

Software Maintenance Capability Maturity Model (SM-CMM): Process Performance Measurement

*Alain April*¹, *Alain Abran*², *Reiner R. Dumke*³

¹ Bahrain telecommunications Company, Bahrain
april_alain@hotmail.com

² École de Technologie Supérieure, Montréal, Canada
aabran@ele.etsmtl.ca

³ Otto von Guericke University of Magdeburg, Germany
dumke@ivs.cs.uni-magdeburg.de

Abstract: *Software maintenance constitutes an important part of the total cost of the lifecycle of software. Some even argue that this might be the most important component of the cost, even though customers often do not perceive the added value of software maintenance. A proposed approach to highlighting the added value of maintenance is to provide the customer with process performance measures aligned with the key activities performed by the maintenance organization. Such performance measures could then form the basis for a clear agreement on the expectations, and outcomes, of these activities.*

Process Performance management and measurement requires that processes be chosen based on their impact on the quality and the performance of the software maintenance organization. It also requires that measures be identified and established and that a reference point (baseline) and a target be set for each measure. Finally, they require that data be collected in order to develop and use process performance prediction models. In this paper, we introduce best practices, for the first three maturity levels, to help the maintainer organization assess its process performance. These practices constitute a subset of our proposed Software Maintenance Capability Maturity Model (SM-CMM).

Keywords: *software maintenance, maintenance measurement, process performance measurement, process maturity.*

1 Introduction

1.1 Software maintenance versus software development

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.

1.2 Contribution

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 [6].

Figure 1 illustrates the second version [5] of a Software Maintenance Capability Maturity Model (SM-CMM) that was initially published in 1996 [14], keeping

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 Rejuvenation and retirement of software

Figure 1: Domain and Key Process Areas of the SM-CMM version 2.0 [5]

many of the attributes and features of the first version of the model and modernizing it with the recent appearance of the CMMi. Our 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. This paper focuses on ‘*Maintenance Process Performance*’, the fourth key process area of the Process Management domain - highlighted in Figure 1. This ‘*Maintenance Process Performance*’ requires the identification, definition and establishment of quantitative targets. By monitoring progress against these targets, management can derive quantitative measures of how well the organization is meeting its business objectives.

To support quantitative management, the software maintenance organization must define and implement:

- a) definitions of measures (the measures are typically divided into two categories:
 - i) external measures, which are visible by customers; and
 - ii) internal measures, which are more technical and aimed at characteristics of the products and of the software itself);
- b) target measures;
- c) reference points (baselines) for each measure;
- d) a measurement repository; and
- e) measurement models which can estimate/anticipate the performance of the maintenance processes.

Maintenance organizations must use and analyse the data collected on the performance of the execution of its processes in order to develop quantitative knowledge of the quality of: a) its deliverables; b) its services; c) the performance of the execution of its processes; and d) the technologies used [13, FM102.HDA102.HDB103.T106].

2 Maintenance Processes

What processes and key activities are likely to be needed to measure software maintenance performance aspects? Which are important to customers and stakeholders? To answer these questions, we propose a model of key software maintenance processes. The most important processes and key activities requiring measurement are, in our opinion, those represented in dark grey and light grey in figure 2.

To provide alignment with the ISO 12207 standard [8], the software maintenance key processes have been grouped into three classes (Figure 2):

- a) primary processes (operational);
- b) support processes (supporting the primary processes); and
- c) the organizational processes that are offered by the IT unit or by other departments of the organization (for example: finance, human resources, purchasing, etc.).

The key Operational Processes (also called primary processes) that a software maintenance organization uses are initiated at software project development initiation, and then maintained subsequently, starting with the transition process. The *Transition process* ensures that the software project is controlled and that a structured and coordinated approach is used to transfer the software to the maintainer. In this process, the maintainer will focus on the maintainability of this new software.

Once the software has become the responsibility of the maintainer, the *Issue and Service Request Management process* handles all the daily issues, problem reports, change requests and support requests. These are the daily services that must be managed efficiently. The first step in this process is to assess whether a request is to be addressed, re-routed or rejected (on the basis of the service level agreement [3], the nature of the request and its size) [4, 11]. Accepted requests are documented, prioritised, assigned and processed in one of the service categories: 1) *Operational Support process* (which typically does not necessitate any modification of software); 2) *Software Correction process*; or 3) *Software Evolution process*.

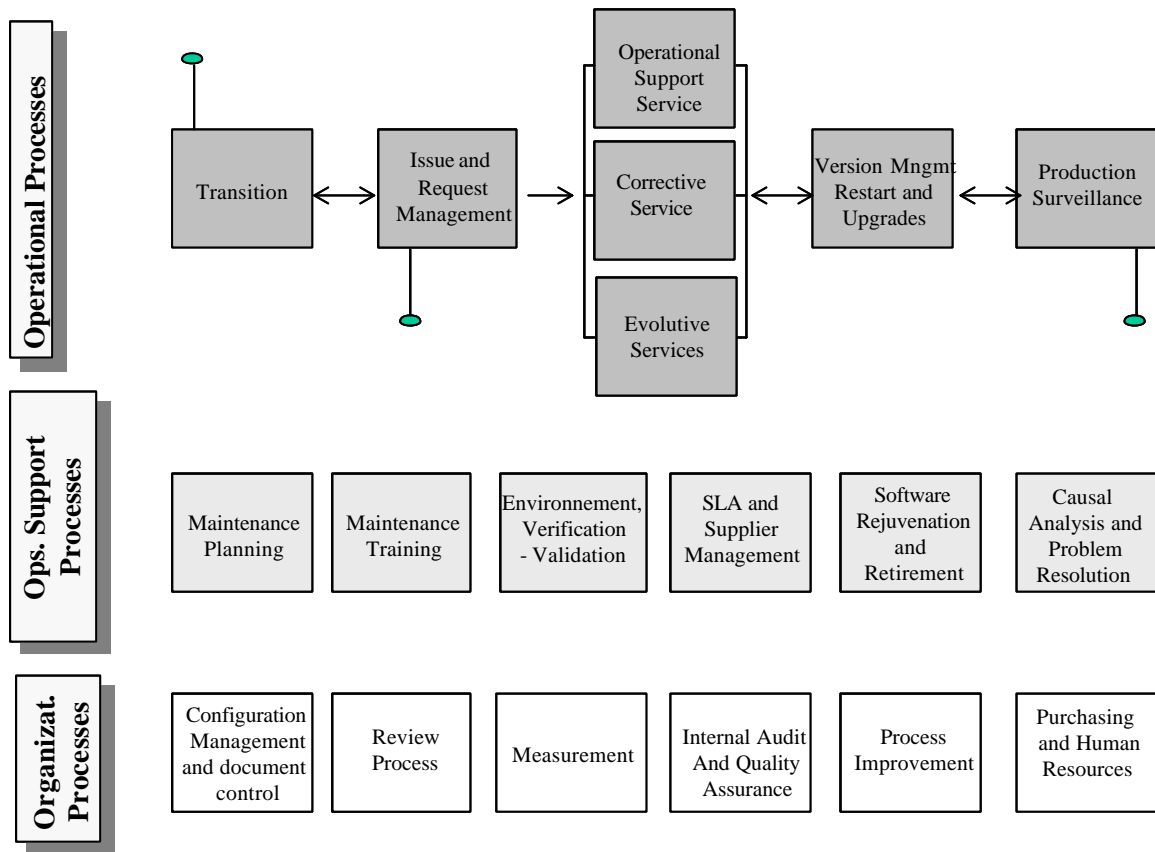


Figure 2: A classification of the Software Maintainer's Key Processes

Note that certain service requests do not necessitate any modification to the software. In our model, we refer to them as 'operational support' activities, and these consist of: a) replies to questions; b) provision of information and counselling; and c) helping customers to better understand the software, a transaction or its documentation.

The last two main operational processes concern the *Version Management process*, which will move items to production, and the *Production Surveillance process*, which will ensure that the operational environment has not been degraded. Maintainers always oversee the behaviour of the operational system and its environments for signs of degradation. They will quickly warn other support groups when something unusual happens (operators, technical support, scheduling, networks and desktop support) and judge whether or not it is an instance of service degradation that needs to be investigated.

A process which is used, when required, by an operational process is said to be an operational support process [8]. In this classification, we include: a) the many maintenance planning processes; b) the maintainer's education and training; c) the maintenance environments and testing; d) management of the contractual aspects and service level agreements; e) rejuvenation or retirement of software; and, finally, f) resolution of problems. These are all key activities, which are available to support some operational process activities.

Organizational processes are typically offered by the IT department and by other departments in the organization (for example: human resources, finance, quality assurance and ISO 9000). While they are important to measure, it is often easier for the maintainer to start defining the operational and operational support processes.

This generic model should help users understand the various key software maintenance processes. What is important is that these processes be classified. The list presented here is only a model (one that the companies that have helped develop the SM-CMM model find useful) and other model users should create their own list based on the terminology and classifications that suit their particular organization.

3 Maintenance - Performance Management Process

This section presents both the goals and target of the Performance Management Process, as well as the detailed practices.

3.1 Goals and targets

Process performance management demands that the processes having the greatest impact on the quality and process performance of the maintenance organization be identified first, followed by a definition of the measures and the establishment of a baseline (a set of references).

The goals of process performance management are:

- To document the rationale for the selection of the most important performance factors impacting processes and their activities, and which are key to achieving the quality targets;
- To measure and communicate the targets and quantitative results of the service levels to both customers and management.

The objectives of process performance management are:

- To identify the processes and the key activities of the software maintenance organization that impact performance, and which are to be used for analysis;
- To establish a baseline consisting of the key software maintenance processes and activities;
- To identify and establish the measures of the performance of the selected processes/activities;
- To establish models for predicting the performance of the software maintenance processes/activities.

This performance management process is not complete in and of itself. It requires practices borrowed from other process areas of the SW-CMM model for it to be operational. The first link is to *Quantitative Maintenance Management* – Figure 1, which provides more information on how to use a reference point (baseline) of the performance of processes and of its models. The second link is to *Measurement, Decision and Causal Analysis* – Figure 1, to obtain more information on how to specify a measure, collect the data and conduct an analysis of the data. Together, these three process areas cover the whole domain of software maintenance measurement.

Once this process has been successfully implemented, it will be observed that:

- The measures of quality and of process performance have been established for the production software, intermediary products and software processes;
- The measures are harmonized and relate to the normalized processes;
- The data collection activities are performed at the operational level and the data are stored in the corporate repository;
- The baseline has been created, validated and documented.

3.2 Detailed practices

The detailed practices are presented next by maturity levels, from 0 to 3.

3.2.1 Level 0 and 1 practices

The individual practices are assigned to one of five levels of maturity. At level 0, there is only one practice;

Pro0.1 The software maintenance organization does not conduct process performance measurement on its processes.

These organizations just perform the daily work of software maintenance.

At level 1, two practices are included, and presented next.

Pro1.1 Individual initiatives of process or product measurement are implemented by individuals and employees interested in this domain.

In these software maintenance organizations, the software maintenance managers or programmers develop process and product measurements on their own initiative. These definitions are personal and are rarely: a) shared with other organizations; or b) used to better manage or improve either process or product. They are used typically to explain an individual event or situation which occurred.

Pro1.2 Some qualitative process and product measures are collected by the software maintenance organization.

The employees of the software maintenance organization have established relationships with their customer counterparts and obtained qualitative performance measures on how they are performing. These comments and observations typically appear in e-mails and circulate between the organizations.

3.2.2 Level 2 practices

At this maturity level, process performance addresses basic considerations and typically differs across the various units which conduct software maintenance in the organization. Qualitative information is normally collected by the manager and his employees and reported in weekly and monthly meetings. When quality and performance targets are established, this information is used for local, short-term improvement and is based on individual priorities.

Pro2.1: Some processes and key products of software maintenance have identified measures.

At this maturity level, the software maintenance organization should have identified a basic set of process measures. These measures are collected and are typically used in: a) the weekly management meeting; and b) communications with customers. Typical measures found at this level are based on what could be called measures for the management of queues [1], which means that process performance has not yet been addressed or completely understood. For example:

- The number of outstanding requests;
- The average waiting time before being serviced;
- The estimated number of days in the queue;
- The number of requests completed;
- The number of requests closed for this period, opened during this period and still pending;
- A comparison of estimates versus actual costs.

Pro2.2 Some quality and performance targets exist in the software maintenance organization.

At this level of maturity the software maintenance organization must identify basic quality and performance targets. Setting quality targets requires that the current performance is looked at and that realistic goals are established. These targets are typically used in the weekly management meeting and in the communications with customers. For example:

- a) the availability percentage of particular software in operations, i.e. 98%;
- b) the degree (%) of satisfaction of customers based on surveys;
- c) the limit of available overtime hours of the maintenance staff; and
- d) the average waiting time for a change request to be completed.

Pro2.3 The reference points (baselines), for the current measures are stored, used and reviewed with the various stakeholders (customers, sponsors, program managers and maintenance employees) for review, discussion and improvement.

At this maturity level, the current measure data are captured, stored and communicated to the many stakeholders, with the objective of explaining the current level of performance of the maintenance activities and production software. At this maturity level, this data is typically accumulated in a local, and sometimes personal, repository, and is normally used by the software maintenance manager to explain and analyse specific situations and events. The objective of this practice is the sharing and acceptance of an agreed-upon and communicated reference points (baseline), which describe the current performance levels.

3.2.3 Level 3 Practices

At this maturity level, the definition of the measures become more precise, and published, and is now standardized among all software maintenance units of the organization. The process performance definition activity must be considered as a project improvement activity to ensure alignment with the overall quality objectives of the organization. The key activities of the standardized software maintenance processes are identified as candidates to be measured. Quality and performance attributes of operational software are also investigated. Measure targets and reference points (baselines) are established and maintained, and a rationale describes the variation limits. The definition of each measure is established, and the data collection and data validation activities are identified. Customers should perceive a harmonization of the maintainer's activities, services and operational software measures. The human resources maintenance personnel are trained in process performance measurement activities. The measurement data is collected, validated and integrated into the corporate measurement repository. Human resources maintenance personnel collect measurement data, daily, as part of their operational activities.

Pro3.1 The software maintenance measurement programme is defined and treated as a project, and special attention is given to risk management in order to minimize the risk of failure.

To achieve this best practice, a measurement programme must be set up (with multiple controlled implementation steps or stages) and its risk managed. It is especially important to manage the measure definition, collection and verification activities, as is done at the operational level. Risk management can be divided into three steps: 1) definition of a risk management strategy; 2) identification and analysis of the risks; and 3) a check of the identified risks, including the implementation of mitigation, if necessary [13, PA148.N104].

Pro3.2 The software maintenance organization identifies its key processes and their activities, and defines quality and performance measures.

This best practice requires that key activities of the software maintenance processes be investigated to understand which contribute the most to quality and performance and how to measure them. Typically, the activities of the normalized processes that will contribute best to performance analyses of software maintenance must be identified and selected [13, PA164.IG101.SP101]. At this maturity level, the measurement activity must be coordinated across all software maintenance units of the organization. It is also necessary to align this activity with other IT measurement initiatives to ensure a cohesive approach for the whole organization. Ideally, all the measurement data should be integrated in one repository. Harmonization of measures in an organization is key [13, PA164.IG101.SP102.SubP103]. The definition of the measures that will be part of the performance analyses of software maintenance must be established and maintained [13, PA164.IG101.SP102]. The process measures are determined according to their perceived value for the maintenance organization and their customers. These measures should cover the whole software life cycle, including maintenance [9 6.4.2, 7 3.4.3]. Measures on which standardized techniques or processes can be used are defined and set up in accordance with a formal procedure [7, 1.5.3.5]. An organization-wide, approved and funded measurement programme supports the measurement activities and the analysis of these measures [9, 6.4] [7, 3.4.3.1]. The standardized software maintenance processes are used to select where data are to be collected and what analysis is required [7, 3.4.3.2].

Pro3.3 The software maintenance organization identifies its key products and production software, and defines their quality and performance measures.

'It is not practically possible to measure all sub-characteristics internally or externally for all parts of a large software product. Similarly it is not usually practical to measure quality in use for all possible user-task scenarios. Resources for evaluation need to be allocated between the different types of measurement dependent on the business objectives and the nature of the product and maintenance processes.' [10] The maintainer must identify which

perspective of the products to measure, i.e. those that have a significant impact on the customer and on the quality of the operational software. To achieve this best practice, the maintainer must identify key product measures and document why measuring them is important. In order to do this, all the software maintenance departments must share the same measures, so that an integrated approach to service level management and reporting will begin to emerge.

The intermediary products (for example: technical documentation, software testing activities) are rarely presented to the customer, but are key to software quality. We introduce here the concept of internal versus external software measures (see Figure 4). Achievement of software product quality is based on the execution of the operational and supporting processes in software product quality that can be measured internally (typically, by static measures on the code) or externally (typically, by measuring the behaviour of the source code when executed).

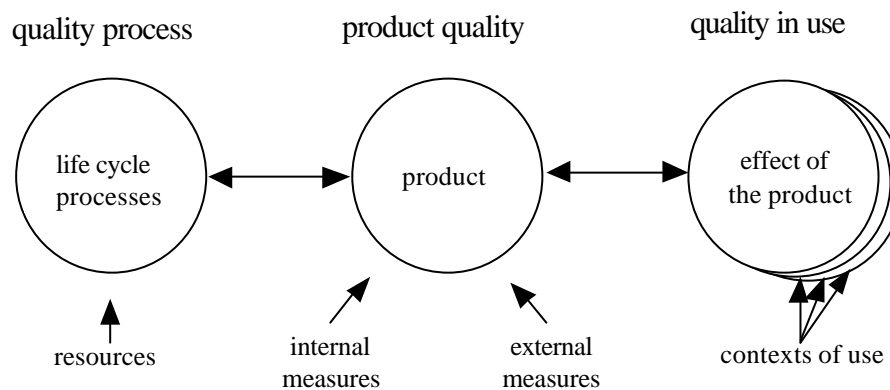


Figure 4: Quality in software life cycle

The objective is for the product to have the required effect in a particular context of use [10]. The external measures are used to reflect the attributes of the quality of the software that are apparent and important to customers. For example, we must measure the availability of the software, as this is important to the customer. Other examples of such important external measures are:

- a) measures of, and follow-up on, delivery times of a specific request [7, 3.4.3.5];
- b) measures of software failure, which are often associated with the identification of which software unit was involved in the failure [7, 3.4.3.7].

Pro3.4 The software maintenance organization's key processes, products and operational software have quality and process performance targets.

In this best practice, the maintainer must set quality and performance targets, and these must be established and maintained [13, PA164.IG101.SP103]. The following attributes are necessary [13, PA164.IG101.SP103.N101]:

- Establishment of targets that take into account the business objectives;
- Establishment of targets that take into account previous performance of the activities and operational software;
- Definition of productivity and process execution aspects in order to judge quality;
- Identification of the process limits (by inherent variability or natural limits) on key processes or key maintenance products.

Pro3.5 A measure of a key process, product or operational software for which the software maintenance organization services has a validated reference point (baseline) which is used in analysis, control and improvement follow-up.

The reference points (baselines) of the software maintenance measures must be documented and include some level of detail [13, PA164.IG101.SP104.N101]. For example, it will be necessary to identify:

- The key activity of the process targeted by the measure;
- The sequence of measurement activities;
- How representative of the software maintenance work this measure is.

It is possible to have more than one reference point for the maintenance organization. These different values can represent the performance of different organizational units of the maintainer when they execute a process [13, PA164.IG101.SP104.N102]. A number of measures, covering maintainability and other key characteristics of products, are collected and used to manage and improve the maintenance processes [9, 6.4.1] [7, 3.4.3.4]. Slight adaptations of a normalized process can lead to a reduction in the possibility for comparison of the various organizational units. Some explanation of the adaptations and of the reference point values must be documented to allow for such comparisons [13, PA164.IG101.SP104.N103].

Pro3.6 Models of the software maintenance process performance are established.

Prediction models for software maintenance activities must be developed to achieve this best practice. These models are typically used to estimate, or predict, the results of maintenance processes based on historical and current data. An attempt is made, therefore, to predict the future behaviour of the processes based on current data [13, PA164.IG101.SP105. N101]. For example [13, PA164.IG101. SP105.N102]: The maintenance organization will be able to

use these models to estimate and predict the delivery of services, and of intermediary products and software product versions. The organization can also attempt to evaluate the return on investment of the process improvement activities. Maintenance employees can use the models to estimate, analyse and forecast the effort and cycle times of different maintenance activities. It will also be possible to attempt to forecast the overtime of employees and the availability of software. In addition, the models can be used to help size a modification and to predict failure rates based on the size of the modification [2]. However, it is important to collect enough data to ensure the statistical validity of the analyses.

The use of the models will grow as their prediction capabilities improve and as the data collected is managed. It will be possible, for example, to use them in situations like the following [13, *PA164.IG101.SP105.N102*]: Analysis of process or product data using a formal procedure [7 3.4.3.9, 10]. We should not, however, lose sight of the fact that the models are best used with the key maintenance activities and products, which have a visible impact with customers and stakeholders [13, *PA164.IG101.SP105.N103*].

Pro3.7 Models of software performance are established.

Performance engineering models profiling the operational software under the maintainer's responsibility must be developed and used to achieve this best practice. These models are used to estimate, or predict, the operational performance of software using historical data. An attempt will be made, therefore, to use these models to predict future behaviour using current data [13, *PA164.IG101.SP105.N101*].

More information can be found in the Performance Engineering maturity model [12] in which it is recommended that the following activities be considered at maturity level 3:

- The Performance Engineering process should be taken into account throughout the entire software process, and all available Performance Engineering methods and tools be used comprehensively with regard to existing performance risk.
- Performance-relevant product and resource metrics should be selected for Performance Engineering use and standardized within the organization, and these metrics be stored and managed in appropriate database systems to guarantee a continuous flow of experiences.
- The performance requirements of the customer, which are defined in the system analysis phase, should be used as success criteria in the final inspection test. Furthermore, they should be arranged in service level agreements (SLA) with the provider of the information system.

- An initial organizational structure for the entire PE process should be defined and introduced step-by-step in level 3.

4 Summary

In the global competitive context, organizations feel more and more pressure from their customers who are becoming increasingly exacting in requesting better quality services, more effective services, services at the lowest possible cost and accompanying support and maintenance services that challenge those of the competition. To satisfy quantity, quality and delivery dates demanded by their customers, the organization must have access to software supporting their operations that is reliable, responsive and therefore well maintained. Maintaining and supporting the software of an organization is not an easy task, and software maintenance managers do not currently have access to analytical tools to evaluate the best strategy for improving specific software maintenance activities.

In this paper, we have presented the Maintenance Performance process area of our Software Maintenance Capability Maturity Model. The focus has been to illustrate how Process Performance Measurement practices are interpreted from the maintainer's perspective. The goals and targets of the process areas have been presented, together with the detailed practices from levels 0 to 3 included.

We mentioned also that these practices must be complemented with the *Quantitative Maintenance Management* and *Measurement Decision and Causal Analysis* practices, in order to offer the maintainer a full range of measurement practices. The level 4 and 5 practices are included in our model and can be made available upon request (please e-mail the author of this paper at april_alain@hotmail.com).

References

- [1] A. Abran, H. Nguyenkim, "Measurement of the Maintenance Process from a Demand-based Perspective", *Journal of Software Maintenance: Research and Practice*, 5 (2), 1993, pp. 63-90.
- [2] A. Abran, M. Maya, "A Sizing Measure for Adaptive Maintenance Work Products", *International Conference on Software Maintenance - ICSM-95*, Opio (France), IEEE Computer Society Press, Los Alamitos, CA, 1995.
- [3] A. April, J. Bouman, A. Abran, D. Al-Shurougi, "Software Maintenance in a Service Level Agreement: Controlling Customer Expectations", *Fourth European Software Conference, FESMA*, Heidelberg (Germany), May 2001.

- [4] A. April, D. Al-Shurougi, "Software Maintenance Productivity", ICB/ASAY "The Role of Quality Maintenance in Cost Minimisation Conference, Bahrain, May 27-28, 2002.
- [5] A. April, "Modèle d'évaluation de la capacité à maintenir le logiciel, v0.5", Rapport technique en Génie Electrique, Ecole de technologie supérieure, DGA 1005, Montréal (Canada), 2003.
- [6] 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.
- [7] Camelia, "Modèle d'évolution des processus de développement, maintenance et d'exploitation de produits informatiques - Version 0.5", Projet France-Quebec, , Montréal (Canada), 1994.
- [8] ISO, ISO/IEC 12207:1995 - Information Technology – Software Life Cycle Processes, International Standardization Organization, Geneva, 1995.
- [9] ISO, ISO/IEC/JTC1/SC7/WG18 "Software and System Engineering – Guidelines for the Application of ISO9001:2000 to Software - version 4.1", 31-5-2001, International Organization for Standardization, Geneva, Switzerland, 2001.
- [10] ISO, ISO/IEC 9126:2001, Information Technology – Software Product Evaluation – Quality characteristics and guidelines for their use, Second edition, International Organization for Standardization, Geneva, Switzerland, 2001.
- [11] M. Kajko-Mattsson, S. Forssander, U. Olsson, "Corrective Maintenance Maturity Model: Maintainer's Education and Training", in Proceedings, International Conference on Software Engineering, IEEE Computer Society Press, Los Alamitos, CA, 2001.
- [12] A. Schmietendorf, A. Scholz, "The Performance Engineering Maturity Model", Metrics News, 4(2), Otto Von Guericke University of Magdeburg (Germany), ISSN 1431-8008, December 1999.
- [13] SEI, "Capability Maturity Model Integration for Software Engineering (CMMi), Version 1.1", CMU/SEI-2002-TR-028, ESC-TR-2002-028,

Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburg, USA, 2002.

- [14] M.Zitouni, A. Abran, "A Model to Evaluate and Improve the Quality of the Software Maintenance Process",6th International Conference on Software Quality Conference, Ottawa (Canada), American Society for Quality Control - ASQC- Software Division, 1996.