Analysis of the knowledge content and classification in the SWEBOK chapter: Software Maintenance

(Alain April[1], Alain Abran[1], Reiner R. Dumke[2])

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

Technical Report 03-001 of ÉTS Software Engineering Laboratory, 2003.

1. Introduction

This document presents an analysis and proposed amendments to the Trial version of SWEBOK Guide describing software maintenance. The scope of the proposed amendments is to broaden the set of maintenance topics identified to improve coverage of the body of knowledge on software maintenance, as well as to recommand additional references.

Figure 1 illustrates the second version [*5*] of a Software Maintenance Capability Maturity Model (SM-CMM) that was initially published in 1996 [*14*], keeping many of the attributes and features of the first version of the model and modernizing it with the recent appearance of the CMMi. This new version of the SM-CMM model presents four (4) software maintenance process domains and nineteen (19) software maintenance process areas. It is strongly aligned to the CMMi to facilitate its use and to enable parallels to be drawn where maintenance processes refer to development processes [*8*], or when they share some similarities.

4 Process domains of software maintenance	Key Process Areas of Software Maintenance
Process Management	1- Maintenance Process Focus 2- Maintenance Process/Service definition 3- Maintenance Training **4- Maintenance Process Performance** 5- Maintenance Innovation and deployment
Maintenance Request Management	1- Event and Service Request Management 2- Maintenance Planning 3- Monitoring & Control of maintenance events and service request 4- SLA & Supplier Agreement Management 5- Quantitative Maintenance Management
Software Evolution Engineering	1- Software Transition 2- Software Support 3- Software Evolution & Correction of software 4- Software Verification and Validation
Support to Software Evolution Engineering	1- Software Configuration Management 2- Process and Product Quality Assurance 3- Measurement, Decision and Causal Analysis 4- Causal Analysis and Problem Resolution 5- Software Rejuvination and retirement of software

Figure 1.0 : Key process areas of SM-CMM [5]

The development of this software maintenance maturity model that has required an exhaustive review of the software maintenance standards, industrial pratices, key references, publications and current research proposals. The list of key process areas of this models already highlights processes that software maintainers use in their day to day work. Some of these have not yet found their way in the 2001 SWEBOK v0.9 draft proposal.

2.0 Suggested clarifications within the current Maintenance taxonomy

The references in this Paper refer to the corresponding section numbers within the Maintenance Chapter of the Trial version of SWEBOK Guide. We first focus on the content of the Maintenance taxonomy of topics, as described in this version of the SWEBOK Guide.

1.1 Software maintenance versus software development

Basili states that software maintenance is a specific domain of software engineering, and that it is therefore necessary to look into its processes and methodologies to take into account its specific characteristics [*Bas96*]. For instance, in a software maintenance organization, it is important to understand how the management of maintenance activities differs from the management of software project activities. While project management is organized towards the delivery of a product within a specific timeframe and by a pre-arranged project closure date, the maintenance organization and processes must be structured to handle ongoing work on a daily basis for its customers with, by definition, no closure date. Key characteristics in the nature and handling of small maintenance requests have been highlighted in [*1*], for example:

- Modification requests come in more or less randomly and cannot be accounted for individually in the annual budget planning process;
- Modification requests are reviewed and assigned priorities, often at the operational level – most do not require senior management involvement;
- The maintenance workload is not managed using project management techniques, but rather queue management techniques;
- The size and complexity of each small maintenance request are such that it can usually be handled by one or two maintenance resources;
- The maintenance workload is user-services-oriented and application-responsibility–oriented.
- Priorities can be shifted around at any time, and requests for corrections of application errors can take priority over other work in progress.

Both maintenance practitioners and researchers have observed that the fit of the project-oriented SW-CMM model is weak and maintenance–oriented models have already been proposed from the maintainers point of view, rather than from a developer's viewpoint of large projects: Kajko-Mattsson [*Kaj01*], Niessik [*Nie02*] or April's [*Apr03*]. We therefore recommend that, in Section 3.2 of the Trial Version, the statement that SW-CMM applies equally to maintenance be amended accordingly.

We also recommend to include in Section 3.2.1 on Maintenance Process Models the following references documenting the best practices for a software maintainer be added. In particular, key maintenance process and activities best practice models have been documented in [*Hin92*, *Hat94*, *Zit96*, *Iso00*, *Iti01*, *Kaj01*, *Nie02*], together with detailed practices that are important to the maintainer:

- Hinley's detailed process model is based on an iterative approach of accepting a stream of change requests (and error reports).
- Hather describes the wider aspects of the software maintenance process, in the form of applications management.
- Zitouni presented the detailed practices of a Software Maintenance Maturity Model.
- Kajko-Mattsson has focusedon a Corrective Maintenance Maturity Model.
- More recently Niessik offers an IT Service Capability Model that applies to maintainers.
- The UK Central Computer and Telecommunications Agency[*Iti01*] has produced a series of guidelines on good practice that apply to the maintainer, in the form of the Information Technology Infrastructure Library.
- It would also be important to note that maintainers are obtaining ISO9001:2000 [*Iso00*] certification which has a significant impact on their process and their customers.

We also recommend to introduce in Section 3.3.1.4 on Maintainability the references to standards related to maintenance, such as ISO9126 [*Iso91a*], IEE1061 [*Iee98*], and IEEE982.2 [*Iee88*] standards. Maintainability and many other measures of quality of a software..

The reference to Colter [*Col87*] work should be quoted in the introduction to Section 3.3.2 on Management Problem: More specifically Colter observed that that the major problem of software maintenance was not technical, but managerial, and discussed in further details the many alignment, staffing and status issues found in software maintenance.

The list of items in Section 3.4 on Techniques for Maintenance is, in our opinion, too restrictive to represent all the techniques used by a maintainer. On the one hand, it should be highlighted that most of the developers design, coding and testing techniques are used by the maintainer and, on the other hand, it might more appropriate to limit the content of this section to the description of the software rejuvenation, migration and retirement subjects specifics to maintainers. Rejuvenation [*Pfl01*] includes typically redocumentation [*Pre97*], restructuration [*Pre97*] as well as reverse engineering and re-engineering. Of course, investing resources in a rejuvenating effort typically will require a business case: The maintainer must develop and present the decision criteria that will trigger and prove the potential savings associated with the rejuvination of a legacy software [*Pre97, Ben93*]. Similarly for the migration or retirement of a software which will require a plan as suggested in ISO12207 sections 5.5.5.2 and 5.5.6.1.

Finally, since the topic of 'impact analysis' is adressed in other areas of the text (ex: 3.2.2.1), we recommend to move the impact analysis text from this section.

3.0 Suggested additions to the taxonomy

We also suggest that a number of topics and sub-topics be added to the currently version of the Maintenane KA taxonomy; specific references are also suggested to document the proposed additions. It is left to the SWEBOK Maintenance associate editor to adapt the stucture of the Trial taxonomy to take into consideration the proposed changes, including recommendation on addition of suggested references to the reference structure publication. The following section presents additional topics observed to be important knowledge required by the software maintainers. It is left to the SWEBOK associate editor to decide if these topics and references are to be added in the main text or only as further readings.

Transition Management : ISO/IEC 14764, in section 6.7 states that it is essential that the maintainers look at what activities, and related output, is, and is not, carried out during the development stage of a software. Many autors have described the problems that occur in maintenance and often trace them back the cause to a weak or missing development activity [*Boe81, Sch87, HP90, Dek92, Wal94, Pig97, Ben00*]. This means that emphasis must be made by the maintainer during development to try to influence quality work.

Software Engineering Process Group : Maintainers are involved in SEPG activities as full members of complex and key support processes of the organisation. Maintainer process improvement includes an important communication activity to publicise the uniqueness of some key maintenance activities [*Fow90, Rif02*].

Process Simulation : Process simulation techniques are used in the maintenance area. These techniques are used for improvement activities to optimize the maintenance processes and case studies are described in [*Bar95*].

Software Maintenance Measurement : Maintainers in more advanced maintenance organization use extensively customers satisfaction surveys to understand how their customers are doing [*But95, Voi00, Cfi02*]. Maintainers use internal benchmarking techniques to compare different maintenance organisations and products to improve their processes internally [*Abr93a, Bou96a*]. External benchmarking of software maintenance organizations is now becoming more popular [*Gui91, Abr93a, Ifp94, Mai02, Isb03*]. Measurement programmes specific to maintainers are also described in publications [*Gra87, Abr91, Abr93, Stp93, Sta94, Mcg95*]. Software estimation models based on the functional size of small maintenance have also been published [*Abr95*]. Pressman [*Pre97*, section 4.5.2] also indicates that we cannot find one measure to reflect the maintainability of a software, and that a number of indicators is required! This leads to external and internal measurement of the maintenability of software done by some organizations using commercial tools [*Boo94, Lag96, Apr00*].

Accept/Reject Maintenance Requests : There is a unique process that the maintainers use to determine if a request is to be accepted or rejected (it has been observed that most often rejected requests are tranfered to a developer) [*Dor97, Apr01*]. This process is often source of confusion by the customers and developers and need to be documented and shared with all the stakeholders.

Maintenance Request Repository : An adequate information system (often shared with the operations help desk area) must be used by the maintainer to manage the workload and to track a large number of users requests . It can become the basis for the effort collection and an important component of the measurement infrastructure [*Gla81, Art88, Iti01* section 4.4.7, Kaj01d, Nie02 activity 3].

Software Maintainer specific training and education : the following references on maintainers training and education address the specifics for software maintainers [*Kaj01c, Kaj01e, Kaj01f, Hum00, Pfl01* section 10.1 and chapter 11]

Service level Agreement (SLA): Maintainers are or course on the front lines of all software support activities. They get the service calls when software fails, and customers are not always interested to understand the details and the hidden complexity. Maintainers need to understand and co-ordinate the service level across many softwares, hardwares, external suppleirs and internal IT groups [*Bou99*, *Apr01*, *Raf02*]. Maintainers should therefore develop formal Service Level Agreements (SLA) with their customers and tie them with their suppliers (also referred to as 'back to back' SLA's) to ensure a common understanding of what can be achieved and to manage service levels proactively [*Mcb96*, *Bou99*, *Iti01* 4.3.3. & 4.3.4, *Apr01*]. The maintainers should also be involved annually in the planning, updating and negotiation of the SLA's [*Iti01* 4.2, *Dmr96*].

Other Maintenance Contracts : Many types of contracts can be inventoried in a maintenance organization and the maintainers must get familiarized with all the contracts impacting his areas of responsibilities, and acquire the relevant expertise when its time to sign or renew them *[Iti01* 4.4.4 & 4.4.5, Mar94]. This, of course, should include design and management of escalation clauses when service/software problems occur [*Raf02*].

Billing of the maintainers services : More and more often, the maintainer must track accurately its work and issue a maintenance billing to the customer organization; this must of course be supported by the development of a billing policy [*Iti01* 5.4.2]. Maintenance service items and prices must be clarified and supported by a software maintenance billing process and supporting systems.

Production systems surveillance: A maintenance organization must also put in place a production systems surveillance set-up to prode, every day, the operational environment for signs of degradation or failures. Such surveillance systems ensure that problems are identified as early as possible(hopefully before the user is aware of it). [*Iti01* section 4.4.8].

Maintenance Planning : [*Apr03*] has identified in the SM-CMM practices in five different planning perspectives used by a software maintainer:
1) Annual Planning (budget, ressources, training, rejuvenation projects,…)[*Bal03*, *Pfl01*, *Iee98a*];
2) Transition Planning of new applications [*Pig97*, ISO14764];
3) Version planning of what is in and out of a new version of a software release [*Mar83*,*Lal96*] ;
4) Disaster Recovery (DR) planning which consist of planning the annual DR tests and
5) Planning at the individual maintenance requests (also called impact analysis)[ISO14764, ISO12207, IEE1219, *Lay90*, *Pfl01*, *Kaj01d*].

Each of those planning activities have their own objectives and different outputs.

4.0 Additional references

In addition to the references already included in the Trial version, we recommend that the followinig seminal references be added:

- Addition in Section 3.1.6 on Categories of Maintenance of Lientz & Swanson [*Lie78*][*Lie8O*] which undertook the first surveys which categorized software maintenance into four different categories.

- Software Maintenance Tutorial of Keith H. Bennet [*Ben00*] be inserted as a key reference as it is an overview of maintenance quite similar to the proposed SWEBOK text. It is a good introduction document to software maintenance.

5.0 References

[Abr91] A. Abran, H. Nguyenkim, Analysis of Maintenance Work Categories Through Measurement. Proceeding of the Conference on Software Maintenance IEEE 1991, Sorrento, Italy, October 15-17, pp. 104-113.

[Abr93] A .Abran, H. Nguyenkim, Measurement of the Maintenance Process from a Demand-Based Perspective, Journal of Software Maintenance: Research and Practice, Vol. 5, no 2, 1993. pp 63-90.

[Abr93a]A.Abran, Maintenance Productivity & Quality Studies: Industry Feedback on Benchmarking. Proceedings of the Software Maintenance Conference, ICSM93, Montréal September 27-30, 1993, Invited paper pp. 370.

[Abr95] A. Abran, M. Maya, A Sizing Measure for Adaptive Maintenance Work Products, ICSM-95, Opio, France, Oct. 1995.

[Apr03] A.April (2003) Modele D'Evaluation de la Capacite a Maintenir le Logiciel, Rapport technique en Genie Electrique, Ecole de technologie Superieure, v0.5.

[Apr01] April, A., Bouman, J., Abran, A., Al-Shurougi D., (2001). Software Maintenance in a Service Level Agreement : Controlling the Customer Expectations, Fourth European Software Conference, FESMA, Heidleberg, Germany, May.

[Apr00] April, A., Al-Shurougi, D. (2000). Software Product Measurement for supplier Evaluation, FESMA 2000, Madrid, 18-20 October.

[Art88] Arthur, L., (1988). Software Evolution : The Software Maintenance Challenge, John Wiley & Sons.

[Bal03] Baldrige National Quality Program. (2003) Criteria for Performance Excellence. [On line] http://www.quality.nist.gov/ (last accessed on 8 January 2003).

[Bar95] Barghouti, N., Rosenblum, D., et al. (1995). Two case studies in modeling real, corporate processes, Software Process : Improvement and Practice, 1(1), 17-32.

[Bas96] V.Basili, L.Briand, S.Condon, et al., "Understanding and Predicting the Process Software Maintenance Releases", Proceedings of the IEEE International Conference on Software Engineering, IEEE-Computer Society Press, Los Alamitos, CA, 1996.

[Ben93] Bennett, K.H., "An Overview of Maintenance and Reverse Engineering," in *The REDO Compendium*, ed. van Zuylen, Wiley, 1993.

[Ben00] Bennett, K.H, (2000). Software Maintenance : A Tutorial, Software Engineering , Dorfman and Thayer, IEEE , 2000, p 289-303.

[Boe81] Boehm, B.W. (1981). Software Engineering Economics, Prentice-Hall.

[Boo94] G.Booch, D. Bryan, Software Engineering with Ada, Third Edition, Benjamin/Cummings Publishing Company, Redwood City, California, 1994

[Bou99] Bouman, J., Trienekens, J., Van der Zwan, M. (1999). Specification of Service Level Agreements, Clarifying Concepts on the Basis of Practical Research, Proceedings of Software Technology and Engineering Practice '99.

[Bou96a] P.Bourque, Maintenance Benchmarking Projects: An Overview. dans International Software Benchmarking: Engineering and Measurement Issues. 1996. Montreal: Université du Québec Montréal.

[But95] Butler, K. (1995). The Economic Benefits of Software Process Improvement, Crosstalk, 8(7), July issue, 14-17.

[Cfi02] Voice of the Customer Surveys, Claes Formell International, [On line] http://www.cfigroup.com/home.htm (last accessed on 8 January 2003) (provided as an example of a survey only)

[Col87] Colter, M., "The Business of Software Maintenance," *Proc. 1st Workshop Software Maintenance*, University of Durham, Durham, 1987.

[Dek92]S.M. Dekleva. Delphi Study of Software Maintenance Problems. Proceedings of the International Conference on Software Maintenance, 1992, p10-17.

[Dmr96]Ducroc, Meilleur, Roy, (1996). DMR Application Maintenance and Support Services, Service Level Agreements Metrics and Benefits, Toronto, Canada, October 1.

[Dor97] Dorfman, M., Thayer, R.H., (1997). Software Engineering. IEEE Computer Society Press.

[Fow90] Fowler, P., Rifkin, S. (1990). Software Engineering Process Group Guide, CMU/SEI-90-TR-24, [On line] http://www.sei.cmu.edu/pub/documents/90.reports/pdf/tr24.90.pdf (last accessed on 6 January 2003).

[Gla81] Glass, R., Noiseux, R. (1981). Software Maintenence Guidebook, prentice-Hall.

[Gui91] H.Yepez Jr., Competitive Benchmarking, IBM Guide session IM-3305G, Guide International Corp .,20-03-91, Anaheim, Ca.

[Gra87] R. Grady, D. Caswell, Software Metrics:Establising a Company wide Program. EnglewoodCliffs, NJ, Prentice-Hall, 1987.

[Hat94] Hather, R, Burd, L., and Boldyreff, C., "A Method for Application Management Maturity Assessment," *Proc. Centre for Software Reliability Conference*, 1994 (to be published).

[Hin92] Hinley, D.S. and Bennett K.H., "Developing a Model to Manage the Software Maintenance Process," *Proc. Conf Software Maintenance*, IEEE Computer Society Press, Los Alamitos, Calif, 1992, pp.174—182.

[HP90] Corporate Engineering, Software Quality and Productivity Guide – Best Practice in Software Development at Hewlett-Packard, 1990.

[Hum00] Humphrey, W. (2000). Managing Technical People – Innovation, teamwork and the software Process, Addison-Wesley.

[Iee88]IEEE Std 982.2-1988 : IEEE Guide for the use of the Ieee Standard Dictionary of measures to produce reliable software, 96 pages, 1988.

[Iee98]ANSI/IEEE Std 1061. IEEE Standard for a Software Quality Metrics Methodology. IEEE Computer Society Press, 1998.

[Iee98a] IEEE Std 1219-1998: Standard for Software Maintenance, IEEE , 1998.

[IFP94] Guidelines to Software Measurement, version 1.0, International Funtion Point User Group, 1994.

[Isb03] International Benchmarking Software group (2003). Data Collection Questionnaire Application Software Maintenance and Support, Version 1.0, [On Line] http://www.isbsg.org.au/html/index2.html (last accessed on 25 January 2003).

[Iso91a] ISO/IEC 9126 (2001). Information Technology – Software Product Evaluation – Quality characteristics and guidelines for their use, Second edition.

[Iso00] ISO9001:2000, Quality Management Systems – Requirements, International Organization for Standardization, Third edition 2000-12-15, International Organization for Standardization, Geneva, Switzerland.

[Iti01] Office of Government of Commerce, (2001), Best Practice for Service Delivery, Managing IT service, isbn 0113300174, Published by the Stationary Office, Norwich, UK.

[Kaj01] Kajko-Mattsson, M. (2001). Towards a Business Maintenance Model, Proceedings of the IEEE International Conference on Software Maintenance, 500-509.

[Kaj01a] Kajko-Mattsson, M., Westblom, U., Forssander, S. (2001). Taxonomy of problem Management Activities Fifth European Conference on Maintenance and Re-Engineering, 1-10.

[Kaj01c] Kajko-Mattsson, M., Forssander, S., Westblom, U., (2001). Corrective-Maintenance Maturity Model (Cm3): Maintainer's Education and Training ICSE, 610-619.

[Kaj01d] Kajko-Mattsson, M., (2001). Corrective Maintenance Maturity Model, partial fulfillment of the requirements for P.H.D, report 01-015, Stockholm University.

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

[Kaj01f] Kajko-Mattsson, M., Forssander, S., Andersson, G., Olsson, U., Developing CM^3: Maintainers' Education and Training at ABB, Journal of Computer Science Education, Swets&Zeitlinger, Vol. 12, No 1-2, pp. 57-89, 2002. (2001).

[Kaj01g] Kajko-Mattsson, M., Forssander, G., Olsson, U., Corrective Maintenance Maturity Model (CM[3)]: Maintainers' Education and Training, in Proceedings, International Conference on Software Engineering, IEEE Computer Society Press: Los Alamitos, CA, 2001.

[Kaj00] Kajko-Mattsson, M., Forssander, S., Andersson, G., (2000). Problem Reporting and Resolution Process at ABB ROP: State of Practice, Journal of Software Evolution and Maintenance, John Wiley & Sons, 12(5), 255-285.

[Lag96] Lague, B., April, A. (1996). Mapping of the ISO9126 Maintainability Internal Metrics to an industrial research tool, SESS, Montreal, October 21-25.

[Lal96] Lalonde, D. (1996). Bell Canada and Bell Sygma presentation. BCCS Platform release Management, release 48 Content, Process Champion Meeting, October 30th.

[Lay90] Layzell, P., Macaulay, L. (1990)., An Investigation into Software Maintenance Perception and Practices, In Proceedings IEEE International Conference on Software Maintenance, 130-140.

[Leh8O] Lehman, MM., "Programs, Lifecycles and the Laws of Software Evolution," *Proc. IEEE,* Vol. 19, 1980, pp. 1060—1076.

[Leh84] Lehman, MM., "Program Evolution," *Information Processing Management,* Vol. 20, 1984, pp. 19—36.

[Lie78] Lientz, B., Swanson, E.B., and Tompkins, G.E., "Characteristics of Applications Software Maintenance" *Comm. ACM,* Vol. 21, 1978, pp. 466—471.

[Lie8O] Leintz, B. and Swanson, E.B., *Software Maintenance Management,* Addison-Wesley, Reading, Mass., 1980.

[Mai02] Maintenance Technology, Using Benchmarking Data Effectively, [On Line] http://www.mt-online.com/current/03-01mm01.html (last accessed on 8 January 2003).

[Mar83] Martin, J., McLure, C. (1983). Software Maintenance, The Problem and Its Solutions, Prentice-Hall, Englewood Cliffs, New Jersey.

[Mcb96]D. McBride, 'Successful Deployment of IT Service Management in the Distributed Enterprise', Hewlett-Packard Professional , 1996.

[McG95] J.McGarry, Practical Software Measurement : A Guide to Objective Program Insight, DoD, Sept. 1995.

[Nie02] Niessink, F., Clerk, V., van Vliet, H. (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 (last accessed on 15 December 2002).

[Pfl01] Pfleeger, S.L. (2001). Software Engineering Theory and Practice, second edition, Prentice hall.

[Pig97] Pigoski, T.M. (1997). Practical Software Maintenance: Best Practice for Managing your Software Investment, Wiley.

[Pre97] Pressman, R.S. (1997). Software Engineering: A practitioner's Approach. Fourth Edition, McGraw-Hill, fourth edition.

[Raf02] Raffoul, W. (2002). The road to outsourcing success: Outsourcing maturity management model, Meta Group, [On line] http://www.techupdate.com/techupdate/stories/main/0%2C14179%2C2851971-2%2C00.html (last accessed on 6 January 2003).

[Rif02] Rifkin, S. (2002). What would I do if I wrote the SEPG guide today, SEPG Conference, Phoenix, Arizona, V0.2, February 19.

[Sch87] Schneidewind. N.F. (1987). The State of the Maintenance. Proceedings of the IEEE.

[Sta94] G.E.Stark, L.C.Kern, and C.V.Vowell. A Software Metric Set for Programme Maintenance Management. Journal of Systems and Software, 1994.

[Stp93] D.St-Pierre, Integration of Maintenance Metrics. in IEEE International Conference on Software Maintenance -

ICSM-93. 1993. Montréal, Québec: IEEE.

[Voi00] Voice of the Customer surveys, [On line] http://www.isixsigma.com/vc/ (last accessed on 8 January 2002). Provided as an example of a survey only.

[Wal94] Walton D.S., "Maintainability Metrics," *Proc. Centre for Software Reliability Conf*, Dublin, 1994. Available from Centre for Software Reliability, City University, London, UK.

[Zit96] Zitouni, M., Abran, A. (1996). A Model to Evaluate and Improve the Quality of the Software Maintenance Process, in Proceedings of 6th International Conference on Software Quality Conference, Ottawa: ASQC- Software Division, 238-258.