Utilizing Usability Evaluating Model in Applying CMM to Improve the Quality of Software Maintenance Process

Amir Mohamed Talib
Faculty of Computer Science & IT, University Putra Malaysia
43400 UPM, Serdang, Selangor, Malaysia
E-mail:ganawa53@yahoo.com

Rusli Abdullah
Faculty of Computer Science & IT, Information System Department,
University Putra Malaysia
43400 UPM, Serdang, Selangor, Malaysia
E-mail: rusli@fsktm.upm.edu.my

Abstract

Maintenance plays an important role in the life cycle of a software product. It is estimated that there are more than 100 billion lines of code in production in the world. As much as 80% of it is unstructured, patched and not well documented. Maintenance can alleviate these problems. The purpose of this paper is to explore the use of the Capability Maturity Model (CMM) to improve the quality of software maintenance process (SMP). The architecture of the CMM model has been retained almost as is while its content, which was specific to the development process, has been either modified or extended to take into account the characteristics specified to the maintenance function, these characteristics were then organized into key process areas (KPAs) of CMM model. This paper applied the definition of (ISO 9241-11, 1998) that examines effectiveness, efficiency, and satisfaction. The emphasis will be given to the SMP activities

Keywords: Software maintenance, CMM, Software life cycle, Software maintenance process, Usability evaluating model, Software development

1. Introduction

Within the budget allocated to software in major corporations, the maintenance process is often taking the greatest share of software corporate resources: for many of these organizations, between 50% and 70% of the software budget is allocated to the maintenance process (Arthur, (1988)); (Swanson and Beath, (1989)); (Sharpe, (1991)). It is felt that software maintenance has not received it proportionate share of management attention and that it has suffered from lack of planning, thus resulting typically in crisis management. Within this context, software maintenance is perceived as expensive and ineffective.

The literature search has not come up with diagnostic techniques to evaluate the quality of the maintenance process of a given organization, nor to identify an improvement path (Arthur, (1988)); (Swanson and Beath, (1989)); (Sharpe, (1991)).

Evaluation models must support the following three management objectives:

- At the operational level they provide a detailed analysis and evaluation of a business process and of its key process.
- At the tactical level, they identify the strengths and weaknesses of each process as well as a progression path, should there be a decision taken within a continuous improvement process program. They also provide a map to develop an action plan to address the strengths and weaknesses within the set of organizational priorities and allocation of resources.
- At the strategic level, they provide to the senior executives the relative positioning of their organizations within their competitive environment. Based on this evaluation, priorities are then set, which lead to the allocation of scarce corporate resources to meet the corporate objectives.

2. Literature Review

2.1 Software Maintenance Process

(Pigoski, 2002) advice is to focus on a few key characteristics when measuring the software and not measure everything. It is better to use a few measures that are well understood rather than many measures that are not.

In an interview with (Swanson, 2006), he claims that not only software systems should be maintained but also the user knowledge of particular systems, (Kajko-Mattsson, 2001).

(Pfleeger, 2001) described a model of SM that includes measured feedback. It is an interesting approach for measuring the impact of a proposed change to decide what the risks are and eight various options and depicts the activities executed when a change is requested. The arrows with text coming from the rectangles pointing down, represents the measurements with information that can be used for decisions about when and how to make a change.

According to (Pigoski, 2007), over 50% of the software maintainer's time is spent (in order to perform corrections and enhancements) on understanding the program and that productivity can be improved vastly if this process is improved.

Further (Pigoski, 2007) demonstrates that corrective SM represents only 20% of the total maintenance effort. The remaining 80% is for enhancements, i.e. the adaptive and perfective categories of maintenance. The opinion is that maintenance costs are too high because of the concept that maintenance is only "fixing bugs or mistakes", but that is the small part of SM (20%).

(Pigoski, 2007) also implies that another reason for the high SM cost is that the needed items are seldom included initially in the development phase because of schedules or monetary constraints but are deferred until the operation-maintenance phase.

(Vaidyanathan, 2002) says that new development approaches (prototyping, incremental and spiral models) still need maintenance. Development continues after delivery of software and it is still subject to corrective and preventive maintenance and may be subject to perfective and adaptive maintenance. Systems are being fielded more quickly and the systems are modified quickly to accommodate user requirements.

2.2 CMM to improve the quality of Software Maintenance Process

(Diane, 2006) describes how CMMI can contribute to measurable improvements. It also may be useful for executives and senior managers who are faced with decisions about the allocation of scarce resources for improvement efforts.

In this study many of the organizations described that have achieved improvements in product quality and customer satisfaction also have achieved higher productivity, cost performance, and schedule performance. Better quality may not always be free, but it can occur with better project performance as a result of disciplined process improvement.

Brief by (Nicole McFarland, 2006) describe the overview of the Capability Maturity Model Integration (CMMI) infrastructure and concepts. CMMI is a process-improvement model for the enterprise that provides a set of best practices that address productivity, performance, cost, and stakeholder satisfaction when producing products or services. CMMI provides the practitioner with common terminology, common training and an integrated appraisal method. CMMI will tell you what to do, but not how to do it or who should do it.

(Carol, D and Barbara, E., 2002) studies the mutual effect of increased process maturity and an organization's maturity in their use of Function Point Analysis (FPA). As a company moves to a higher maturity level according to the Capability Maturity Model® Integration SM (CMMISM), its measurement maturity should also increase. The tie between the CMMI's process areas and FPA is not well understood, yet there is a direct connection that can be made between the model and FPA.

(Raffo, 1999) have provides a brief background on alternative quantitative approaches, describes the model currently being utilized by the SBMS Melbourne site, and provides an example of its use in analyzing a potential process change.

In order to justify and obtain management support for proposed process improvements, process analysis tools that support quantitative methods and address process level issues are needed.

(Viktor, C and Frank N, 2004) introduces the IT Service Capability Maturity Model (IT Service CMM®) and gives an overview of the model itself, and describes how to use it to improve the maturity of an IT service organization. Also describes the contents of the IT Service CMM, the main ways in which you can apply it to improve an organization, and the background you need to understand the IT Service CMM specification.

2.3 Usability Evaluating Model

(Virzi, 1992) described usability according to five basic parameters; 1) easiness and speed of learning of system use 2) efficiency to use 3) easiness to remember system use after certain period of time 4) reduced numbers of user errors and easy recovery from them and 5) subjective satisfaction of users. Many authors further the study of research in usability based on these 5 characteristics.

(Faulkner, 2005) further suggested that 15 users can uncover 90-97% of usability problems. (Carstens D. S., and Patterson P., 2005) mentioned that there should be specific usability guidelines for specific type of web application.

(Kostaras N. and Xenos M, 2004) suggested the heuristic evaluation method which is to be carried out by usability experts as it is easy to use, fast, relatively cheap and it can be employed in systems that are completed and fully operational. Heuristic evaluation is mainly based on rules of thumb by evaluators. Usability specialists will judge the compliance between user interface and usability principles called Heuristics either based on their own point of view or observation.

3. Overview of CMM model

The CMM was originally developed to assist the U.S. Department of Defense (DoD) in software acquisition. The rationale was to include likely contractor performance as a factor in contract awards. This model for determining likely contract performance also became a guide or framework for software process improvement. It used to judge the maturity of the software processes of an organization and to identify the key practices that are required to increase the maturity of these processes. It describes the principles and practices underlying software process maturity and is intended to help software organizations improve the maturity of their software processes in terms of an evolutionary path from ad hoc, chaotic processes to mature, disciplined software processes (Herbsleb, 1994).

3.1 Benefits of CMM

There is now substantial evidence of the business benefits of CMM-based software and a growing understanding of the factors that contribute to a successful improvement effort. In the past years, numerous empirical studies of organizations using CMM have been conducted by SEI and others. The most broadly supported claim is that CMM-based SPI has substantial business benefits for those moving from the initial through the defined levels (Herbsleb, 1997). Extensive studies have showed that organizations implementing CMM have achieved tremendous benefits in terms of productivity, calendar time, quality, and business value

The CMM represents a "common sense engineering" approach to software process improvement. The maturity levels, key process areas, common features, and key practices have been extensively discussed and reviewed within the software community. It represents a broad consensus of the software community and is a useful tool for guiding software process improvement efforts. The CMM provides a conceptual structure for improving the management and development of information systems products in a disciplined and consistent way. It can improve the likelihood with which a software organization can achieve its cost, quality, and productivity goals. However, CMM does not address all the issues of information systems development. Rigorously following the practices and procedures in CMM sometimes would make information systems development process too rigid and structured. Moreover, the overall model is too large and resource demanding for small businesses. The complexity also makes it difficult to tailor the model for specific project.

CMM cannot possibly fit the needs of all organizations equally and cannot be interpreted literally. An organization needs to use common sense when applying the CMM structure and enhance its own practices such that they meet the CMM goals” (Johnson and Brodman, 2000). It’s important for an organization to rigorously follow the key practices and procedures in CMM to achieve higher level of capability maturity level therefore increase its market competitive advantage. But at the same time, practitioners need to make their own judgments to flexibly implement the model to achieve the best result.

3.2 Quality in CMM model

Quality of software maintenance varies in different organizations at different CMM levels. There are many factors influencing the maintenance process, which in turn affect the final product quality. In an immature organization, there is no objective basis for judging product quality or for solving product or process problems. Therefore, product quality is difficult to predict. Activities intended to enhance quality such as reviews and testing are often curtailed or eliminated when projects fall behind schedule. When hard deadlines are imposed, product functionality and quality are often compromised to meet the schedule. In a mature organization, managers monitor the quality of the software products and the process that produced them. There is an objective, quantitative basis for judging product quality and analyzing problems with the product and process.

3.3 The KPAs of the CMM

Figure 1 (Note 1) lists the KPAs for each maturity level in the CMM. The CMM categorizes the overall company process maturity into five levels of maturity. The maturity levels, 1 to 5, indicate the overall effectiveness of the company's software engineering practices. Each increasing level is based on achieving the attributes of the previous low levels. The five levels are describing as follows (Paulk, 1993):

- Level 1: Initial – The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.
- Level 2: Repeatable – The project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on project with similar applications.
- Level 3: Defined – The software processes for both management and engineering activities is documented, standardized, and integrated into a standard software process for organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.
- Level 4: Managed – Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.
- Level 5: Optimizing – Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

Each maturity level has associated KPAs that describe the software engineering attributes that must be present to satisfy that particular level. As the maturity level increases, quality and productivity increase and risk of unsuccessful and unpredictable projects decreases.

By definition, KPAs are expressed at a single maturity level. There are, however, relationships between the key process areas, and improvements in a specific management or technical area need not be restricted to a single key process area. Figure 2 (Note 1) illustrates these relationships. Organizations may work on higher level KPAs before they have achieved lower level maturity levels, and attention must continue to be focused on lower level KPAs even when KPAs at higher maturity levels have been achieved.

The KPAs are categorized in Figure 2 (Note 1) into three broad categories: Management, Organizational, and Engineering processes. The Management process category contains the project management activities as they evolve from planning and tracking at Level 2, to managing according to a defined software process at Level 3, to quantitative management at Level 4, to innovative management in a constantly changing environment at Level 5. The Organizational process category contains the cross-project responsibilities as the organization matures, beginning with a focus on process issues at Level 3, continuing to a quantitative understanding of the process at Level 4, and culminating with the management of change in an environment of continuous process improvement at Level 5. The Engineering process category contains the technical activities, such as requirements analysis, design, code, and test, which are performed at all levels, but that evolve toward an engineering discipline at Level 3, statistical process control at Level 4, and continuous measured improvement at Level 5.

Note that at Levels 4 and 5 there are KPAs that span these process categories. This helps identify potential new KPAs for CMM v2 as Levels 4 and 5 become better understood.

4. Methodology

4.1 Utilized Usability Model

Usability is a multidimensional construct and can be assessed using various criteria. This project applies the definition of (ISO 9241-11, 1998) that examines effectiveness, efficiency, and satisfaction. (ISO 9241-11, 1998) defines usability and explains how to identify the information which is necessary to take into account when specifying or evaluating usability of a visual display terminal in terms of measures of user performance and satisfaction. Guidance is given on how to describe the context of use of the product (hardware, software or service) and the relevant measures of usability in an explicit way. The guidance is given in the form of general principles and techniques, rather than in the form of requirements to use specific methods.

The guidance in (ISO 9241-11, 1998) can be used in procurement, design, development, evaluation, and communication of information about usability. (ISO 9241-11, 1998) includes guidance on how the usability of a product can be specified and evaluated. It applies both to products intended for general application and products being acquired for or being developed within a specific organization.

(ISO 9241-11, 1998) also explains how measures of user performance and satisfaction can be used to measure how any component of a work system affects the whole work system in use. The guidance includes procedures for measuring usability but does not detail all the activities to be undertaken. Specification of detailed user-based methods of measurement is beyond the scope of (ISO 9241-11, 1998).

(ISO 9241-11, 1998) is considered as one of the most widely adopted and cited definitions of usability is that of the International Organization for Standardization, which identifies usability with the ability to use a product for its intended purposes: 'the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use (ISO 9241-11, 1998).

According to the benefits and importance of (ISO 9241-11, 1998), this project proposes an evaluation model for assessing usability of SMP. As reflected in the definition, three central criteria for usability are the effectiveness, efficiency and satisfaction with which users can achieve specified goals.

4.1.1 Effectiveness

The first criterion, effectiveness, suggests that specified goals are to be achieved with completeness (ISO 9241-11, 1998). Effectiveness can be understood as "how good a system is at doing what it is supposed to do" (Preece, Rogers and Sharp, 1992) and is related to the "utility" of the system (Grudin, 1992), and "to the extent to which the system provides the right kind of functionality so that users can do what they need or want to do" (Preece, Rogres, and Sharp, 1992). Effectiveness can also be defined as the accuracy and completeness with which users achieve certain goals.

Indicators of effectiveness include quality of solution and error rates. Moreover, quality of solution is used as the primary indicator of effectiveness, i.e. a measure of the outcome of the user's interaction with the system. In this project, effectiveness is to evaluate if the system as a whole can provide information and functionality effectively and is measured by how many answers are correct.

4.1.2 Efficiency

The second criterion, efficiency, suggests that specified goals are to be achieved with as little expenditure of resources as possible (ISO 9241-11, 1998). But in another way, measures of efficiency relate the level of effectiveness achieved to the expenditure of resources (Bevan, 1995).

According to (Bevan, 1995), resources may be 'mental or physical effort, which can be used to give measures of human efficiency, or time, which can be used to give a measure of temporal efficiency, or financial cost, which can be used to give a measure of economic efficiency'.

Moreover efficiency, which is the relation between (1) the accuracy and completeness with which users achieve certain goals and (2) the resources expended in achieving them. Indicators of efficiency include task completion time and learning time. In this study, we use task completion time as the primary indicator of efficiency. In addition, efficiency is likewise to evaluate if the system as a whole can be used to retrieve information efficiently and is measured.

4.1.3 Satisfaction

The third criterion, satisfaction, suggests that users should feel comfortable with, and have positive attitudes towards the use of the system (ISO 9241-11, 1998). In this sense, satisfaction relates to concepts such as ease of use, user satisfaction and usefulness (Davis, 1989) and (Mathieson and Keil, 1998). The 'specified context of use' includes users, tasks, equipment and the physical environment, where 'task' is defined in terms of activities required to achieve a goal (ISO 9241-11, 1998).

Satisfaction is the users' comfort with and positive attitudes towards the use of the system. Users' satisfaction can be measured by attitude rating scales. And can measure by use preference as the primary indicator of satisfaction.

Finally, Satisfaction looks into the areas of ease of use, organization of information, and visual attractiveness, and error handling and is measured by questionnaires. Ease of use is to evaluate user's perception on the ease of use of the system. Organization of information is to evaluate if the system's structure, layout, and organization meets the user's satisfaction. Labeling is to evaluate from user's perception if the system provides clear labeling and if terminology used are easy to understand. Visual attractiveness evaluates the site's design to see if it is visually attractive. Contents evaluate the authority and accuracy of information provide.

From previous studies, we found interlocking relationships among effectiveness, efficiency, and satisfaction. The longer it took to perform a task, the greater the number of steps involved. The greater the number of steps involved in completing a task, the lower the satisfaction. The more time spent on completing a task, the lower the satisfaction. Further, incorrect answers entailed more steps, while correct answers entailed fewer steps. This means that when we knew how to locate the answer, it take fewer steps. Although there are interlocking relationships, effectiveness, efficiency, and satisfaction are three distinct criteria and should be measured separately. One cannot replace the other.

For that a usability evaluating model has been developed in order to identify a comprehensive set of usability components needed to evaluate various aspects of is how the CMM KPAs improve the quality of SMP shown in Figure 3 (Note 1)

4.2 Respondents

The respondents of this study are two companies in Malaysia and these companies were already familiar with the SMP and its applications and the companies respondents was chosen to fill the questionnaire of this study.

4.3 Procedure

In the beginning, the respondents was received a short, scripted verbal orientation explaining the purpose of the usability evaluating. Then they were asked to complete a short background questionnaire to collect their demographic characteristics. The respondents were asked to perform a set of information via a questionnaire sheet (Note 3) about how to improve the quality of SMP using the CMM KAPs. The tasks were written on a sheet of paper that included a space where respondents were asked to indicate their answers. Once the tasks were completed, respondents were asked to complete a short respondent satisfaction questionnaire to collect and test their own perceptions towards SM.

4.4 Tasks

Respondents were complete three tasks:

1. They completed a background/experience questionnaire that *including name, gender, age, education level, Major/Department, and years of experience.*

2. They performed tasks using the questionnaire's sheet.

3. There was also a post-survey questionnaire that specifically examines CMM KAPs techniques. After completing a task, the respondents were asked to rank satisfaction and to write down comments.

4.5 Data Collection

This evaluation model considers both quantifying elements of performance (experience and experiment) as well as subjective empirical. If the answer is wrong, or he/she not familiar with this question then skip to the second question until all the question will be solved. However, the authors will record whether respondents are able to complete tasks successfully. The criteria for successful task completion are:

- Respondent is able to give a correct answer based on his own information about the system. Any guessed or assumed answers, whether correct or not, are not record as successfully completed tasks.

- Respondent is able to give a definite answer to the question. Where respondents indicated they are unsure about the answer or would seek clarification, the task will record as not successfully completed.

4.6 Questionnaires

The purpose of the questionnaire is to prove:

- Handle the interpretation of the term KAPs of CMM and the company's key objective in SM.

- Handle the aspects that come into play in KAPs, such as the existence of a strategy, the processes of quality control of data, the content that is being managed.

- Identify the KAPs technique of willingness of cooperation for research work.

Basically, there are two types of questionnaire that we prepared as part of usability evaluating for the respondents for the level of the questions.

4.6.1 Pre-Survey Questionnaire (background)

A series of questions designed to collect demographic information about the respondents to assess their level of his information about the system.

4.6.2 Post-Survey Questionnaire

After the test subject completed each scenario, he/she should answer a specific questions related to the tasks. To indicate whether the tasks was clear and completed successfully. After the test subjects complete all the scenarios, he/she will answer eighteen points.

5. Result

The result was conducted according to the methodology described in the previous section. It is starts with an overview of data collected by analyzing trends. Correlation was measured by Pearson's Correlation process of SPSS 16.0, the Quantitative and Qualitative analysis for CMM KPA's questions were described.

This study applies CMM model to evaluate and improve the quality of the SMP.

Whichever usability evaluation method was used, it is necessary to consider several kinds of usability factors in a systematic manner. Moreover, in the case of expert evaluation using analytical methods, comprehensive consideration of usability factors are very important to overcome its limitations.

Several types of data were collected to assess user's performance and user's perceptions, as follows:

- ✓ Effectiveness was measured by the number of tasks successfully completed.
- ✓ Efficiency was measured by the amount of time taken to complete all tasks.
- ✓ Satisfaction was measured by a rating scale for several satisfaction elements.

In addition, we divided the result based on two parts; Quantitative Analysis of CMM KPAs in Software Maintenance and Qualitative Analysis of CMM KPAs in Software Maintenance. Any task contained 9 questions (tasks).

5.1 Effectiveness

As we can notice all respondents were able to complete all the tasks. Effectiveness was measured by the number of tasks successfully completed. For other tasks respondents were able to complete in success percent ranging between (50% & 100%). The Successful task completion for the individual tasks is summarized in Figure 4 and Figure 5 (Note 2) below for the respondents. As we can see the average of successfully completion task are high, according to this results the successfully completion task that presented the effectiveness, achieved correctly. Moreover these results of the tasks successfully completed are high.

Figure 4 (Note 2) shows that Q2, Q6, Q8 and Q9 are completed answered successfully (100%) and the others questions got more than 70%.

Figure 5 (Note 2) shows that Q1, Q5 and Q9 are completed answered successfully (100%) and the others questions got more than 60%.

5.2 Efficiency

Completion time is the one factor used for measuring efficiency in this paper. Efficiency was measured by the amount of time taken to complete all tasks. An average of 44 minutes and 3.5 seconds per respondent was taken to complete the all tasks. However, there was much variation among the respondents, for example, the fastest respondent took only 18 minutes and the slowest took 37 minutes and 9 seconds which are about three times longer. Pearson's product-moment correlation analysis was conducted to see if the respondents' completion time is related. The results showed that total completion time is independent, (see Figure 6 and Figure 7) (Note 2).

As a result, efficiency was measured by evaluating completion time used in this survey by each respondent. Respondents who they were familiar with the systems in general tended to use less time to complete their tasks. When the respondent knows how to get the answer, it takes them fewer time while when they don't know how to use the system, they take more time.

5.3 Satisfaction

Respondents Satisfaction measured by using the two scales (YES= respondent agreed, NO= respondent not agreed). Satisfaction was measured by a rating scale for several satisfaction elements.

According to the result below, the satisfaction for the respondents were in moderate level.

Figure 8 (Note 2) shows the respondent satisfaction for Quantitative analysis for MAS. The Q2 and Q8 give high satisfaction (Mean=5.10) out of (Maximum =6) and Q6 give high satisfaction (Mean=4.70), and Q7 also give high satisfaction (Mean=4.40) out of (Maximum=5). All respondents feel satisfied with the system when they fail to perform the task correctly.

Figure 9 (Note 2) shows the respondent satisfaction for Qualitative analysis for MAS. Q8 give high satisfaction (Mean=5.40) and Q9 give high satisfaction (Mean=5.50) out of (Maximum =6) and Q7 give high satisfaction (Mean=4.20) out of (Maximum=5). 30% of respondents feel less satisfied with the system when they fail to perform the task correctly.

6. Comparison with the CMM Model

It can be seen at first glance the content of the CMM Model which was used as an input to the adaption process was extensively modified. When version 1.0 of the maintenance model is compared to the original CMM Model, it can be seen in table 1 (Note 3) that four of KPAs is unchanged, six have been modified and eleven have been added.

7. Conclusion

As there are various methods of developing software, different approaches of maintenance activities are adopted for each different development method. Because CMM focuses on changing the system to make it better able to adapt to user requirements throughout the software's lifetime, maintenance becomes easier. It is also good to use an iterative method for the maintenance phase because iteration is done for each change made to the system and each bug that is fixed. This allows for smaller and more frequent releases. This ensures that each added part of the product is fully working before moving onto the next change. Using smaller releases also makes it easier to code and debug each part. In addition, it allows the developers to have a working product at the end of each iteration. On the other hand, component based software design is built on the idea of developing software systems by selecting appropriate off-the-shelf components and then assembling them with a well-defined software architecture. The maintenance of such systems is done at the component level rather than at the source code level. Therefore, if a defect is found in a specific part, the maintainers can seamlessly replace that module with a working one. This is done easily by modifying the glue code. Finally, open source development allows each user to serve as a co-developer. This leads to the creation of better software and allows users to customize a program on their own. Any of these development methodologies can be used to make the maintenance phase more productive and efficient.

References

Arthur. L. J. (1988). "Software Evolution-The Software Maintenance Challenge" John Wiley & Sons

Bevan, N. (1995). "Measuring usability as quality of use Software Quality", *Journal of Usability Measuring,* 45: 567-575.

Carol, D and Barbara, E. (2002). "CMMI Capability Maturity Model Integration: What has changed?" The Journal of Defense Software Engineering, Vol, 15 No 2

Carstens D. S., Patterson P. (2005). "Usability Study of Travel Websites", Journal of Usability Studies, Issue 1, volume 1, 2005, pp. 47-61.

Davis, F.D. (1989). "Perceived usefulness, perceived ease of use, and user acceptance of information". MIS Quarterly, pp 318.

Diane, L. Gibson, D, Goldenson. D and Keith, K. (2006). "Performance Result of CMMI-Based Process Improvement" TECHNICAL REPORT, CMU/SEI-2006-TR-004, ESC-TR-2006-004.

Faulkner, L. (2005) "Beyond the five-user assumption: Benefits of increased sample sizes in usability testing". Behavior Research Methods, Instruments and Computers, 35(3), pp. 379-383.

Gurdin, J. (1992). "Groupware and Social Dynamic" *Eight Challenges for developers. Communications of the ACM 37, 1, pp. 92-105.*

Herbsleb, A. Carleton, J. Rozum, et al. (1994). *Benefits of CMM-Based Software Process Improvement: Initial Results*, retrieved November 26, 2007 from http://www.sei.cmu.edu/pub/documents/94.reports/94.tr.013.html

Herbsleb, J., Zubrow, David, D. and Goldenson, et al. (1997). "*Software Quality and the Capability Maturity model",* Communications of the ACM, 40, 6

International Standards Organization. (1998). ISO document 9241- Ergonomic requirements for office work with visual display terminals.

Johnson, D.L. and Brodman, J.G. (2000). "*Applying CMM Project Planning Practices to Diverse Environments,* Software", IEEE, Volume: 17 Issue: 4 , July-Aug, pp 40-47.

Kajko-Mattsson, M. (2001). “A Conceptual Model of Software Maintenance” Journal of Software Maintenance 14(3): 197-227

Kostaras N., Xenos M. (2004). “Assessing Educational Web-site Usability using Heuristic Evaluation Rules”, in 11th, Panhellenic Conference in Informatics, pp. 543-550.

Mathiesom. K. Keil. M. (1998). “Beyond the interface: Ease to use and Task/Technology Fit”, pp.579-592

Nicole McFarland. (2006). “Introduction to CMMI course participant” Version 1.1, Vol 3, No 1, pp 37-45

Paulk, Mark C. (1993). Report: “Capability Maturity Model for Software, Version 1.1”,

Pfleeger, S. L. (2001). “*Software engineering - theory and practice*” Upper Saddle River: Prentice Hall. NJ, USA. JCS&T Vol. 5 No. 1

Pigoski., T. (2002). “*Software Engineering – A practitioner’s approach*” IEEE Software archive, Vol 19 No 4, pp 100 - 106

Pigoski., T. (2007). “*Practical Software Maintenance*”, John Wiley & Sons. Vol 49 No 5, pp 515-529

Preece. J, Rogers. E, and Sharp. B.A. (1992). “Evaluating a teen preventing of Usability Web”, *Journal of Victorian Behaviour.* pp. 227-247.

Raffo, Vandeville, and Martin. (1999). “Software Process Simulation to Achieve Higher CMM Levels,” Journal of Systems and Software, Vol. 46, No. 2/3

Sharp, S, Haworth, D.A, et Hale, D. (1991). “Characteristics of Empirical Software Studies: 1980-1989”. Journal of Software Maintenance and Practice. Vol 3, PP. 1-5.

Swanson. (2006). “A review of past and current overall models for maintenance management”. *Quality in maintenance engineering journal,* Vol.6 No.2.

Swason. E.B, and Beath, CM. (1989). “Maintaining Information System in Organization” John Wiley & Sons. Vol 1 No 1, pp 47 - 58

Vaidyanathan. (2002). “Analysis of inspection-based preventive maintenance in operational software systems”, *Article in Reliable Distributed Systems, Proceedings.* 21st IEEE Symposium on, pp.286-295

Viktor, C and Frank, N. (2004). “IT Service CMM, a pocket guide” DTPresto Design & Layout, Zeewolde-NL, ISBN: 90-77212-35-3

Virzi, R. A. (1992). “Refining the test phase of usability evaluation”: How many subjects is enough? Human Factors, 34, pp. 457-468.

Notes:

Note 1.

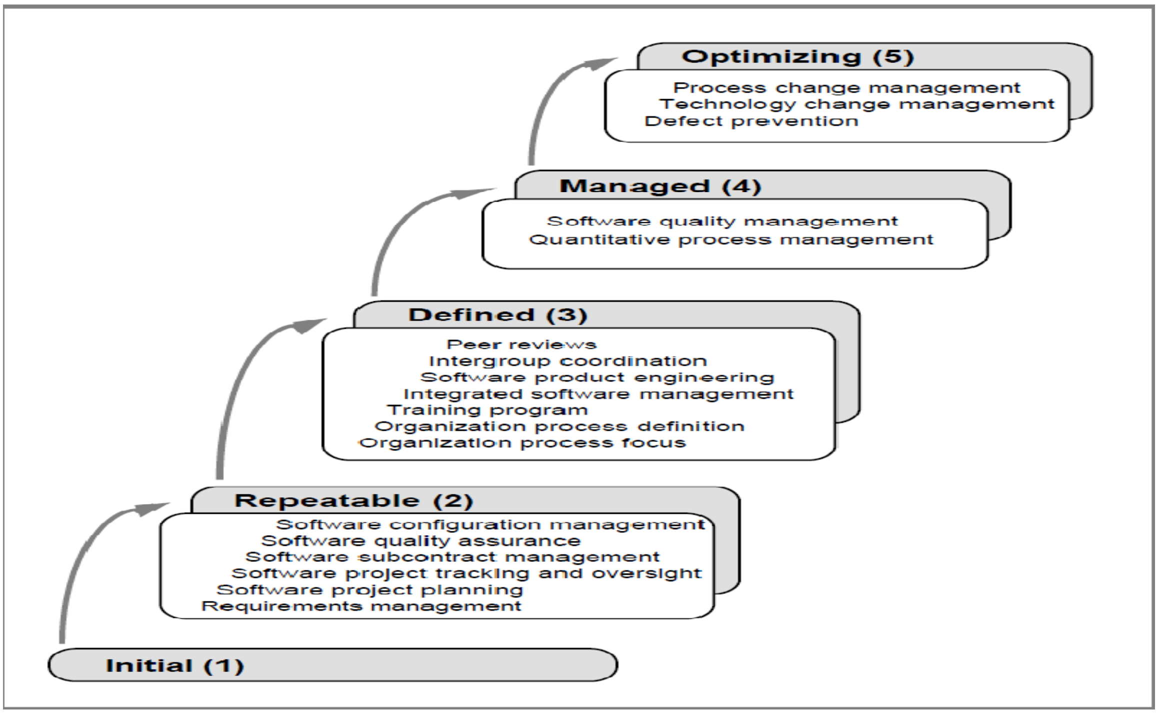
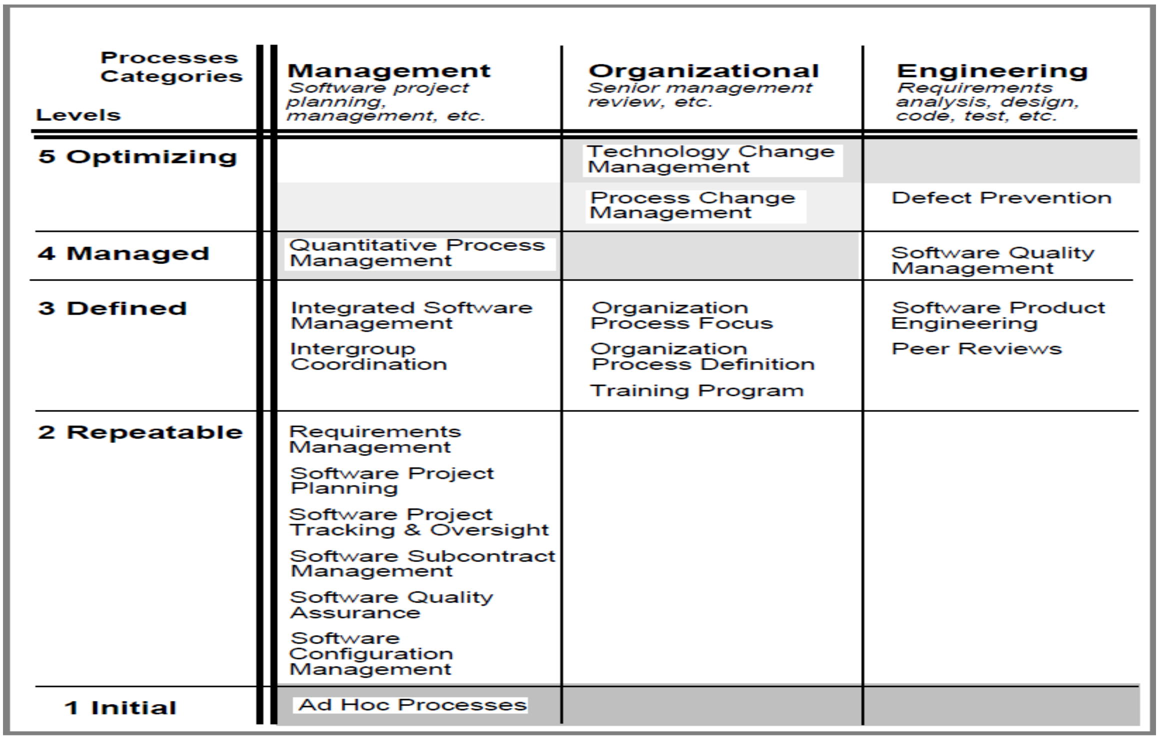

Optimizing (5)
Process change management
Technology change management
Defect prevention
Managed (4)
Software quality management
Quantitative process management
Defined (3)
Peer reviews
Intergroup coordination
Software product engineering
Integrated software management
Training program
Organization process definition
Organization process focus
Repeatable (2)
Software configuration management
Software quality assurance
Software subcontract management
Software project tracking and oversight
Software project planning
Requirements management
Initial (1)

Figure 1. The KPAs by Maturity Level



Processes Categories Levels	Management *Software project planning, management, etc.*	Organizational *Senior management review, etc.*	Engineering *Requirements analysis, design, code, test, etc.*
5 Optimizing		Technology Change Management Process Change Management	Defect Prevention
4 Managed	Quantitative Process Management		Software Quality Management
3 Defined	Integrated Software Management Intergroup Coordination	Organization Process Focus Organization Process Definition Training Program	Software Product Engineering Peer Reviews
2 Repeatable	Requirements Management Software Project Planning Software Project Tracking & Oversight Software Subcontract Management Software Quality Assurance Software Configuration Management		
1 Initial	Ad Hoc Processes		

Figure 2. The KPAs Assigned to Process Categories

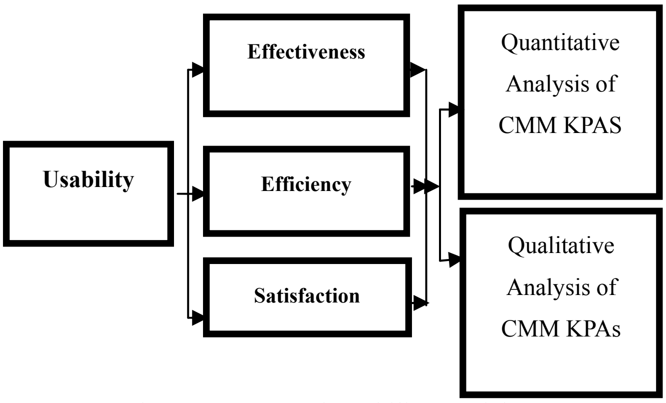

Usability
Effectiveness
Efficiency
Satisfaction
Quantitative Analysis of CMM KPAS
Qualitative Analysis of CMM KPAs

Figure 3. Proposed Usability Model

Note 2.

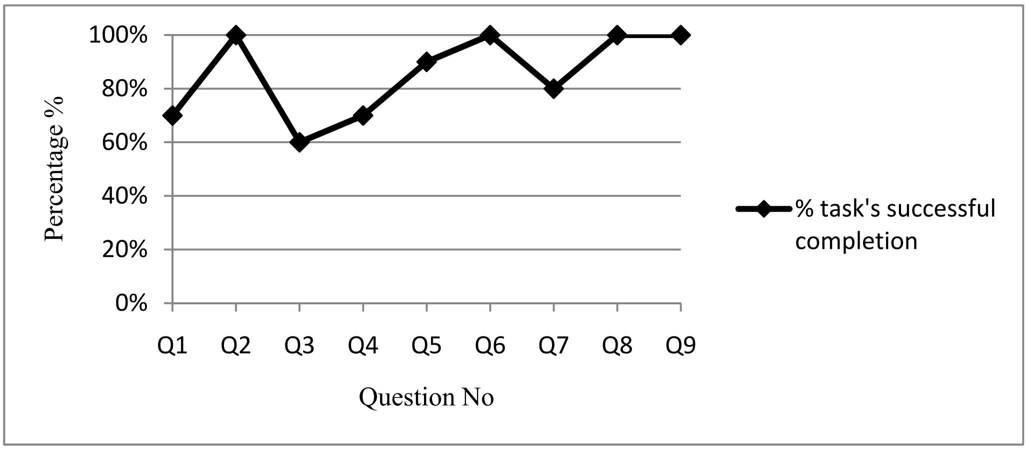


Figure 4. Completed Tasks Successfully for Quantitative analysis

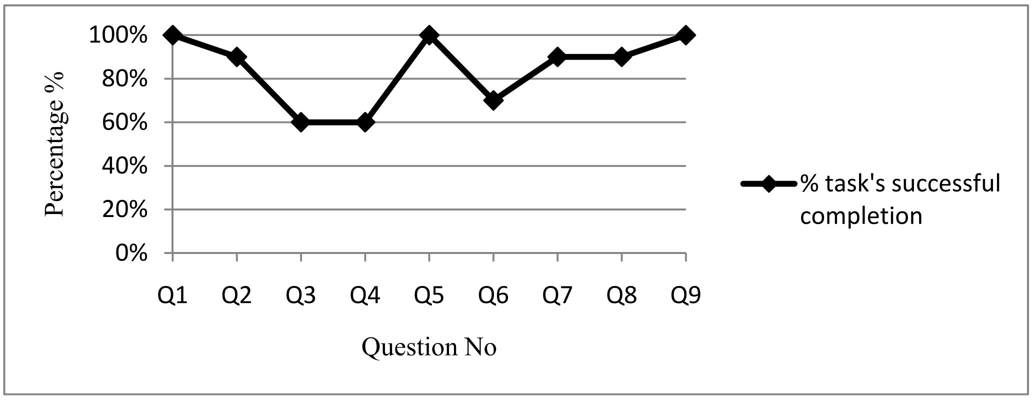


Figure 5. Completed Tasks Successfully for Qualitative analysis

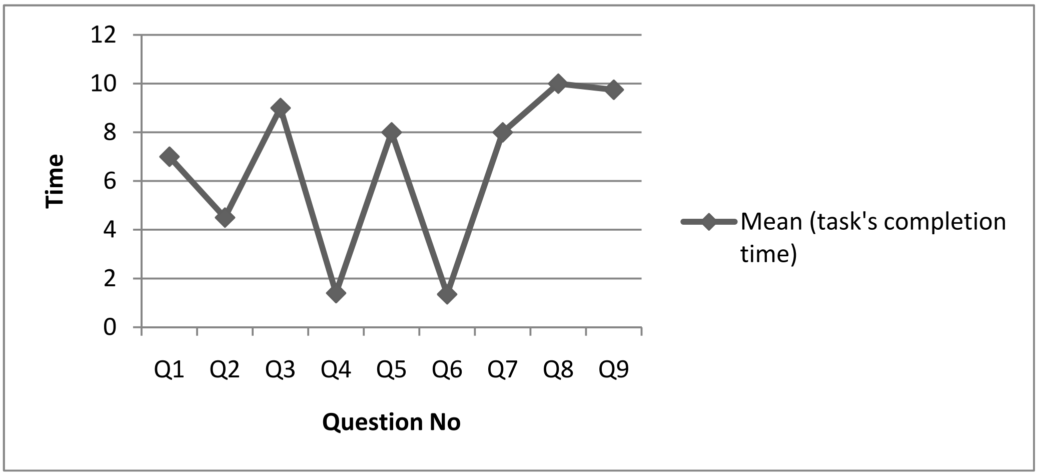


Figure 6. Time used to completed for Quantitative analysis

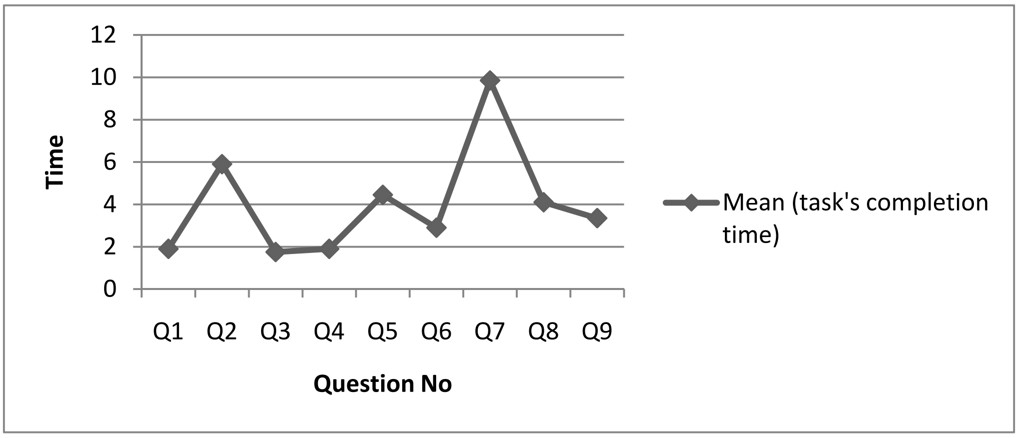


Figure 7. Time used to completed for Qualitative analysis

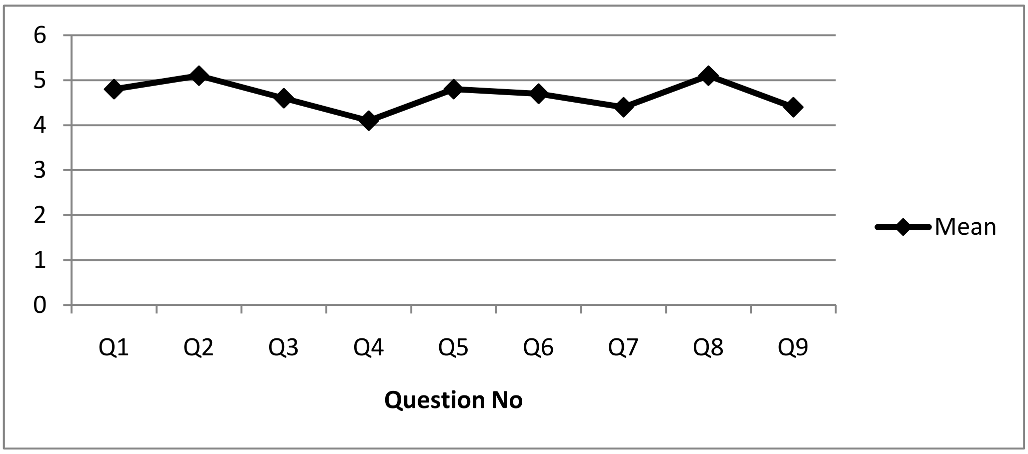


Figure 8. Respondent satisfaction for Quantitative analysis

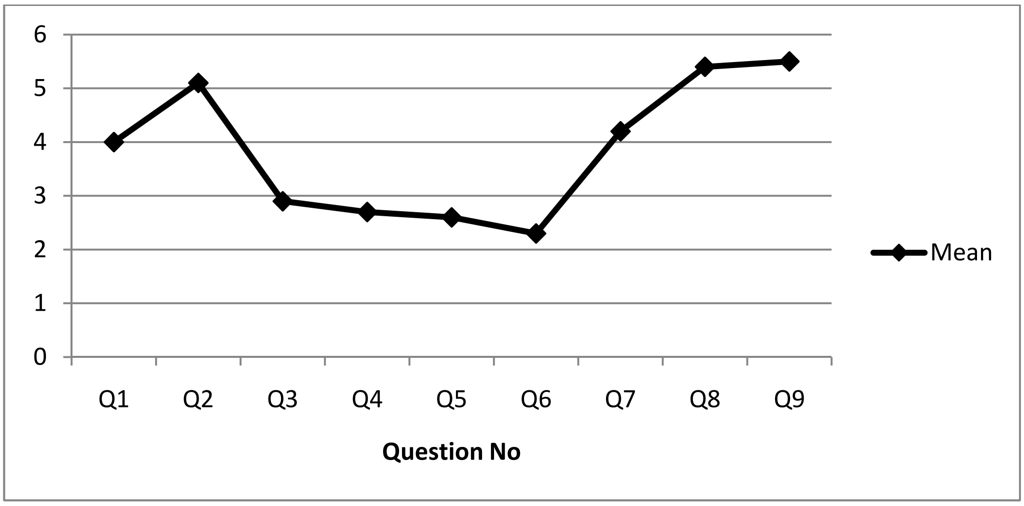


Figure 9. Respondent satisfaction for Qualitative analysis

Note 3:

Table 1. Comparison of KPAs and CMM Model

Level	Maintenance Kea Process Area (KPAs)	Key Practices for Our New CMM Model		
		Unchanged	**Modified**	**New**
Initial				
Repeatable	1. Requirement Management			x
	2. Software Maintenance Planning		x	
	3. Software Maintenance Tracking		x	
	4. Software Subcontract Management		x	
	5. Software Quality Assurance			x
	6. Software Configuration Management		x	x
	7. Transfer to Maintenance Management			
	8. Problem Management			x
	9. User Support Management			x
	10. Organization of Maintenance Function	x		
	11. Role of the Interviewing around the Maintenance			x
Defined	12. Organization Process Focus		x	
	13. Organization Process Function	x		
	14. Training Program	x		
	15. Integrated Maintenance Management			x
	16. Software Maintenance Portfolio engineering			x
Managed	17. Quantitative Management of the Process			x
	18. Software Quality Management		x	
Optimizing	19. Defect Preventive			x
	20. Process Change Management	x		
	21. Process Improvement Management			x
Total	**Twenty one KPAs**	**4**	**6**	**11**

We observed from table 1 above, some KPAs were derived from the CMM model and modified to map more closely to maintenance characteristics. Our own results support the validity of separate dimensions corresponding to maturity levels 2 and 3 respectively. The existence of an additional dimension may reflect the difference between organizational maturity as characterized by the CMM and process capability as characterized by ISO/IEC 15504. Note too that both the 15504 studies and our own suggest the existence a similar "Quantitative Process Implementation" construct. Such similarities based on disparate methods and data provide additional confidence about both sets of results. Although the KPAs at levels 4 and 5 map to the same dimension in our current analysis, we can safely conclude that each stage of the CMM corresponds to a dimension of organizational maturity and that the KPAs in each maturity level are good items for measuring the maturity of a common underlying construct.

QUESTIONNAIRE SHEET

Appendix A.1 Pre-Survey Questionnaire

Thank you very much for agreeing to participate in this experiment. All of your personal data that we collect will be entirely confidential. I would like to gather a bit of background information about u.

Respondent Name____

Gender: ______Male ______Female

Date________________________

How old are you? 20-29 30-39 40-49 50-59 60 or above

Level of education:

_____Certification Bachelor ________ Certification Diploma

__________Degree Postgraduate

Appendix A.2 Usability Evaluating Questions

The goal of this Survey to ensure that the CMM KPAs (Key process areas) will improve the quality of Software maintenance process

I will ask you a series of questions and would like you to think out loud while you look for the answer. Please remember that we are testing the effectiveness of the CMM and its KPAs and this is not a test of you. The whole test should take less than one hour. Thank you

Kindly be informed that this survey generalized for all the four part of the Software maintenance:

Corrective: a detected error does not block the normal operation of the system and the solution time is not critical.

Predictive: when new functionalities are added to the system.

Preventive: consists of the software modification to improve its maintainability and quality properties.

Adaptive: when the system will change its execution environment.

Description for How to Answer the Question:

Evaluation of the matrix: Assign yourself the following points for each

NA = 0, where 0 is doing nothing at all = NONE and

1 = Don't Know, Not Sure or Can't Say = NO

2 = Not Important or as Not been Addressed = MINIMALLY

3 = Partially Beneficial or somewhat Effective or Less Scope for Overall Improvement =

PARTIALLY

4 = Important or May not be effective but other associated necessary actions being taken =SUBSTANTIALLY

5 = Critical or already in place and effective = FULLY

Also, the scale can generally be summarized as follows for majority situations

'NA 1 2 3 4 5' is calibrated as in

'5 (Always) 4 (Often) 3 (Sometimes) 2 (Occasionally) 1 (Never)'

NA (Not Applicable), (Note: "NA" and "1" scale values are equivalent.)

QUESTIONNAIRE - Part One (Quantitative Analysis)

1. Does your organization follow a documented plan for developing and improving its software process in the whole process of Software maintenance?

NA 1 2 3 4 5

2. Does the standard software development process documentation describe the use of tools and techniques at any Software maintenance process project?

NA 1 2 3 4 5

3. Is Software maintenance process mechanism used for assessing existing designed and code for reuse in the new application?

NA 1 2 3 4 5

4. For each Software maintenance process project involving software development, is there a designed software manager?

NA 1 2 3 4 5

5. Are Software maintenance process prototype methods used in designing the critical performance elements of the software?

NA 1 2 3 4 5

6. Are internal design review standard applied in Software maintenance process project?

NA 1 2 3 4 5

7. Are statistics on software design errors gathered at any level of Software maintenance process project?

NA 1 2 3 4 5

8. Do people throughout your organization participate in software process improvement activities in Software maintenance process (e.g., on teams to develop software process improvements)?

NA 1 2 3 4 5

9. Does the Software Quality Assurance (SQA) function have a management reporting channel separate from the Software maintenance project management?

NA 1 2 3 4 5

QUESTIONNAIRE- Part Two (Qualitative Analysis)

1. Have your organization conducted a new technologies evaluated to determine their effect on quality and productivity at any level of the Software maintenance process?

Yes No

2. Are the Software maintenance process project's actual results (e.g., schedule, size, and cost) compared with estimates in the software plans?

Yes No

3. Does your organization follow a written organizational policy to meet its training needs in Software maintenance process project?

Yes No

4. Is a mechanism used for initiating error prevention actions at any level of Software maintenance process?

Yes No

5. Is review efficiency analyzed for any Software maintenance process project?

Yes No

6. Does the Software maintenance process project follow a written organizational policy for performing peer reviews?

Yes No

7. Are estimates (e.g., size, cost, and schedule) documented for use in planning and tracking the Software maintenance process project?

Yes No

8. Are the Software maintenance process project's actual results (e.g., schedule, size, and cost) compared with estimates in the software plans?

Yes No

9. Does your organization follow a written organizational policy to meet its training needs in Software maintenance process project?

Yes No

Appendix A.3 Post-Survey Questionnaire

Thanks again for participating in this experiment. This questionnaire gives you an opportunity to tell us your reactions to the system you used. Please circle a number on the scale to indicate your reactions. Thank you

QUESTIONNAIRE - Part One (Quantitative Analysis)

1. Is a documented procedure used for selecting subcontractors based on their ability to perform the work in the Software maintenance process?

NA 1 2 3 4 5

2. Is a mechanism used for error cause analysis at any level of Software maintenance process?

NA 1 2 3 4 5

3. Is a formal procedure used to make estimates of Software maintenance process size?

NA 1 2 3 4 5

4. Are defect prevention activities planned in the Software maintenance process project?

NA 1 2 3 4 5

5. Are the activities for managing software quality planned for the Software maintenance process project?

NA 1 2 3 4 5

6. Does the system requirement in the Software maintenance process activities allocated to software change; are the necessary adjustments to software plans, work products, and activities made?

NA 1 2 3 4 5

7. Is the project planned and managed in accordance with the project's defined Software maintenance process?

NA 1 2 3 4 5

8. Are software configuration management activities planned for the Software maintenance process project?

NA 1 2 3 4 5

9. Are automated tools used to analyze the complexity of Software maintenance process?

NA 1 2 3 4 5

QUESTIONNAIRE- Part Two (Qualitative Analysis)

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

Yes No

2. Are target computer memory utilization estimates and actual tracked at any Software maintenance process project?

Yes No

3. Has your organization developed, and does it maintain, a standard Software maintenance process?

Yes No

4. Are Software maintenance process prototype methods used in designing the critical elements of the machine interface?

Yes No

5. Are automated tools used to analyze cross references between modules in Software maintenance process?

Yes No

6. Does the system requirement in the Software maintenance process activities allocated to software used to establish a baseline for software engineering and management use?

Yes No

7. Does SQA (Software Quality Assurance) provide objective verification that Software maintenance process and activities adhere to applicable standards, procedures, and requirements?

Yes No

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

Yes No

9. Does the organization follow a plan for managing technology changes in the Software maintenance process project?

Yes No

Comments: ____

____.