Egypt–SPIN Newsletter

Issue 4, Oct. – Dec., 2003

Sponsored by SECC

From the Editor

Welcome to our 4th issue of Egypt –SPIN newsletter. In each issue we are trying to put together relevant information in the form of articles and recaps from the previous 3 months events hoping to provide our members of Egypt – SPIN with information to support their current interests.

In this issue you will find articles presenting **"Software Engineering Competence Center (SECC) program to support the local software development Small & Medium companies SMEs" Dr. Gamal Mohamed Aly** SECC director will provide the reader with brief account of the SPI / SMEs program.

Dr. Gamal Darwish SECC program coordinator will present our efforts to **"Build National Software Process Asset Library for SMEs"**

The team members of SPI/SMEs program are presenting the general features of each key process area which will be taken as mandatory in the software process improvement program.
Dr. Amr Kamel is writing **"Developing Requirement Management for SMEs".**
Eng. Elaraby is writing **"Software Configuration Management for SMEs"**
Dr. Hoda Hosney is writing **" SQA within the SPI framework of Tailoring the CMM for SMEs"**
Mr.Marian Tadrous is writing **" Peer Reviews in the local certification model"**
Eng. Madiha A. Hassan is writing **"Software Product Engineering for SMEs"**

Dr. Adel ghanam is also continuing to give us his valuable thoughts in the article **"Can we continue without Product Quality Accreditation".**

Eng. Omar Kamal will present a review of the **"Mastering Process Improvement"** workshop

We hope we succeed to give you an idea about what is going in our community. Please write to the editor your comments about our progress. we always ask you to submit short articles for publication that deal with your experience in defining, developing and managing software efforts as well as process improvement experience. Remember that our goal is to encourage an interchange between our readers. As reminder you can email me mad_abdalla@mcit.gov.eg. Also if you have a colleague who wants to receive an electronic mail of our newsletter he/she can send an email to spin@secc.org.eg asking for registration as Egypt-SPIN member.

Table of Contents

SECC Software Process Improvement Program For Small & Medium Companies (SMEs)

By: Dr. Gamal Aly

The vision of the Ministry of Communication and Information Technology (MCIT) in supporting the implementation of software quality standards in the software development local industry became a reality by establishing the Software Engineering Competence Center (SECC) early 2001. A decision was made to initiate the activities by implementing the CMM® because it is a common sense application of process management and quality improvement concepts to software development and maintenance. It represents a broad-based consensus of good engineering and management practices agreed upon by the international software community.

To ensure the successful implementation of Software Process Improvement in Egypt SECC had assessed the readiness of the local companies for implementing CMM® We found only 20% of the local organizations were ready to start. Others have expressed their concerns of the enormous cost & time required to implement Software Process Improvement program. But the positive point was their interest and need to start implementing a more realistic model to serve the small & medium organizations.

The software local community associations (CIT , EHITA, ...) have discussed this issue coming to a conclusion that this solution should be an adopted model based on the CMM®. This model should be taken as a base for regulating the local software industry through standard mechanism. This will create a competitive edge in the local market and provide a solid infrastructure for the growth of the software industry. The local adopted model will serve as basis of local certification which will be considered as an intermediate step towards the international accreditation and recognition. This local certification (quality mark) will be a must for doing any local business.

SECC with its strategic partner Motorola and the Software Community representatives had formulated a working group on October 2003. A field survey conducted by a team from Motorola and SECC to four SMEs working in the local market. The result of the survey is a set of recommendations. To develop an adopted model of excellence based upon CMM® and IEEE standards. This set of adapted standards should include tailored processes, procedures and guidelines for the basic software processes, which are considered mandatory for any improvements. Criteria for tailoring each specific SME or small project should be included based on the suitable life cycle model. Risk criteria and patterns will be focused on shortage of resources, short life cycles, etc… with suggestions to minimize their potential impacts. This means the model will focus on barriers and suggest practical solutions to overcome them. Each tailored set of key process areas includes recommended methodologies, minimum metrics, required tools and simplified work flow diagrams to summarize the steps of each mandatory process. These tailored practices maintain the intent of CMM® by keeping the normative component of the selected KPAs. These tailored practices should not remove aspects which violate the CMM® principles. Rather, it should highlight them with practical and alternatives driven from community culture. Collectively the model intended to re-word CMM® practices to be applicable for SMEs.

SECC has developed the action plan of this project. It includes three phases (Preparation – Development – Implementation). The stakeholders of these phase are the local expertise, steering committee board (Local Industry Associations) and an international body to give guidance and review.

Building National Software Process Asset Library For SMEs

Dr. Gamal Darwish

Since the CMM® describes what a process should address rather than how it should be implemented there are many implementation details to be worked out. Also the CMM® is to be tailored or interpreted to different business environment in the context of implementation.

As a part of SECC program for software process improvement for SMEs, we are targeting a rigid way of implementation. To facilitate the implementation the we are going to build a process asset library to communicate the templates & checklist to the national software companies. This process asset library will serve as a training tool which can reduce the cost of training for its users.

<u>For SMEs getting started with this national process asset library will give them:</u>

- The roadmap for organizing the basic processes of any quality management system.
- A readily available processes to projects' managers for startup.
- Tools and templates to aid in the population of processes.

The Software Process Asset Library will include the basics needed to start the software process improvement program including the related resources for the basic set of the following key process areas:

1. Software Project Management.
2. Software Requirement Management.
3. Software Configuration Management.
4. Software Quality Assurance.
5. Software Peer Reviews.
6. Software Product Engineering.

The output hierarchy of each key process area will be as following:

1. Goals to achieve.
2. Actions to be followed.
3. Assets to be used.

The general feature of each set of goals, actions and assets will simplify the rules, prioritize actions, be cost effective for SMEs by managing their limited resources and complementing their short life cycles. Training materials needed to satisfy the goals of each KPA will be included. Simple documentation mechanism will be introduced with case studies in each KPA.

Both international partners and steering community members will review the outcomes. Pioneers in the SPI program from the local community will share in mentoring the implementation of the library elements.

Requirements Management in Small and Medium Enterprises – Developing an Egyptian Framework, conforming to the CMM model

By: Amr Kamel

Keywords

Requirements Management, SPI

INTRODUCTION

In 1994 Standish Group surveyed more than 8000 projects and found that the top three reasons that projects ran over schedule, over budget and with less functionality than desired all are related to Requirements Management (RM) (Standish Group, 1994). This type of surveys and knowledge has motivated Software Process Improvement (SPI) programs such as the Capability and Maturity Model (CMM) to include RM as a mandatory Key Process Area (KPA) for any organization to escalate up the CMM maturity levels.

Unfortunately, the CMM SPI model and many of similar models were built with large organizations in mind, making it difficult for SMEs to adopt. Furthermore, needs of the local IT community mandate further elaboration on the CMM model before it can be implemented. In this adaptation of the RM KPA in the CMM SPI, we will elaborate on the "how" aspects of RM rather than the "what" aspects defined in the CMM.

Requirements Management Processes

Overview

The objective of the RM process is to establish and maintain a common understanding with the customer on the requirements addressed by the project. This agreements covers both technical and non-technical (e.g. delivery dates) requirements and forms the basis for the planning, estimating and tracking of the remaining phases of the project development activities. Furthermore, industry experts estimate that a typical project will experience 25% change in its requirements through out its life time (McConnell 1996); a second objective of the RM processes is to manage these changes and minimize their impact on the overall project's schedule and budget.

Scope

The RM process described in this report presents a framework that supports:

- An evolutionary, incremental approach to requirements development.
- User/Customer involvement in requirements elicitation processes.
- A disciplined approach to requirement development.
- Assure requirements traceability to facilitate RM.

Processes

The following processes and templates to support their execution, will be defined as part of the SECC's support RM in the local market:

1. Planning – This process starts with a project statement or a CFP. It is concerned with defining the project goals, constraints and acceptance criteria.
2. Elicitation – By eliciting customer requirements, an understanding of the software operational scenarios is documented.
3. Analysis & Negotiation - In this process, the requirements are qualitatively analyzed and the project risks are identified and analyzed.
4. Specification – In this process, the requirements, rationale and their traceability are documented from the perspective of the development team.
5. Verification & Validation – this process is used to baseline the requirements for the project's development phase by validating the requirements with the customer and verifying that the

specification captured all user requirements.

6. Acceptance – The acceptance process provides the project stakeholders a milestone where critical decisions about the project can be taken (e.g. go no-go).
7. Postmortem – The postmortem process ensures a graceful end for the requirements phase by performing several housekeeping activities, e.g. updating the project/enterprise database ensuring that all required forms are complete and properly filled, etc. suggestions with supporting data for improving the RM process are collected, compiled (but not analyzed) at this point.

As shown in Figure 1, the planning is the entry point to the requirements management KPA. From the customer perspective, the process ends upon the acceptance of the requirements, Nevertheless, the process ends after the postmortem process for the software team.

Activity Interactions

The deployment of the above processes is expected to execute in order. However, interactions between neighboring processes may take place during progression through the requirements development cycle. However, the nature of the clarification or issue to deal with, dictates the feedback path to follow through the process. For example, as seen in figure 2, issues discovered during verification could be addressed in elicitation, analysis, or specification, a combination of them.

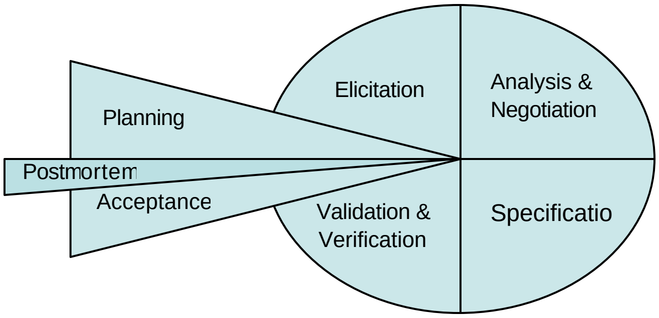
Document templates

In addition to the process definitions, templates for the following documents will be provided:

1. SRS – Software requirements specifications.
2. Requirements management policy.

Suggested measurements and metrics related to the Requirements phase will be proposed. A survey of available RM tools will be provided as well.

Figure 1



.

REFERENCES

1. McConnell, Steve. Rapid Development: Taming wild software schedule, Microsoft press 1998.
2. Standish Group, The. Taming the seas of Information Technology.1994

Biography

Dr. Amr Atef Kamel is an Assistant Professor, Department of Computer Science, Faculty of Computers and Information, Cairo University. He is also an Adjunct Professor, University of Alberta, Edmonton, Alberta, Canada. His research and professional interests include software quality management and process improvement.

Establishing a Software Configuration Management (SCM) Key Process (KP) - For Small & Medium Companies

By: Abd El Meguied Al Araby

Introduction

Software is a living object, which evolves over time. A control scheme should be used to manage its evolution and control changes to it, during its life cycle. Software Configuration Management is the process responsible for these activities.

The Software Configuration Management Process is addressed at the Capability Maturity Model for Software (SW-CMM) Level 2.

The SW-CMM is not adequate for small and medium companies, since these companies have not enough software developer to fill all the roles proposed by the model. It is wiser, that the small and medium companies tailor the SW-CMM model to their needs and their limited resources.

This article discusses the activities required to establish a Software Configuration Management Process for small and medium companies.

Guiding Principles

The basic principles that will guide the establishing of Software Configuration Management for small and medium companies are:

- The SEI Capability Maturity Model for Software (SW-CMM)
- IEEE Software Engineering Standards.

Approach

The approach to the development of Software Configuration Management Process shall include the following:

- Understanding the characteristics of small and medium organizations.
- Analyzing the SW-CMM level 2 Software Configuration Management Key Process Area, IEEE standards and a literature search for reference to SCM implementation at small organizations.
- Establishing the Software Process Improvement Plan for the Software Configuration Management Process.
- Implementing the Software Process Improvement Plan for the Software Configuration Management Process.
- Integrate the SCM KPA with other KPAs for the proposed model for small and medium companies.

Activities

During the establishing phase, the following activities should be addressed:

- Define the SCM Policy.
- Define the SCM Process.
- Identify the SCM KPA relationships with the other selected KPAs.
- Simplifying SCM Planning.
    - o Derive a simple SCM plan template
    - o Document the SCM process.
- Define abbreviated roles & responsibilities and propose who will be in charge for it.
- Select the required SCM procedures and document it.
- Consolidate all KPAs Policies into one.
- Consolidate the SCM Plan, Software Quality Assurance Plan and Peer Review Plan into the Software Development Plan.
- Define a simple metrics scheme for SCM.
- Propose SCM tools.

Deliverables

- SCM Policy.
- SCM Process.
- SCM Plan Template.
- SCM Procedures.
- SCM Metrics
- List of SCM tools.

References:

1. ANSI/ IEEE Std. 1042-1987 IEEE guide to Software Configuration Management.
2. IEEE Std. 828-1990 IEEE Standard for Software Configuration Management Plans.
3. Paulk M. C. et al: The Capability Maturity Model guidelines for improving Software Process, Addison-Wesley, 1995.
4. Paulk M. C.: Using the Software CMM in Small Organizations, Carnegie Mellon University, 1998.
5. Terttu Orci, Capability Maturity Model for Extra Extra Small Organizations, Level 2, 2000, UMINF 00.12.
6. Terttu Orci, Capability Maturity Model for Extra Small Organizations, Level 2, 2000, UMINF 00.13.
7. Terttu Orci, Capability Maturity Model for Small Organizations, Level 2, 2000, UMINF 00.20.

Biography

Eng. Elaraby is a freelance software engineering consultant. He has over 28 years experience in software engineering in the military, government and private sectors.

SQA within the SPI framework of Tailoring the CMM for SMEs

By: Hoda M. Hosny

The software Capability Maturity Model (CMM) has gained wide acceptance in software process improvement everywhere in the world. The gains of using CMM in software process improvement have also been widely publicized by its owner, the Software Engineering Institute (SEI) through various means. However, implementing CMM in an organization is a difficult and expensive task, as is the case with any improvement effort, requiring another way of thinking and more activities to be performed by the employees than they are used to. Software companies which have been functioning in the Egyptian market for sometime and aspiring for international certification would surely want to discover the way towards CMM maturity levels. Newly established software development organizations with small numbers of developers and who undoubtedly experience some symptoms of chaos at the beginning caused by immaturity, would seek the way to internationally recognized process improvement models. It is advisable for all such companies to start on a software process improvement program. Such a program may adopt the CMM or some other model publicly available, or it can be approached using a local model developed for that purpose. CMM, for example, presents a number of maturity levels stepwise and to enough detail, thereby offering a roadmap, which is based on sound management practices, and has been historically proven to lead to success with high probability [2]. In terms of process improvement, the model introduces more and more engineering and management practices. However, the existing models, such as CMM, are not directly applicable in any organization. CMM, for example, proposes more than 25 organizational roles, with various tasks and responsibilities. In a small or medium size organization, there is not enough people to fill the roles proposed, neither is there any need for many of those roles [2]. Furthermore, the models are usually described on a huge number of text pages, being quite tedious and time consuming to comprehend [2]. Therefore, the models need to be scaled down to the local needs and responsibilities of such companies. At the same time, it is important that such a model is applicable continuously in the course of time, while the organization grows, otherwise it will not be useful any more.

The objective of the SPI program for SMEs is to develop a software process improvement framework for small organizations which is usable from the very start of its adoption, dynamic in the sense of supporting growth, and easy to get oversight. The framework is intended to be applicable fom the start of the business (or from the start of its first introduction into the business) and to continue with the organization as it grows in size by which time the original CMM, or any other publicly available model, should be an adequate tool for further improvement.

The evolution from a small or medium organization to a large organization may be extremely fast, e.g. as is currently the case in organizations providing Internet solutions and services, or it may be more moderate, occurring over a few years [2]. In either case, the SPI framework is intended to help SMEs get started with a limited number of key process areas. One of the essential KPAs is the software quality assurance process area.

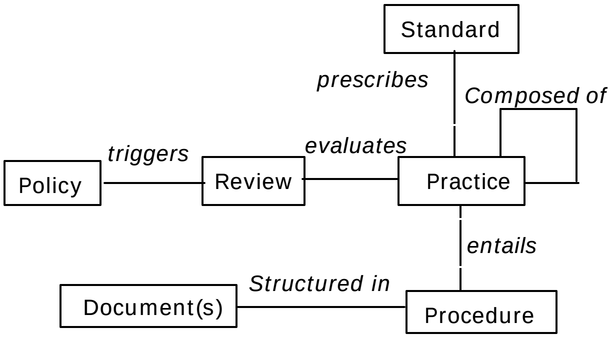
The purpose of Software Quality Assurance within the SPI framework is to provide management with appropriate visibility into the process being used by the software project and of the products being built. SQA involves reviewing and auditing the software products and activities to verify that they comply with the applicable procedures and standards and providing the software project manager and other appropriate managers with the results of these reviews and audits.

The software quality assurance group, according to CMM, is the collection of individuals (both managers and technical staff) who plan and implement the project's quality assurance activities to ensure the software process steps and standards are followed i.e. they take the responsibility of managing compliance within the organization . Fig. 1 (adapted from [1]) depicts the various issues in the compliance management process and how they relate to standards.

The software quality assurance group works with the software project during its early stages to establish plans, standards, and procedures that will add value to the software project and satisfy the constraints of the project and the organization's policies. By participating in establishing the plans, standards, and procedures, the software quality assurance group helps ensure they fit the project's needs and verifies that they will be usable for performing reviews and audits throughout the software life cycle. The software quality assurance group reviews project activities and audits software work products throughout the life cycle and provides management with visibility as to whether the software project is adhering to its established plans, standards, and procedures.

Compliance issues are first addressed within the software project and resolved there if possible. For issues not resolvable within the software project, the software quality assurance group escalates the issue to an appropriate level of management for resolution.

Fig. 1 Managing Compliance



The SQA within the SPI framework is made simple enough for adoption by SMEs while maintaining the structure of the CMM. The process activities and procedures within these activities are briefed down to the level of checklists that would be easier to follow as tailored practices. Some helpful templates and guidelines drawn from international standards are also provided for essential items such as the SQA plan, metrics methodology and risk management.

References

[1] Wolfgang Emmerich, W., Finkelstein, A., Montangero, C., Antonelli, S., Armitage, S. and Stevens, R., “Managing Standards Compliance”, IEEE Transactions On Software Engineering, Vol. 25, No. 6, November/December 1999

[2] Laryd, A. and Orci, T. “ Dynamic CMM for Small Organizations”, Proceedings of ASSE 2000, the First Argentine Symposium on Software Engineering, Tandil, Argentina, Sep 2000, 133-149.

[3] Software Engineering Institute, “The Capability Maturity Model” http://www.sei.cmu.edu/

“Peer Review” As One Of The Local Certification Key Process Area

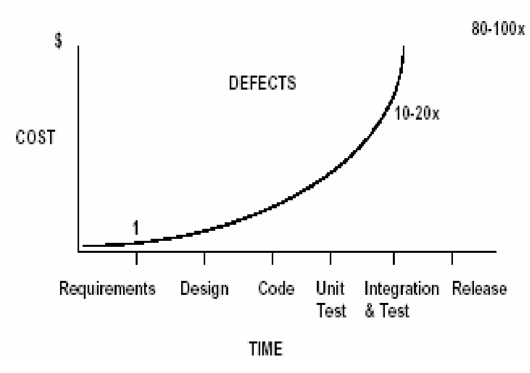
By: Marian Tadros

A peer review is a methodical examination of a product by the author’s peers to identify defects and areas where changes are needed. The specific products that will undergo peer reviews are identified in the project’s plans and scheduled as part of the project planning activities. Data concerning numbers of defects and types are collected.

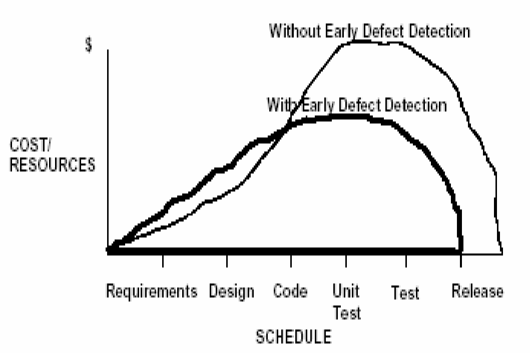
Why Peer Review Is Included As One Of The Local Certification KPAs?

From the graphs ad table below we can see the importance of peer review for managing and improving the software process.

Defects cost less to fix when detected earlier in the process



Early defect detection can shorten the schedule



World-Class “Peer Reviews”

MEASUREMENT	WORLD-CLASS BENCHMARK
Costs of Poor Quality: Total Savings	Reduced from ~ 33% to 15% ; Total Saving: $7.5-$45 Million
Defect Removal Efficiency	70-90% defect removal before test
Inspection Cost/Savings	Average cost $2,500; Savings $25,000
Post-Release Defect Rate	.01 Defect per KSLOC
Productivity	Double (e.g. in 3 years at ~ 33% a year)
Return on Investement	7:1 – 12:1 ROI
Schedule / Cycle Time	Reduce by 10-25 % (e.g., per year)

Scope Of Peer Review

Peer review types:

Based on technical researches and survey results, we founded inspection and walk through are suitable review types for the small end medium enterprises as we can replace the formal and management reviews by the quality assurance and project tracking KPAs.

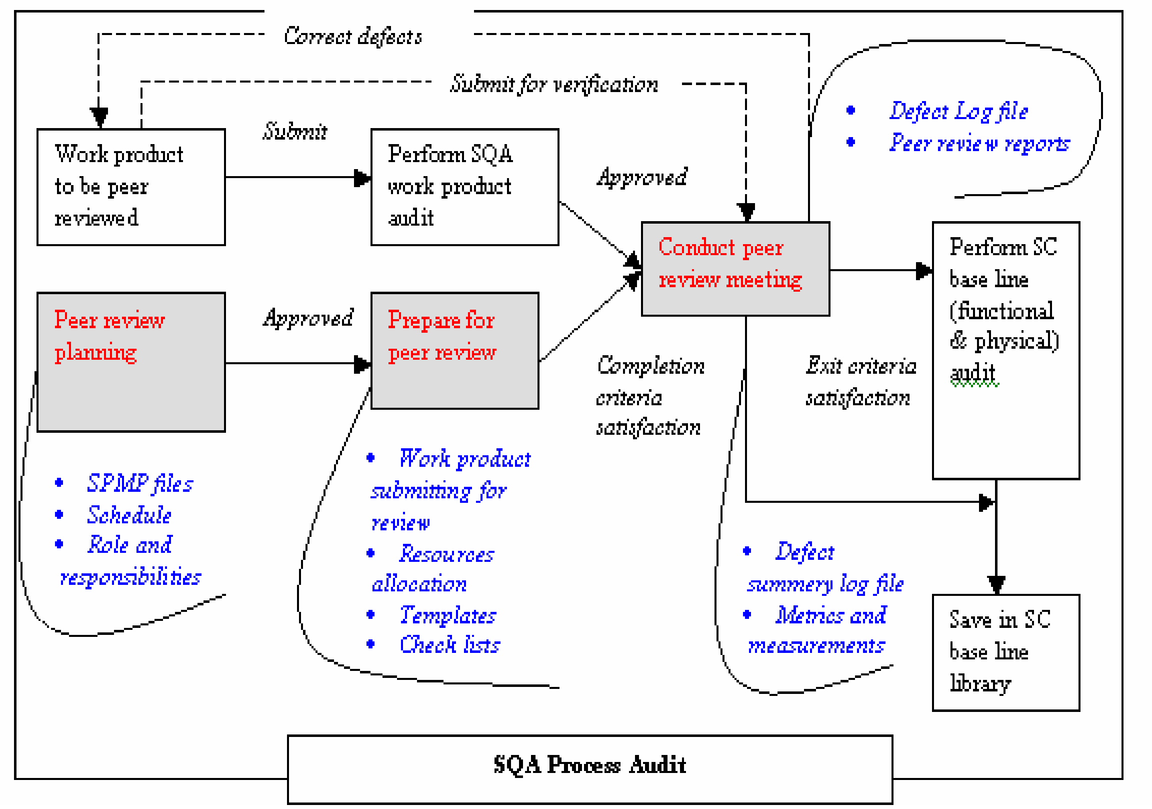
Work products to be peer reviewed

- Requirements definition work products
- Estimation work products
- Product engineering work products

Project Key Deliverables

- Pear review policy document
- Peer Review Process Model Document
- Peer Review Roles & Responsibilities Document
- Templates for peer review work products
- Check lists for peer review .
- Tools to be used
- Metrics and measurements
- Pee review process improvement Guide lines
- Mapping to CMM L3 peer review KPA goals and practice

Tentative View For Positioning Peer Review In The Software Life Cycle Process


Correct defects
Submit for verification
Submit
Work product to be peer reviewed
Perform SQA work product audit
Approved
Conduct peer review meeting
Defect Log file
Peer review reports
Perform SC base line (functional & physical) audit
Peer review planning
Approved
Prepare for peer review
Completion criteria satisfaction
Exit criteria satisfaction
SPMP files
Schedule
Role and responsibilities
Work product submitting for review
Resources allocation
Templates
Check lists
Defect summery log file
Metrics and measurements
Save in SC base line library
SQA Process Audit

Risks to be considered and managed in Pear Review KPA modelling

- Security "Managements are fear from sharing the critical component know haw among the organization".
- Project limited schedule and recourses
- Lack of expertise either in peer review or technical
- Human factors " People do not accept of being their work products reviewed".

Software Product Engineering Key Process Area

By: Madiha A. Hassan

One of the most significant challenges facing our software engineering community is to recognize an effective way to improve the quality of their products, services and the productivity with which they work. They need to capitalize on software development and maintenance experiences. Capitalizing involves application of technologies already widely in use, as well as the definition and adoption of standards. One goal of software engineering practitioners is to make sure that all those activities, which generally can be classified as process and product improvements are based upon understanding of the appropriate way of their goals and the appropriate implementation according to their specific environment.

Methods for quality improvement of software development can be classified along different lines according to their approach. Much effort is dedicated to improving the quality of the software development process (Software Process Improvement or SPI). The starting point in this approach is that an improved development process automatically leads to better software products that can be delivered within estimated time and budget constrains. Models supporting this approach are (CMM®, ISO, BOOTSTRAP, SPICE/ISO 15504, ….).

Other approach is to concentrate effort towards software product quality. How to specify and validate the quality of software products. ISO 9126 gives handles to focus on product quality such as the functionality, usability, maintainability, reliability, efficiency and portability of the software product. These six characteristics are considered as the main required attributes, which must be addressed in the output product. Sub-characteristics are also defined and used by international organizations as a guidance to specify their products' characteristics.

There is a strong evidence and explanations of the direct relation between using the CMM® as a backbone model with the software product quality attributes described in ISO 9126.

In fact common features and key practices of all the KPAs will influence product quality, but some will influence more than others. The KPAs, which are selected in the local model, are chosen because they are supporting this aspect than others. So they have to be undertaken as a basis for the quality improvement program.

According to my point of view I am relating these KPAs to ISO 9126 attributes which can be extended more than that. Software Requirement Management KPA will influence the functionality of the product. Software Configuration Management KPA is related to the capability of the software product for the identification of its components to be modified which influence the maintainability (as reusability and manageability is a sub-attribute of it in ISO 9126). Software Quality Assurance KPA has a direct effect to the efficiency attribute of the product. Peer Reviews KPA aims to avoid failure as a result of faults, a mature application of this KPA will give reliable software (fault tolerance and recoverability). Software Product Engineering KPA focuses upon the correctness and consistency of the product, this is the capability to provide the right or agreed upon product, which deals directly with the efficiency of the product (the efficiency is a sub- characteristic of the functionality).

The adopted Software Product Engineering KPA will take into consideration factors to like external view of the software product (as viewed by users), Criteria to build (internal view of the software product) and metrics to control the defined process.

The assets used will be The CMM® Software Product Engineering KPA as a guideline of "what to do" and IEEE standards to cover "how to do".

Implementation phase of this KPA should consider the nature of SMEs, the applicability of approaches and the positive aspects we can build on.

Tasks to be covered:

1. Ensure that software requirements are developed, maintained, documented and verified.
2. Ensure that software design is developed, maintained, documented and verified to accommodate the software requirements to form the framework for coding.
3. Ensure that the software code is developed, maintained, documented and verified to implement software requirements and software design.
4. Perform software testing according to a well-defined process.
5. Develop and maintain the documentation that will be used to operate and maintain the software according to standard formats.
6. Collect and analyze data on defects identified in peer reviews and testing according to a well-defined process.
7. Maintain consistency across software products including software requirements, software design & code (the tool is tractability matrix, which should be governed and controlled by an adequate process).

Deliverables

1. Simple governing policy to ensure the success of the process which includes the required resources and structure.
2. Clear & simple definitions for the processes.
3. Detailed procedures for the tasks to be performed.
4. Guidelines for tailoring each process & procedure according to the type of the product.
5. Risk criteria and mitigation procedure for each.
6. SDLC approaches and guidelines for selecting the appropriate one.

Can we continue without Product Quality Accreditation?

By: Dr. Adel Ghannam

1-Market Justification

With the current high competition and price war , Egyptian Software Companies cannot afford the "high –cost" for "after –sales" support. Product quality , bilingualism and support speed are their differential edges, and should be tackled at the very early phase of the development. In the same time, the agenda to build embedded quality should be simple enough to fit the resources capabilities.

It will also be difficult to promote the software export without having an entity to certify the product quality. This entity should be a state entity , to have the accreditation power

A software testing center will accredit the Egyptian software products , and allow printing the "software quality sign" on the CD , manual ,demos , Ads…etc.

The OBJECTIVE of the testing Unit

a. Product quality improvement for the Sector development

b. Creation of industry cluster , based on quality.

c. Create a normalized national quality metric, to increase export

d. Improve the International competitiveness

e. Industry standards setting

2- Scope of The Services

The main services of the center will be divided into :

1. Validate the design specifications against the requirement specifications.
2. The unit will be responsible to validate that, the product presented for testing is fully documented up to the international standard
3. Test Case and script Design based on Documentation
4. Running the unit test, Module integration test, Application integration test, and system test. This latter test will be done in association with the business analyst and the designer.

Besides these deliverables there will be the following supportive functions:

1. Test Result Archiving
2. Test tools procurement that fit different platforms, and run training for the test Engineers of the center
3. The final accreditation authority

3-The Target Market

The global target market are all software companies that need to create a quality image whether in the local market or the international market. We assume that the number of these companies are 50% of the registered software companies in the Chamber of Communication and Information technology ,of the federation of Egyptian industries.

To encourage these companies to use the services of the testing entity, the following actions are required:

1. Promote the quality sign
2. Give priority in the governmental purchase for product with the quality sign
3. Exemption from any export expenses for product having this sign
4. Companies having this sign should be part of the export subsidizing scheme , managed by the ministry of foreign trade
5. Companies having this sign , can exhibit in one local and two regional exhibitions ,free of charge. The MCIT will carry the cost

4-The Environment Forces and the test model

Positive Environmental factors that affect the testing unit:

1. The image of state recognized product

2. The low demand in the local market, which will force many software companies to search External markets.
3. Managers of software companies need tangible and quantifiable tools to measure their product quality
4. Hosting in the smart village

Negative environment factors:

1. Absence of confidence from the customer side , regarding their IPR
2. Poor documentation by the customer
3. Inadequate Budget for the development phase of the testing center
4. Week local know-how on testing techniques.

Conclusion:

The product quality is not any more a factor , that a company can use to differentiate its product in the market. Local and international users are assuming the availability of good quality for grant.

To position our self on the map of the International IT, we should talk about "our offering is the best cost/performance " to satisfy user requirements. Quality image , of Egyptian software, needs hard and smart approaches.

5- The development plan and cost estimate

The target organization should be capable to perform comprehensive testing ,in short time for different applications and different platforms .The target entity ,therefore , should be operated with test automation tools.:

Manual testing can be described as a situation where a person initiates each test, interacts with it,

and interprets, analyzes, and reports the results. I generally call test cases automated when all of

the following elements are present. If one or more elements are absent I consider the tests semi automated.

- Ability to run a subset of all the automated test cases
- ?No intervention is needed after launching the tests
- ?Automatically sets-up and records the relevant test environment parameters
- ?Runs the test cases
- ?Captures the relevant results
- ?Compares actual with expected results and flags differences
- ?Analyzes and reports pass/fail for each test case and for the test run
- Ability to run on different platform

The development phase of the testing unit includes:

1. Procurement of the hardware and basic software to cover most popular international platforms.
2. Procurement of the software testing tools to serve business solutions, embedded software, and mobile computing, Other test tools will be added, upon the needs
3. Prepare the premises (Hosting the unit in the smart village will be an asset)
4. Staff recruitment
5. Staff training on:

    i. Design test cases and scripts

    ii. Project management on testing projects

    iii. Incremental testing techniques, to reduce the time of Go-To-Market

    iv. Set up the electronic Archiving of software products tested and their test results, for future ease-of-accessibility

    v. Run a pilot

Mastering Process Improvement.
"A journey that never ends"

By: Omar Kamal

(14-18)Dec,2003 the "Mastering Process Improvement" workshop developed by the Software Engineering Institute (SEI) was carried out on Cairo by Dr. Charles R. Myers. Chuck is a Visiting Scientist at the Software Engineering Institute (SEI). He has over 35 years of professional experience in the military and industry as a manager, engineer, consultant, and teacher. Over 16 years of his experience has been closely related to Capability Maturity Models (CMM), with a constant focus on approaches for successfully implementing process improvement.

The course was organized and sponsored by the Software Engineering Competence Center (SECC) in collaboration with USAID/ICT. The attendees came from IBM, Fujitsu, Mentor Graphics, SAS, MCIT, ICT, DMS, SPI, ITWorx, OMS, United Ofoq, Link.Net, Giza Systems, ItSoft, and Quicktel. Although the attendees vary in their experience level, and their exposure to the CMM appraisal process, it was clear that most of the attendees didn't have the exact same view of process improvement as the SEI did.

The courses previously sponsored by the Software Engineering Competence Center (SECC) covered the CMM model in general, as-well as quite a good number of specific key process areas related to such model, but a number of questions remained un-answered.

- How to manage the change?
- How to asses current processes?
- How to build on areas of strength?
- How to narrow down the gap between specific CMMI PAs and the current project/organization practices?

The course revealed the importance of learning the process improvement itself as a discipline together with the set of necessary knowledge, practices, skills, tools, and infrastructure needed to bring in any process improvement project to success. Measuring and narrowing the performance gap won't be clear if the process improvement framework remains fuzzy.

This course focused on the Establishing, Acting, and Learning phase of the IDEAL model which is a high-level description of the phases of process improvement. At the beginning, some introductory concepts and guiding principles were introduced. The role of each of the Change Agent, the Management Steering Group (MSG), and the Technical Working Group (TWG) and the relation between all different groups was explained and discussed interactively all over the course. The discussion included the desired characteristic of each group regardless of the organization size. Which by then triggered a series of inquires most of them were about how efficient small organizations are capable of mapping the roles and functions of those multiple groups into their small organizations.

The Process Change Methodology (PCM) was introduced and explained in details. The PCM is divided into eight major steps each has a number of sub-activities that efficiently manage the change. Refer to the PCM Snap Shot to get the feeling. For more information visit the course information on the SEI website http://www.sei.cmu.edu/products/courses/master-process-improve.html.

The concept I enjoyed the most, is including the project adoption characteristics in the project partner selection criteria. Picking the appropriate project partner -to pilot the new solution with- becomes more accurate and the chance of successful institutionalization is higher. The idea of classifying the organization personal into (innovators, early adopters, early majority, late majority, and laggards) according to their adoption characteristics was really smart. Although it has its routes in marketing management, but it can be adopted also in evaluating and deploying the pilot solution.

Finally, the instructor went through a brief introduction to the Capability Maturity Model Integration (CMMI), explaining the difference between the staged and continuous representation for the CMMI. The benefit gained from having different representation is great, because it allow the organization to find its own way through the integrated model. An organization may choose to reach a higher maturity level as an objective for its process improvement endeavor. Another may choose to reach higher capabilities in specific function within its product life cycle. The idea seems to make a lot of sense, as the improvement effort can bring a number of tangible achievements sooner rather than later.

The following is a list of all course materials that were offered through out the course:

1. Clear and organized handouts
2. The exercises workbook: contain a number of exercises, guidelines, work aids, and reference aids.
3. The MPI Resources Folder Inventory: includes useful articles, presentations, templates, checklist, and other valuable guiding materials. The handouts, exercises, and resources folders are categorized according to the PCM Phases for simplifying cross-referencing.
4. The skilled facilitator book by Roger Schwarz:http://www.schwarzassociates.com/sfp.htm
5. The team performance book by David sibbet:http://www.grove.com/services/tool_modeltp.html
6. A complete sample of Software Engineering Process Group (SEPG) action plan.
7. Software Project Planning, Tracking and Oversight Report: A report that explains aspects of software project planning tracking and oversight as performed by one of the mature development centers.

The following list briefs the Process Change Methodology (PCM) activities necessary to implement process improvement in any organization.

1. Organize and Prepare.

The Process Groups PGs are formed, identifying their own set of processes and procedures that govern their activities and communicating it to the MSG.

2. Conduct Organizational Scan.

The PGs know more about there own organization areas of strength, how much projects vary, what is common across the organization projects, and what are the areas that could serve as starting points for any improvement efforts.

3. Establish Technical Working Groups.

The best possible team/teams that fit for carrying-out the improvement process in the area/areas identified by the PG/PGs is formed under the name of Technical Working Group (TWG)/TWGs.

4. Understand Project's Current State.

The TWG/TWGs starts by understanding the ongoing processes whether they are documented or not. Team members may need to get trained on how to gather, interpret, and realize what is happening and how dose it happen? The current processes are documented and a consensus is reached on how things are actually happing.

5. Redesign the Processes.

The TWG/TWGs begins by studying the current processes and carrying out the gap analysis. The new design for the process takes into account eliminating any non-value added activities or sub-activities and may add new necessary ones that were often overlooked or ignored.

6. Develop Solution.

The new process is part of the solution not all the solution. The complete solution includes different work products that are offered to project partners to ease the institutionalization of the process.

7. Conduct Pilots & Evaluate.

Next, the solution is piloted and the actual effectiveness of the new process is measured against the planned one. The pilot may end up by adopting the new process, altering and re-examining it again, or simply drop the solution.

8. Facilitate Organizational Learning.

The solution may then be institutionalized across the organization and its support infrastructure is established.

Finally, the new journey of improvement starts again by hunting for new improvements opportunities.

Biographies:

Omar Kamal has 7 years of experience in wireless telecommunications, software development, training, and software quality management. Omar is working as a senior software engineer in QuickTel Research and Development. He used to work with Lucent Technologies, Hewlett Packard, and Eitisalat. He holds a bachelor's degree in telecommunications engineering from Cairo University, and master's degree in business administration from city university. He is also Certified Quality Manager by the American Society for Quality.