A Software Architecture Evaluation Model[1]

Juan C. Dueñas, William L. de Oliveira[2], Juan A. de la Puente

Department of Engineering of Telematic Systems,
Technical University of Madrid
ETSI Telecomunicación, Ciudad Universitaria, s/n, E-28040 Madrid
E-mail: {jcduenas, william, jpuente}@dit.upm.es

Abstract: The fulfilment of quality requirements is fundamental for the success of software-intensive systems. This fact forces companies to quantify the quality requirements at the moment of their specification, and to evaluate these requirements in all the results of the design process, both the by-products and the end system. The definition of the software architecture is one of the most important and early decisions of the design process, with a strong influence on the final quality of the product; therefore its evaluation should be made as early as possible, before the design is complete. This paper presents a software architecture evaluation model considering the software architecture as a final product itself and also as an intermediate product of the design process.

1. Introduction

The quality of equipment, services and systems is an essential element for the competition in the global market. Customers ask for increasingly more complex and demanding systems that must meet high quality standards. This fact drives software development companies to monitor the quality of both their products and their processes. Quality consists in the fulfilment of the system requirements, although as important as the "functional" aspect, is the fulfilment of other characteristics of its operation, more difficult to evaluate, such as the resource efficiency, the scalability, the maintainability, et cetera, that usually receive the name of "quality requirements".

[1] This work is partially funded by the ESPRIT program under contract EP20477, ARES project (Architectural Reasoning for Embedded Software), and by Spanish CICYT, under contract TIC96-0614.

[2] The work of William L. de Oliveira in the DIT/UPM is partially funded by CNPq-Brasil, with reference 260.022/95-0 and COPEL/LAC-Brasil.

The scientific and industrial communities recognise that, in the case of software intensive systems, the first phases of the design and the decisions made then have a fundamental impact on the final quality. Not only because the errors made in the design cycle have more expensive reparation as they propagate, but because the early decisions will impose a strict threshold for the capacities and quality of the final system, especially in relation to these no-functional requirements. Being one of the first decisions made, the Software Architecture (SA) must be checked against these quality requirements. This approach is just to be applied in companies that design software systems with strict quality requirements.

A brief definition given by Garlan and Perry [9] establishes that SA is “the structure of components in a program or system, their interrelationships, and the principles and guides that control the design and evolution in time”. This reveals the most important issues of SA:

- The focus on structural information, instead of algorithmic or behavioural models, because the approach is based on the definition and usage of high abstraction level components and their relationships (connectors in architectural terms).
- The information about “what to do” and “when” (i.e. the development process), is still part of the SA. This includes the know-how of the company about the product line.
- All the information included in the SA is oriented towards controlling the product evolution in time.

The draft-standard ISO/IEC 14598-1 specifies the elements required for quality evaluation of software: the quality model, the method for evaluation, the metrics and the supporting tools. Thus, in order to develop good software, quality requirements must be specified, the software quality assurance process should be planned, implemented and controlled, and all intermediate as well as end products must be evaluated. A suitable method to perform objective software quality evaluations is the measurement of the quality attributes of the software. This approach will be applied to the architecture of software systems: this article presents a quality model, methods for evaluation and metrics adapted to the assessment of SA: in the second section the general software quality assessment process is defined and in the third one the SA evaluation model. The article finishes with some comments about the current and future work in this area.

2. The General Software Quality Assessment Process

The result of any software development phase (including the last one) can be considered as a software product, with its own requirements that express the needs of its users, and that must be defined prior to the development. Besides, in the same way that global requirements evolve with respect to the development phase, they are also partitioned following the software product decomposition into its major components (which may require different evaluation criteria). An example of this at the SA level is the need for different quality requirements for the critical components in a hard real-time system, whose safety and predictability requirements are more strict than the requirements for the logging subsystems. Therefore, in order to keep consistent the quality specification for each subsystem with the one of the final product and throughout all the phases of the development process, quality requirements need to be specified in terms of a common quality model.

One available quality model is defined in the standard ISO/IEC Draft 9126-1 – Information technology – Software quality characteristics and metrics Part 1: Quality characteristics and sub-characteristics [7]. This part of ISO/IEC 9126 specifies a quality model that categorises software quality into six characteristics, which are further divided into sub-characteristics. These are defined by means of externally observable attributes for each software system. In order to ensure its general application, this standard does not cover which are these attributes, nor how can they be related to the sub-characteristics.

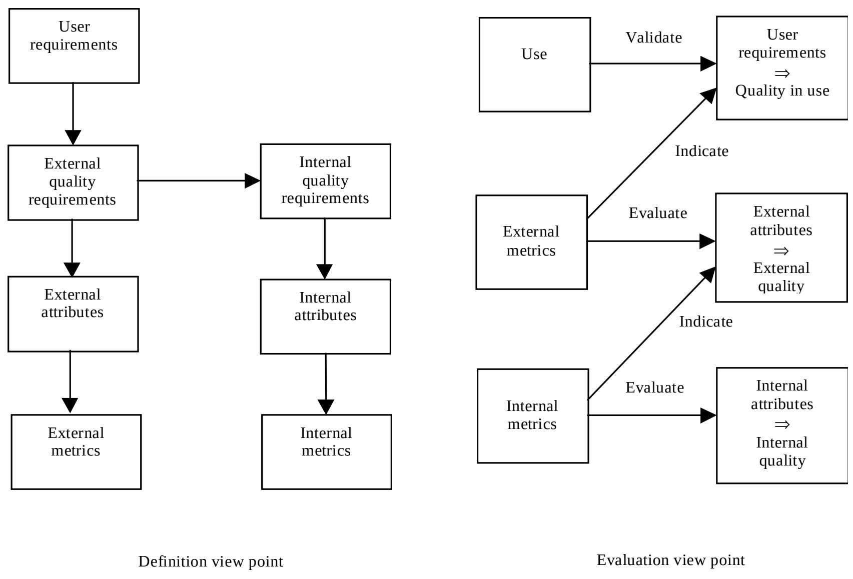


Figure 1: General quality process.

Once the quality model has been chosen, the general quality process starts (see Figure 1). The user requirements for the product (the result of any development phase) are studied by the development team, who specifies the product external quality requirements for each relevant quality characteristic, in order to establish the extent to which a product satisfies the user needs when used under specified conditions. A specific set of externally observable attributes (such as response time is used to define the efficiency of a computer system) are useful for this purpose. The completeness and correctness of the quality requirements specification need to be evaluated then to ensure that all the necessary requirements have been specified and unnecessary requirements excluded. The developers will evaluate the product against these requirements before delivery.

The external quality can be assessed by means of the external metrics applied under usage conditions. The external evaluation of the quality characteristics should therefore take place under conditions that emulate as closely as possible the expected conditions of use. It is important at this point to note that "metrics" in this context denote any kind of quantitative result that follows certain conditions about allowed values, scale and the measurement procedure. For example, the results of simulation, provided those items, can be categorised as metrics.

At next, the developers must specify the internal quality requirements of the product. The internal quality is the whole set of internal attributes of a product that determine its ability to satisfy its needs. The difference between external and internal attributes is that those reflect the influence of the environment of usage of the product, while internal ones focus only on intrinsic properties of the product (such as modularity and complexity), more related to its structure and to the way it is being built. In any case, the specification of the internal quality should use the same quality model used for the external quality specification, so the external and internal attributes are kept coherent.

The internal quality of each product can be evaluated by internal metrics, which measure these internal attributes. One of the most important usages of the internal quality evaluation is to help in choosing and improving the development process for a product line. Another main usages are the evaluation of internal attributes for technical decision making on several building options during development, and their usage like external quality indicators.

3. The Software Architecture Evaluation Model

Following these general considerations for the development and evaluation process, the specific process for the SA evaluation, shown in Figure 2, is based on the consideration that SA is a software product, and it should follow and adapt the general software quality process. In the next description, the taxonomy used for metric, measure, measurement, quality and attributes is that defined in [7].

3.1 Quality Specification

Following the definition given by the standard ISO 8402, the quality specification must contain the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs. The user view is expressed in the external quality specification and the developer view is expressed in the internal quality specification.

The external quality specification

The users and the developers specify the external quality of the final product or system and the external quality of the SA using the quality reference model presented in ISO/IEC 9126-1. The specifications must contain the selected quality characteristics and their optimum and allowed values under the user viewpoint. These requirements will be evaluated before the product delivery through testing techniques.

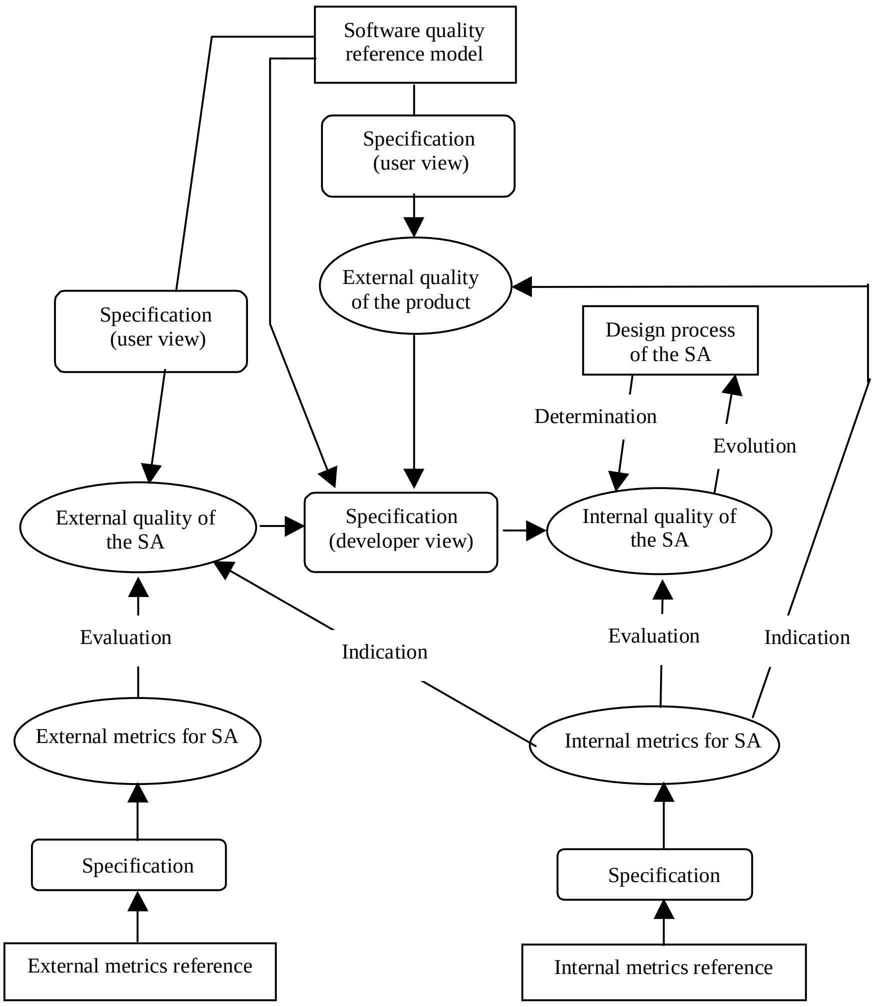

Software quality reference model
Specification (user view)
External quality of the product
Specification (user view)
Design process of the SA
Determination
Evolution
External quality of the SA
Specification (developer view)
Internal quality of the SA
Evaluation
Indication
Evaluation
Indication
External metrics for SA
Internal metrics for SA
Specification
Specification
External metrics reference
Internal metrics reference

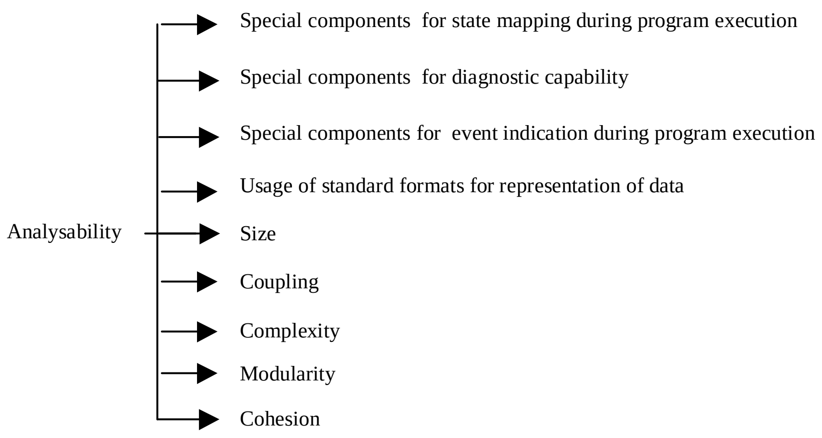
Figure 2: Quality evaluation model for SA.

The internal quality specification

The developers specify the internal quality of the SA based on the external quality of the product and the external quality of the SA, using the same quality reference model (ISO/IEC 9126-1) and by means of the definition of a set of software internal attributes. The specification must contain the software internal attributes selected for each quality characteristics and also their optimum and allowed ranges of values.

The developers need to decide how to particularise the specified quality requirements on the SA. For this purpose, they need to map these quality requirements to internal attributes that will be present on the SA. The mapping is based on the expert's knowledge or company accumulated data, depending on the maturity of the process implanted on the company.

The internal attributes are composed by special elements (such as functional elements or data elements) denoting quality characteristics, and intrinsic properties resulting from the development process (such as size, modularity, complexity, coupling and cohesion). For example, considering for analysability the following correspondence with internal attributes, the developers need to establish the relative importance between these attributes and their values. Some techniques such as the Quality Function Deployment (QFD) [8] are suitable for this purpose.


Analysability
Special components for state mapping during program execution
Special components for diagnostic capability
Special components for event indication during program execution
Usage of standard formats for representation of data
Size
Coupling
Complexity
Modularity
Cohesion

3.2 Metrics Specification

The metrics specification must contain: the selected measure for each specified quality characteristic and internal attribute, a measurement scale, and the set of available methods for measurement, including procedures to categorise qualitative data. The metrics will be external when evaluate the external quality or internal when evaluates the internal quality.

A particular measurement is useful when the measure helps in understanding a process, its resources, or any of its products [3]. Then, the assessment team must choose some of the available metrics (some will be presented in this section); or define new metrics following a certain reasoning process, such as the Goal-Question-Metric (GQM)[2], composed by several activities:

- The first step is to define the goal in terms of purpose, perspective and environment. In the present case, the purpose is related with the SA quality evaluation, indication and comparison, and the end product quality

prediction. The perspective depends on the aims of the assessment and it is closely related to the role of the evaluation staff: developer, user, management, maintainer, et cetera. There are two suitable environments: the SA representation considered like an intermediate design product or like an end product itself.
- The second step is to establish the questions that indicate the attributes related with the goal.
- In the third step each question is answered, and a proposal for new or existing metrics is carried out.

The external metrics specification

The purpose of the external metrics is to provide data for the evaluation of SA as a product in itself, not related to the final product, but focusing on its usage during the development; its users are often the staff of the next development phase. External metrics depend on the real use of the SA (for example, for evolution analysis of a product line), so it has to be evaluated as part of a working environment for the intended usage. One case of external measure applicable to the SA for measuring its analysability is:

Mean time for analysis =
Sum of times between analysis request and its execution /
Number of analysis to be carried out

A common case of analysis is to trace the fulfilment of a certain function point in the SA.

The internal metrics specification

The purpose of the internal metrics is to provide data for the software product evaluation regardless its environment. Internal metrics provide users, evaluators, testers, or developers the chance to evaluate the product quality before its usage. Then, internal metrics can be used:
- to evaluate the SA internal quality.
- to indicate that the software satisfies external quality requirements.

The measurement of internal metrics often use figures of amount or frequency of appearance of special software elements in the product representation (such as graph representation or tables of control, data flow, state transition structure or documentation) and therefore the figures can be obtained without execution. However, few available metrics for SA have been found in the literature. Thus, some common metrics defined for design representation and source code must be adapted to the elements that appear in SA representations, that are expressed using an Architectural Description Language (ADL) [3] [4] [5] [6]. One common measurement procedure is the architectural walkthrough using the ADL model and the description of its components.

Some examples of measures that detect the presence or absence of special elements in the SA are:

- Security measure:
  Data encryption ratio =
  Number of data components defined with data encryption-decryption facility in the SA /
  Specified number of data components requiring data encryption-decryption facility in the internal quality specification.
- Analysability measure:
  Diagnostic functions ratio =
  Number of diagnostic functional components in the SA /
  Specified number of diagnostic functions in the internal quality specification.
- Changeability measure:
  Parameterisation ratio =
  Number of parameterisation data components in the SA /
  Specified number of parameterised data components in the internal quality specification.

The intrinsic properties (such as size, modularity, complexity, coupling and cohesion) are determined by combination of measures directly applied on the SA representation. These measures can be further divided into those referring to the number of interface and implementation elements, and those about the SA configuration, defined as counts of graphs and interaction elements in the model.

3.3 Quality Evaluation

The quality evaluation consists of the data collection, the measurement and the analysis of results.

The external quality evaluation

For external quality evaluation, the goal of metrics is to find the values specified in the external quality specification for each software characteristic. Then, an external quality specification must exist which defines the expected software characteristics with their values. The measures are applied and the results are compared with the expected values.

On the other hand, if there are no available specifications for external quality, the results of application of internal metrics on the SA can be used as an indication or foresight of the presence of a certain quality characteristic in the end product. The metrics are applied and the results identify the presence or value of an internal characteristics set (such as the presence of special function or size) that can be associated with some external quality characteristic (such as modularity and diagnostic function can be associated with maintainability).

The internal quality evaluation

For the internal quality evaluation, the goal of metrics is to find whether certain internal attributes meet the values specified in the internal quality specification for each software characteristic. Then, a previous internal quality specification must exist which defines the expected internal attributes with their values and their evaluation procedures. The measures are applied and the results are compared with the expected values.

The presence or absence of special elements is, usually, detected by several questioning techniques [1] (such as scenarios, questionnaires and checklists) and inspection techniques (such as SA models walkthrough). This information, usually, is not formalised in an ADL, but it is attached to the ADL model in the form of annotations, attributes or natural language descriptions. The intrinsic properties, however, can only be detected by measuring techniques [1] applied on the SA representation formalised through an ADL.

The SA development process used constraints the internal attributes of the SA, so the measurement result can be used as feedback for the improvement of the SA development process. Another peripheral use for software internal attribute measures is to normalise external measures in order to allow the comparison between SAs. For example, using the internal attribute "size" for normalisation of analysability measures:

Analysability metric normalised by size:
Density of time analysis = Mean time analysis / Size

4. Conclusion and Future Work

This paper presents a research work whose aim is to establish the basis for the SA quality evaluation and prediction of the final system quality, since it is accepted that SA, as a product of the design process, has a great influence onto the final product quality. For this purpose, a quality model based on well-know standards has been chosen, and a conceptual framework that relates quality requirements, metrics and internal attributes of the SA and final system has been proposed.

Additionally, a rigorous formalisation of the evaluation process of the quality requirements of the SA, especially in relation to metrics has been advanced. Although it is an especially novel work because of youth of the work area, the adoption of design models based on components (strongly based on the SA approach) will show in the future more work in the same line.

Same aspects that still need elaboration are:

- The formalisation of the relationship between internal quality attributes and quality characteristics/sub-characteristics, which must be studied for specific application domains, development processes and ADLs. QFD will be used to organise and identify the weighted relationship between quality characteristics and internal quality attributes, allowing to make the technical decisions more appropriate, even in presence of contradictory quality requirements and attributes.

- The definition of internal metrics for the internal attributes measurement. As mentioned, GQM will be used to identify the metrics from goals established with respect to internal quality attributes through a reasoning process.

An application experiment of these concepts for architectonic evaluation in the telecommunication system area is being carried out, after a previous phase of architectural recovery. Some metrics are being applied on the obtained SA. The measurement results will also be evaluated by system experts.

This work does not pretend to establish the optimal values for the internal quality attributes, because they are context-dependent, but to formalise a framework that provide the means to meet these values and the way for predicting the quality of the final product from the SA evaluation.

5. References

[1] G.Abowd, L.Bass, P.Clements, R.Kazman, L.Northrop, A.Zaremski, *Recommended Best Industrial Practice for Software Architecture Evaluation,* Technical Report CMU/SEI 96-TR-025 (1997).

[2] V.R.Basili, H.D. Rombach, *Goal/Question/Metric Paradigm: The TAME Project: Towards Improvement-Oriented Software Environments*, IEEE Transactions on Software Engineering, Vol. 14, n. 6 (1988).

[3] N.E.Fenton, S.L.Pfleeger, *Software Metrics – A Rigorous & Practical Approach – Second Edition,* International Thomson Computer Press (1997).

[4] IEEE Standard, *IEEE Std 1061 – Standard for a Software Metrics Methodology* (1992).

[5] IEEE Standard, *IEEE Std 982.1 – Standard Dictionary of Measures to Produce Reliable Software* (1988).

[6] IEEE Standard, *IEEE Std 982.2 – Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software* (1988).

[7] ISO/IEC Standard, *ISO-9126 Software Product Evaluation - Quality Characteristics and Guideline for their Use* (1991).

[8] B.M.Reed, D.A.Jacobs, *Quality Function Deployment for Large Space Systems,* National Aeronautics and Space Administration, (1993).

[9] M. Shaw, D. Garlan, *Software Architecture. Perspectives on an Emerging Discipline,* Prentice Hall 1996.