SM^{CMM} MODEL TO EVALUATE AND IMPROVE THE QUALITY OF THE SOFTWARE MAINTENANCE PROCESS

Alain April[1], Alain Abran[1], Reiner R. Dumke[2]
[1]École de Technologie Supérieure, Montréal, Canada
[2]Otto von Guericke University of Magdeburg, Germany
aapril@ele.etsmtl.ca , aabran@ele.etsmtl.ca , dumke@ivs.cs.uni-magdeburg.de

Abstract

*Software maintenance function suffers from a scarcity of management models that would facilitate its evaluation, management and continuous improvement. This paper is part of a series of papers that presents a Software Maintenance Capability Maturity Model (**SM**^{CMM}). The contributions of this specific paper are: 1) to describe the key references of software maintenance; 2) to present the model update process conducted during 2003; and 3) to present, for the first time, the updated architecture of the model.*

1. Introduction

Maintenance still suffers today from a scarcity of best practice proposals that can readily be applied in the industry. Aside from the Kajko-Mattsson [*Kaj01e*] corrective maintenance evaluation model, a large number of software maintenance best practices still need to be recognized and better described for technology transfer to the industry at large. It is felt that software maintenance still does not receive an adequate share of management attention and that it is suffering from lack of planning, as illustrated by its typically crisis management style and, within this context, it is still perceived as being expensive and ineffective.

For the software development function, there already exist many management models for evaluating the quality of the development process and proposing improvements. However, for the software maintenance function, there is no published comprehensive model that takes into account the specific characteristics of the maintenance process. Recognizing the importance of software maintenance and the limitations of process assessment models, which emphasize development over maintenance, an initial draft of a comprehensive maintenance evaluation model was published in 1996 [*Zit96*]. This paper presents the updated Software Maintenance Capability Maturity Model – ***SM***^{CMM} – as well as documenting, and providing traceability to, the contributions of other models.

Section 2 presents the findings and contributions from an extended literature review. Section 3 presents a discussion of what is missing in the CMMi to reflect a maintainer's point of view. Section 4 describes the model components, architecture and the steps followed to create this new version. Finally, work in progress and conclusions are presented in section 5.

2. Prior Contributions

2.1. Researchers' contributions

The literature review has not revealed any comprehensive diagnostic techniques for evaluating the quality of the maintenance processes of an organization, or for identifying improvement paths. Table 1 presents an inventory of the recent software engineering process evaluation and assessment models. Each of these models where analyzed to find specific and detailed contributions that could help maintainers in general. Out of the thirty-four proposed models in this inventory, only a handful (shown in **bold** in Table 1) offer publicly available maintenance practices, which can be useful to the software maintainers' specific context. However, none of these models cover the entire set of concepts specific to software maintainers, as documented in the Guide to the Software Engineering Body of Knowledge (SWEBOK) [*Abr01*].

The second version of the ***SM***^{CMM} introduces a much larger number of mappings to: a) standards; b) relevant software engineering CMM proposals; and c) recognized software maintenance references. From these mappings, a large number of detailed best maintenance practices have been identifies and included.

Table 1: Software Engineering CMM proposals, sorted by year of publication

Year	Software Engineering CMM proposals
1991	Boo91
1992	Tri92
1993	Sei93
1994	**Cam94**, Kra94
1996	Bur96, **Zit96**, Dov96
1997	Som97
1998	Esi98, Top98, Baj98
1999	Wit99, Vet99, Sch99
2000	**Cob00**, Str00, Bev00, Lud00
2001	**Kaj01d & 01e**, Ray01, Sch01, Luf01, Tob01, Sri01
2002	**Sei02**, **Nie02**, Mul02, Vee02, Pom02, Raf02, Sch02, Ker02, Cra02

The key mappings and references used are:

- Software maintenance standards ISO12207 [*Iso95*], ISO14764 [*Iso98*] and IEEE1219 [Iee98];
- The most widely recognized quality models ISO9001: 2000 [*Iso00*] and the CMMi [*Sei02*];
- The process evaluation model standard ISO/IEC TR 15504 (SPICE) [*Iso98a*].

The revised **SM^{CMM}** also includes inputs from, and references to, other maturity models and best practices publications, which tackle a variety of software maintenance-related topics. The intention is to have a unique source, which will reference other references when needed:

- Cm3-Corrective Maintenance Maturity Model [*Kaj01d*];
- Cm3-Maintainer's Education and Training Maturity Model [*Kaj01e*];
- ITIL Service Support and Service Delivery best practices [*Iti01*];
- IT Service CMM [*Nie02*];
- CobIT [*Cob00*];
- Malcolm-Baldrige [*Mal03*];
- CAMELIA Maturity Model [*Cam94*];
- **SM^{CMM}** version 1 [*Zit96*].

Some of the **SM^{CMM}** improvements have been documented in [*Apr02*], and had been implemented in the *Camélia* model initially developed by Bell Canada and Nortel. Another refinement is derived from the CMMi [*Sei02*] adoption of the continuous representation, which in turn can be traced back to its successful use in the past by other models, such as: Bootstrap [*Boo91*], *Camélia* [*Cam94*] and ISO/IEC TR 15504 (Spice) [*Iso98a*]. These improvements to the **SM^{CMM}** have provided the following benefits: a) conformity to SPICE recommendations; b) a more granular rating for each roadmap and domain; and c) identification of specific practices across maturity levels, together with a path from level zero (absent) to a higher level of maturity. Furthermore, the **SM^{CMM}** has been aligned to the CMMi model and to many of the best practices documented in the software maintenance literature.

3. SM^{CMM} and CMMi model

The initial version of the **SM^{CMM}** model included only two references: (a) [*Swa89*]; and b) [*Ball90*]) in its literature review. Version 2 of the **SM^{CMM}** has benefited from a much larger number of references, each carefully reviewed to ensure a wider and more representative coverage of the software maintenance literature. This review has also confirmed that some maintenance processes are unique to maintainers and are not present in the software development function (see Table 2).

When these key processes are compared to the CMMi model content, it can be observed that the CMMi model, being highly centered on software development, does not explicitly address these aspects of software engineering. With its primary focus on project management, it does not explicitly address the issues specific to the software maintenance function [*Apr03*].

Table 2: Software management key process areas (P = present, A = absent)

Some Maintenance Key Processes	Software management (maintenance)	Software development (creation)
Management of problems (Problem resolution interfacing with a help desk)	P	A
Acceptance of the software	P	A
Managing transition from development to maintenance	P	A
Service Level Agreements (SLAs)	P	A
Maintenance planning activities (versions, SLAs and impact analysis)	P	A
Event and Service Request Management	P	A
Software management (operational support)	P	A
Software rejuvenation	P	A

For example, in the CMMi:

- The concept of maintenance maturity is neither recognized nor addressed;
- There is not sufficient inclusion of maintenance-specific practices as process improvement mechanisms;
- Maintenance-specific issues are not adequately addressed;
- Rejuvenation-related plans such as the need for re-documentation, re-documentation, reengineering, reverse engineering, software migration or retirement are not satisfactorily addressed.

Depending of the source of the maintenance requests, maintenance activities are handled through distinct processes. This is illustrated in Table 3, with some examples. For each request source, a maintenance service/process is initiated, together with due registration of the related maintenance work categories. For example, when users are the source of requests, then a change request related to operational use of the software and the work to be carried out can be classified into one of three maintenance categories: corrective, adaptive and operational support. In some instances, a supporting process will be needed. A typical one is the need for service level agreement information as part of the operational support activities.

The absence in the CMMi of some of the specific processes used by the maintainers in everyday situations has been documented as far back as 1996 [*Zit96*] and this is still valid with the new CMMi version, since it maintains a developer's view of the software production process. We have also presented [*Apr04*] that the use of CMMi with small maintenance groups creates specific difficulties as the CMMi has omissions and gaps concerning specific software maintenance processes and activities.

4. SM^{CMM}: components and architecture

Maintainers, more than ever, need support in, and guidance on, improving their own processes, and a maturity model can help maintenance organizations of all sizes, maturity levels and using all types of methodologies (i.e. Xtreme maintenance) [*Apr04*].

The ***SM**^CMM^* model remains flexible and uses a continuous representation (see Figure 1) for its new version 2. New additions are: a) a grouping of generic and specific goals together and b) the addition of roadmaps. Roadmaps are an additional categorization of processes within a specific domain.

Figure 1: SM^{CMM} high-level model architecture

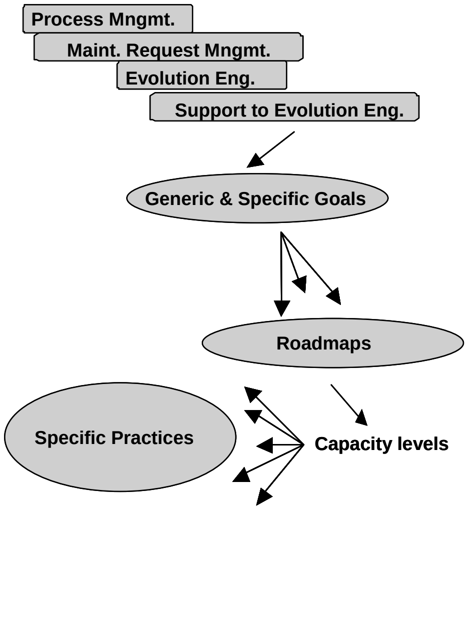

Table 3: Activities and categories of maintenance work

Source of Requests	Example of a Key Maintenance Service/Process	Assignment to a Maintenance Category of service for effort collection
Project Managers	Management of the transition from development to maintenance	Operational Support for project
Project Managers	Provide knowledge of existing legacy systems	Operational Support for project
Users	Ask for a new report or complex query	Operational Support for users
Users	Ask for new functionality	Adaptive
Users	Report an operational problem	Corrective
Users	Quarterly account management meeting with the customer	Operational Support for users + Service-Level Agreement
Software Operations	Change to a systems utility	Perfective
Rejuvenating Studies	Software impact analysis	Leads to a project, major enhancements or re-development which is outside the scope of daily maintenance

The new architecture of the ***SM***^{CMM} includes an overview of the strategy and the approach developed to update content specific to the maintenance function, as well as the steps taken to map them into the framework of a maturity model which complements the CMMi.

4.1 Model Update Process

A previous publication [*Apr04*] describes how the participating organizations updated the content of version 2 of the ***SM***^{CMM}. In summary, a 9-step process is executed, which uses key software maintenance references, one by one, and proceeded to map each relevant practice to the maturity model. This task has been quite time-consuming, taking more than a year to carry out. This approach often required restructuring of the overall model whenever a new process (or practice) could not be fit into the classification provided by the evolving model. Reordering, renaming or creating detailed roadmaps within a key process area had numerous editing impacts. One component, which evolved significantly, is the description of the process characteristics. We are currently attempting a better alignment with the ISO 15504 (SPICE) requirement that processes need specific attributes to help with the assessment. Therefore, an important step was to establish generic attributes, which characterize a specific level of maturity.

Version 2 of the ***SM***^{CMM} currently includes a total of 4 software maintenance process domains, 18 key process areas (KPAs), 74 roadmaps and 443 practices [see Apr04 for details]. As discussed previously, some key process areas are aligned with the CMMi model and slightly adapted to reflect the software maintenance specific characteristics.

4.2 Work in progress

Six participating companies of the telecommunications industry have used the model. During 2003 more than 50 software maintenance practitioners have helped improve and review this new version. While the initial version (1) of the model included only two references ([*Swa89*]; [*Ball90*]), the new ***SM***^{CMM} (version 2) benefits from a much larger number of references, each of which reviewed to ensure a wider and more representative coverage of the software maintenance literature. We have also used this inventory to supplement the current software maintenance body of knowledge proposed by the new version of SWEBOK [*Apr03*]. Further research work is required to improve version 2. For example, it would be most appropriate and useful to analyze the results of empirical studies in order to make adjustments to the model. This would ensure that key practices suggested by maintenance experts or described in the literature are positioned at the correct level within the maintenance evaluation model. Empirical studies could also be set up to study the efficiency of the model as a tool for continuous improvements in maintenance management. The empirical studies would contribute to a better understanding of the problems of the software maintenance function and in the validation of the proposed model.

5. Conclusion

This article has presented a second version of a software maintenance model developed to evaluate and improve the quality of the maintenance process. This model is based on the model developed by the SEI of Carnegie Mellon University in Pittsburgh to evaluate and improve the process of software development. To identify key differences between the development and the maintenance functions, larger seminal works in the literature were reviewed. This made it possible to subsequently enhance the software maintenance model that maps the characteristics of the software maintenance function to the model structure. The research work reported here involved identifying, describing, structuring, modeling and inserting maintenance-specific components.

6. References

[Abr01] Abran, A., and J.W. Moore Executive Editors; P. Bourque and R. Dupuis: Editors; L.L. Tripp Chair of IAB, "Guide to the Software Engineering Body of Knowledge - Trial Version," IEEE Computer Society Press, Dec. 2001.

[Apr02] April, A., D. Al-Shurougi, (2002). “The Role of Quality Maintenance in Cost Minimisation,” Software Maintenance Productivity Conference, ICB/ASAY, Bahrain, May 27-28.

[Apr03] April, A., A. Abran, P. Bourque (2003). Analysis of the knowledge content and classification in the SWEBOK chapter: Software Maintenance, Position paper accepted for the 11th International Workshop on Software Technology and Engineering Practice – STEP 2003, Amsterdam Sept. 2003.

[Apr04] April, A., A. Abran, Dumke, R; "Assessment of Software Maintenance Capability: A model and its Design Process," IASTED 2004 Conf, on Software Engineering, Innsbruck (Austria), Feb. 16-19, 2004.

[Baj98] Bajers, F. (1998). How to introduce maturity into software change management, Technical report R98-5012, Department of Computer Science, Aalborg University, Denmark.

[Bal90] Ball, R.K. (1990) Software Maintenance / Management from The Seminar of Software Management Institute, North York, Ontario, April.

[Bev00] Bevans, N. (2000). Serco Ltd. *Introduction to Usability Maturity Assessment*, [On-line]. www.usability.serco.com (tested on 11-11-02).

[Boo91] Bootstrap (1991). Esprit project #5441, European Commission, Brussels, Belgium.

[Bur96] Burnstein, I., et al., (1996). Developing a Testing Maturity Model: Part II, Crosstalk, September issue, [On line]. http://www.improveqs.nl/pdf/Crosstalk%20TMM%20part%202.pdf (tested on 11-09-03).

[Cam94] Camélia. (1994). Modèle d’évolution des processus de développement, maintenance et d’exploitation de produits informatiques, Projet France-Québec, Version 0.5.

[Cob00] IT Governance Institute, (2000). CobiT, Governance, Control and Audit for Information and Related Technology, 3rd Edition, July issue.

[Cra02] Crawford, J.K. (2002). Project Management Maturity Model, Providing a proven path to project management excellence, Marcel Dekker/Center for business practices.

[Dov96] Dove, R., Hartman, S., Benson, S., (1996). A Change Proficiency Maturity Model, An Agile Enterprise Reference Model with a Case Study of Remmele Engineering, Agility Forum, AR96-04, December.

[Esi98] European Software Institute (ESI), TeleSpice & R-Spice, [On line]. www.esi.es (Tested 15-11-02).

[Iee98] IEEE Std 1219, (1998). Standard for Software Maintenance, IEEE .

[ISO95] ISO/IEC 12207, (1995). Information Technology – Software Life Cycle Processes, ISO.

[ISO98] ISO/IEC 14764, (1998). Software Engineering-Software Maintenance, International Organization for Standardization.

[ISO98a] ISO/IEC TR 15504-2, (1998). Information Technology – Software Process Assessment – Part 2 A reference model for process capability , International Organization for Standardization.

[Iso00] ISO9001:2000, (2000). Quality Management Systems – Requirements, International Organization for Standardization, Third edition, December 15.

[Iti01] Information technology Infrastructure Library, (2001). Central Computer and Telecommunications Agency, Service Support 09/2000 and Service Delivery 04/2001, HSMO Books, London, UK.

[Kaj01d] Kajko-Mattsson, M., (2001). Corrective Maintenance Maturity Model, report 01-015, Stockholm University.

[Kaj01e] Kajko-Mattsson, M., S. Forssander, U. Olsson, (2001). Corrective Maintenance Maturity Model: Maintainer’s Education and Training, in Proceedings, International Conference on Software Engineering, IEEE Computer Society Press.

[Ker02] Kerzner, H. (2002). Strategic Planning for Project Management Using a Project Management Maturity Model, John Wiley & Sons.

[Kra94] Krause, M.H. (1994). Software – A Maturity Model for Automated Software Testing, Medical device & Diagnostic Industry Magazine, December issue.

[Lud00] Ludescher, G., Usrey, M.W. (2000). E-commerce Maturity Model, Proceedings of the 1st International Research Conference on Organizational Excellence in the Third Millennium, R.Edgeman editor, Estes Park, CO, August 2000, 168-173.

[Luf01] Luftman, J. (2001). Assessing Business-IT Alignment Maturity, Communications of AIS, 4(2).

[Mal03] Malcolm Baldrige National Quality Program. (2003) Criteria for Performance Excellence. [On-line] http://www.quality.nist.gov/ (Tested on 8-01-03).

[Mul02] Mullins, C. (2002). *The Capability Model – from a data perspective*, The Data Administration Newsletter, [On line]. www.tdan.com/i003fe04.htm, (Tested on October 30, 2003).

[Nie02] Niessink, F., V. Clerk, H van Vliet, (2002). *The IT Service Capability Maturity Model*, release L2+3-0.3 draft, [On-line]. http://www.itservicecmm.org/doc/itscmm-l23-0.3.pdf (Tested on 15-11-03).

[Pom02] Projxsoft, *Project Organization Maturity Model* (POM2), [On-line]. http://www.projxsoft.com/default.asp?nc=2053&id=4 (Tested on 11-10-03).

[Raf02] Raffoul, W. (2002). The Outsourcing Maturity Model, Meta Group, [On-line]. http://techupdate.zdnet.com/techupdate/stories/main/0%2C14179%2C2851971-2%2C00.html#level1 (Tested on October 9, 2003).

[Ray01] Rayner, P., et al. (2001). *The Programme Management Maturity Model*, The PM Group, February 15, [On-line].http:// www.eprogramme.com/events/pmmm.htm(Tested on 15-10-13).

[Sch99] Schmietendorf, A., Scholz, A. (1999). The Performance Engineering Maturity Model at a glance, Metrics News, 4(2), December issue.

[Sch01] Scheuing, A.Q., Fruhauf, K. (2000). Maturity Model for IT Operations (MITO), [On-line]. www.software.saq.ch , (Tested on December 16, 2002).

[Sch02] Schlichter, J. (2002). *PMI Organizational Project Management Maturity Model*, [On-line]. http://www.pmi.org/prod/groups/public/documents/info/pp_opm3.asp (Tested on October 11, 2003).

[Sei93] Software CMM, (1993). Version 1.1, CMU/SEI-93-TR-24, ESC-TR-93-177, Software Engineering Institute, Carnegie Mellon University.

[Sei02] Capability Maturity Model Integration for Software Engineering (CMMi), (2002). Version 1.1, CMU/SEI-2002-TR-028, ESC-TR-2002-028, Software Engineering Institute, Carnegie Mellon University.

[Som97] Sommerville, I., Sawyer, P. (1997). Requirements Engineering: A Good Practice Guide, John Wiley & Sons.

[Sri01] Sribar, V., Vogel, D. (2001). The Capability Maturity Model for Operations, Metagroup, [On-line] http://www.metagroup.de/cgibin/inetcgi/jsp/displayArticle.do?oid=15213(Tested on October 10, 2003).

[Str00] Stratton, R.W. (2000). The Earned Value Management Maturity Model, [On-line]. http://www.mgmt-technologies.com/evmtech.html (Tested on October 8, 2003).

[Swa89] Swanson, E.B., Beath, C.M. (1989). Maintaining Information Systems in Organizations. John Wiley & Sons.

[Tob01] Tobia, E., Glynn, M. (2001). E-business, can we control it? , E-business Maturity Model, 14th Utility Coal Conference, PricewaterhouseCoopers, [On line]. www.utilitycoalconference.com (Tested on 10-11-02).

[Top98] Topaloglu, N.Y. (1998). Assessment of Reuse Maturity, 5th International Conference on Software Reuse, Victoria, BC, June 2-5.

[Tri92] Trillium, (1992). Model for the Telecom Product Development & Support Process Capability, Bell Canada, version 2.2, large distribution for the first time.

[Vee02] Veenendaal, V., Swinkels, R. (2002). Guideline for testing maturity: Part 1: The TMM model, in: Professional Tester, Vol. 3, Issue 1, [On-line]. http://www.improveqs.nl/tmmart.htm (Tested on October 8, 2003).

[Vet99] Vetter, R. (1999). The network maturity model for Internet Development - IEEE Computer, 132(10), 117-118.

[Wit99] Wichita State University, (1999). Enterprise Engineering Presentation, Capability Model of BPR, course IE80I.

[Zit96] Zitouni, M., Abran, A. (1996). A Model to Evaluate and Improve the Quality of the Software Maintenance Process. in 6th International Conference on Software Quality Conference. Ottawa: ASQC-Software Division.