

Software Quality Models: A Brief Survey


AHMED ABDELSALAM



Software Quality

- ❑ **Quality:** is a general term that gives an idea about the degree of correctness of something according to some criteria.
- ❑ It can be measured by having some quantitative or qualitative metrics that are measured in subjective or objective manner.
- ❑ The quality of a software systems was recognized to be an important issue in software development.
- ❑ It has a great impact on the total costs of the software system development

Software Quality

- Are the required functions available in the software ?
 - How reliable is the software system?
 - Is the software system easy to use?
 - How efficient is the software system?
 - How easy is to modify the software system?
 - How easy is the transfer the software system to another environment?
- 

Software Quality

□ ISO-9126 defined a quality model that can be used to measure the quality of any software. This quality model suggested Six quality characteristics that are desired to exist in every kind of software.

- ❖ Functionality
- ❖ Reliability
- ❖ Usability
- ❖ Efficiency
- ❖ Maintainability
- ❖ Portability

Existing Quality models

SOME EXAMPLES OF EXISTING QUALITY MODELS



Example 1

- ❑ **Quality Model:** A model with the objective to describe, assess and/or predict quality.
- ❑ **Quality Meta Model:** A model of the constructs and rules needed to build specific quality models.
- ❑ **Quality Modelling Framework:** A framework to define, evaluate and improve quality. This usually includes a quality metamodel as well as a methodology that describes how to instantiate the metamodel.
- ❑ The software quality models (DAP) can be a definition model (define quality), assessment model (assess the quality of a given system) or prediction models (predict quality)

Example 1

❑ Quality models weaknesses

- ❑ **General:** One of the main shortcomings of existing quality models is that they do not conform to an explicit metamodel. Another problem is that today quality models do not address different views on quality. In the field of software engineering, the value based view is typically considered of high importance. This view is largely missing in current quality models.
- ❑ **Definition models:** Existing quality models lack clearly defined decomposition criteria that determine how complex concepts of quality are to be decomposed. Most definition models depend on a hierarchical decomposition of quality attributes. This decomposition does not follow defined guidelines and can be arbitrary. The ambiguous decomposition in many quality models is also the cause of overlaps between different quality attributes.

Example 1

❑ Quality models weaknesses

- ❑ **Assessment models:** The unclear decomposition of quality attributes is in particular a problem for analytical quality assurance. The given quality attributes are mostly too abstract to be straightforwardly checkable in a concrete software system. The existing quality models neither define checkable attributes nor refinement methods to get checkable attributes, they are hard to use in measurement.
- ❑ **Predictive models:** Predictive quality models often lack an underlying definition of the concepts they are based on. Most of them rely on regression using a set of software metrics. This regression then results in equations that are hard to interpret

Example 1

❑ Quality models usage scenarios

- ❑ **Definition models:** are used in various phases of a software development process. During requirements engineering, they define quality attributes and requirements for planned software systems and thus constitute a method to agree with the customer what quality means. During implementation, quality models serve as basis of modelling and coding standards or guidelines. They provide direct recommendations on system implementation and thus constitute constructive approaches to achieve high software quality. Furthermore, quality defects that are found during quality assurance are classified using the quality model. they can be used to communicate software quality knowledge during developer training or student education.

Example 1

❑ Quality models usage scenarios

- ❑ **Assessment models:** extend quality definition model usage scenarios to control compliance. During requirement engineering, assessment models can be used to objectively specify and control stated quality requirements. During implementation, the quality model is the basis for all quality measurements, i.e. for measuring the product, activities and the environment. This includes the derivation of guidelines for manual reviews and the systematic development and usage of static analysis tools. During quality audits, assessment models serve as a basis of the performed audit procedure.
- ❑ **Prediction models:** are used during project management. More specifically, such models are used for release planning and in order to provide answers to the classical when to stop testing problem.

Example 1

❑ **Quality models requirements**

- ❑ **General requirements:** Any quality model shall define how it can be integrated with the development tasks. These range from requirements engineering to quality assurance tasks.
- ❑ **Definition models:** It is necessary for definition models to have an explicit metamodel that defines the constructs and rules used to build and extend the definition model. In order to clarify the further discussion, this allows to separate the requirements into model and metamodel requirements.

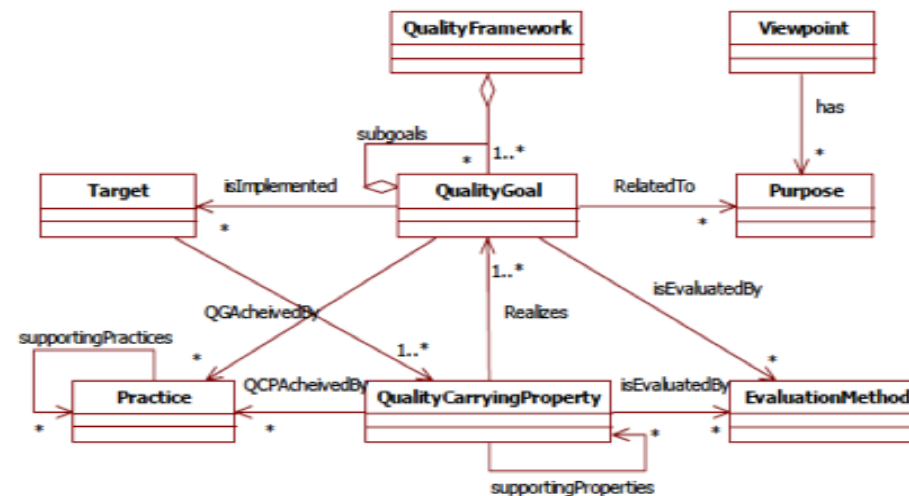
Example 1

❑ Quality models requirements

- ❑ **Assessment Models** : An assessment model also contains quality criteria but in a way suitable for quality assessments. Hence, all quality criteria in an assessment model shall be assessable.
- ❑ **Prediction models:** For prediction models, all requirements to assessment models hold as well. Prediction models are often built using statistical regression of a set of metrics. Depending on the regression method, this can lead to rather complex and artificial equations. For a sensible use of prediction models in practice, however, the equations need to be comprehensible. Hence, the content of a prediction model shall be easy to interpret

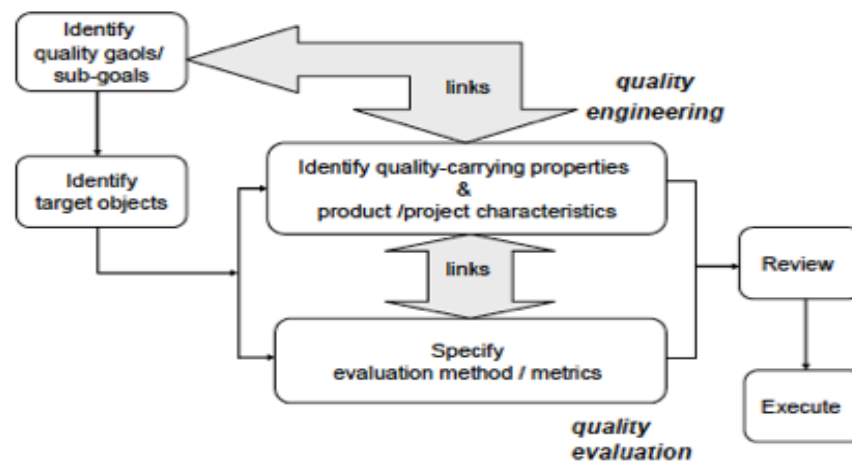
Example 2

- ❑ Quality framework is proposed to evaluate the models used to develop a software system.
- ❑ The metamodel for proposed framework that defines the relations between its main constructs



Example 2

- The required steps to develop a quality framework based on the previous metamodel



Example 2

❑ Tool support

- ❖ The quality framework is modeled using the StarUML, it is a general modelling tool and it is one of the most popular UML tools in the world. But the problem with the StarUML tool that models become rapidly complex by adding links and elements.
- ❖ The authors provided another tool support for the metamodel. The tool will give support developers to specify quality for different domains and different purposes. The tool implemented on the Eclipse platform and using Graphical Model Framework (GMF)

More Examples

- ❑ If you are more interested about software quality models here some useful references that have more examples of quality models..
 - ❖ Wagner, Stefan, and Florian Deissenboeck. "An integrated approach to quality modelling." Proceedings of the 5th International Workshop on Software Quality. IEEE Computer Society, 2007.
 - ❖ Broy, Manfred, Florian Deissenboeck, and Markus Pizka. "Demystifying maintainability." Proceedings of the 2006 international workshop on Software quality. ACM, 2006.
 - ❖ R. G. Dromey. A model for software product quality. IEEE Transactions on Software Engineering, 21(2):146-162, 1995
 - ❖ D. Coleman, B. Lowther, and P. Oman. The application of software maintainability models in industrial software systems. J. Syst. Softw., 29(1):316, 1995.
 - ❖ McCall, Jim A., Paul K. Richards, and Gene F. Walters. Factors in software quality. volume i. concepts and definitions of software quality. GENERAL ELECTRIC CO SUNNYVALE CA, 1977.

Questions



Thanks

