Collaboration and the Knowledge Economy: Issues, Applications, Case Studies
Paul Cunningham and Miriam Cunningham (Eds)
IOS Press, 2008 Amsterdam
ISBN 978–1–58603–924-0

An Overall Framework for the Evaluation of Research & Development Projects Systems

Androklis MAVRIDIS, Adamantios KOUMPIS
ALTEC S.A, M. Kalou 6 str, Thessaloniki, 54629, Greece
Tel: +302310595646, Fax: +3023210595640, Email: mavr@altec.gr, akou@altec.gr

Abstract: European Research and Competitive funded projects are large and complex structures that involve a constellation of members, such as universities, research centres, business companies, etc, where the promised result is the outcome of collective research and development effort. When the expected outcome is software then, apart from testing the offered functionalities, there is the need to ensure that the offered system satisfies quality attributes, such as operability, usability, maintainability etc, imposed by end users. In most cases there are few, if any, resources spent on evaluation tasks, as the consortium is limited by time constraints ordered by strict work plans.

To cope with this reality we have developed a unified framework aiming to evaluate the systems' architecture, the developed software, and the prototypes offered to end users. The driving force behind each mentioned evaluation is the set of quality goals ordered by the system's stakeholders. The key element behind this methodology is the common set of quality attributes against which the various project's deliverables are assessed.

Keywords: Research and Development projects, Evaluation, Architecture, Software

1. Introduction

European Research and Competitive funded projects are large and complex structures that involve a constellation of members, such as universities, research centres, business companies, etc, where the promised result is the outcome of collective research and development effort. When the expected outcome is software then, apart from testing the offered functionalities, there is the need to ensure that the offered system satisfies quality attributes, such as operability, usability, maintainability etc, imposed by end users. In most cases there are few, if any, resources spent on evaluation tasks, as the consortium is limited by time constraints ordered by strict work plans.

Through our participation 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) we have developed a unified framework aiming to evaluate the proposed architecture, the developed software, and the prototypes offered to end users. The process follows a sequence of steps. It starts with the evaluation of the systems' architecture under selected quality attributes. It then goes 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.

The driving force behind each mentioned evaluation is the set of quality goals ordered by the system's stakeholders. The key element behind this methodology is the common set of quality attributes against which the various project's deliverables are assessed. To

Copyright © 2008 The Authors

achieve this we employ the Architecture Tradeoff Analysis Method (ATAM) [1] method and the ISO 14598 and ISO 9126 2-4 standards [2].

2. Objectives

The study towards system evaluation performed during our participation in the SAPHIRE project. The project aims 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.

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/results without providing proper justification of the system's quality, the appropriateness and overall acceptance by the involved stakeholders. The dedicated resources spend on systems' evaluation and evaluation tasks are only adequate for software evaluation and proof of concept through end users' participation in prototypes and prototypes, which often happen too late in the project's life cycle.

Facing this reality and in the scope of 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 as early as possible the end users in the evaluation process 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.

3. Methodology Used

The proposed framework performs three different evaluations, namely the system's architecture, software and prototypes evaluation. To achieve this we employ Architecture Tradeoff Analysis Method (ATAM) for the architectural part, and the ISO 14598 and ISO 9126 (2-4) for the software and prototypes evaluations.

The evaluation is performed in a sequence of steps. The results of ATAM provide the input to the software and to the prototype assessments as the employment of ISO 9126 measurements and metrics are guided by the same quality attributes that have been identified in ATAM, shown in the following figure:

Copyright © 2008 The Authors

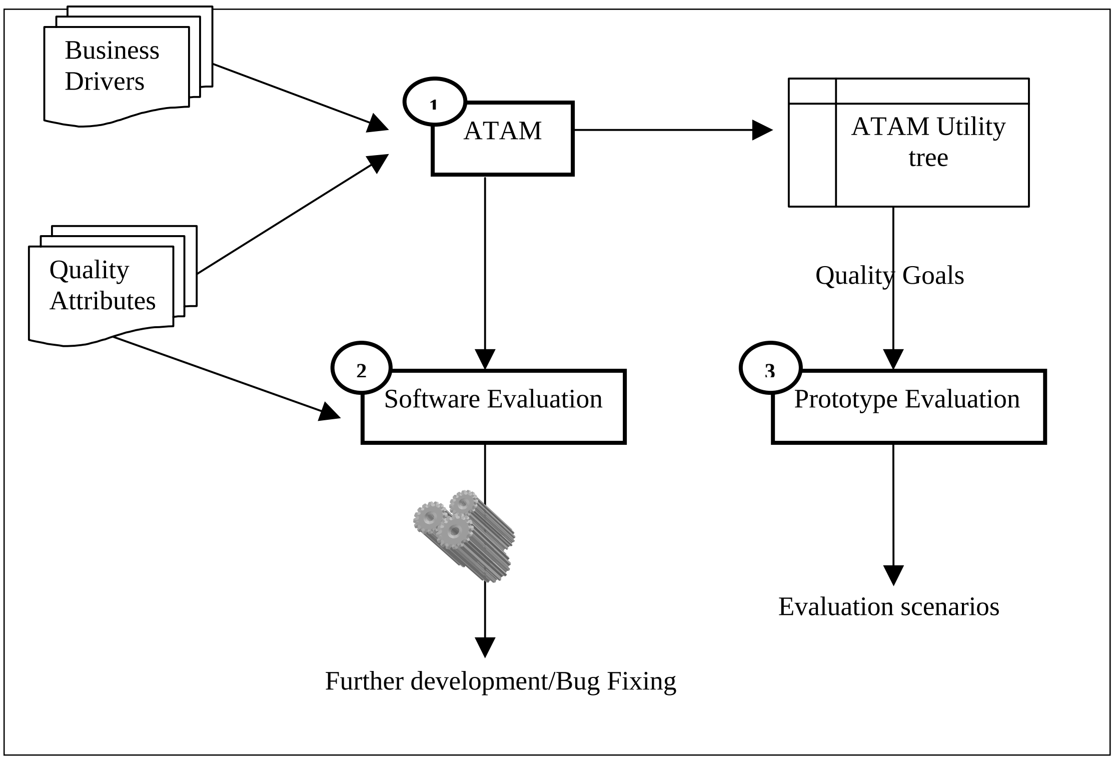

Figure 1: Evaluation Steps

3.1 Architecture Evaluation

For evaluating the system's architecture we employ the well known 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. Evaluating an architecture using the ATAM, the goal is to understand the consequences of architectural decisions with respect to the quality attribute requirements of the system. A system is motivated by a set of functional and quality goals.

ATAM focuses on quality attribute requirements. Quality attribute characterizations answer the following questions about each attribute:

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 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."

Copyright © 2008 The Authors

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.

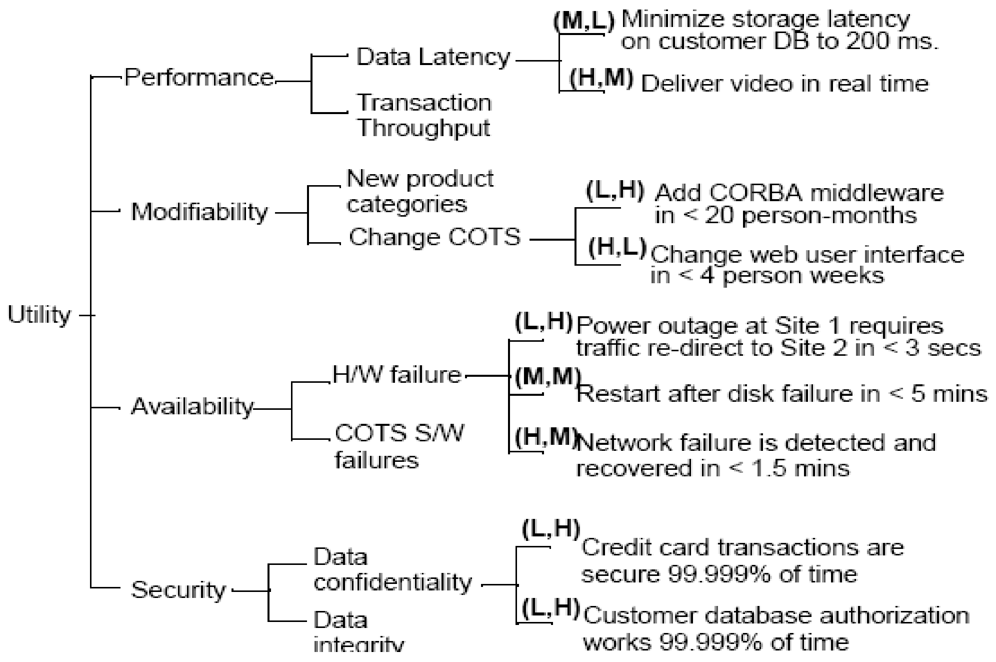
Utility
Performance
Data Latency
Transaction Throughput
(M,L) Minimize storage latency on customer DB to 200 ms.
(H,M) Deliver video in real time
Modifiability
New product categories
Change COTS
(L,H) Add CORBA middleware in < 20 person-months
(H,L) Change web user interface in < 4 person weeks
Availability
H/W failure
COTS S/W failures
(L,H) Power outage at Site 1 requires traffic re-direct to Site 2 in < 3 secs
(M,M) Restart after disk failure in < 5 mins
(H,M) Network failure is detected and recovered in < 1.5 mins
Security
Data confidentiality
Data integrity
(L,H) Credit card transactions are secure 99.999% of time
(L,H) Customer database authorization works 99.999% of time

Figure 2 Utility Tree Example

3.2 *Software Evaluation*

In our framework, we employ the ISO 14598 standard which provides our 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:

4. Establish evaluation requirements
5. Specify the evaluation
6. Design the evaluation
7. Execute evaluation

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.

There are three classes of evaluation requirements and their associated metrics recognized in the ISO 9126 (2-4) standards: internal metrics, external metrics and quality in use metrics.

Copyright © 2008 The Authors

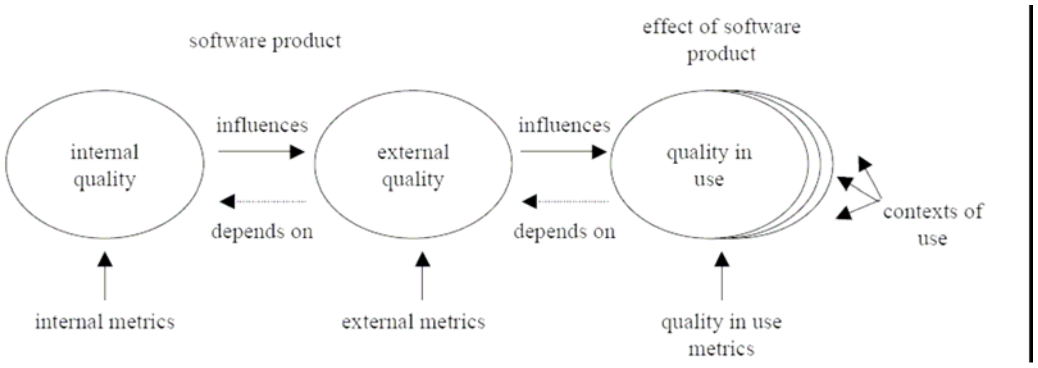

Figure 3: ISO9126 2-4 Standards

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.

Clearly enough, 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 Prototype Evaluation

As stated above, the first step in prototyping evaluation is the quality in use metrics selection from the pool of ISO 9126 – 4 standard, in a similar approach to this of the software evaluation. The second outcome of the ATAM employment, the utility tree, acts as 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.

The set of these evaluation scenarios is the final outcome of the proposed framework. Different techniques and presentations for scoring end users satisfaction can be employed at this stage as it is out of scope of the current study.

Copyright © 2008 The Authors

4. SAPHIRE Results

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

4.1 SAPHIRE Architecture Analysis

The resulting ATAM utility tree is shown below:

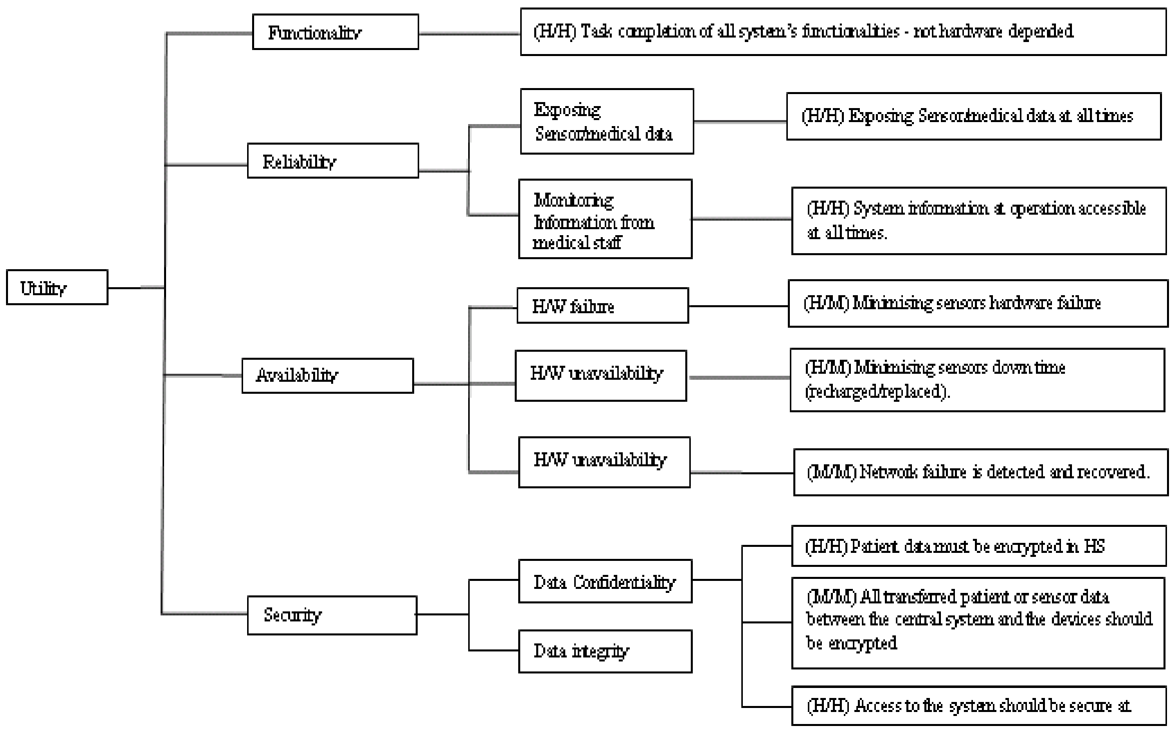
Utility
Functionality
(H/H) Task completion of all system's functionalities - not hardware depended
Reliability
Exposing Sensor/medical data
(H/H) Exposing Sensor/medical data at all times
Monitoring Information from medical staff
(H/H) System information at operation accessible at all times.
Availability
H/W failure
(H/M) Minimising sensors hardware failure
H/W unavailability
(H/M) Minimising sensors down time (recharged/replaced).
H/W unavailability
(M/M) Network failure is detected and recovered.
Security
Data Confidentiality
Data integrity
(H/H) Patient data must be encrypted in HS
(M/M) All transferred patient or sensor data between the central system and the devices should be encrypted
(H/H) Access to the system should be secure at

Figure 4 SAPHIRE Utility Tree

4.2 SAPHIRE Software Evaluation Metrics

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

1. Accuracy expectation metric
2. Computational Accuracy metric
3. Precision metric
4. Data exchangeability (User's success attempt based) metric
5. Data corruption prevention metric
6. Interface standard compliance metric

Reliability Compliance metrics:

1. Failure density against test cases metric
2. Failure resolution
3. Breakdown avoidance metric
4. Incorrect operation avoidance metric
5. Availability metric
6. Mean down time

Usability metrics:

1. Operation Understandability metric
2. Understandable input and output metric

Copyright © 2008 The Authors

Effectiveness Compliance metrics:

1. Task effectiveness
2. Task completion
3. 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 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:

1. Completion Rate:
2. Errors
3. Assists

Efficiency:

1. Task time
2. 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

Limited by time constraints we tried to blend state of the art methodologies and standards in order to achieve a unified framework, able to assist developers in efficiently testing the various software components delivered by various partners (often a cumbersome task due to cultural differences and remote collaboration) and to satisfy the end users quality goals.

Sticking to the idea that end users requirements 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 precious 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 fairly easy task having a set of system's requirements and architecture analysis. The 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. As a consequence, from the proposed evaluation framework employment in SAPHIRE project we were able to refine the metrics selection mechanisms and the linkage of those, with quality goals ordered by the participating stakeholders, enabling us to map the overall evaluation process to be used in other software projects in a more automatic manner. For the SAPHIRE and similar eHealth related funded projects point of view, the employment of this framework provides a clear evaluation path to be followed by partners according to their role in the development product life cycle, easing in such way the 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.

Copyright © 2008 The Authors

Nevertheless, 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 the system should satisfy. Having these, we can apply the framework to assess the software, the architecture and the system prototypes against the appropriate measurements and metrics selected accordingly from the pool of ISO 9126 2-4 standards.

6. Conclusions

In this paper we presented the proposed framework for the evaluation of the R&D projects systems. We justify our choice for adapting the ISO 14598 and ISO 9126 standards and ATAM for our overall evaluation framework by presenting the evaluation process these tools offer. We provided detailed description of the overall methodology, the steps to be taken upon application at each development phase, the outcomes of each evaluation step and the tools/techniques employed. 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.

References

[1] R. Kazman, M. Klein, P. Clements, “ATAM: Method for Architecture Evaluation”, CMU, 2000.

[2] IEEE Standard for a Software Quality Metrics Methodology, IEEE Std. 1061-1992, 1992.

[3] Babar, M, L. Zhu, and R. Jeffery, Framework for Classifying Software Architecture Evaluation Methods, Australian Software Engineering Conference (ASWEC'04), 2004.

Copyright © 2008 The Authors