A Glimpse into CM³: Problem Management

Mira Kajko-Mattsson

Department of Computer and Systems Sciences, Royal Institute of Technology and Stockholm University, Electrum 230, SE-16440 Kista, Sweden
mira@dsv.su.se

Abstract. In this paper, we give a very short glimpse into a process model for software problem management. This model is the result of a long-term study of current generic process models and of industrial processes at ABB.

1. Introduction

CM³: Problem Management is a process model for reporting and resolving software problems within corrective maintenance. It is the result of a long-term study of standard process models and of two industrial processes utilised at ABB Robotics and ABB Automation Products. Its primary role is to provide guidance to industrial organisations in the process of building and improving their problem management [4].

As depicted in Figure 1, *CM³: Problem Management* has the following constituents: (1) *CM³: Definition of Maintenance and Corrective Maintenance*, (2) *CM³: Taxonomy of Activities* listing process activities [5], (3) *CM³: Maturity Levels* indicating the degree of organisations' capability to manage problems and provide feedback to defect prevention and process improvement, (3) *CM³: Conceptual Model* identifying problem management concepts [1], (4) *CM³: Roles* designating roles of individuals involved in problem management [5], (6) *CM³: Roadmaps* visualising the maintenance process [2-3], and, finally, (7) *CM³: Maintenance Elements* providing explanations of and motivations for the above-mentioned constituents using the structure presented in Figure 2 [7].

Due to the restricted amount of space in this paper, we limit our presentation of *CM³: Problem Management* to only the designation of *CM³: Maturity Levels* (*Initial*, *Defined,* and *Optimal),* and some of their most important *CM³: Maintenance Elements.* For more information about our process model, we recommend the interested readers to survey our other research contributions [1-7].

2. CM³: Problem Management: Level 1 (Initial)

At the *Initial* level, the process is implicitly understood by maintenance engineers. Usually, this is due to the following reasons: (1) the process is not defined, (2) the process is defined, but not documented, (3) the process is defined and documented, but the documentation is either outdated or inconsistent, or (4) the process is defined and properly documented, but not consistently adhered to.

K.R. Dittrich, A. Geppert, M.C. Norrie (Eds.): CAiSE 2001, LNCS 2068, pp. 469–472, 2001.

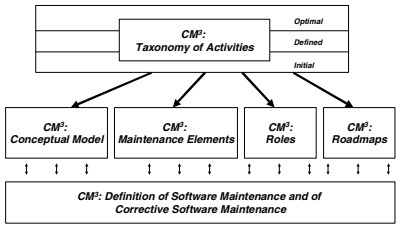

Fig. 1. Structure of CM³ [4, 6].

Maintenance Element:	A statement of a maintenance element.
Goal:	The goal to achieve after having implemented the element.
Motivation:	Motivations for why the maintenance element should be implemented. Different problems and arguments are described, if any.
Measures:	A list of suggestions to be conducted in order to implement the maintenance element.

Fig. 2. CM³: Maintenance elements [7].

At this level, the process offers little visibility, or gives a distorted view of its status. At the very most, the organisation has insight into the amount of software problems that have been reported and resolved. But, it cannot follow and follow up the problem resolution process, and make any kinds of reliable process analyses. The feedback provided by the process is not always meaningful. Success at this level may depend on the combination of the following factors: (a) competence of the maintenance engineers, (b) their dedicated overtime, (c) low staff turn-over rate, (d) the stability of the applications maintained, and (e) the repetitive nature of conducting similar tasks.

At this level, *CM³: Problem Management* strongly recommends to consider the following maintenance process elements as a starting point on the organisations' voyage towards process excellence.

- *Software problems are communicated in writing*: to enable communication about problems in a formal, disciplined and tractable way, and to achieve control over all the maintenance requirements (reported problems in our case).
- *One problem is reported in one and only one problem report*: to facilitate communication on software problems, to enable follow and follow up of the problem resolution, and to enable problem validation and verification.
- *A problem submitter is identified*: to enable the delivery of the problem solution to the right customer.
- *Maintenance requirements are categorised*: to enable prioritisation of the maintenance requirements, to enable different kinds of process analyses, and to provide basis for assessing product quality.
- *Problems are classified as internal or external*: to be able to assess the quality of software products from the customer perspective, and to assess the effectiveness of quality assurance and quality control procedures.
- *Problems are classified as unique or duplicate*: to improve and measure productivity, and to provide correct statistics of the number of unique problems.

3. CM³: Problem Management: Level 2 (Defined)

At the *Defined* level, a coarse-grained process model for managing software problems is defined and documented. The process covers the most rudimentary process phases and activities essential for managing software problems. These phases/activities offer

visible milestones for following the progress and for making different kinds of intermediary decisions.

At this level, the process is consistently adhered to. Process compliance with its documentation is checked on a regular or event-driven basis. Although simple, the data and measurement provided by the process is meaningful, appropriately reflecting the process. Due to the coarse-grained process visibility, however, the assessment of effort and resources still corresponds to a rough estimation. Some of the maintenance elements applicable at this level are the following:

- *A template for how to structure a description of a software problem is institutionalised*: to develop a maximal support to problem submitters for describing their maintenance requirements (software problems). A proper description of a problem is the most important prerequisite for its effective resolution.
- *Correctness, consistency, and completeness of the problem report data is continuously checked and improved*: to maximise an objective understanding of a software problem, and to provide correct, consistent, and complete feedback for problem validation and verification, and for making different kinds of statistics and process analyses.
- *Sources (submitter and maintainer's) of problem description and problem report data are separated*: to enable efficient problem validation and verification, correct and reliable statistics, and to enable planning of maintenance work.
- *Problem management process, its phases, results and executing roles are documented:* to track the problem resolution process to their process phases, results, and the roles.
- *Process variances (allowed process deviations) are defined and institutionalised:* to enable process flexibility, and to provide feedback to process refinement and improvement.
- *The suggestion for a problem solution and the plan for its realisation is approved by a Change Control Board, before it gets implemented:* to assure that software is treated like a common organisational asset by commonly discussing and choosing the most optimal solution to the problem.

4. CM³: Problem Management: Level 3 (Optimal)

At the *Optimal* level, the problem management process allows a fine-grained visibility into its status and progress. We have clear insight into every process step, and its results. The detailed process knowledge helps us make a thorough impact analysis during which the complexity and ramifications of a corrective change are recognised. This substantially helps assess the amount of effort and resources required for its resolution. In contrast to Level 2, the discrepancy between the planned and actual effort is strongly reduced due to the more detailed process feedback. The process does not only suffice for managing software problems, but also provides useful feedback to process improvement and defect prevention. The following maintenance elements apply at this level:

- *Causes of problem defects are defined and classified*: to be able to assess the product quality, and to enable a root cause analysis.
- *Traceability of change is ensured on the documentation/source code line level*: to correctly measure the modification size due to software problems, and to enable the tracking of all modifications to problem reports and vice versa.
- *Impact analysis is conducted*: to determine the complexity and ramifications (including ripple effect) of a software problem in order to correctly assess the amount of work, effort and resources required for its resolution.
- *A model for conducting a root cause analysis is defined and followed*: to identify the original sources of defects in order to provide feedback to process improvement and defect prevention.

5. Epilogue

In this paper, we have presented *CM³: Problem Management* process model. Our model does not only handle the resolution of software problems, but also provides a basis for quantitative feedback important for assessing product quality, crucial for continuous process analysis and improvement, and essential for defect prevention. Our aspirations are to provide a framework to the organisations building and improving their problem management processes, and to provide a pedagogical tool for universities and organisations in the process of educating and training their students and software engineers within the area of corrective maintenance.

References

[1] Kajko-Mattsson M. *Common Concept Apparatus within Corrective Software Maintenance*, In Proceedings, International Conference on Software Maintenance, IEEE Computer Society Press in Los Alamitos CA, 1999, pp. 287-294.

[2] Kajko-Mattsson, M., *Maintenance at ABB (I): Software Problem Administration Processes (The State of Practice),* Proceedings, International Conference on Software Maintenance, IEEE Computer Society Press: Los Alamitos, CA, Sep 1999.

[3] Kajko-Mattsson, M., *Maintenance at ABB (II): Change Execution Processes (The State of Practice),* In Proceedings, International Conference on Software Maintenance IEEE Computer Society Press: Los Alamitos, CA, Sep 1999.

[4] Kajko-Mattsson, M., *Corrective Maintenance Maturity Model (CM³),* Technical Report, No. 00-010, Department of Computer and Systems Sciences (DSV), Stockholm University and Royal Institute of Technology, December 2000.

[5] Kajko-Mattsson, Mira, *Taxonomy of Problem Management Activities*, In Proceedings 5th European Conference on Software Maintenance and Reengineering, IEEE Computer Society Press in Los Alamitos CA, 2001.

[6] 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 in Los Alamitos CA, 2001.

[7] Kajko-Mattsson, M., *CM³: Problem Management: Maintenance Elements*, Technical Report, No. 2001-010, Department of Computer and Systems Sciences (DSV), Stockholm University and Royal Institute of Technology, December 2001.