Ministry of Communications and Information Technology

National Information Center

General Department of Information Technology

Information Systems Department

Section Policies and Standards

Software Process Improvement Standards& Specification

Commission on software standards

December 2009

First version NIC/SD/SPISS V1.0 :2009

Contents

A. Introduction 3

B. Scope 5

C. Document Setup & Purpose 6

Guideline 1: Identify business objectives 6

Guideline 2: PRISMS Assessment Method 8

Guideline 2-1: PRISMS Assessment Approach 9

Guideline 2-2: PRISMS Assessment Flexibility 10

Guideline 2-3: Answering Criteria 10

Guideline 2-4: Summary 11

Guideline 2-5: Scoring Criteria 12

Guideline 2-6: Prioritizing KPAs method 13

Guideline 3: Describing Best Process Improvement Practices: A guide-based approach for Knowledge sharing 16

Guideline 3-1: Capturing Process Improvement Best Practices Guide 17

Guideline 3-2: Prioritized areas for improvement 18

Guideline 3-3: Identify best practices: 18

Guideline 3-4: Modify Best Practices 20

Guideline 3-5: Knowledge sharing best practice store 20

Guideline 4: Mapping Process Improvement Best Practices to areas for improvement 21

Guideline 4-1: Selecting and defining Measures for software process 23

Guideline 4-2: PRISMS approach for software measurement 23

Guideline 4-3: Identifying Process Issues 28

Guideline 4-4: Guidelines for Identifying Process Issues 28

Guideline 4-5: Identify Process Measure 30

Guideline 4-6: Collecting and retain data 34

Guideline 4-7: Analyze software process behavior 34

Bibliography 36

Appendix 48

A. Introduction

The dynamic competitive forces driving almost all the software industrial sectors are forcing an increasing number of Organizations to realize continuous organizational changes. Businesses are starting to look towards improving their software processes and product quality in order to compete in the software market. Product quality is normally influenced by quality of the process that is used to produce the product (Sommerville, 2001). The ability to produce reliable and usable software within time and budget is still difficult to achieve for many organisations. A British Computer Society Review 2001 paper indicated that only 13% of 1027 surveyed IT projects were ‘successful’ (Taylor, 2001). Another report indicates that although there has been some recent improvement, 23% of their surveyed projects were considered total failures and only 28% totally successful (on time, within budget and all functionality) (Johnson *et al*., 2001) and now many of these, search for solutions from software process improvement (SPI).

Several SPI models have been invented during the last decade. In recent years these SPI models have gained a lot of momentum. The most influential models for SPI are the Capability Maturity Model (CMM) (Zahran, 1998), Capability Maturity Model Integration (CMMI) (Ahern, Clouse and Turner, 2003) both of these were developed by the Software Engineering Institute (SEI), funded by US department of Defense. The Software Process Improvement Capability Determination (SPICE) model was an international attempt for standard process improvement framework (Zahran, 1998). All of these SPI models provide frameworks and road-maps for SPI programs. Such models typically define the levels of process maturity from 1- 5 through the identification and the assessment of various key attributes.

There are potentially large gains to be made within the industry by wider application of SPI, but as yet the use of models such as CMM and CMMI within smaller organizations has been limited. Most of the existing SPI models have been accused of being cumbersome when implemented in small software enterprises (Kautz, Hansen and Thaysen, 2000, Lyard and

Orci, 2000, Dyba, 2000, Johnson and Brodman, 1997). This is mainly due to the massive resources that are needed, prior to, or within the application of the SPI program. Further to that CMM, CMMI and their like have been accused of being bureaucratic models that force their clients to go through their maturity ladder without ensuring that they will achieve a quality product or meet their business objectives (Tim, 2004). For example, why is it necessary that the SPI program should start with key process areas in level 2, and not level 5, or key process areas in level 3 without the completion or the satisfaction of all the key process areas in level 2? However, CMMI and SPICE appear to have considered this problem by inventing other models such as continuous representation (Ahern, Clouse and Turner, 2003) and the process capability dimension model (Emam, Drouin and Melo, 1997) in order to let the SPI applier have flexibility of choice by being able to select a particular category or particular process area that needs to be improved. But these models do not specify which category or process area to start with even if it is known what process areas need to be improved i.e. how to prioritize the areas for improvement. This problem may be common not just for SMEs but for all businesses sizes as Glib (Glib, 2003) describes CMM, CMMI SPICE and their like as not a 'Good medicine' for even very large systems engineering projects and that they are, also, overly complex for most IT projects.

In general there is a consensus that the current standard SPI framework such as CMM cannot be applied unmodified to small organizations (Johnson and Brodman, 1997). In addition, a survey by SEI suggests that over two-thirds of the people responsible for SPI know what needs to be improved, but need more guidance on how to do it (Paulk, 1998).

Paulk (1998) also concludes that "the issues associated with interpreting the software CMM for the small organization are different in degree but not in kind." Paulk also emphasizes the need to tie the SPI programme to the organization's business goals. However, there is no explicit mechanism for this in CMM.

Achieving business objectives is one the important recipe for Information Technology (IT) business success (Johnson *et al.*, 2001). The SPI standard models such as CMM, CMMI, SPICE and their like tend not to focus their software process on the organization business objectives at the early stages of their SPI program, few of them such as CMMI tend to focus the process to the business objective at very late stages of its SPI program or road-map (Chrissis, Konrad and Shrum, 2003, Ahern, Clouse and Turner, 2003, Reifer, 2003). Whilst it is difficult for SMEs to reach the higher levels in the standard SPI models' ladder (Reifer, 2003, Ward, Fayad and Laitinen, 2001, Lyard and Orci, 2000, Paulk, 1998), it will also be difficult for them to let their SPI focus or contribute to their business objectives. This may mean that they may not achieve their business objectives when they implement the SPI program.

B. Scope

Encouraging though these results from the previous section are, many questions remain to be answered. Typically, SMEs operate within tight financial constraints. They require low-risk strategies which relatively quickly show results for any investment of resources. Key areas that still need to be addressed are:

1. Which generic model provides the most reliable way to achieve these results?
2. How can SPI be tailored to the organization's business goals?

The PRISMS (PRocess Improvement for Small to Medium Software enterprises) was set up in order to address these outstanding area, as can be seen below.

The research project addresses the following objectives:

- ♦ To define a generic process model for software process improvement that is suitable for SMEs. The model enables individual SMEs to tailor the software process improvement to the organization's business objectives. Hence the business objective drives the improvement wheel.

C. Document Setup & Purpose

This documents setup in shape of guidelines meant to help software Organizations within Sudan wishing to adopt standard SPI programme bear in mind challenges of software industries within the country such as resources or size of software Organizations.

The guidelines below can be seen as initial attempt from the author to standardize the SPI programme within the country, however regular updates to this documents is a crucial mandatory to cope with the new standards or software engineering practices that may come up during or after writing this document.

Guideline 1: Identify business objectives

CMM-SW provides a guideline for good management and engineering practices, with a strong emphasis on management, communication, and co-ordination for development and maintenance of software processes. However CMM-SW not considers business objectives as ultimate goal needed to be; SMEs should look at achieving a successful combination between software processes and business objectives achieved (Kautz, Hansen and Thaysen 2002) (Chrissis, Konrad and Shrum, 2003).

As PRISMS approach main objective is tailor SPI program to the business objectives, therefore, Identifying business objectives is crucial activity in PRISMS approach. Business objectives is the derive wheel for PRISMS approach. Most of PRISMS later activities which are come in later sections depend on the Identification of the SME business objectives. Figure shown some links between previous and later activities.

Identifying the business objectives activity is responsible for gathering individual SME business objectives. The objectives may vary from one SME to another. In order to do this, we started by conducting a survey with our SME partners and also, in conjunction with this, of the literature to define what the business objectives are. As a result of this we found that the business goals criteria were based on functionality, cost, time to market, product quality and customer satisfaction.

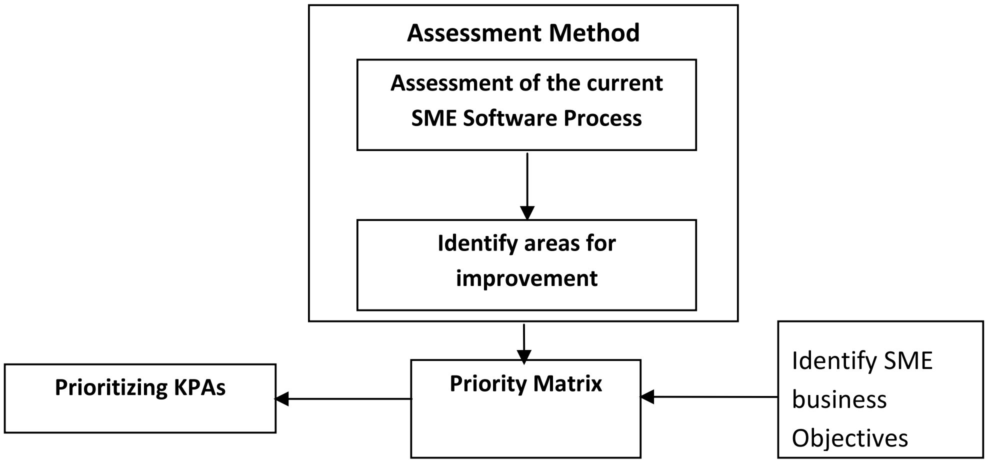


Figure 3-2 SMEs Area's for Improvement Prioritization Framework

As can be seen in Figure 3-2 the main purpose of doing this is to assess how to prioritize the areas for improvement, which have been identified by PRISMS assessment method, to the SMEs business objectives. Therefore we came with following common business objectives explained by the questioning the impact of the areas for improvement that would associate with in criteria versus the areas for improvement is as follows:

- **Impact on development cost –** how important the areas for improvement are to make an impact on reducing the cost of development but still maintaining the Quality.
- **Impact on Quality Software-** how important the areas for improvement are to make an impact on your product and process Quality.
- **Customer Impact –** how important the areas for improvement are to satisfying or attracting more customers.
- **Reduce time to market impact-** how importance the areas for improvement are to delivering your product on- time or quicker.
- **Impact on Changeability –** how important the areas for improvement are to achieving changeability or maintainability in the context of software or organisational change, in the context of introducing a new policy into your Company.

- **Increase functions impact-** how important the areas for improvement are to increase functions.
- **Impact on Motivation –** how important the areas for improvement are to provide motivation to your employees.
- **Business impact –** how important the areas for improvement are to make an impact on your business process.

This business objective activity allows each individual SME to select or modify the objectives that their company may wish to achieve.

As mentioned earlier it is vital to identify business objectives to support prioritizing the areas for improvement as will be shown in the next section.

Guideline 2: PRISMS Assessment Method

The CMM-SW is standard for software process assessment. The main objective from the CMM-SW assessment is to assess the capability of an organization and identify the key process areas as opportunities for improvement.

The main objective of this assessment method is to identify areas for improvement and our approach to achieve this is to use CMM-SW assessment questionnaire i.e. the software process maturity questionnaire as an assessment tool in order to identify areas for improvement or key practices in each individual KPA. In CMM-SW, Key Process Areas (KPA) identify the issues that must be addressed to achieve a maturity level in CMM-SW. Each KPA identifies a cluster of related activities or practices that, when performed collectively; achieve a set of goals considered important for enhancing process capability. These related activities are called the Key Practices. The key practices for the collection and analysis of process data evolve across the maturity levels.

Also, identifying other opportunities or areas for improvement that differ from CMM-SW are also considered; by giving the chance for the SME to identify their own area for improvement that not considered as a key practice in the CMM-SW. The SW-CMM assessment in this project is tailored to suite SMEs needs or objectives such as allowing the elimination of a particular KPA(s) or practices within the KPA that the SMEs think is/are not necessary or important for their businesses, thus the assessment method is not rigid, involving filling unnecessary questionnaires or questions within questionnaires. This may, from the SME perspective, be due to these KPA(s) being irrelevant or the activity within a KPA may be considered to be not necessary.

Guideline 2-1: PRISMS Assessment Approach

Due to the limited resources of SMEs the assessment approach in this project is proposed to suite the SMEs environment which is self-assessment approach. Self-assessment is the most common way of performing a software process assessment (Dutta et al. 1996). The popularity for self-assessment lies in its low cost, good accessibility and ownership of the results.

Also we considered the use of the automated assessment in this project as the next Report will explain how this assessment method could be automated. The use of the automated assessment is promising method. Using software measurements as supporting evidence for the assessment of software capability opens new possibilities for assessment. Firstly, the interruptions for the personnel at their work can be reduced as more information is acquired online. Secondly, assessment can be carried out more frequently as measurement data are gathered frequently continuously. Thirdly, providing an audit trail for assessment is easier as more judgements are based on measurement data. Finally, assessment cost is reduced; as fewer people are needed to perform an assessment and assessments themselves become an integral part of the job.

Guideline 2-2: PRISMS Assessment Flexibility

Considering barriers such as company size and resources mean that such flexibility is important. Some of the KPAs or activities could be useful for the SME but may not be affordable due to the small size of the company, lack of resources. It may also be that some of the KPAs or activities are not used or not necessary for the SME, for example sub-contractor management or integrated software management key process areas.

PRISMS assessment method allow the SME is to eliminate the Key process area or the key practice that they think is not applicable or important to their business without considered in the scoring criteria.

Guideline 2-3: Answering Criteria

In our questionnaire the answering criteria in the assessment are tailored from the CMM-SW questionnaire criteria which are: Yes-No- Not Applicable (N/A)-Don’t Know as standard criteria in CMM-SW assessment to “Yes- Partially- No-Does Not Apply (N/A)”. The “don’t know” answer has been replaced by the ‘partially’ answer. This is considered important because ‘don’t know’ means very little, whereas ‘partially’ permits the assumption that part of the process or work may have been performed or if performed then not fully addressed. ‘N/A’ only selected by the SME when they think the key practices are not possible to be implemented may be due to their lack of resources or not important to their business. From observations with our SMEs partner it was found that due to the small teams used within SMEs, there is good communication among the team members and everyone within the team knows what each member is doing. Due to this factor the answering criteria has been tailored to Yes-Partially-No-N/A.

The areas improvement can be identified if the answer is ‘No’, ’Partially’ or “Does Not Apply (N/A)” and stored in a data store to be used later in the other activities. Evidence should be shown when the answer selected is ‘YES’-‘PARTIALLY’ to verify or justify the selection. Therefore the use of the **Comments** spaces for any elaborations or qualifications about the answers to the questions is vital. Also all the questions must be answered per each individual KPA.

Guideline 2-4: Summary

As can be seen in the appendix “A” to the right of each question there are boxes for the four possible responses: **Yes**, **Partially**, **No** and **Does Not Apply**.

The SME check **Yes** when:

- The practice is well established and consistently performed.
- The practice should be performed nearly always in order to be considered well-established and consistently performed as a standard operating procedure.

Check **No** when:

• The practice is not well established or is inconsistently performed.

- The practice may be performed sometimes, or even frequently, but it is omitted under difficult circumstances.

Check **Does Not Apply** when:

• You have the required knowledge about the project or organization and the question asked, but you feel the question does not apply to the project and not wanted to be counted within the score algorithm.

- For example, the entire section on “Software Subcontract Management” may not apply to the project if you don't work with any subcontractors.

The SME should use the **Comments** spaces for any elaborations or qualifications about their answers to the questions.

The SME should check of the boxes for each question and answer all of the questions.

Guideline 2-5: Scoring Criteria

The areas improvement can be identified if the answer is 'No', 'Partially' or "Does Not Apply (N/A)" and stored in a data store to be used later in the other activities. Evidence should be shown when the answer selected is 'YES'-'PARTIALLY' to verify or justify the selection.

The percentage of each answer criteria is calculated as follows:

Answering Criteria Percentage (KPA satisfaction level) =

$$\frac{\sum \text{(Number of answering per criteria) x 100}}{\text{(The Total number of the questions –the total number of N/A answers)}}$$

Table 3.1 explains the general idea of analysing the questionnaire:

Answering Criteria	Count	Questions No	Percentage of Each Criteria	KPA rating %
Yes	4		(4*100/9)	
Partially	2		(2*100/9)	(Yes)%+(1/2)x(Partially)%
No	1	**10**	(1*100/9)	
N/A	2		(2*100)/9	

Table 3.1

The practices that their selected answer is: "No" or "Partially" will be treated as areas for improvement.

During an assessment, an SME can cover only the subset of KPAs that are relevant to its business needs. In most cases, it is not necessary to assess all of the KPAs in the capability level. The capability of each level is determined the KPA within in the level. For example to determine whether the SME software process achieved level 2 or not, it necessary to determine the rating or the satisfaction level achieved by the KPAs within level 2.

As can be seen in table 3.2 each KPA is measure by ordinal rating ‘F’ *(Fully)*, ‘L’*(Largely)*, ‘P’ *(Partially)*, or ‘N’ (*Not Achieved*) that represents the extend of the achievement of KPA. A software process is defined to be at capability level k if all the KPAs are satisfy the rate “F” and the level k key practices are rated as “F” or “L”.

Acronym	**Achievement/satisfaction of the defined KPA**
N (Not achieved)	0% to 15%: there is a little or no evidence of achievement of the defined key practices in the assessed KPA.
P (Partially achieved)	16% to 50%: there is evidence of a sound systematic approach to and achievement of the defined key practices in the assessed KPA. Some aspect of achievement may be unpredictable.
L (Largely achieved)	51% to 85%: there is evidence of a sound systematic approach to and significant achievement of the defined Key practices in the assessed KPA. Performance of the key practice may vary in some areas or work units.
F (Fully achieved)	86% to 100%: There is evidence of a complete and systematic approach to and full achievement of the defined key practices in the assessed KPA. No significant weaknesses exist across the defined organization unit.

Table 3.2

Guideline 2-6: Prioritizing KPAs method

It is important for SMEs to focus their SPI process, by contributing to their business objectives, from the early stages in the SPI program. Hence the SPI program will have definite objectives to achieve besides reducing the risks in wasting resources.

In this project, a priority matrix proposed to aid to prioritize areas for improvement. A Priority Matrix will be a useful tool for this research. A Priority Matrix is a simple two-axis matrix in which, in this research scenario an area for improvement will be rated or weighted and prioritized according to its importance to the business objectives.

The purpose of the priority matrix is to insure areas for improvement that are identified as opportunities for improvement from the assessment are aligned with SME's business objectives, and that the highest priority areas for improvement take precedence over lower priority areas for improvement.

Areas for improvement/ business objectives	Reduced development costs	Better quality software	Increase functions	**Weighted Total**
The project is not following a written organizational policy for managing the system requirements allocated to software.	2	4	1	7
The activities for managing allocated requirements on the project are not subjected to SQA review.	5	5	1	11
Are the project's actual results (e.g., Schedule, size, and cost) compared with estimates in the software plans?	5	5	1	11

Table : Priority Matrix

Similar priority matrix approaches have been used elsewhere, for example a priority matrix was described by the officials from GTE Directories in their presentation in February 1995 describing how they won the Baldrige Award (Carlson, 1995). Also a matrix called a "strategic improvement matrix," was used by the officials from Armstrong Building Products Operations in their presentation to the February 1996 Baldrige Award conference (Wellendorf, 1996). A similar matrix was also used in several GWU student group projects in the late 1990s, and a similar matrix was used by visiting scholars at GWU in December 2000 to identify how the US Department of State's Junior Faculty Development Program might be improved (Naoumova and Umpleby, 2001). Another one was used by members of the GWU Department of Management Science in May 2001 (Umpleby and Melnychenko, 2001).

Priority Matrix has proved to be a useful method for achieving data-driven decision-making (Crow, 2003). The priority matrix lists the KPAs down the left hand side of the matrix. The header of each column contains an SME business objective. The rating criteria of the priority matrix starts with smallest score, which indicates this particular process area is relatively unnecessary or unimportant whilst the highest score indicates this process area is highly important , desired and necessary for the business(Crow, 2003). Therefore the rating criteria for this research priority matrix were developed in order to determine the level of the importance by developing the following score criteria:

1. Score 1 indicates that process is difficult to implement and to justify and has a very little impact on the business.
2. Score 2 indicates that process is may be difficult to implement but it has a good business reason.
3. Score 3 perhaps easy to implement and has the direct impact on the business.
4. Score 4 easy to implement and has immediate benefits to the business.
5. Score 5 the process is very easy to change or it has a big impact on the business.

A priority matrix should prioritize the list with only one KPA at each priority level. This is the key to this tool (Umpleby and Karapetyan, 2003). Each item must have its own position in the priority structure. There cannot be two number 1 priorities (Umpleby and Karapetyan, 2003). In this research priority matrix the same techniques were applied as those mentioned by Umpleby and Karapetyan (2003).

Like other priority matrix, in this research priority matrix a separate "Weighted Total" column acts to calculate each KPA against the business objectives. The highest score will indicate that the SME should focus the attention to that KPA as the one with the highest priority.

Senior staff within an SME should prioritize those areas for improvement identified by the use of the assessment method against their business objectives. Once all the areas for improvement have been prioritized, they may be circulated among the stakeholders of the SME. Any disagreements about the order of priority may be addressed and the final list approved by everyone.

The goal is to gain general agreement as to which areas for improvement should take priority over which other areas for improvement. The finalized Priority Matrix then becomes a decision-enabling tool helping decide which areas for improvement to focus on as the process unfolds.

Guideline 3: Describing Best Process Improvement Practices: A guide-based approach for Knowledge sharing

As PRISMS model contain process improvement plan activity where, it comes after prioritizing the areas for improvement, it is vital in this activity and prior to process improvement implementation is to identify which best practices that could aid process improvement implementation to achieve their objectives. Therefore we suggest to is re-use knowledge (i.e. best practices) that has been proven successful in solving problems (i.e. areas for improvement) as an objective for process improvement plan activity which is well worth the effort to investigate in detail. The idea of "guide" as method for capturing such knowledge is gaining popularity within the software engineering community.

In this research, we are expanding the scope of the "guide" idea to be used as a way of recording organizational process improvement knowledge in the form of "best process improvement practices", in accordance with objectives of the SME.

The guide has mainly been concerned with capturing and re-using knowledge. The first concerns is how the process improvement knowledge will be captured, the second concern is how process improvement knowledge stored and the third concern is how this knowledge mapped to the areas for improvement. This guide is mainly concerned for solving particular problems those have been discovered via PRISMS assessment method and prioritized via PRISMS priority matrix. This report presents other issue regarding the reusability and modification of the knowledge as presented here as “best practices” that proven have best effect for solving problems. More over this guide tends to generalize its activity (as seen here below) in case of change management for reusing them in similar situations in the SMEs.

Guideline 3-1: Capturing Process Improvement Best Practices Guide

In this report, we define guide to accumulated experiences in order to capture best process improvement practices, which is useful for solving problems under similar situations and similar circumstances. Figure 1 explains how SMEs can capture and map Process Improvement Best Practices.

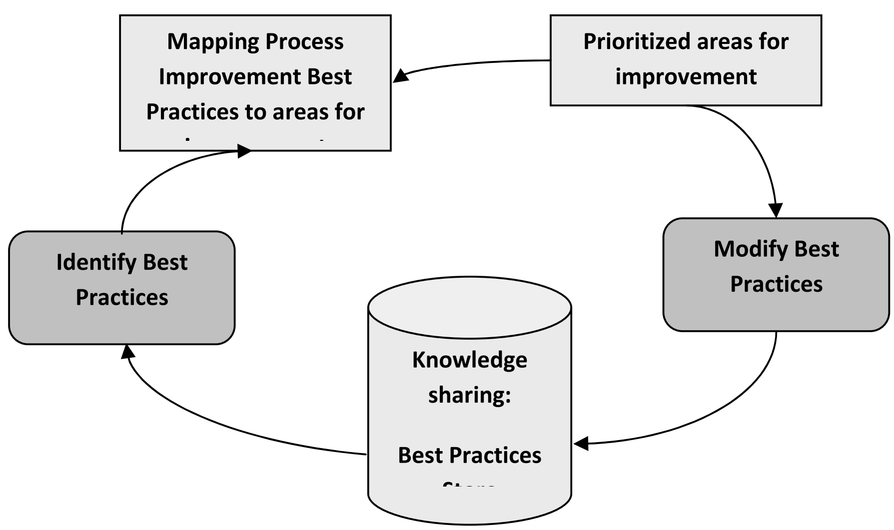


Figure 3-1Capturing and mapping Process Improvement Best Practices

The activities in figure above are represented the guide as explained here below:

Guideline 3-2: Prioritized areas for improvement

At the early stage of PRISMS model, emphasising on analysing the current process in order to identify opportunities for improvement. And then those opportunities are prioritized to act as SPI goals. Hence the plan is to put in place to precede these goals. As part of SPI plan it vital to know which practices that could achieve the SPI goals and to help SPI implementation. This guide is attempting to show how this should be tackled and generalized.

Guideline 3-3: Identify best practices:

As can be seen in the Figure 1 above the idea in this activity is to identify best practices that suitable for the areas of improvement. This task is vital and in the same time is not easy to process. The practices in the current SPI models have a little to offer in terms of engineering practices in contrast of managerial or management practices. In order to apply SPI program you need practices to aid in SPI implementation which is the thing that via experience as many of SMEs do.

Therefore we suggest for the SPI team is to identify the practices that has been successful from their previous experience in solving the particular problems parallel with other engineering practices that could found in extreme programming XP (Beck, 1999). XP practices proven could be mapped to SPI program implementation and suitable for SMEs environment (Reifer, 2003;Koch, 2003; Paulk, 2001).

XP is a light-weighted (or agile) methodology for software development (Beck, 1999). XP practices improve a software project in four essential ways or values: communication (Individuals and interactions over processes and tools), simplicity (Working software over comprehensive documentation), feedback (Customer collaboration over contract negotiation) and courage (Responding to change over following a plan) (Williams, 2003).

(Paulk, 2001) described the relation between XP and CMM that XP practices can addresses and satisfy, most of the practices in CMM level 2 and 3. A summary of Paulk description in the following table:

Level 2 KPAs	Satisfaction	Level 3 KPAs	Satisfaction
RM	✓✓	OPF	✓
SPP	✓✓	OPD	✓
SPTO	✓✓	TP	--
SQA	✓	ISM	--
SCM	✓	SPE	✓✓
SSM	--	IC	✓✓
		PR	✓✓

Table 1: the satisfaction level of CMM KPAs (level 2, 3) by XP practices (Paulk, 2001)

Where:

RM: Requirement Management

SPP: Software Planning Process

SPTO: Software Process Tracking and Oversight

SQA: Software Quality Assurance

SCM: Software Configuration Management

SSM: Software Subcontract Management

OPF: Organisation Process Focus

OPD: Organisation Process Definition

TP: Training Program

ISM: Integrated software management

SPE: Software Product Engineering

IC: Intergroup Coordination

PR: Peer Reviews

✓✓**:** Largely addressed in XP

✓**:** Partially addressed in XP

Guideline 3-4: Modify Best Practices

XP advocate many good engineering practices, however, a business cannot prepared for all situation arises, “practices” can contribute with partial with particular set of description of problems and solutions at various level of abstraction. In XP There are set of solutions as “practices” for solving arises in the implementation of SPI however is not yet sufficient as you can see from the above table. In this project we suggest that this could be improved by through iteration of sharing the knowledge among SPI team to identify best practices to solve SPI problems that could be arised. It is impossible for us to identify a set of best practices due to the time and funds available in this project. However we give a process or guide for capturing them via iterated experience that comes from experiences of various projects and supported by an automated tool.

In this activity we stressing of the vital role of SPI team is to share experience that will help in modifying best practices that have been experienced as suitable or even not suitable to be avoided in similar situation. In the case of adding new practices this should be followed by a detailed description including the procedures of those practices in order to be used for similar situation in the future.

Guideline 3-5: Knowledge sharing best practice store

Knowledge is the most decisive factor in software process improvement. Advancement strategies, thus, depend on learning the creation of new knowledge. Such knowledge creation is not simply a matter of compiling facts; it is rather a dynamic human process that is related to the practice of individuals and groups. Therefore iterated or continuous approach as can be seen in Figure 1.

Good and robust knowledge normally comes via iteration and try and error concept and sharing ideas therefore our approach in this guide is to keep and store the knowledge that came via experience for store and retrieval purposes.

Guideline 4: Mapping Process Improvement Best Practices to areas for improvement

In this section we present a suggested translation of the Best Practices that described in the identify best practices section into the areas of the process improvement that identified as specified in the Prioritized areas for improvement section We suggest where each Best Practice should be applied to the areas for improvement based from the literature and the experiences of the SPI tem. The motivation is to transfer the accumulated knowledge from one domain to another. We argue that the key process areas consists of many of the same steps as development of a software solution. Therefore the accumulated process knowledge as captured in the Best Practices should be transferable.

KPA	Areas for improvement	The planning game	Small releases	Metaphor	Simple design	Unit testing	Rfractoring	Pair programming	Collective ownership	Continuous integration	On site customer	Coding standards	commitments
1. Requirement Management	1.1 Allocate Adequate resources and non technical requirements.												
	1.2 The SQA group reviews and/or audits the activities and work products for managing the allocated requirements and reports the result.												
2.2 Software project plan	2.1 All estimates such as size, cost, and Schedule are documented for use in planning and tracking the software project.												
	2.3 Affected groups and individuals												

	agree to their commitments related to the software project.													

Table (2) An Example of Mapping Process Improvement Best Practices to areas for improvement

Guideline 4-1: Selecting and defining Measures for software process

Software Process Measurement is a vital process or activity within the software process. Measures aid in managing, controlling or stabilizing the software process in order to ensure the software processes within the organization are performing or perform as expected.

For software processes measure to be cost effective and objectively derived, they must be designed to support the business goals. Therefore, the right measures are linked to business goals and give insight into whether the goal is being accomplished or not .

PRISMS Project aim is to tailor SPI program to the business objectives, therefore in this project software process measures should be implemented on the basis of achieving the business objectives.

Guideline 4-2: PRISMS approach for software measurement

PRISMS software process measures approach is similar to the CMM-SW process measures approach (Baumert and McWhinney, 1992), and adapting Measuring for Process Management and Improvement approach (Florac and

Park, 1997) as can be seen in Figure 2. However, some adjustments and modification in parts of these software measures approaches are needed to make software process measurement suited to the SMEs environment. The following points explain how SMEs should measure their software process with the conjunction of the these approaches

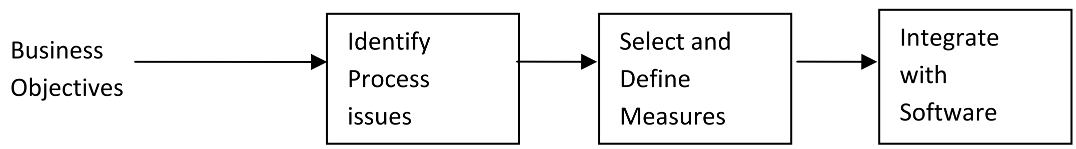


Figure 2 Measurement planning activities (Florac and Parrk, 1999)

The first point that is raised here is what processes the SMEs want to measure. The SMEs should be specific in selecting the processes that they want to measure or else it would be resource consuming. As the PRISMS project prioritizes the areas for improvement that most contribute to the business objectives, it will be likely to select the KPAs which those prioritized areas lie upon to measure their improvement. This will thus contribute back to the business objectives.

The second point is to find out how these selected KPAs are doing now with respect to the measurable attributes of quality, quantity, time and cost i.e. to see and detect the variation from acceptable performance (Florac, 1999). All processes are designed to produce results **(Florac, 1999)**. The products and services they deliver and the ways they deliver them have measurable attributes that can be observed to describe the quality, quantity, cost, and timeliness of the results produced (Baumert, 1992; Florac, 1999). If the SME knows the current values of these attributes, and if a process is not delivering the qualities the SME desires, the SME will have reference points to start from when introducing and validating process adjustments and improvements. The product and process measures may well be the same measures . Measurements of product attributes such as efforts, rework, and degree of code reuse that are tracked over time and used to investigate trends in process performance are just a few examples. The SMEs may seek reasons for variations in process performance by measuring attributes of the resources or environments that support the process such as, for example, the experience and training levels of the software engineers or the amount of computer time or memory available. In each case, the key to measuring process performance is to choose attributes and measures that are relevant to the particular process and issues under study as will be explain in the third point. To measure process performance, SMEs should measure process-performance attributes at several points in time to obtain sequential records of their behaviours. It is these sequential records that are the basis for establishing statistical control and, hence, for assessing the stability of the process.

The third point, due to the lack of resources within the SMEs, measurement could be time and cost consuming if the SMEs are not selective of the process issues that contribute to their business objective. The choices of attributes to measure will change from process to process and from issue to issue. Within the KPA process there could be many measurable issues and attributes, however SMEs should carefully map process issues and attributes that contribute to their objectives as section 3.1 explains. For example as can be seen in table 3-1, if our business goals are based on function, cost, time to market, and quality, SMEs can identify process issues (concerns) that relate to achieving these goals. Hence process performance can be quantified by measuring attributes of products produced by our processes as well as by measuring process attributes directly. Table 3-1 lists some typical business goals that often concern SMEs and relates these goals to corresponding process issues and examples of attributes that can be measured to assess the performance of a process with respect to these issues.

Measurements of attributes or indicators like those shown in column four of Figure 3-1 are important not just because they can be used to describe products and processes, but because they can be used to control the processes that produce the products, thus making future process performance predictable. Measurements of product and process attributes can also be used to quantify process performance, running the performance overtime tell us the stability of the process and guide SMEs in making process improvements. This in turn helps keep SMEs operations competitive and profitable.

Business Objective	Process issues	Measurement attributes

Reduced development costs	Rework Productivity Efficiency	Product size Product complexity Effort Number of changes Requirement satiability
Better quality software	Predictability Problem recognising Root cause analysis	Number of defect introduced. Effectiveness of defect detection activities.
Increase functions	Product Conformance	Number of requirements Product size Product complexity Rates of change Percent nonconforming
Reduce time to market	Production rate Responsiveness	Elapsed time, normalised for a product characteristics.

Table 3-1: Business Goals, Process Issues, and

Related Measurable Attributes

The next section discusses the guidelines of identifying process issues that need to be measured in the process concerning the business objectives that relate to risks that threaten the SMEs ability to meet goals, responsibilities, or commitments. Goals and issues serve to identify and focus the measurements needed to quantify the status and performance of software processes.

Guideline 4-3: Identifying Process Issues

The process or sub process that is being managed has a purpose or objective that can be traced back to the software organization's business goals.

Experience has shown that it is important to identify the critical factors that determine whether or not your processes succeed in meeting the goals that the SME has set. These critical factors often arise from concerns, problems, or issues that represent levels of risk that threaten the SME's ability to meet its goals. We refer to critical factors collectively as issues. Note that issues are not necessarily problems. Rather, based on the SME understanding and experience with the processes and the products the processes produce, they describe situations that require attention.

Guideline 4-4: Guidelines for Identifying Process Issues

The following guidelines outline a straightforward approach for identifying process issues.

1. **Use the identified business goals or objectives**. Use the identified business objectives already carried out as an activity within the PRISMS project. If the SME has a resource difficulty at the time when measurements are needed, it may be important to prioritize the business objectives based on the level of importance attached to them by the business. In most cases, business goals and objectives that are tied to cost, quality, or time can be mapped readily to the appropriate software processes.

2. **Group the identified areas of improvement into their common KPAs**. This will help to identify issues that can be described and quantified by closely related measurements.

3- **List the objectives for each KPA**. Listing objectives in terms of process performance will help identify potential problem areas where it can be easy to elicit or to induce the measurement attributes from the KPA process objective. The best way to do this is in terms of the attributes of the products or processes that the SME wants to control or improvements it wants from its software process. An example of this can be seen in table 3-2.

KPA	Goal
Software Project Planning:	The schedules of activities for the project are consistent with each other and with the software requirements and the software development plan. The actual performance of the project does not indicate schedule slippage and the necessity of a re-planning effort.
Software Project Tracking	Actual results and performance of the software project

and Oversight:	are tracked against documented and approved plans.
Software Subcontract Management:	The prime contractor tracks the subcontractor's actual results and performance against the commitments.
Software Requirement Management	The requirement process is controlled in terms of stability i.e. the no. of requirement changes over time.

Table 3-2 an example of some KPAs and their related objectives lists in terms of process performance

Guideline 4-5: Identify Process Measure

As mentioned in the previous section each areas of improvement to should be grouped back their relative KPA. Each KPA associated with a capability level within the CMM-SW. A set process measurement attributes or indicators describes by (Baumert, McWhinney, 1992) which is categorize each level within CMM-SW with the possible set of indicators, an example of the repeatable and defined levels indicators could be seen in table 3-5.

The choices of attributes or indicators to measure changes from process to process and from issue to issue. This is where team knowledge of the SME's processes comes into play.

Process Measurement Attributes or Indicator Category	Repeatable Level	Defined Level

Progress	Progress Actual vs. planned completions. Gantt chart	Actual vs. planned completions with ranges. Gantt chart PERT chart
Efforts	Actual vs. planned staffing profiles	Actual vs. planned staffing profiles with finer granularity
Cost	Actual vs. planned costs Cost and schedule variances	Cost Actual vs. planned costs ranges Cost and schedule performance indices
Software Quality Assurance Audit Results	Status of non-compliance issues	Status of non-compliance issues Audit information Sampling size information
Trouble Reports	Status of trouble reports Number of trouble reports opened, closed, unevaluated during Trouble report density Comparison of trouble reports and Number of trouble reports	Status of trouble reports Number of trouble reports compared with historic data Length of time trouble reports remain open Number of trouble reports per product

Peer Review		Results Number of opened and closed defects Number of defects by product defect density Pareto analysis of defects Preliminary control charts
Defect Prevention		
Requirements Stability	Number of requirements changes and requirement clarifications Distribution of requirement over releases.	Number of requirements changes and requirements clarifications with ranges Distribution of requirements changes by requirement type Length of time requirements. change requests remain open Number of waivers requested and approved from requirements
Size Stability	Size growth Distribution of size over releases	Size growth with ranges Distribution of size over releases

Process Stability		
Computer Resource Utilization	Actual vs. planned profiles of computer resource utilization	Actual vs. planned profiles of computer resource utilization
Training		Actual vs. planned number of classes offered Actual vs. planned attendance Course quality profiles Number of waivers from training.

Table 3-2. Indicators for the Repeatable and Defined Levels (Baumert, McWhinney, 1992, p.18)

The choices of attributes or indicators to measure changes from process to process and from issue to issue. This is where team knowledge of the SME's processes comes into play. Due to the SMEs resources barriers, SMEs should be careful in selection of their measurement indicators, the following guidelines help SMEs to select or identify measurement indicators:

- Measures should be related closely to the process issue under study.
- Have high information content. Pick measures of product or process qualities that are sensitive to as many facets of process results as possible (Florac, 1999)
- Permit easy and economical collection of data.

- Permit consistently collected well-defined data.
- • Show measurable variation. The measure of the attribute must show variation and must provide information about the process.
- • As a set, have diagnostic value. They should be able to help the SME identify not only that something unusual has happened, but what might be causing it.

However, below, are some guidelines suggested by {Florac, 1999 #137}Florac and Carleton (1999). These are set guidelines of how to pickup Measures that could be used to characterize process performance:

• As a set, have diagnostic value. They should be able to help the SME identify not only that something unusual has happened, but what might be causing it.

Guideline 4-6: Collecting and retain data

The operational activities of measurement begin with the collecting and retaining of data. The procedures that SMEs defined for collecting and retaining data must now be integrated into the SMEs' software processes and made operational. This means putting the right people, sensors, tools if it is possible, and practices into the processes in the right places. It also means capturing and storing the data for subsequent use in analysis and process improvement.

Guideline 4-7: Analyze software process behavior

In this section we describe how to apply measures in terms of performance and stability. This document recommend to use statistical process control

(Radice, 2000) to assess software process behavior hence software process stability.

Bibliography

1. Williams, L. (2003), Guest Editor's Introduction: The XP Programmer--The Few-Minutes Programmer, **IEEE Computer Society**, 20(3) May/June 2003, 16-20.
2. Wells, D. (2003) **Extreme Programming: A gentle introduction.** [Internet] Available from: <http://www.extremeprogramming.org/index.html>[Accessed 15-05-2003].
3. Umpleby, S. and Karapetyan, A. (2003) **HOW A QUALITY IMPROVEMENT PRIORITY MATRIX REVEALS CHANGE IN A UNIVERSITY DEPARTMENT** [Internet] Available from: <http://www.gwu.edu/~umpleby/recent%20papers/quality_improv_matrix.htm>[Accessed 10-2-2004].
4. Reifer, D. J. (2003), XP and CMM, **IEEE Computer Society**, 20(3) May/June 2003, 14-15.
   Rational (2003) **Rational Unified Process** [Internet] Available from: <http://www.rational.com/products/rup/index.jsp?SMSESSION=NO>[Accessed 17-05-2003].
5. Rasmusson, J. (2003), Introducing XP into Greenfield Projects: Lessons Learned, **IEEE Computer Society**, 20(3) May/June 2003, 21-28.
6. Murru, O., Deias, R. and Mugheddu, G. (2003), Assessing XP at a European Internet Company, **IEEE Computer Society**, 20(3) May/June 2003, 37-43.
7. Lang, S. (2003),**Sound Technology Business Decisions**, IN: APTA TransITech Conference**,** May 2003,USA.
8. Harwood, R. (2003),**Rational Unified Process**, IN: SPIN Conference**,** 26-05-2003,Milton Keynes-UK, British Computer Society

9. Dyba, T. (2003),**Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context.**, IN: Proceedings of the 9th European Software Engineering Conference held jointly with 10th ACM SIGSOFT international symposium on foundations of software Engineering**,** September 2003,Helsinki, Finland, ACM Press, 148-157.
10. Done (2003) **concerning the definition of micro, small and medium-sized enterprises** [Internet] Available from: <http://europa.eu.int/euroex/pri/en/oj/dat/2003/l_124/l_12420030520en00360041.pdf>[Accessed 6 May 2003].
11. Crow, J. (2003) **Project Management Tips** [Internet] Available from: <http://www.projectkickstart.com/html/tips8.htm>[Accessed 12 Mar. 2004].
12. Chrissis, M. B., Wemyss, G., Goldenson, D., Konrad, M., Smith, K. and Svolou, A. (2003) **CMMI® Interpretive Guidance Project: Preliminary Report** [Internet] Available from: <http://www.sei.cmu.edu/pub/documents/03.reports/pdf/03sr007-bodyrevised.pdf>[Accessed 01-06-2004].
13. Chrissis, M. B., Konrad, M. and Shrum, S. (2003) **CMMI: Guidelines for Process Integration and Product Improvement**, UK, Addison Wesley.
14. Boehm, B. (2003), Value-Based Software Engineering, **ACM SIGSOFT Software Engineering Notes**, 28(2) March 2003, 3-15.
15. Ahern, D. M., Clouse, A. and Turner, R. (2003) **CMMI Distilled: A Practical Introduction to Integrated Process Improvement**, 2nd, uk, Addison Wesley.
16. Webmasters, I. (2002) **ITS Priority Matrix** [Internet] Wednesday, March 13, 2002. Available from:

<http://www.usfca.edu/its/support/priority_matrix.html>[Accessed 10-02-2004].

17. Umpleby, S. and Melnychenko, O. (2002),**Using a Quality Improvement Priority Matrix in a University Department**, IN: in Customer Satisfaction Management Frontier Confrence**,** Fairfax, VA, Quality University Press, 6.1-6.12.
18. SEI (2002) **Capability Maturity Model Integration (CMMI) Version 1.1., CMU/SEI-2002-TR-012,** [Internet] March 2002. Available from: <http://www.sei.cmu.edu/publications/documents/02.reports/02tr029.html> [Accessed 16 May 2003].
19. Institute, B. (2002) **BOOTSTRAP** [Internet] Available from: <www.bootstrap-institute.com/home.htm>[Accessed 12-11-2002]. Eldon Y. Li, H. C. T. L. (2002), Software process Improvement of top Organizations in Taiwan:a comparative study, **Total Quality Management**, 13(5) 701-713.
20. Conradi, R. and Fuggetta, A. (2002), Improving Software Process Improvement., **IEEE Software**, 19(4) 92-99.
21. Conradi, R. and Fuggeta, A. (2002), Improving Software Process Improvement, **IEEE Computer**, 19(4) 92-99.
22. Buchholz, J. (2002) **Technology in Interpretation** [Internet] Available from: <http://www.uwsp.edu/cnr/schmeeckle/interp/Tech/resultsmatrix.htm>[Accessed 11-2-2004].
23. Boehm, B., Port, D., Jain, A. and Basili, V. (2002) **Achieving CMMI Level 5 Improvements with MBASE and the CeBASE Method** [Internet] May 2002. Available from:

<http://www.stsc.hill.af.mil/CrossTalk/2002/may/boehm.asp>[Accessed 11 October 2002].

24. Wolak, R. G. (2001) **DISS 725 – System Development: Research Paper 4 Software Process Assessment and Improvement Models** [Internet] Available from: <http://www.scisstudyguides.addr.com/papers/rwDISS725researchpaper4.pdf>[Accessed 5 May -2002].
25. Ward, R. P., Fayad, M. E. and Laitinen, M. (2001), Software Process Improvement in the Small, **Communication of the ACM**, 44(4) April 2001, 105-107.
26. Sommerville, I. (2001) **Software Engineering**, 6th, London, Pearson Education Ltd.
27. Richardson, I. (2001), Software process matrix: a small company SPI model, **Software Process: Improvement and Practice**, 6(3) September 2001, 157-165.
28. Reason, P. and Brabury, H. (2001) **Hand Book of Action Research**, London, SAGE Publication.
29. Persse, J. R. (2001) **Implementing the Capability Maturity Model**, USA, John Wiley & Sons Inc.
30. Paulk, M. C. (2001) **Extreme Programming from a CMM Perspective** [Internet] Available from: <www.sei.cmu.edu/activities/cmm/papers/xp-cmm.pdf>[Accessed 10-7-2003].
31. Leung, H. K. N. and Yuen, T. C. F. (2001), A Process Framework for Small Projects, **Software Process: Improvement and Practice**, 6(2) June 2001, 67-83.
32. Williams, L., Kessler, R. R., Cunnigham, W. and Jeffries, R. (2000) **Strengthening the Case of Pair-Programming** [Internet] Available from:

<http://www.cs.utah.edu/~lwilliam/Papers/ieeeSoftware.PDF>[Accessed 29th May 2003].

33. Parzinger, M. J. and Nath, R. (2000), A study relationships between total quality management implmentation factors and software quality, **Total Quality Management**, 11(3) 353-371.
34. O'Hara, F. (2000),**European experiences with software process improvement**, IN: Proceedings of the 22nd international conference on Software engineering**,** June 2000.
35. Lyard, A. and Orci, T. (2000),**Dynamic CMM for small organizations**, IN: Proceedings ASSE 2000, the first Argentine Symposium on Software Engineering**,** September 2000,Tandil, Argentina, 133-149.
36. Kautz, K., Hansen, H. W. and Thaysen, K. (2000),**Applying and Adjusting a Software Process Improvement Model in Practice: The use of the IDEAL Model in a Small Software Enterprise**, IN: Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000)**,** June 4-11-2000,Limerick, Ireland, ACM Press, 626-633.
37. Kautz, K., Hansen, H. W. and Thaysen, K. (2000),**Applying and Adjusting a Software Process Improvement Model in Practice: The Use of the IDEAL Model in a Small Software Enterprise**, IN: Winter Simulation Conference**,** June 2000,USA-Pittsburgh, Society for Computer Science Simulation International, 386-393.
38. Horvat, R. V., Rozman, I. and Gyrks, J. (2000), Managing the complexity of SPI in small Organizations, **Software Process: Improvement and Practice**, 5(1) March 2002, 45-54.
39. Hilburn, T. (2000), Software quality and the capability maturity model, **Communications of the ACM**, 40(6) pp.30-40.

40. Han, V. L. (2000), An Early SPICE Experience, **Quality Process**, 33(2) Feb 2002, 99-105.
41. Habra, N., Niyitugabira, E., Lamblin, A.-C. and Renault, A. (2000),**Software Process Improvement for Small Structures : First Results of a Micro-Assessment Framework**, IN: Proceedings of the European Conference on Software Process Improvement**,** September 16-11-1998,Barcelona, 1-12.
42. Dyba, T. (2000), Improvisation in Small Software Organizations, **IEEE Software**, 17(5) September 2000, 82-87.
43. Curtis, B. (2000),**Tutorial Notes: Software Process Improvement: Best Practices and Lesson Learned**, IN: International Confrence On Software Engineering**,** 4-11 June 2000,Limerick, Ireland, University of Limerick, Arnold, D. N. (2000) **The Explosion of The Ariane 5** [Internet] Available from: <http://www.ima.umn.edu/~arnold/disasters/ariane.html>[Accessed 03-04-2002].
44. Thorwart, K. (1999), The AMETIST process improvement experiment: towards efficient team development in small Organizations, **Software Process: Improvement and Practice**, 4(1) Feb. 1999, 11-18.
45. Solingen, R. v., Kusters, R. J., Trienekens, J. J. M. and Uijtregt, A. v. (1999), Product-focused software process improvement (P-SPI): concepts and their application, **Quality and Reliability Engineering International**, 15(6) November/December 1999, 475-483.
46. Raynus, J. (1999) **Software process improvement with CMM**, London., Artech House.
47. Paulk, M., Weber, C., & Chrissis, M. (1999), The Capability Maturity Model for Software. In K. Emam & N. Madhavji (Eds.), Elements of

Software Process Assessment and Improvement, **IEEE Computer Society Press**, 3-22.

48. Paulk, M. C. (1999),**Practice of High Maturity Organisation**, IN: SEPG Conference**,** Atlanta, Georgia.
49. Florac, W. A. and Carleton, A. D. (1999) **Measuring the Software Process: Statistical Process Control for Software Process Improvement**, Reading, Massachusetts, Addisson Wesley Longman, Inc.
50. Emam, K. E. and Madhavji, N. H. (1999) **Elements of Software Process Assessment & Improvement**, USA, IEEE Computer Society. Dorling, A. (1999) **The SPICE Trials** [Internet] 1999. Available from: <http://www.sqi.gu.edu.au/spice//>[Accessed 05-05-2003].
51. Beck, K. (1999) **Extreme Programming Explained: Embrace Change**, USA, Addison Wesley Professional.
52. Baskerville, R. L. (1999), Investigation Information Systems With Action Research, **Communication of the Association of the Information Systems**, 2(19) October 1999, 1-32.
53. Zahran, S. (1998) **Software Process Improvement: Practical Guidelines for Business Success**, USA, Addison-Wesley.
54. Zahran, S. (1998) **Software Process Improvement. Practical Guidlenes for Business Success**, USA, Addison-Wesley.
55. Paulk, M. C. (1998),**Using the Software CMM in Small Organizations**, IN: The Joint 1998 Proceedings of the Pacific Northwest Software Quality Conference and the Eighth International Conference on Software Quality**,** 13-14 October 1998,Portland, OR, Software Engineering Institute, 350-361.
56. Kim, C. (1998) **CMM implementation guide. choreographing software process improvement.**, Reading, Mass.. Harlow, Addison-Wesley.

57. Kautz, K. (1998), Software Process Improvement In Very Small Enterprises: Does It Pay Off?, **Software Process: Improvement and Practice**, 4(4) December 1998, 206-226.
58. Dunaway, D. K., Goldenson, D., Monarch, I. and White, D. (1998),**How Well is CBA IPI Working? User feedback**, IN: Software Engineering Process Group Conference**,** March 1998,Chicago, Illinois, The Dunway Group.
59. Wang, Y., Court, I., Staples, M. R. G., King, G. and Dorling, A. (1997),**Quantitative Evaluation of the SPICE,CMM, IS0 9000 and BOOTSTRAP**, IN: 3rd International Software Engineering Standards Symposium (ISESS '97)**,** Centre for Syst. Eng., Southampton Inst., UK, IEEE Computer, P.57.
60. Richardson, I. (1997),**Quality Function Deployment - A Software Process tool?**, IN: Third Annual International QFD Symposium**,** October, 1997,Linkoping University, Linkoping, Sweden.
61. Johnson, D. L. and Brodman, J. G. (1997) In *IEEE Computer Society*, Vol. 8, pp. 1-5.
62. Emam, K. E., Drouin, J. N. and Melo, W. (1997) **SPICE: The Theory and Practice of Software Process Improvement and Capability Determination**, USA, Willey Publication Series.
63. Brodman, J., & Johnson, D (1997),**A software process improvement approach for small organizations and small projects**, IN: International Conference on Software Engineering**,** Boston, MA.
64. Bemberger, J. (1997), Essence of the Capability Maturity Model, **IEEE Computer Society**, pp.112-114.

65. Atkins, M. H. and Lowe, J. F. (1997), Sizing up the small firm: UK and Australian experience, **International Small Business Journal**, 15(2) Feb. 1997, 42-56.
66. van de Velden, M. J., Vreke, J., Wal, B. V. D. and Symons, A. (1996), Experiences with the Capability Maturity Model in a Research Environment, **Software Quality Journal**, 5(87-95.
67. Ho, K. (1996) **UI work destroyed in blast** [Internet] Available from: <http://www.kevin-ho.com/Text/DI%20Archiv/archiv_DI_1rocket.htm>[Accessed 01-03-2002].

    Herbsleb, J. D. and Goldenson, D. R. (1996),**A systematic survey of CMM experience and results**, IN: Proceedings of the 18th international conference on Software engineering**,** May 1996,Berlin, Germany, IEEE Computer Society, 323 - 330.
68. Briand, L. C., Differding, C. M. and Rombach, H. D. (1996), Practical guidelines for measurement-based process improvement, **Software Process: Improvement and Practice**, 2(4) December 1996, 253-280.
69. Wellendorf, J. A. (1995),**Armstrong Building Products Operations: Information and Analysis**, IN: The Official Conference of the Malcolm Baldrige National Quality Award**,** February 5-7, 1995,Washington, DC, The Quest for Excellence VIII.
70. Paulk, M. C., Weber, C. V., Curtis, B. and Chrissis, M. B. (1995) **The Capability Maturity Model for Software: Guidelines for Improving the Software Process (SEI)**, USA, Addison Wesley.
71. Finkelstein, A. (1995) **Report of the Inquiry Into The London Ambulance Service** [Internet] Available from:

72. <http://www.cs.ucl.ac.uk/staff/A.Finkelstein/las/lascase0.9.pdf>[Accessed 01/10/2002].

Finkelstein, A. and Dowell, J. (1995) **A Comedy of Errors: the London Ambulance Service case study** [Internet] Available from:

73. <http://www.cs.ucl.ac.uk/staff/a.finkelstein/papers/lascase.pdf>[Accessed 05-05-2002].
74. Cohen, L. (1995) **Quality Function Deployment: How to Make QFD Work for You**, Reading, Massachusetts, Addison-Wesley Publishing Company.
75. Coallier, F. (1995), TRILLIUM : A Model for the Assessment of Telecom Product Development and Support Capability, **IEEE Software Process Newsletter**, 2(3-8.
76. Kuvaja P and Biceggo A (1994), A European assessment methodology, **Software Quality Journal**, 3(2) May 1995, 117 - 127.
77. Institute, S. E. (1994) Software Engineering Institute, pp. 1-42.
78. Paulk, M., Curtis, B., Chrissis, M. and Weber, C. (1993) **Software Process Maturity Questionnaire: Capability Model version 1.1** [Internet] 11-November 19994. Available from:

<http://www.sei.cmu.edu/publications/documents/93.reports/93.tr.024.html>[Accessed 12-11-2002].

Appendix

Total Capability Assessment Report

Assessment Capability Detailed Report

SMEs Capability Assessment Tool

Level 2: Key Process Areas

Requirement Management
Software Planning Process
Software Process Tracking and Oversight
Software Quality Assurance
Software Configuration Management

Level 3: Key Process Areas

Organisation Process Focus

Total Capability Assessment Report

Assessment Capability Detailed Report

Organisation Process Definition
Training Program
Integrated Software Management
Software Product Engineering
Inter-group Co-ordination
Peer Reviews

PRISMS Main Pages

Level 2: Software Project Planning

Are estimates (e.g., size, cost, and Schedule) documented for use in planning and tracking the software project?

- Yes
- No
- N/A
- Partially

Comment:

Do the software plans document the activities to be performed and the commitments made for the software project?

- Yes
- No
- N/A
- Partially

Comment:

Do all affected groups and individuals agree to their commitments related to the software project?

- Yes

- No
- N/A
- Partially

Comment:

Does the project follow a written organizational policy for planning

a software project?

- Yes
- No
- N/A
- Partially

Comment:

Are adequate resources provided for planning the software project

(e.g., funding, training and experienced individuals)?

- Yes
- No
- N/A
- Partially

Comment:

Are measurement used to determine the status of the activies for planning the software project (e.g., completion of milestones for the project planning activies as compared to the plan?

- Yes
- No
- N/A
- Partially

Comment:

Does the project manager review the activities for planning the software project on both a periodic and event-driven basis?

- Yes
- No
- N/A
- Partially

Comment:

Submit Reset

Level 2: Software Project Tracking and Oversight

Are the project's actual results (e.g., Schedule, size, and cost) compared with estimates in the software plans?

- Yes
- No
- N/A
- Partially

Comment:

Is corrective action taken when actual results deviate significantly from the project's software plans?

- Yes
- No
- N/A
- Partially

Comment:

Are changes in the software commitments agreed to by all affected groups

and individuals?

- Yes
- No
- N/A

Partially

Comment:

Does the project follow a written organization policy for both tracking and

controlling its software development activities?

Yes

No

N/A

Partially

Comment:

Is someone on the project assigned specific responsibilities for tracking

software work products and activities(e.g., effort, schedule, and budget)?

Yes

No

N/A

Partially

Comment:

Are measurement used to determine the status of the activities for software

traqcking and oversight(e.g., total effort expended in performing tracking and

oversight activities)?

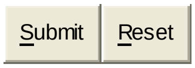
- Yes
- No
- N/A
- Partially

Comment:

Are the activities for software project tracking and oversight reviewed with senior management on a periodic basis(e.g., project performance, open issues, risks, and action items)?

- Yes
- No
- N/A
- Partially

Comment:


Submit
Reset

Level 2: Software Quality Assurance

Are SQA activities planned?

- Yes
- No
- N/A
- Partially

Comment:

Does SQA provide objective verification that software products and activities adhere to application standards,procedures, and requirements?

- Yes
- No
- N/A
- Partially

Comment:

Are the results of SQA reviews and audits provided to affected groups

and individuals (e.g., those who performed the work and those

who are reponsible for the work)?

- Yes
- No
- N/A
- Partially

Comment:

Are issues of noncompliance that are not resolved within the software

project addressed by senior management(e.g., deviation from

applicable standards)?

- Yes
- No
- N/A
- Partially

Comment:

Does the project follow a written organizational policy for implementing

SQA?

- Yes
- No
- N/A
- Partially

Comment:

Are adequate resources provided for performing SQA activities (e.g.,

funding and a designated manager who will recieve and act on

software noncompliance item)?

- Yes
- No
- N/A
- Partially

Comment:

Are measurements used to determine the cost and schedule status of the activities performed for SQA)e.g., work completed, effort and funds expended compared to the plan)?

- Yes

- No
- N/A
- Partially

Comment:

Are activities for SQA reviewed with senior management on a periodic basis?

- Yes
- No
- N/A
- Partially

Comment:

Submit Reset

Level 2: Software Configuration Management

Are Software Configuration management activities planned for the project?

- Yes
- No
- N/A
- Partially

Comment:

Has the project identified, controlled, and made available the softwork products through the use of configuration management?

- Yes
- No
- N/A
- Partially

Comment:

Does the project follow a documented procedures to control changes to configuration items/units?

- Yes

No

N/A

Partially

Comment:

Are standard reports on software baseline (e.g., software configuration

control board minutes and change request summary and status reports)

distributed to affected group individuals?

Yes

No

N/A

Partially

Comment:

Does the project follow a written organizational policy for implementing

software configuration management activities?

Yes

No

- N/A
- Partially

Comment:

Are project personnel trained to perform the software configuration management activities for which they are possible?

- Yes
- No
- N/A
- Partially

Comment:

Are measurements used to determine the status of the activities for software configuration management management(e.g., effort and funds expended for software configuration management activities)?

- Yes
- No
- N/A
- Partially

Comment:

Are periodic audits performed to verify the software baselines conform to the documentation that defines them (e.g., by the SCM group)?

- Yes
- No
- N/A
- Partially

Comment:

Submit Reset

Level 3: Organization Process Focus

1 - Are the activities for developing and improving the organization's and project's software processes coordinated across the organization (e.g., via a
software engineering process group)?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

---

2 - Is your organization's software process assessed periodically?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

---

3 - Does your organization follow a documented plan for developing and improving
its software process?

Yes ☐ No ☐

N/A Partially

Comment :

4- Does senior management sponsor the organization’s activities for software
process development and improvements (e.g., by establishing long-term plans, and by
committing resources and funding)?

Yes No

N/A Partially

Comment :

5- Do one or more individuals have full-time or part-time responsibility for the
organization’s software process activities (e.g., a software engineering process group)?

Yes No

N/A Partially

Comment :

6- Are measurements used to determine the status of the activities performed to
develop and improve the organization's software process (e.g., effort expended for
software process assessment and improvement)?

Yes No

N/A Partially

Comment :

7- Are the activities performed for developing and improving software processes
reviewed periodically with senior management?

Yes No

N/A Partially

Comment :

Reset Submit

Level 3: Organization Process Definition

1 - Has your organization developed, and does it maintain, a standard software process?

Yes No

N/A Partially

Comment :

2 - Does the organization collect, review, and make available information related to the use of the organization's standard software process (e.g., estimates and actual data on software size, effort, and cost; productivity data; and quality measurements)?

Yes No

N/A Partially

Comment :

3 - Does the organization follow a written policy for both developing and maintaining its standard software process and related process assets (e.g., descriptions of approved software life cycles)?

Yes No

N/A Partially

Comment :

4- Do individuals who develop and maintain the organization's standard software process receive the required training to perform these activities?

Yes No

N/A Partially

Comment :

5- Are measurements used to determine the status of the activities performed to define and maintain the organization's standard software process (e.g., status of schedule milestones and the cost of process definition activities)?

Yes No

N/A Partially

Comment :

6- Are the activities and work products for developing and maintaining the organization’s standard software process subjected to SQA review and audit?

Yes No

N/A Partially

Comment :

Reset Submit

Level 3: Integrated Software Management

1 - Was the project's defined software process developed by tailoring the organization's standard software process?

Yes No

N/A Partially

Comment :

---

2 - Is the project planned and managed in accordance with the project’s defined software process?

Yes No

N/A Partially

Comment :

---

3 - Does the project follow a written organizational policy requiring that
the software project be planned and managed using theorganization’s standard software process?

Yes No

N/A Partially

Comment :

4- Is training required for individuals tasked to tailor the organization’s standard software process to define a software process for a new project?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

5- Are measurements used to determine the effectiveness of the integrated software management activities (e.g., frequency,causes and magnitude of replanning efforts)?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

6- Are the activities and work products used to manage the software project subjected to SQA review and audit?

Yes No

N/A Partially

Comment :

Reset Submit

Level 3: Software Product Engineering

1 - Are the software work products produced according to the project's defined software process?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

---

2 - Is consistency maintained across software work products (e.g., is the documentation tracing allocated requirements through software requirements, design, code, and test cases maintained)?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

---

3 - Does the project follow a written organizational policy for performing
the software engineering activities (e.g., a policy which requires the use of appropriate methods and
tools for building and maintaining software products)?

Yes ☐ No ☐

N/A Partially

Comment :

4- Are adequate resources provided for performing the software engineering tasks (e.g., funding, skilled individuals, and appropriate tools)?

Yes No

N/A Partially

Comment :

5- Are measurements used to determine the functionality and quality of the software products (e.g., numbers, types, and severity of defects identified)?

Yes No

N/A Partially

Comment :

6- Are the activities and work products for engineering software subjected to SQA reviews and audits (e.g., is required testing performed, are allocated requirements traced through the software requirements, design, code and test cases)?

Yes No

N/A Partially

Comment :

Reset Submit

Level 3: Intergroup Coordination

1 - On the project, do the software engineering group and other engineering groups collaborate with the customer to establish the system requirements?

Yes No

N/A Partially

Comment :

---

2 - Do the engineering groups agree to the commitments as represented in the overall project plan?

Yes No

N/A Partially

Comment :

---

3 - Do the engineering groups identify, track, and resolve intergroup issues (e.g., incompatible schedules, technical risks, or system-level problems)?

Yes No

N/A Partially

Comment :

4- Is there a written organizational policy that guides the establishment of interdisciplinary engineering teams?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

5- Do the support tools used by different engineering groups enable effective communication and coordination (e.g., compatible word processing systems, database systems, and problem tracking systems)?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

6- Are measures used to determine the status of the intergroup coordination activities (e.g., effort expended by the software engineering group to support other groups)?

Yes No

N/A Partially

Comment :

7- Are the activities for intergroup coordination reviewed with the project manager on both a periodic and event-driven basis?

Yes No

N/A Partially

Comment :

Reset Submit

Level 3: Peer Reviews

1 - Are peer reviews planned?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

---

2 - Are actions associated with defects that are identified during peer reviews tracked until they are resolved?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

---

3 - Does the project follow a written organizational policy for performing peer reviews?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

4- Do participants of peer reviews receive the training required to perform their roles?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

5- Are measurements used to determine the status of peer review activities (e.g., number of peer reviews performed, effort expended on peer reviews, and number of work products reviewed compared to the plan)?

Yes ☐ No ☐

N/A ☐ Partially ☐

Comment :

6-Are peer review activities and work products subjected to SQA review and audit (e.g., planned reviews are conducted and follow-up actions are tracked)?

Yes No

N/A Partially

Comment :

Reset Submit