J. of Comp. and I.T. Vol.1(2), 181-185 (2010).
www.compitjournal.org

ISSN 2229-3531 (Print)
2455-9997 (Online)

Using Quality Models to Engineer Quality Requirements

VIVEK MISHRA[1]

[1]I. E. T. Department of M. C .A. Dr. R. M. L. Awadh University Faizabad (INDIA)
(mca.vivek@rediffmail.com)

(Acceptance Date 11th October, 2010)

Abstract

It is known to all that quality requirements are of many kinds. A quality model gives a foundation based on structure. In this model I identify, analyze and give a special name to these quality requirements.

Key words: Quality models, Quality factors, Quality criteria, Quality measures.

Introduction

Quality requirements are critical to the success of any application. Some quality requirements are critical to the success of any application. But which ones? The situation is relatively straight forward with functional requirements. They are relatively uniform in nature and have several well known methods (e.g., functional decomposition and use case modeling) that can be used to successfully identify, analyze, and specify them. But there are a large number of quite different types of quality requirements, and these different types of quality requirements require quite different types of analysis methods.

Quality models :

One role of a quality model, a concept that comes from the quality and measurement communities: to make the general term "quality" specific and useful when engineering requirements. A quality model first decomposes the general concept of quality to create a taxonomy of its component quality factors and subfactors (i.e., aspects, attributes, or characteristics). The quality model then provides specific quality criteria (i.e., descriptions) and measures (i.e., means of measurement) that can be used to turn these general high level quality factors into detailed and specific measurable descriptions that can be used to specify the associated aspect of quality or to determine during testing if that aspect of quality actually exists.

Scope of Quality Models :

The typical scope of a quality model is one or more related applications. Thus, a quality model is used to document or analyze the required or actual quality of an application.

Another common scope for a quality model is one or more related components, whether these components are being bought or developed, either as part of a larger application or as generally reusable components. A quality model can also have another kind of scope. A quality model can be:

- A general industry standard quality model, such as the ISO 9126 quality model[1] or the OPF quality model[2].
- An organizational quality model, such as one that is used on all projects within a given development organization (e.g., all applications within a program or related projects or a product line of applications).
- An endeavor-specific quality model, such as one that is used on a single project developing a single application.

Components of a Quality Model :

As illustrated in the following figure, a quality model consists of a hierarchy of the following components:

- **Quality Factor Groups**, which are groups of related quality factors.
- **Quality Factors**, which are high-level aspects, attributes, or characteristics of an application, component, center, or business organization.
- **Quality Subfactors**, which are lower-level quality factors that are components of other quality factors or subfactors.
- **Quality Criteria**, which are specific descriptions that provide evidence either for or against the existence of a specific quality factor or subfactor.

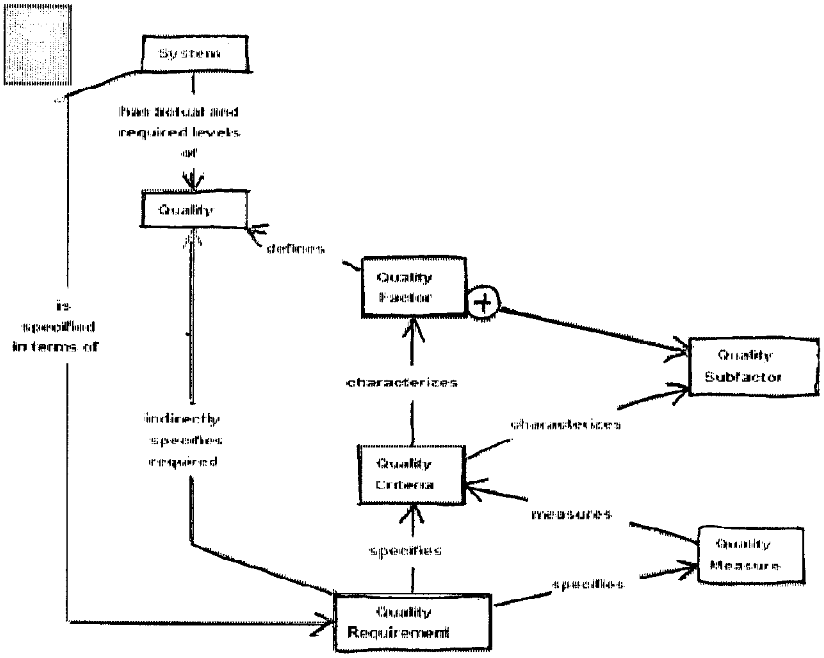

Fig. 1. Relationships Involving Components of a Quality Model

- **Quality Measures**, which are metrics that quantify quality criteria and thus make them measurable, objective, and unambiguous. Therefore, producing and documenting a quality model is an endeavor's first step towards producing a complete and consistent set of quality requirements.

Quality Factor Groups :

A quality factor group is a major grouping of related quality factors. Quality factor groups provide the highest-level decomposition of quality models into their component parts.
For example, the quality factors might be grouped as follows:

- **Developer-Oriented Quality Factors**, which are quality factors that primarily affect members of the development and maintenance organizations during the current or future development of an application or component.

- **User-Oriented Quality Factors**, which are quality factors that primarily affect members of the user organizations during actual usage of an application or component.

Quality Factors :

A quality factor (a.k.a., quality attribute) is an important high-level aspect, attribute, or characteristic of the quality of one or more work products (e.g., application, component, or documents). A quality factor thus characterizes one part of the overall quality. There are many different quality factors such as availability, extensibility, performance, reusability, security, or usability. Because many of the quality factors end in the letters "ility", they are often referred to as the "ilities". Notice that quality factors are typically written either as individual nouns or as noun phrases.

Quality Subfactors :

A quality subfactor is a quality factor that is a component part of another quality factor or higher-level quality subfactor. As more detailed kinds of quality factors, quality subfactors are also important high-level aspects, attributes, or characteristics of one or more work products (e.g., application, component, or documents) that characterizes part of their overall quality. As quality factors, they are also typically designated with nouns or noun phrases.

Quality Criteria :

A quality criterion is a specific description that provides evidence either for or against the existence of a specific quality factor or subfactor. Thus, quality criteria go a long way towards making the high-level quality factors detailed enough to be unambiguous and testable. They lack only the addition of quality measures to make them sufficiently complete and specific to form the basis for detailed quality requirements.

Quality Measures :

A quality measure is a metric that quantifies a quality criterion. Quality measures thus provide numerical values specifying or estimating the quality of a work product or process by measuring the degree to which it possess a specific quality factor or subfactor. The measurement community has published many documents[3] concerning how to measure the quality criteria of the various kinds of quality factors.

Quality Requirements :

To specify a quality requirement is to specify a minimum acceptable amount of a quality factor, and this is done by specifying a minimum amount of a quality measure for a quality criterion for that quality factor.

Engineering quality requirements using quality models :

To engineer the quality requirements for an application, one could perform the following process in an iterative, incremental, parallel, and time-boxed manner:

1. *Select Quality Model:* Select a relevant quality model on which to base the quality requirements. If possible, reuse an existing quality model. Extend or tailor the quality model as required.

2. *Select Relevant Quality Factors and Subfactors*: For each quality factor and subfactor in the quality model, determine its relevance to the current endeavor. Consider the functional, data, and interface requirements. Brainstorm with stakeholder representatives and subject matter experts.

3. *Produce Quality Criteria:* For each quality factor and subfactor, evaluate the relevant functional, data, and interface requirements to identify/determine associated quality criteria. Where practical, reuse parameterized quality criteria types that can be instantiated to produce endeavor-specific quality criteria. Brainstorm with stakeholder representatives and subject matter experts.

4. *Select Related Quality Measures:* For each relevant quality factor and associated quality criteria, determine the appropriate quality measure.

5. *Specify Quality Requirements:* For each quality criteria and associated quality measure for each quality factor and subfactor, determine a minimum acceptable amount of that quality measure for that quality criteria2 and specify the associated requirement using a standard format.

6. *Evaluate the Quality Requirements:* Validate the resulting quality requirement with its stakeholders. Verify the quality of the quality requirement against associated standards and guidelines (e.g., quality characteristics such as completeness, feasibility, implementability, lack of ambiguity, testability, and understandability, etc.). Ensure that the resulting requirements specifications / repositories are complete in the sense that adequate quality requirements exist for all relevant quality factors and subfactors in the quality model.

Conclusion

Quality requirements can and should be based on a standard quality model with standard quality factors, subfactors, criteria types, and measures. Using such a quality model produces the following benefits:

- All important and relevant types of quality requirements are engineered, resulting in a more complete requirements specification.

- The resulting quality requirements are organized into a logical and understandable hierarchy that is easy to use and learn.

- Communication among stakeholders regarding the quality requirements uses standardized terms with clearly documented meanings.

- The quality requirements properly capture the different types of quality.

Acknowledgement

I deem it my inevitable duty to express my heartily gratitude to Dr. Vinodani Katiyar, Professor in the Department of Computer Science, SIMT, Lucknow, India.

References

1. ISO/IEC 9126-1: Information technology - Software quality characteristics and metrics- Part 1: Quality characteristics and subcharacteristics, International Organization for Standardization, International Electrotechnical Commission, Geneva (1991).
2. OPEN Process Framework Quality Model available at http://www.donald-firesmith.com/Components/WorkProducts/ModelSet/ QualityModel/QualityModel.html
3. ISO/IEC 9126-2: Information technology - Software quality characteristics and metrics- Part 2 : External metrics, International Organization for Standardization, International Electrotechnical Commission, Geneva, (1991).