

# Towards a Maturity Model for Software Product Evaluations

- Position paper on software evaluation and certification -

Allan Baktoft Jakobsen, Michael O'Duffy and Teade Punter

## ABSTRACT

*This paper is about a model that describes the maturity of software products and their evaluation. The model is called the SCOPE Maturity Model (SMM)™. It is based on the assumption that different evaluation levels can be defined for software products. The model provides acquirers and users as well as producers of software a clear overview of the characteristics of product evaluation on a specific level. The maturity model is a 'vehicle' for improving software product quality in a company and reflecting this in the form of evaluation. Moving from one step to the next in this maturity model offers significant added value to software product quality.*

## 1. Introduction

Software is important in our society. Many of our basic processes, like power distribution and transport, depend on software. Industry invests significant resources in software development and its maintenance. Also the number of products that partially contain software is high. An example is a television set, of which the effort spend to develop a new generation consists for more than 70% of software development resources (Rooijmans et al, 1996). And also the commercial software market has evolved in the past twenty years from a niche market into a market place where as many as 2,000 new software packages are released every month (Bresnahan, 1996).

With the growing importance of software, the impact of software failures on society has increased too. An example of the important effect of software failures is the threat of the 'Y2K bug' when entering the new millennium.

An example of a failure was the aborted Ariane 5 launch in 1996. It was 'caused by the complete loss of guidance and attitude information 37 second after start of the main engine ignition sequence – 30 seconds after lift-off. This loss of information was due to specification and design errors in the software of the inertial reference system. The extensive reviews and tests carried out during the Ariane 5 development program did not include adequate analysis and testing of the inertial reference systems or of the complete flight control system, which could have detected the potential failure' (ESA, 1996)

This example shows the importance of evaluation to gain good software product quality. Software product evaluation is defined as the assessment of software product characteristics according to specified procedures, reference ISO/IEC 14598-1 (1999).

Software product evaluation can be conducted in many different ways. Examples are:

- testing of system's usability by using checklists,
- static analysis of software modules to predict system's maintainability
- counting number of faults to predict system's reliability
- acceptance testing to confirm functionality

- conformance checking of software documentation according to ISO 12119 (1994) standard.

Apart from the fact that each of these examples is related to a specific quality characteristic – an overview of characteristics can be found in ISO/IEC FCD 9126 part 1 (version 2 FCD, 1998) - each of these evaluations differs in the intensity of conducting the evaluation. For example: applying a checklist is less intense and will cost less than conducting static analysis. We think that evaluation customers will have their own arguments to design and conduct evaluation(s). To structure the evaluation possibilities, we propose a *maturity model for software product quality and evaluation, the SCOPE Maturity Model (SMM)*.

This model – SMM - was developed by members of the EuroScope consortium. The EuroScope consortium is a network of European evaluators; it was formed in 1993 and includes member organisations from Denmark, France, Hungary, Ireland, Italy, Netherlands, Spain, and United Kingdom.

This consortium has discussed the possibilities of software product certification. During these discussions it was felt that the software market – producers and acquirers/users of software - do not see the benefits of certification. It is usually not obvious to the certification's customer what can be expected of the service and how a certification relates to testing or measuring the software. To address this problem the SMM was defined to situate software product evaluations. Besides the SMM definition, a certification service has been defined called SCOPEmark level-2. It will be conducted on level 2 of the SMM and will be launched in April 1999.

This paper reports on maturity issues in software product quality evaluation in section 2. The levels of the model are presented in section 3. The scheme for SCOPEmark level-2 certification is described in section 4.

## **2. Maturity of software product evaluations**

The objective of the SCOPE Maturity Model (SMM) is to enable evaluation customers as well as evaluating bodies to use a framework for the design and execution of an appropriate software product evaluation.

In software development, the maturity concept is usually related to evaluations of process capability. The CMM (Capability Maturity Model) (Paulk et al, 1993) is the oldest and best-known framework for process assessments and it has inspired other models like BOOTSTRAP (Kuvaja et al, 1994) and SPICE (Kuvaja et al, 1995). Process assessments differ from product evaluation: product evaluations focus on the output of a project, i.e. the project artefacts, whereas process assessment considers the capability of transforming inputs to outputs.

The essential characteristic of a maturity model as extracted from the CMM, is obviously the five capability maturity levels. These levels serve two major purposes: standards for external benchmarking and vehicles for internal improvement. Highly mature behaviour seems to be the ability to interact with the surroundings in a systematic and optimal way. In the CMM, for example, less mature companies are driven by individual vision whereas a profound understanding of their market and business drives more mature companies.

Projecting the concept of a maturity model onto software product evaluations involves introducing maturity levels that serve as external benchmarks as well as drivers for internal improvement of product quality.

### 3. The levels

In this section we introduce the framework for the SCOPE Maturity Model. The purpose is not to reinvent the concept of a maturity model, obviously, but to apply it to *product quality evaluations*, and the CMM will consequently be our primary source of inspiration in doing so.

If a software product is assessed as a *level n* product in the maturity model, this means that it satisfies the demands not only at this level, but also at all previous level *n-1 etc.* Each level will consist of a set of demands to the evaluation process and to the results of the evaluation as it is carried out.

Below, we explain the underlying ideas of each level. The lower levels are based on international standards, but no such standards really exist for the higher and more advanced levels of software quality evaluations.

#### **SMM-1, the initial level**

The initial level is the default level. This means that there are no explicit demands to the degree of product quality at this level. However, if a product is assessed as a level 1 product, the *implicit* requirement is that the product has in fact been evaluated. This makes the initial level an *awareness* level for the model.

Since level 1 means not level 2 or higher, a level 1 product will be characterised by

- no real control regarding the quality,
- few real activities defined that directly relate to software quality: a thorough software product depends on individual effort and heroics,
- no specific reference to requirements, quality in use or risk analysis

#### **SMM-2, the repeatable level**

*Repeatable* product quality means that product expectations as specified in basic requirements are being *reassured* by testing with satisfactory results. The essential ingredient at level 2 is the ISO 12119 standard providing assurance to the user that the product is working as promised and documented. The standard demands that

- basic requirements are specified to a satisfactory degree
- basic testing according to the specification has been carried out.
- documentation is satisfactory

Obviously, the test results must be satisfactory, and faults must be removed through proper regression testing.

The higher levels are characterised by a more interactive view on the product quality.

#### **SMM-3, the defined level**

At level 3, software product quality is addressed as *quality in use* (Bevan, 1997) instead of the quality according to specifications/requirements. *In use* situations are analysed in a *well-defined* way and the results are used to direct the product quality effort.

The essential standard at this level is the ISO 9126 which is used to define the appropriate quality characteristics for the product, this is, in terms of functionality, reliability, usability, efficiency, maintainability, and portability.

Level 3 demands a more proactive role of the evaluator. At level 2, the process of setting up the evaluation is that of looking for some common product artefacts. Setting up the evaluation at level 3 is a dynamic process interacting much more with the product and its context of use. Use of the ISO/IEC 14598 (FDIS, 1999) standard, together with the ISO/IEC 9126 standard, is used to raises the control of the evaluation to a defined level.

**SMM-4, the managed level**

At level 4, the holistic view of product quality is expanded as the evaluation is based on *quantitative measures* and *risk analysis*. Metrics are collected widely to manage the set-up and execution of the evaluation and to measure both the internal and external software attributes using ISO/IEC 9126 parts 2 and 3 (PDTR).

Risk can be addressed by reference to a quality profile of the product. Such a profile describes the relevant quality characteristics and the evaluation levels, which expresses the extent of risk, for the software product. The profile reflects the notion of quality for a certain software product.

ISO 9126 quality characteristics	Evaluation level				
	-	D	C	B	A
FUNCTIONALITY			✓		
RELIABILITY		✓			
USABILITY			✓		
EFFICIENCY	X				
MAINTAINABILITY		✓			
PORTABILITY		✓			

Figure 1 Example of a (summary) quality profile for a software product

**SMM-5, the optimising level**

At level 5, the evaluation process is continuously being *optimised* in order to *improve* software product quality. This means built-in feedback mechanisms based on quantitative measures to optimise the evaluation, as well as a translation of evaluation results to improve software product quality. To illustrate the difference between level 4 and level 5, problems report and coverage measures collected at level 4 are being used proactively at level 5 to evaluate further. The results are then used to improve the software product quality.

**4. SCOPEmark level-2 certification**

SCOPE (Software Certification Of Products in Europe) originates from the ESPRIT project of the same name that determined a method for software product evaluation. This method is now incorporated into the ISO/IEC 15498 standard.

SCOPEmark is the certification and quality marking of a software product. SCOPEmark will be provided at each level in SMM.

Evaluation at SCOPEmark Level-2 aims at ensuring that product expectations are met by testing against basic requirements and achieving satisfactory results. This is carried out by checking product’s conformance to the ISO 12119 standard.

The EuroScope consortium has developed a certification scheme that uses the ISO 12119 standard. The process of this SCOPEmark level-2 certification is presented in the following figure:

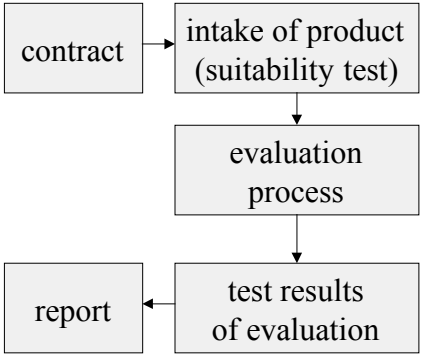


Figure 2 Basic SCOPEmark certification scheme

The contract defines the suitability of the software product to be evaluated for a SCOPEmark level-2. Fail safe systems and mission critical systems are not suitable for certification at this level.

During the evaluation, the product is checked on its conformance to the ISO 12119 standard. Two questions of one of the evaluation checklists are presented below:

Product description		Score	ISO 12119 clause
1	Is a product description available? <i>1: Product description exists,            0: If no product description exists.  <b>If no product description exists, skip all questions of this section: questions 2 - 22.</b></i>		3.1
2	Is the product description free from internal inconsistencies? <i>1: ① Every term should have the same meaning throughout the complete description and ② a method of testing should -concerning all statements in the product description- be defined and ③ in order to define a method of testing it should be possible to identify input, the conditions under which it can be tested, the process to be tested and ④ the output and none of the statements in the product description are identified as incorrect within what is technically possible.            0: If product description does not conform to all of the four statements set before.</i>		3.1.1.

Figure 3 Example of questions of ISO 12119 conformance checklist

SCOPEmark level 2 evaluation focuses upon the clauses about product description, user documentation and programs and data. The ISO 12119 standard also specifies what to test and how test cases, -records and -results have to be documented.

When the evaluation is conducted successfully, a report about the evaluation results is produced. This will be associated with a certificate of attestation. The certification has validity for a limited period of time.

## 5. Conclusion

This position paper presents the basic concepts of a Maturity Model for Software Product Evaluations (SMM) as proposed by the EuroScope consortium. The model can be used as a vehicle to improve software product evaluation in a company.

The paper also presents an overview of the SCOPEmark level-2 certification scheme. This type of software product evaluation should be mapped to level-2 of the SMM.

## 6. References

- Bevan N., Quality and usability: a new framework, in: vanVeenendaal and McMullan (eds.), *Achieving Software Product Quality*, Den Bosch, UTN Publishers, 1997.
- Bresnahan J., Mission Impossible, *CIO Magazine*, October 15, 1996
- ESA, Joint ESA/CNES press release, N33-96, Paris, 23 July 1996
- ISO/IEC 9126, *Software Quality Characteristics*, FCD 1998
- ISO 12119, *Information Technology – Software Packages – Quality requirements and testing*, 1994
- ISO/IEC 14598, *Software Product Evaluation*, parts 1-6, 1998
- Kuvaja P, Similia J, Krzanik L, Bicego A, Koch G, and Saukkonen S, *Software Process Assessment and Improvement: The Bootstrap Approach*, London, Blackwell Publishers, 1994
- Kuvaja P, Bicego A, and Dorling A, SPICE: The software assessment model, in: *Proceedings of ESI-ISCN conference*, Vienna, 1995
- Paulk M, *The Capability Maturity Model*, Addison Wesley/SEI, 1995
- Rooijmans J, Aerts H, vanGenuchten M, *Software Quality in Consumer Electronic Products*, *IEEE Software*, January 1996.