An Evaluation Framework for EU Research and Development e-Health Projects' Systems*

Androklis Mavridis, Stamatia-Ann Katriou, and Adamantios Koumpis

ALTEC S.A, M. Kalou St 6, Thessaloniki, 54629, Greece
Tel.: +302310595646; Fax: +3023210595640
{mavr,akou}@altec.gr

Abstract. Over the past years it has become evident that an evaluation system was necessary for the European Research and Competitive funded projects which are large and complex structures needing constant monitoring. This is especially so for e-Health projects. The race to complete assignments means that this area is usually neglected. A proposed framework for the evaluation of R & D project systems using ATAM, ISO 14598 and ISO 9126 standards is presented. The evaluation framework covers a series of steps which ensures that the offered system satisfies quality, attributes such as operability, usability and maintainability imposed by the end users. The main advantage of this step by step procedure is that faults in the architecture, software or prototype can be recognised early in the development phase and corrected more rapidly. The system has a common set of attributes against which the various project's deliverables are assessed.

Keywords: Evaluation, e-Health, Research and Development projects, Software.

1 Introduction

In most European Research and Competitive funded projects, the strict time plans and time limitations often lead consortia to focus mostly on delivering the proposed system without providing proper justification of the system's quality, the appropriateness and overall acceptance by the involved stakeholders. The dedicated resources spent on a system's evaluation tasks are only adequate for software evaluation and proof of concept through end users' participation in prototypes, which often happens too late in a project's life cycle. This lack of proper evaluation often hinders a project moving from research into its applied form. This is especially important when coping with the peculiarities of e-Health projects.

With the cooperation of EU partners, ALTEC has developed a unified framework aiming to evaluate the proposed architecture, the developed software, and the prototypes offered to end users of the systems that are the product of European Research and Development projects. These projects, funded by the EU,

* eHealth 2008, September 8th and 9th, 2008, City University, London EC1.

D. Weerasinghe (Ed.): eHealth 2008, LNICST 1, pp. 9–16, 2009.

consist of large and complex Research structures, which involve a constellation of members, including universities, research centres and business companies who cooperate collectively on each research project.

When the expected research outcome is software the main evaluation task is to test the offered functionalities, but few, if any, resources are spent on evaluation tasks, as the consortium is usually limited by time constraints ordered by strict work plans. Yet there is the need to ensure that each offered system satisfies quality attributes, such as operability, usability, maintainability, etc, imposed by end users.

The framework for evaluating these complex structures was developed while ALTEC was participatiing in the SAPHIRE project (IST- 27074 SAPHIRE “Intelligent Healthcare Monitoring based on a Semantic Interoperability Platform” PRIORITY 2.4.13 Strengthening the Integration of the ICT research effort in an Enlarged Europe Focus: eHealth). The SAPHIRE project was to develop an intelligent healthcare monitoring and decision support system on a platform integrating the wireless medical sensor data with hospital information systems. The resulting system is employed on two pilot medical prototypes, namely the Homecare and Hospital Prototypes and the operation involves real patients and real healthcare data to be handled by the system. It was thus, a major requirement to have our system evaluated in order to assure not only its intended functionality but also its acceptance by the end users, which in this case, are the patients and the medical staff operating the system.

2 Aims and Objectives

Our aims and objectives were to assess the various deliverables of the project against a set of quality goals ordered by the system’s stakeholders. To achieve this we employed the Architecture Tradeoff Analysis Method (ATAM) [1] method and the ISO 14598 and ISO 9126 2-4 standards [2].

For the evaluation of the SAPHIRE system, our objectives were:

1. To provide a unified evaluation framework able to accommodate the evaluation of architecture, software and developed prototypes.
2. To have this framework as generic as possible - not focusing only to the specific needs of the SAPHIRE project - to adopt to other projects easily without modifications.
3. To involve the end users in the evaluation process as early as possible by stating their true requirements from their perspective.
4. To exploit the results from the evaluation steps early on, providing valuable assistance to the development team.

The evaluation process follows a sequence of steps starting with the evaluation of the system’s architecture under selected quality attributes. It then moves on to examine the pieces of software developed against a set of quality criteria, and finishes by evaluating the final integrated prototype through the employment of scenarios and metrics desired by end users.

3 Methodology

The proposed framework performs three different evaluations, namely the system's architecture, software and prototypes evaluation. Architecture Tradeoff Analysis Method (ATAM) was used for the architectural part, and the ISO 14598 and ISO 9126 (2-4) for the software and prototypes evaluations (Figure 1). The step sequence is described in the following paragraphs.

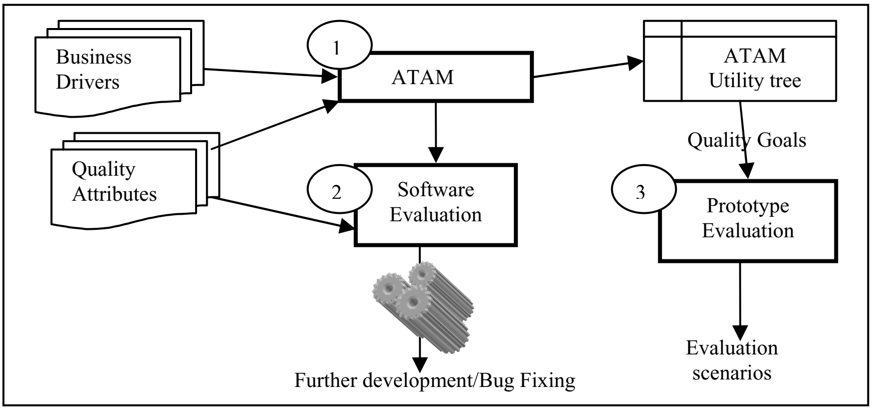

Fig. 1. The evaluation steps

3.1 STEP ONE: Evaluating the System's Architecture

For evaluating the system's architecture we employ the recognised methodology, ATAM. Apart from offering the most complete and assistive approach [3], ATAM ideally fits in our framework as it is driven by quality attributes that must be met. ATAM reveals how well an architecture satisfies particular quality goals (such as performance or modifiability), and provides insight into how those quality goals interact with each other—how they trade off against each other. Such design decisions are critical. By using this methodology, poor architecture is exposed early in the developmental sequence.

ATAM focuses on quality attribute requirements. Quality attribute characterisations answer the following questions:

1. What are the triggers/stimuli inputs to which the architecture must respond?
2. What is the measurable or observable definition of the quality attribute by which its achievement is judged?
3. What are the key architectural decisions that impact on achieving the attribute requirement?

The consequence of using the ATAM is a clarification and concretization of quality attribute requirements, achieved in part by eliciting scenarios from the stakeholders that clearly state the quality attribute requirements in terms of triggers and responses. The process of brainstorming scenarios also fosters stakeholder communication and consensus regarding quality attribute requirements.

Scenarios are the second key concept upon which ATAM is built. Based on these scenarios and refinements of quality attribute goals the team builds the quality utility tree. Utility trees translate the business drivers of the system under examination into concrete quality attribute scenarios. For example: "security is central to the success of the system since ensuring the privacy of the patients' data is of utmost importance"; and "usability is central to system's acceptance since we need to assure the patients' satisfaction."

Before assessing the architecture, these system goals must be made more specific and more concrete. The team needs to understand the relative importance of these goals versus other quality attribute goals, such as performance, to determine where we should focus our attention during the architecture evaluation.

The primary aim of ATAM, is to record any risks, sensitivity points, and tradeoff points that may be found when analyzing the architecture. Risks, sensitivity points, and tradeoff points are areas of potential future concern with the architecture. The output of this first step is a list of quality attributes and the scenarios identified in the utility tree. These, feed the next step of software evaluation.

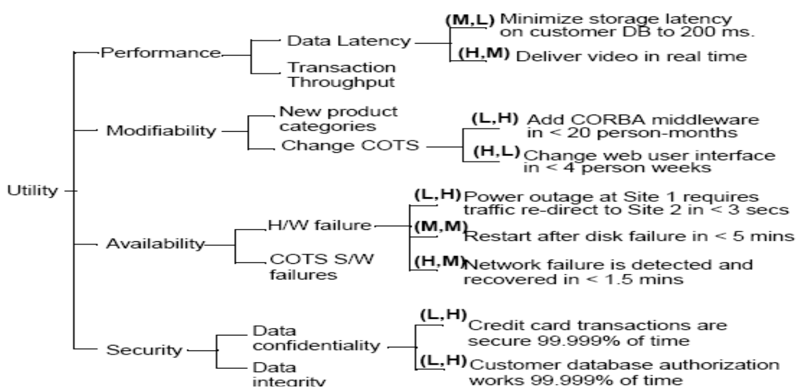

Fig. 2. An example of a Utility Tree

3.2 STEP TWO: Evaluating the System's Software

To evaluate the software in our framework, the ISO 14598 standard is used, providing an overall software evaluation quality model. This model orders how, when, whom and what is to be measured, defining as the primary tools for assessments the Quality in use measures. The process as adopted from the ISO 14598 standard involves the use of quality characteristics and it orders four stages:

1. Establish evaluation requirements
2. Specify the evaluation
3. Design the evaluation
4. Execute evaluation

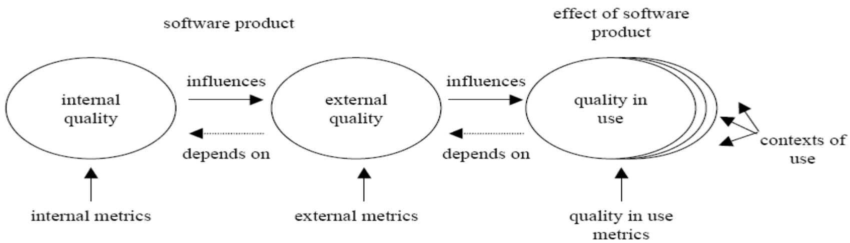

Fig. 3. ISO9126 2-4 standards

The first two stages can easily be performed by exploiting the set of desired quality attributes and through the scenarios identified in the utility tree, already accomplished in the ATAM employment. Designing the evaluation is achieved, with the help of the quality model specification, where one needs to set quality goals for the system under study. The ISO 9126 (2-4) standards are divided into three classes of evaluation requirements: internal metrics, external metrics and quality in use metrics.

The internal metrics may be applied to a non-executable software product during its development stages (such as request for proposal, requirements definition, design specification or source code). Internal metrics provide the users with the ability to measure the quality of the intermediate deliverables and thereby predict the quality of the final product. This allows the user to identify quality issues and initiate corrective action as early as possible in the development life cycle.

The external metrics may be used to measure the quality of the software product by measuring the behaviour of the system of which it is a part. The external metrics can only be used during the testing stages of the life cycle process and during any operational stages. The measurement is performed when executing the software product in the system environment in which it is intended to operate.

The quality in use metrics measure whether a product meets the needs of specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction in a specified context of use. This can be only achieved in a realistic system environment. The internal and external metrics are intended for developers performing the software evaluation.

In our methodology we employ the ISO 9126 - (2 & 3) External and Internal metrics, intended for developers performing the software evaluation, and the ISO 9126 - 4 Quality in use metrics indented for the prototypes evaluation performed by the end users. The selection of measures and metrics is carried out in relation to the goals set by the evaluators and in relation to the quality goals ordered in ATAM in the previous step. The context of use is very important, as it constrains the interpretation of the quality of use. Given a certain type of user, in particular, the quality in use is then related to particular quality characteristics. We use Functionality, Reliability, Usability, Portability, Efficiency and Maintainability as the main evaluation characteristics. The development team can start the software stress tests based on the selected metrics and the results can feed the bug-fixing and further development activities ensuring the quality of the final software result.

3.3 STEP THREE: Evaluation of the Prototype

As stated above, quality in use metrics selection from the pool of ISO 9126 – 4 standard is used in the prototype evaluation. The second outcome of the ATAM employment is the utility tree which acts as a blueprint for the identification of the scenarios employed during the prototyping evaluation. The quality goals set in the utility tree, can easily be related to the architectural components responsible for delivering these goals. Having these components and their related desired quality attributes, the team can build meaningful assessment scenarios to deliver to end users in order to verify the overall system's quality, which constitutes the final outcome of the proposed framework.

4 SAPHIRE Results

The proposed evaluation framework was developed to assess the SAPHIRE system. Driven by the need to assure its overall effectiveness, we focused on measuring specific quality characteristics ordered by end users, in our case, both the patients and medical staff.

4.1 Architecture Analysis of SAPHIRE

The resulting ATAM utility tree is shown below:

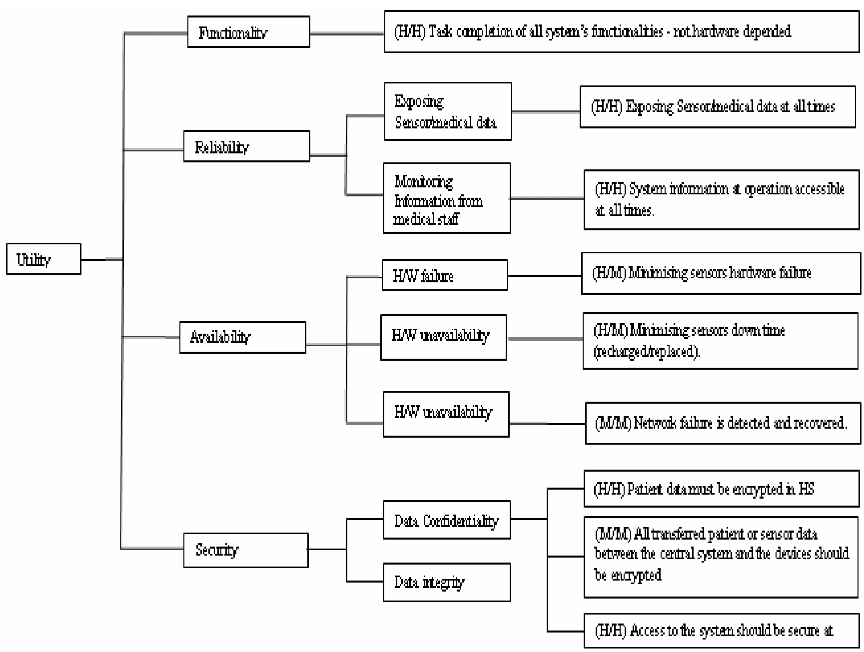

Fig. 4. The SAPHIRE Utility Tree

4.2 SAPHIRE Software Evaluation Metrics

In SAPHIRE we employed the following ISO 9126-2 & 3 metrics.

- Functionality Compliance metrics: Accuracy expectation metric, Computational Accuracy metric, Precision metric, Data exchangeability (User's success attempt based) metric, Data corruption prevention metric, Interface standard compliance metric.
- Reliability Compliance metrics: Failure density against test cases metric, Failure resolution, Breakdown avoidance metric, Incorrect operation avoidance metric, Availability metric, Mean down time.
- Usability metrics: Operation Understandability metric, Understandable input and output metric.
- Effectiveness Compliance metrics: Task effectiveness, Task completion, Error frequency.

4.3 SAPHIRE Prototype Evaluation

We selected metrics that would be easy to apply and to measure. Our metrics were user-oriented, meaning that they aimed to monitored the user's behaviour by using the system in the way each scenario dictated. We adopted from the quality in use metrics pool the Effectiveness, Efficiency and Satisfaction metrics categories.

- Effectiveness: Completion Rate, Errors, Assists.
- Efficiency: Task time, Completion Rate/Mean Time-On-Task.
- Satisfaction: Questionnaires to measure satisfaction and associated attitudes were built using Likert and semantic differential scales. Depending on the case, whether an external, standardized instrument is used or a customized instrument is created, it is suggested that subjective rating dimensions such as Satisfaction, Usefulness, and Ease of Use be considered for inclusion, as these will be of general interest to customer organizations.

5 Business Benefits – Conclusions

The main business benefit behind our approach is the focus on the end-users' quality requirements. These can be translated into quality goals which will drive different evaluation tasks (architecture, software, prototypes) performed by different stakeholders. We manage to increase the confidence of developers, while most importantly minimise the end users involvement (in our case real patients' capacity and medical staff's time-restricted resources).

In employing the framework we took advantage of the work performed in already completed tasks and work packages, namely those of requirements engineering. Employing ATAM was a fairly easy task because it already had a set of system's requirements and architecture analysis. The main difficulties faced were primarily in persuading developers to learn how to employ the ISO 9126 2-3 measurements and metrics to test the delivered software components.

For the SAPHIRE, and similar eHealth related funded projects, the employment of this framework provided a clear evaluation path to be followed by partners according to their role in the development product life cycle thereby easing testing and validation tasks, while providing more time to focus on the critical health/technological issues to be tackled, and thus able to allocate more effort and money to development and refinement tasks.

In addition, we believe that the application of the proposed framework is not limited to e-health related systems. The core concept is the early identification of the desired quality attributes that the system being devised should satisfy. Having these, we can apply the framework to assess the architecture, the software and the system prototypes against the appropriate measurements and metrics selected accordingly from the pool of ISO 9126 2-4 standards. We acknowledge that further work should focus on extracting the quality attributes from the requirements engineering phase in a more automated and traceable manner. Currently we are in the requirements elaboration phase, planning to build a software toolkit, able to offer the proposed evaluation steps and approach. It is our intention to offer this toolkit to partners participating in EU-funding consortia once the project is finalised and critically evaluated.

References

1. Kazman, R., Klein, M., Clements, P.: ATAM: Method for Architecture Evaluation. Carnegie Mellon University (2000)
2. IEEE Standard for a Software Quality Metrics Methodology, IEEE Std, 1061–1992 (1992)
3. Babar, M., Zhu, L., Jeffery, R.: Framework for Classifying Software Architecture Evaluation Methods. In: Australian Software Engineering Conference (ASWEC 2004) (2004)