Towards a Framework for Software Product Maturity Measurement

Mohammad Alshayeb[1], Ahmad Khader Abdellatif[1], Sami Zahran[2] and Mahmood Niazi[1]
[1]: Information and Computer Science Department
King Fahd University of Petroleum and Minerals
Dhahran, 31261 Saudi Arabia
e-mail: {alshayeb, aabdellatif, mkniazi}@kfupm.edu.sa

[2]: Intelligent Consultancy & Training (ICT) Limited, United Kingdom
e-mail: Sami_zahran@hotmail.com

***Abstract*—Capability Maturity Model Integration (CMMI) is a software process improvement model that aims at improving the processes of the software development. CMMI focuses on the "process quality" instead of "product quality". Studies have shown that focusing on "process quality" alone does not guarantee the quality of the produced software, whereas equal attention to product quality is also essential for ensuring the overall software quality. The objective of this paper is to present the initial structure of the framework we propose to measure and assess the software product maturity level. The measure we use for the product maturity is the level of the product compliance with the internal and external quality attributes defined in the stakeholders' requirements. In this framework, we focus on the quality of the product of the process. The proposed framework will help assess the quality of the software product through assessment of the final software deliverable. Successful implementation of the proposed framework will provide a better insight of the software product quality, hence its maturity. We refer to any deliverable code as a product.**

Keywords-Software Product Quality; Software Product Maturity; Product Maturity Assessment; Product Maturity Levels; Product Maturity Model Integration (PMMI); Product Maturity Assessment Method (PMAM).

I. Introduction

The Software Engineering Institute (SEI) of Carnegie Mellon University (CMU) defines the Capability Maturity Model Integration (CMMI) as a process improvement approach that provides organizations with the essential elements of effective processes to improve their software development performance. CMMI process improvement includes identifying the organization's process strengths and weaknesses and making process changes to convert weaknesses into strengths [1]. CMMI consists of best practices that help organizations to improve their software development effectiveness, efficiency, and quality [2].

CMMI defines three constellations, which are collections of best practices and process improvement goals that organizations use to evaluate and improve their processes. These goals and practices are organized into different process areas. The three constellations are:

1. The CMMI for Acquisition (CMMI-ACQ): provides guidance to organizations that manage the supply chain to acquire products and services that meet the needs of the customer.
2. The CMMI for Development (CMMI-DEV): provides process improvement guidance to organizations that develop products and services.
3. The CMMI for Services (CMMI-SVC): provides guidance to organizations that establish, manage, and deliver services that meet the needs of customers and end users.

CMMI aims at improving the process of the software development, however, that does not guarantee the quality of the produced software as the focus in CMMI does not cover "product quality". Previous research have shown that dealing with only "process quality" is not sufficient and that assessment of "product quality" is also required for the improvement of overall software quality [3]. Our proposed framework described in this paper focuses on the quality of the product instead of the process. The quality/maturity of the software product can be assured through the assessment of deliverables of the major phases of the software development lifecycle. Our proposed framework adopts a method for technical product evaluation and quality assessment as the basis for establishing the product's level of maturity. The level of product maturity measured by the degree of its compliance with the internal and external quality attributes defined in the stakeholders requirements. We call this framework Technical-CMMI (T-CMMI). The proposed framework along with the assessment method will: 1) enable software companies to assess their software products to ensure they meet the desired quality before they release it to their clients, 2) enable clients to evaluate the quality of the product before purchasing it and 3) provide the clients with the ability to compare between the quality of different software products.

The rest of this paper is organized as follows: we present the related work in Section 2. In Section 3, we describe the proposed framework. Finally, in Section 4, we present the conclusions and future work.

II. Related Work

In this section, we review the literature on developing maturity models and in software certification for quality assessment.

A. Software Product Maturity Models

Researchers proposed different product maturity models. Al-Qutaish et al. [4] proposed a software product quality maturity model (SPQMM) for assessing the quality of the software product. The proposed model is based on ISO 9126, Six Sigma, and ISO 15026. The model uses the characteristics, sub-characteristics, and measurements of ISO 9126. The values are combined into a single value, which are converted to six sigma. After that, the integrity level of the software product using ISO 15026 is calculated. Finally, the maturity level of the software product is identified. SPQMM is limited to the quality attributes and metrics defined in ISO/IEC 9126 standard.

The EuroScope consortium [5] proposed SCOPE Maturity Model (SMM), a maturity model of software products evaluation. The model has five maturity levels: initial, repeatable, defined, managed, and optimizing. SMM levels 2, 3, and 4 use ISO 12119, ISO/IEC 9126, and ISO 14598 standards. SMM is a measure of the quality in terms of matching stated specifications or requirements; tests are executed to assess the degree to which a product meets the required specifications. SMM requires the process to be documented to ensure the product matches the specifications. Thus, SMM does not focus on the final product quality (code).

April et al. [6] proposed the Software Maintenance Maturity Model (SMmm) However, SMmm focuses only on maintainability. Alvaro et al. [7] proposed a Software Component Maturity Model (SCMM) that is based on ISO/IEC9126 and ISO/IEC 14598 standards. SCMM contains five levels. SCMM depends mainly on the component quality model (CQM). SCMM measures only the maturity of the components and it cannot assess different types of product such as enterprise applications, web-services. Golden et al. [8] proposed the Open Source Maturity Model (OSMM) which helps in assessing and comparing open source software products to identify which one is the best for a defined application. OSMM evaluates the maturity of open source products only without assessing the quality of these software products. OSMM is not primarily used to assess software product quality attributes or product maturity but to help organizations perform a comparison between open source systems.

These three models above either

- Use limited set of quality attributes [4], do not focus on measuring the final software quality [5], or
- Have limited scope [6]-[8].

Therefore, the proposed model will overcome all these limitations.

Our proposed model is designed to be flexible to enable the assessor(s) to define their own set of quality attributes and metrics (based on the stakeholders requirements). In addition, it is generic enough to be applicable to any type of software domain, size or development method.

B. Software Product Certifications

Our proposed model can also serve in certifying software products. Software certification can be granted for different types of software such as final software products [9-13] and components [14]. Certification can be provided by independent agencies, which function like other quality agencies. Involving external agencies in providing the certificate increases the trust in the certification as indicated by Voas [15] "completely independent product certification offers the only approach that consumers can trust". Most of the certification methods are process-based [16], from the process they can determine the quality of the final product. However, certifying the software development process only does not guarantee the quality of the final product [3].

III. Framework for Software Product Maturity Model Integration

In this section, we describe the proposed product maturity assessment framework that can be used to assess the maturity of software products. T-CMMI follows the CMMI approach in defining a reference model and assessment method. T-CMMI consists of two parts:

1. Reference Model that describes the common basis for the assessors to assess the maturity of software products. The reference model describes a scale of the maturity/capability levels of the software product based on its degree of compliance with a set of quality attributes and metrics defined in the stakeholders' requirements.
2. Assessment Method that describes how to use the reference model in assessing the final software product. It also provides guidelines and checklists that help in the assessment process and to ensure a common base of judgment.

Both reference model and the assessment method of the T-CMMI are shown in Figure 1.

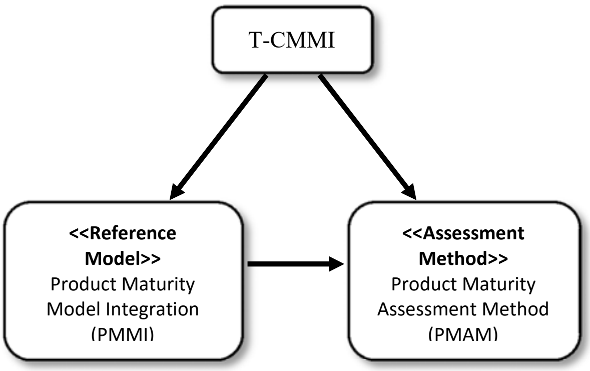
T-CMMI
<<Reference Model>>
Product Maturity Model Integration (PMMI)
<<Assessment Method>>
Product Maturity Assessment Method (PMAM)

Figure 1. T-CMMI Architecture

We adopted CMMI structure for the development of T-CMMI architecture, which contains a reference model and an assessment method. The reference model for the T-CMMI is called Product Maturity Model Integration (PMMI), which contains the capability and product maturity levels. PMMI contains a predefined set of quality attributes and metrics to measure these quality attributes. PMMI adopts the focus-area maturity model structure as opposed to the fixed levels maturity model structure adopted by CMMI. PMMI has two focus-areas, which concentrate on the internal and external quality attributes of the product. The purpose of the Reference Model is to provide a platform and a focus for gathering evidence for product quality indicators that will be used to assess the product maturity level during the Product Maturity Assessment.

The assessment method is called Product Maturity Assessment Method (PMAM). PMAM defines the steps for assessing the final software product against the reference model maturity levels. PMAM contains guidelines and checklists to illustrate how the assessors follow the guidelines in order to measure the capability level and product maturity level for both of PMMI's focus-areas, which concentrate on the internal and external quality attributes. The purpose of the PMAM is to provide a standard method for assessing the product maturity/capability by assessing the degree to which the product conforms to the stakeholders required quality attributes. Below, we discuss these two components in details.

A. Product Maturity Model Integration (PMMI)

PMMI defines a reference model for assessing product maturity and capability. The scope of the PMMI reference model covers integrated view to the end-to-end lifecycle starting with product requirements and ending with product integration, testing and release. The lifecycle is divided into two stages, the DEV stage and the REL stage. These two stages are separate Functional Domains (containing all activities and actors that are involved in the set of activities defined in the development methodology being followed). Each of the DEV & REL stage will have its own Set of Stakeholders and product quality attributes. These two functional domains are defined as follows:

- The DEV stage: covers all the processes and activities for software development, integration and testing (both software unit and software integration testing) of the product. The outcome of the DEV stage is a product ready to be transitioned to the REL stage.
- The REL stage: covers system integration and product pre-release testing

Figure 2 illustrates the PMMI structure showing the DEV and REL stages. Figure 2 shows the main components of each PMMI stage. On the left side are DEV-Stage components, which focus on measuring internal quality attributes, while on the right side are REL-Stage components, which focus on external quality attributes. Product maturity assessment component contains the metrics for each quality attribute that are measured and their results are collected to calculate the capability level for each quality attribute. Then, the capability level of all quality attributes will be fetched into PMMI internal/external quality attributes components. In PMMI internal/external quality attributes component, the weighted average capability values of all quality attributes is calculated to measure the stage maturity level. Finally, the calculated maturity level will be the input to Aggregated DEV/REL Stage Maturity Level component where it is rounded down to calculate the stage maturity level.

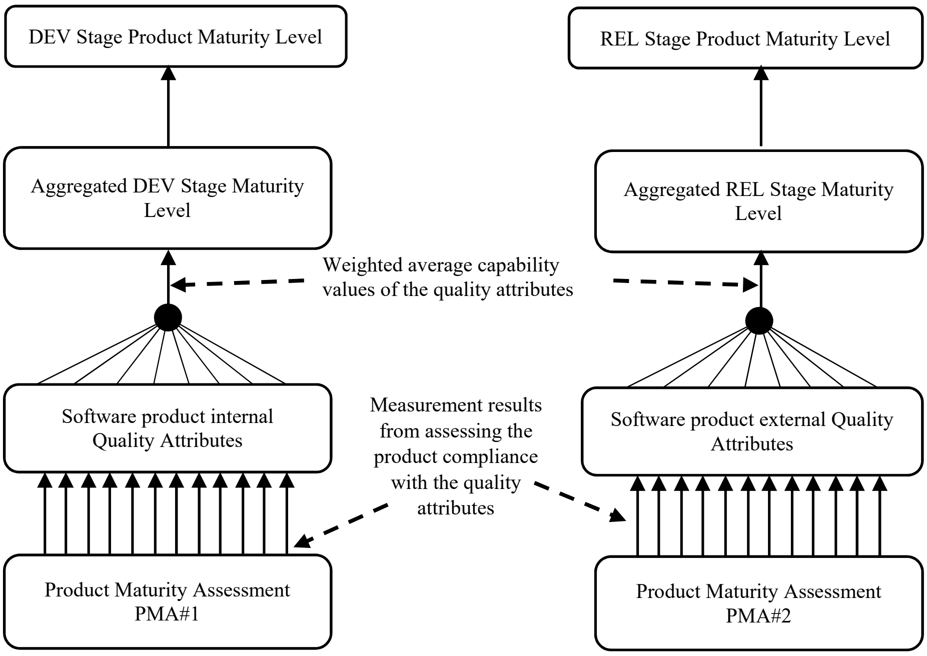
DEV Stage Product Maturity Level
REL Stage Product Maturity Level
Aggregated DEV Stage Maturity Level
Aggregated REL Stage Maturity Level
Weighted average capability values of the quality attributes
Software product internal Quality Attributes
Measurement results from assessing the product compliance with the quality attributes
Software product external Quality Attributes
Product Maturity Assessment PMA#1
Product Maturity Assessment PMA#2

Figure 2. Components of the Product Maturity Model Integration (PMMI)

B. Product Maturity Assessment Method (PMAM)

The PMAM assessment method covers the activities necessary to determine the extent of a product capability to perform in a full compliance with stakeholders' quality requirements. The scope of the assessment is to assess a software product's degree of compliance with the quality attributes defined by the stakeholders (agreed in advance with the assessment sponsor) that covers an integrated view of the end-to-end lifecycle starting with the product and ending with product integration, testing and release. The purpose of the PMAM is to provide a standard method for assessing the level of the product maturity/capability by assessing the degree of the product's conformance with the stakeholders required quality attributes. The PMAM method is compliant with "Guidance on Performing an Assessment" ISO model (ISO 15504-3) [17] framework for software assessment in specifying and defining:

1. Assessment Input.
2. Assessment Process.
3. Assessment Output.
4. Identity of assessment sponsors
5. Identity of Assessors.
6. Responsibilities of each PMAM team member.
7. Expected assessment output and minimum data that should be included in the final assessment report

C. T-CMMI Flexibility

Both components of T-CMMI (PMMI and PMAM) are designed to be flexible and independent of the specific development methodology. In PMMI, assessors can 1) define the quality attributes of interest to the relevant stakeholders with no limits as ISO 9126 defines six attributes only, 2) select the metrics used to measure these quality attributes and 3) define the target capability and maturity levels and their threshold.

PMAM is also designed to be flexible. PMAM process, 1) is applicable to all software domains, 2) can be applied to all software with any size and complexity, and 3) is applicable to all software development lifecycles regardless of the process (or the development methodology) used to build it.

IV. Conclusion and Future work

This paper presented an approach towards developing a software product maturity model. The proposed framework gives the ability to measure the maturity of a software product of any size and domain. It is also applicable to all software regardless of the process used to build it. T-CMMI framework is designed to be flexible, however, assessors can always use the pre-defined set of quality attributes and metrics (which will be supplied with the model) if they wish without customization.

T-CMMI will complement CMMI as CMMI assesses the process quality while T-CMMI assesses product quality. We expect that companies with higher CMMI level should produce better products measured by T-CMMI framework.

In our future work, we plan to complete the development and evaluation of the framework. We will also develop a website to automate the assessment process.

Acknowledgement

The authors would like to acknowledge the support provided by King Abdul-Aziz City for Science and Technology (KACST) through the Science & Technology Unit at King Fahd University of Petroleum & Minerals (KFUPM) for funding this work through project no. 12-INF3012-04 as part of the National Science, Technology and Innovation Plan.

References

[1] SEI-CMU. *Capability Maturity Model Integration*. Accessed (August 2015), Available: http://www.sei.cmu.edu/cmmi/

[2] SEI-CMU. *CMMI Process Areas*. Accessed (August 2015), Available: http://cmmiinstitute.com/cmmi-overview/cmmi-process-areas/

[3] T. Maibaum and A. Wassyng, "A product-focused approach to software certification," *Computer,* vol. 41, 2008, pp. 91-93.

[4] R. Qutaish and A. Abran, "A maturity model of software product quality," *Journal of Research and Practice in Information Technology,* vol. 43, 2011, pp. 307-327.

[5] A. B. Jakobsen, M. O'Duffy, and T. Punter, "Towards a maturity model for software product evaluations," in *Proceedings of 10th european conference on software cost estimation (ESCOM'99)*, 1999.

[6] A. April, J. Huffman Hayes, A. Abran, and R. Dumke, "Software Maintenance Maturity Model (SMmm): the software maintenance process model," *Journal of Software Maintenance and Evolution: Research and Practice,* vol. 17, 2005, pp. 197-223.

[7] A. Alvaro, E. S. de Almeida, and S. L. Meira, "A Software Component Maturity Model (SCMM)," in *Software Engineering and Advanced Applications, 2007. 33rd EUROMICRO Conference on*, 2007, pp. 83-92.

[8] B. Golden, *Succeeding with open source*: Addison-Wesley Professional, 2005.

[9] R. Baggen, J. P. Correia, K. Schill, and J. Visser, "Standardized code quality benchmarking for improving software maintainability," *Software Quality Journal,* vol. 20, 2012, pp. 287-307.

[10] J. P. Correia and J. Visser, "Certification of technical quality of software products," in *Proc. of the Int'l Workshop on Foundations and Techniques for Open Source Software Certification*, 2008, pp. 35-51.

[11] P. Heck, M. Klabbers, and M. van Eekelen, "A software product certification model," *Software Quality Journal,* vol. 18, 2010/03/01 2010, pp. 37-55.

[12] P. M. Heck, "A Software Product Certification Model for Dependable Systems " Eindhoven: Technische Universiteit Eindhoven 2006.

[13] J. H. Yahaya, A. Deraman, and A. R. Hamdan, "SCfM_PROD: A software product certification model," in *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, 2008, pp. 1-6.

[14] A. Alvaro, E. S. d. Almeida, and S. L. Meira, "Towards a Software Component Certification Framework," in *Proceedings of the Seventh International Conference on Quality Software*, 2007, pp. 298-303.

[15] J. Voas, "Developing a usage-based software certification process," *Computer,* vol. 33, 2000, pp. 32-37.

[16] J. Morris, G. Lee, K. Parker, G. A. Bundell, and C. P. Lam, "Software component certification," *Computer,* vol. 34, 2001, pp. 30-36.

[17] ISO/IEC, "15504-3: Information Technology - Process Assessment - Part 3 - Guidance on Performing an Assessment No. 15504-3," 2004.