedule
Small	3-5	6 months
Very small	2-3	4 months
Tiny	1-2	2 months
Individual	1	1 week
Ridiculous	1	1 hour

Each step or iteration of the improvement work should be realised in *improvement projects* with clearly defined goals, plans, deadlines, and milestones. An improvement project may focus on one KPA, for example configuration management. In an improvement project, a process for the KPA in question is defined, documented, applied and evaluated. The newly defined process, i.e. the *development process*, is applied and evaluated in a *development project*, which serves as a pilot. If the evaluation indicates some weaknesses, the development process is modified, used and evaluated again.

4.3 Process Assessment

According to Paulk (1998), the assessment of the processes in a small organisation should be

streamlined rather than follow the CMM-based appraisal. The two-week appraisal and its formality are probably excessive and not suitable for small organisations/projects. Instead, the emphasis should be on efficiently identifying important problems, also with the risk of lacking in rigor. Focusing on the institutionalization practices to establish the organizational culture is recommended by Paulk, as well as explicitly tying process improvement to business needs.

4.4 Interpreting the CMM

Environments where interpretation and tailoring of the CMM are needed include, among others, small projects and organizations. Intelligence and common sense are required to use the CMM correctly and effectively.

The interpretation should be discussed in the context of the types of components of the model. The (key) practices for example are not prescriptive. They only suggested ways to fulfil the goals of the corresponding KPAs. The goals are the prescribed parts. Alternative implementations are possible. However, the practices provide insight into how to achieve the goals and they implement repeatable, defined, measured and continually improving software processes.

4.5 Mapping of the CMM Terminology and the Language of the Organisation

In this context, the organizational structures, roles and their interrelationships and the formality of processes need to be mapped onto their equivalents in the current organization. The CMM explicitly mentions independent groups, for example the SQA group, testing groups and the SCM group. Appropriate organizational terminology for roles such as project manager and project software manager should be specified. People may fill multiple roles, and explicitly stating these roles (together with the corresponding responsibilities) makes the interpretation of the CMM easier and more consistent.

4.6 Invariants and Context Dependence

The invariants of the CMM are the key process areas and goals, with a possible exception of SSM in case subcontractors do not exist. The practices part must be objective for professional competent judgment. For example, formal methods might replace peer reviews on Level 3. Trained, experienced assessors are crucial. The context dependency regards the implementation of the practices, while the practices' contexts are invariant. For example, all organizations need documented requirements and processes, agreed-to commitments and some kind of planning. Similar reasoning can be done for roles/groups. Project management, configuration management and change control need to be done in any project irrespective its size. It is not, however, necessary to appoint formal roles for all. The intent of the practices is invariant, the implementation context dependent.

4.7 Senior Management Commitment

If the organization as a whole wants to change its performance, senior management must actively support the change. In small organizations, a respected champion might have influence enough to move the entire organization, including the president. The president, typically the founder of the small organization, is also often working in other roles.

4.8 Measurement

All data collection and analysis should fulfil an obvious purpose. People need to understand what the data means to the organisation and how it can be analysed in a meaningful way. Measurement should be initiated by collecting simple sets of useful data to get visibility for the critical aspects of a project or an organisation. Since people pay more attention to things that are measured, it is important not to draw attention away from any critical aspects. It is therefore not advisable to measure something simply because it is easy to collect certain kinds of data.

4.9 SEPG

A Software Engineering Process Group (SEPG) is not formally required on Level 2. However, there must be some role responsible for the coordination of process definition, improvement, and deployment activities. Even small organisation should have explicit responsibilities for improvement activities that are explicitly assigned and monitored.

4.10 Priorities

It is recommended to begin with "as is" processes instead of "should be" processes. The existing processes have evolved, because of the people and their need to get a job done. With the organizational focus on process management and improvement, the "as is" and "should be" processes will converge.

4.11 Documentation

There are three main reasons for documenting processes and products; to communicate, to understand and to encourage consistency. Documented processes/procedures support organizational learning and prevent them from reinventing the wheel for common problems. Documented processes/procedures put repeatable processes in place. Documentation need not be lengthy or complex, but be must be tailored to the specific purpose.

4.12 Process Tailoring and Formality

Not only the documentation, but also the processes themselves should be tailored to the specific needs of a project or organisation. Unreasonable administrative overhead can lead to significant resistance and processes might be abandoned. It is not necessary to define separate "documented procedures" for each key practice.

4.13 Deployment of a Buy-in

The tailoring and documentation of processes does only make sense when people are willing to follow these processes. Affected personnel should participate in the definition and implementation of improvement activities. SPI should not be forced onto them. Formal mentoring programs are recommended to establish and sustain new practices and procedures. Management training is particularly important; management support is a key factor to the success of SPI.

4.14 Planning and Risk Management

The intentions of the CMM as well as traditional project management are to establish stable requirements for effective planning and managing. A key to successful development and improvement activities is planning. The organization sets priorities and packages plans according to its needs and context. Incremental or evolutionary life cycles are recommended in general, since they better support active risk management. However, for very small projects or organizations, a waterfall-like life cycle may be more adequate.

4.15 Peer Reviews

Formal peer reviews are a cost-effective tool to improve document quality (O'Neill, 1997). Any form of systematic review is recommended. Inspections for example, are not only useful to detect faults. They have many benefits besides that, like on-the-job training of new employees and risk mitigation of loosing critical business knowledge when people leave the project or organisation (Doolan, 1992).

4.16 Other Issues

There are other issues like how to perform assessments or whether improvement should focus on projects or the whole organisation. Abbot (1997) raises the following issues for SPI in small organisations:

- senior management support
- adequate staffing
- applying project management principles to process improvement
- integration with ISO9001
- assistance from process improvement consultants
- focus on providing value to projects and to the business

It can be noted that all but the last of these issues are directly related to the implementation of SPI activities on quite high a level. There are few relationships to how SPI affects actual project work. This organisational focus has the apparent risk of lack of buy-in (see 4.13).

Brodman and Johnson (1995) on the other hand take a project view and identify the following challenges:

- handling requirements
- generating documentation
- managing projects
- allocating resources
- measuring progress
- conducing reviews

- providing training

In their model, the LOGOS Tailored CMM (Brodman & Johnson, 1995), they focus on the key practices. They clarify existing practices, introduce alternative practices, and align them with the needs of small organizations and projects.

4.17 Summary

Summarizing this section, the implementation of any SPI work involves the following steps:

- Entry point selection
- Defining the improvement infrastructure
- Training
- Improvement project
  - Setting priorities
  - Definition and documentation
  - Pilot
  - Evaluation.

Each improvement project realises a plan-do-evaluate cycle. *Plan* corresponds to setting the priorities in addition to other normal project planning activities. *Do* corresponds to definition and documentation, and the application of a defined development process in a pilot project. *Evaluate* is the evaluation phase.

5. Discussion

5.1 Entry Point to the Dynamic CMM

The basic parameters for selecting an appropriate model are the number of employees and the number of concurrent projects or products/product versions. The number of employees includes development, management, and marketing and sales staff, but excludes administrative and human resources staff. The number of products/product versions is applicable for market-driven development, while the number of projects is applicable for custom development (for specific customers). Table 3 summarizes appropriate entry points to our model.

Table 3: Appropriate entry points to Dynamic CMM.

	1-2 empl	**3-15 empl**	**>15 empl**
1 product	XXS	XS	S
2-5 products	-	XS	S
>5 products	-	S	S

The alternatives marked with "-" are uncommon, if not unrealistic. For example, an organisation with 2-5 products/product versions, with 1-2 employees might have a chance to stay on the market with a large number of subcontractors. However, it might be more a rule than an exception that such a company will grow in the number of employees. If the organisation's characteristics are outside the limits of the table, it is assumed that the original CMM can be used.

5.2 Defining the Improvement Infrastructure

This stage concerns the definition of an infrastructure for the *improvement* work. This infrastructure is composed of a *steering group*, *the SEPG*, *working groups*, and *process users* (Caputo, 1998). The *steering group* decides which processes to improve, defines working groups, and appoints the SEPG. In a very small organisation, the steering group may consist of the senior manager only. The *working groups* do the main work in defining, institutionalizing and evaluating new processes. Their work includes the coaching of *process users*, who finally are supposed to use the new processes. However, in very small organisations all employees are probably involved in one or more working groups. The SEPG collaborates with the working groups in process improvement. It coaches working groups and process users and provides support and motivation for using the process(es). A *SEPG* is not formally required until maturity level 3 in the (SW-) CMM. However, there should be someone with an interest in SPI and knowledge of both SPI and software development. In a very small organisation, the SEPG could be external.

Please note that SM, PM, etc. are roles associated with *development* and not the actual improvement work. Development roles are assigned, according to our XXS, XS, and S models, independently of the actual improvement work.

5.3 Training

The training should concern the underlying ideas and motivations for SPI work in general and the Dynamic CMM in particular. The role models, lists of responsibilities, activities and documents are intended to serve the purpose. The activity diagrams correspond to the development processes and give a comprehensive picture of the KPAs.

5.4 Setting Priorities

The original CMM does not define in which order the KPAs should be introduced. However, there are certain dependencies. At least one KPA needs to be in place orderly to make SQA work meaningful. Project tracking is important, since it has direct consequences for budget and schedule. However, in XXS organisations, project tracking is usually handled informally. The order of the formally handled KPAs, for example Requirement Management and Software Configuration Management, is not generally determined. Which one is introduced first is a question of the situation at hand. For example, if configuration management is experienced as a big problem, it would be natural to start to bring some order in that. At the other hand, if the requirements are volatile, and problems are encountered by introduction of new requirements, requirements management should probably get the highest priority.

To provide a sensible basis for a decision on priorities it is recommended to perform an assessment. CMM-based appraisals are quite formal and time consuming (see section 4.3). To efficiently identify important problems our model provides materials to perform "light-weight" appraisals.

5.5 Definition and Documentation

This activity involves definition and documentation of a *development process*, for example policies and procedures for configuration management. Caputo (1998) proposes workshop as a suitable technique to reach agreement. Development processes should conform to the activity diagrams for the respective KPAs defined in our model. More details can be found in the checklists provided for each KPA (Orci et al, 2001). The checklists define Entry conditions, Roles, Activities, and Exit conditions and have been inspired by the process scripts used in the Personal Software Process[SM] (Humphrey, 1995). An Entry conditions checklist defines entry requirements for the processes related to a KPA, for example the existence of a written organisational policy for the KPA. An Activities checklist lists required and recommended activities/practices for the KPA. The Roles checklist lists all roles that are referred to in the other checklists related to the same KPA. These roles need to be assigned before application of the processes. The Exit conditions checklist is used to verify whether activities have been performed and documented properly. It should be noted that there is no explicit requirements in the SW-CMM to perform exactly the activities presented in the model. Rather, the activities are examples of activities towards fulfilling the goals of the KPA. The same applies to the Dynamic CMM.

An example of a checklist is presented in table 4. The Ref. column contains references to other parts in the model, where details on a rationale for the activities/practices can be found.

Table 4: Software Process Planning (SPP) Checklist - Entry Conditions.

Entry conditions		Ref.
A written organisational policy for SPP exists	❑	C2
A documented procedure for SPP exists	❑	AC6
There is a PM responsible for SPP and SPTO (Software Project Tracking and Oversight)	❑	C1
Adequate resources and funding are available for the planning of the software project	❑	AB3
Tools to support the activities are made available	❑	AB3
PM, SWM, SE, and other individuals are trained in software planning and estimating	❑	AB4
Allocated requirements are documented	❑	AC6.1
Cost/effort data is available	❑	AC10.2
Historical productivity data is available	❑	AC9
Statement of work is available	❑	AB1

A defined and documented development process must be approved by the working group(s) and SEPG before it can be applied in a pilot project. If an approved status cannot be reached directly, another workshop should be arranged to resolve the problems.

5.6 Pilot

A pilot development project is an application of the improved development processes in a realistic context. The pilot project is an adoption test. It will likely generate a list of issues that need to be resolved before the improved processes can be institutionalized.

5.7 Evaluation

The new development process is evaluated after the pilot development project. In addition to the opinions of the workers, the Exit condition checklists are used to validate whether the exit conditions are fulfilled. If the development process cannot reach the approved status, a workshop

should be arranged to resolve the problems and to modify the development process accordingly. The intention is that the development process is used in every development project after its definition.

6. Experiences

So far, the model has been used by us to perform SPI work in four organisations. In all cases, we proceeded as follows; after a general introduction to SPI and maturity models (with a focus on the CMM), we performed an assessment using the materials developed for the Dynamic CMM (Orci & Laryd, 2000a; Orci et al, 2001). The results of these assessments have been documented and presented and certain SPI activities were proposed based on the assessment results. The organisations did then get help to define a package of measures based on the results. It was then up to the organisations to decide whether or not to continue and which measures to implement.

In only one case (an XS organisation, just beyond S size) a second assessment was performed to measure the results of the improvements work. As can be seen from figure 6, the total performance of the organisation has improved in all KPAs. However, the KPAs SQA (Software Quality Assurance) and SM (Subcontract Management) were excluded from the first assessment, since these activities were not relevant (SM) or not practiced (SQA), respectively.

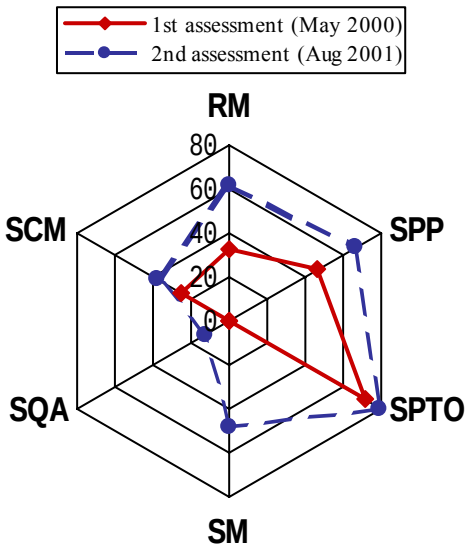

Figure 6: Assessment results from a XS organisation.

Although it seems that the organisation has improved in all areas, some activities/practices actually deteriorated between the first and second assessment. Details about the assessment results and proposed improvement measures can be found in (Mattsson, 2001; Ågren, 2001).

- Competence in the RM KPA has improved in general, but not so for the responsible person.
- General agreement with commitments is lower in the second assessment (SPP and SPTO).
- Affected groups/individuals are less informed about status and contents in "baselines" (SCM).
- Metrics are collected to a lesser degree than before (SPP).

Some of these problems can however be attributed to the introduction of a problem reporting tool that was not generally accepted by all affected groups/individuals.

7. References

Abbot, J.J. (1997). Software Process Improvement in a Small Commercial Software Company. *Proceedings SEPG'97*.

Brodman, J.G., Johnson, D.L. (1995). *The LOGOS Tailored CMM for Small Businesses, Small Organizations, and Small Projects*. LOGOS International, Inc.

Byrnes, P., Phillips, M. (1996). Software Capability Evaluation, Version 3.0, Method Description. Technical Report CMU/SEI-96-TR-002, ESC-TR-96-002. Software Engineering Institute, Carnegie Mellon University, USA.

Caputo, K. (1998). *Implementation Guide—Choreographing Software Process Improvement*. Addison Wesley.

Curtis, B., Hefley, W.E., Miller, S. (1995). *People Capability Maturity Model.* Software Engineering Institute, Carnegie Mellon University.

Doolan, E. (1992). Experience with Fagan's Inspection Method. *Software—Practice and Experience* **22** (2), 173-182.

Dunaway, D.K., et al. (1999). Why Do Organizations Have Assessments? Do They Pay Off? Technical Report, CMU/SEI-99-TR-012, ESC-TR-99-012, Software Engineering Institute, Carnegie Mellon University, USA.

Hayes, W., Zubrow, D. (1995). Moving On Up: Data and Experience Doing CMM-Based Process Improvement. Technical Report CMU/SEI-95-TR-008, ESC-TR-95-008. Software Engineering Institute, Carnegie Mellon University, USA.

Herbsleb, K., Zubrow, D., Goldenson, D., Hayes, W., Paulk, M. (1997). Software Quality and the Capability Maturity Model, *Communications of the ACM* **40** (6), 30-40.

Humphrey, W.S. (1995). *A Discipline for Software Engineering*. Reading, MA: Addison Wesley.

ISO/IEC (1998). *ISO/IEC TR 15504 Parts 1-9:1998 Information Technology—Software Process Assessment*. Geneva: International Organization for Standardization.

Kahoe, R., Jarvis, A. (1995). *ISO 9000-3—A Tool for Software Product and Process Improvement*. New York: Springer.

Kautz, K. (1998). Software Process Improvement In Very Small Enterprises: Does It Pay Off? *Software Process - Improvement and Practice* **4**, 209-226.

Laryd, A., Orci, T. (2000). Dynamic CMM for Small Organizations. *Proceedings ASSE2000*.

Mattsson, J. (2001). Utvärdering och evaluering av QMSE. In Swedish. Internal Report UMNAD 331/01. Umeå University, Sweden.

O'Neill, D. (1997). Issues in Software Inspection. *IEEE Software* **14** (1), 18-19.

Orci, T., Laryd, A. (2000a). CMM for Small Organizations, Level 2. Technical Report UMINF-00.20. Umeå University, Sweden.

Orci, T, Laryd, A. (2000b). CMM for Small Organizations—Implementation Aspects. *Proceedings EuroSPI'00*.

Orci, T., Laryd, A., Börstler. (2001). CMM for Small Organisations, Level 2—The Supplements. Technical Report UMINF-01.19. Umeå University, Sweden.

Paulk, M.C. (1998). Using the Software CMM in Small Organizations. *Joint Proceedings of the Pacific Northwest Software Quality Conference and the Eighth International Conference on Software Quality*. 350-361.

Paulk, M.C., Weber, C.V., Curtis, B., Chrissis, M.B. (Eds.) (1994). *The Capability Maturity Model—Guidelines for Improving the Software Process.* Reading, MA: Addison-Wesley.

Stelzer, D., Mellis, W. (1998). Success Factors of Organizational Change in Software Process Improvement. *Software Process - Improvement and Practice* **4**. 227-250.

Zahran, S. (1997). *Software Process Improvement. Practical Guidelines for Business Success*. Addison-Wesley.

Ågren, O. (2001). Quality Management for Small Enterprises: Second Assessment of "Företaget AB". In Swedish. Unpublished Internal Report. Umeå University, Sweden.