Usability Promotion in a Technical Project with Sparse Resources – a Case Study

Kaarina Karppinen
VTT Technical Research Centre of Finland
Kaitoväylä 1, PO Box 1100
FI-90571 Oulu, Finland
+358 40 5487 058
kaarina.karppinen@vtt.fi

Marja Liinasuo
VTT Technical Research Centre of Finland
Vuorimiehentie 3, Espoo, P.O. Box 1000
FI-02044 VTT, Finland
+358 400 912711
marja.liinasuo@vtt.fi

ABSTRACT
In this paper, we describe how the usability of software functionalities are promoted and evaluated during the design phase of a software project developing security-related functionalities in a middleware. The paper describes our work-in-progress in GEMOM project, challenges faced in the beginning of the project, and our plan to overcome those challenges with a clearly defined usability implementation plan.

Categories and Subject Descriptors
D.2.9 [**Software Engineering**]: Management – *Software quality assurance (SQA).*

General Terms
Design, Human Factors

Keywords
software design, usability, scenarios, acceptability

1. INTRODUCTION
The usability of a software product is becoming a widely recognised quality attribute in software development [1]. However, the conception of usability realised in projects is often quite narrow and, pertaining software design, restricted to attributes that are conceived as becoming topical to produce only in the later phases of software development. Anyhow, no effective product can be designed without taking into account also the "soft" human and context-related complexities, broadly speaking human factors, already in the beginning of the system development.

Furthermore, in the Human-Computer Interaction (HCI) community, software usability has primarily been concerned with the presentation of information, more precisely with user interface [2]. User interface can denote the visible part of the system and, less frequently, the interaction part of the system, i.e., the coordination of the information exchange between the end user and the system in both directions [3]. Either way, the easily neglected fact is that the usability of the tool is not only about the interface but also depends on other attributes rooted deeper in the character of the tool, e.g. the tasks it performs,.

Middleware is a specific type of computer software that connects various software components or applications together. Typically other applications are conceived as users of a middleware, and it has no direct human users, except for systems specialists or the like who usually install and maintain complex IT systems. Hence, usability in a middleware system development is harder to promote than of applications which have a direct interface towards a human user. As a consequence, the lack of "proper" users and the tradition of software development methodologies, which may see a user only as a means to elicit requirements, easily results in software development without any usability perspective. In the ongoing GEMOM project these obstacles are planned to be prevailed and one of the project's aims is better usability of the end product without risking the security of it; a task that is proved to be hard to perform [4].

2. OVERCOMING THE CHALLENGES FOR USABILITY
GEMOM (Genetic Message Oriented Secure Middleware) is a recently launched research project, lasting for 2½ years, that is co-funded by the European Commission and involves ten industry and research partners across Europe [5]. GEMOM is developing a prototype of a secure, self-organizing and resilient messaging platform, which enables reliable message sourcing and delivery in applications.

In GEMOM, five case studies, where the new PS-MOM (Publish-Subscribe variant of Message Oriented Middleware) will be used, are defined. Each case study represents a different application area with diverse demands on security and usability; hence, no common definitions can be produced.

2.1 Defining the Challenges
The challenges concerning usability promotion within the project included several issues: usability was to be promoted in a deeply technical project and among technically oriented project members; the task for usability was not clearly defined; the focus of the work, which is the development of a middleware, lacked direct end users; and additionally, a very limited amount of person months were allocated for human factor studies, thus excluding the possibility of a usability

I-USED'08, September 24, 2008, Pisa, Italy

specialist to e.g. interview or lead workshops in various countries by herself.

This working context resulted in two practical questions, involving also some matters of principle, with no direct answer for the usability expert participating in a pre-defined project. Firstly, how to motivate usability studies in a project without direct end users? Secondly, how to perform usability studies with sparse resources?

2.2 Creating Motivation

The first problem to be solved was the motivation for usability studies, related with the problem of having no direct end users for a middleware. Hence, the eventual end users as well as the outline of a plan for usability promotion had to be defined.

In order to clarify the definition of a user in our project, we started by creating a more detailed picture about the various users. The preliminary version of users was based on the usage distance between the user and the middleware. Three levels of users were found: (1) users that were provided some IT-related service, being the furthest away from the middleware; (2) users that provided the service in question; and finally (3) users that maintained the software providing the service, including the middleware, and thus being situated closest to the system.

The predefined project plan stated that user acceptance shall be obtained with the help of scenarios; the new technical solutions would be interpreted into scenarios of usage, which would then be evaluated with the users. This way user acceptance, i.e. the worth of the technical solutions planned to be realised, as experienced by the future users, could be found out. With no user interface to evaluate, a reasonable choice was to concentrate on the functionalities of the middleware as seen by the human user. This choice was also meaningful regarding the method chosen, as it is easier to describe verbally the chain of events than the attributes of a user interface.

2.3 Overcoming the Lack of Resources

The other problem, sparse resources for usability studies, could be compensated by harnessing technical experts to assist in usability evaluation. Consequently, usability study had to be planned extremely carefully as no prior knowledge of usability could be expected from other project members. In this project, case studies provide the human users for usability studies. The usability expert acts as a supervisor who plans and analyses the usability implementation and its results. For instance, she instructs the case study leaders to reflect with the user representatives what aspects regarding usability and security are important from the viewpoint of the user in their case study.

3. USABILITY IMPLEMENTATION

The theme throughout the usability plan is to realise it mainly by non-usability experts. Hence, a stepwise approach was chosen. The main idea is to perform usability studies as early in the project as possible so that the studies could have an actual effect on the middleware functionalities perceivable by human users. The process steps described below are accompanied by practical instructions produced by the usability expert so that the tasks in question can be performed.

1. Case study representatives are to define and describe who are the users affected by the functioning of the middleware in their case study.
2. Technical experts are to describe the technical solutions from the perspective of the users, i.e. the effect of the solution as can be perceived by the human users.
3. Leader of each case study is to produce the scenarios with the users. For that purpose, a description about the functionalities from the human point of view is provided..
4. The case study leaders are to send the scenarios to the usability expert who will check their meaningfulness and return the checked and possibly corrected scenarios with focused questions related with each scenario.
5. Users in each case study are to answer the questions, and the answers will be sent to the usability expert who will analyse them and produce a report about user acceptance.

So far, after having finished the first step of the process, challenges have mainly been related with the understanding of terms that have different meanings in HCI and SE (Software Engineering) approaches. Hence, special care has been taken when discussing about users or scenarios in this project. “User” means human users for HCI but may mean applications for SE. “Scenario” in turn denotes short stories describing relatively freely working process from the human user’s viewpoint in HCI [e.g. 6], compared with the system description that is more technically oriented in SE [e.g. 7].

This paper describes a work-in-progress, and more will be learned when the project is progressing.

4. REFERENCES

[1] Abran, A., Khelifi, A., Suryn, W., and Seffah, A. 2003. Usability meanings and interpretations in ISO standards. Journal of Software Quality 11, 325-338.. DOI= http://dx.doi.org/10.1023/A:1025869312943

[2] Rafla, T., Robillard, P.N., and Desmarais, M. 2007. A method to elicit architecturally sensitive usability requirements: its integration in to a software development process. Software Quality Journal, 15(2), 117-133. DOI= http://dx.doi.org/10.1007/s11219-006-9009-9

[3] Ferré, X., Juristo, N., Windl, H., and Constantine, L. 2001. Usability basics for software developers. IEEE Software 18(1), 22-29. DOI= http://dx.doi.org/10.1109/52.903160

[4] Cranor, L., and Garfinkel, S. 2005. Security and Usability. O’Reilly Media, Inc.

[5] GEMOM website (July 10, 2008): http://www.gemom.eu

[6] Go, K., Carroll, J.M., 2004. The Blind Men and the Elephant. Views of Scenario-Based System Design. Interactions 11(6), 44-53. DOI= http://dx.doi.org/10.1145/1029036.1029037

[7] Uchitel, S., Kramer, and J. Magee, J. 2003. Synthesis of Behavioral Models from Scenarios. IEEE Transactions on Software Engineering, 29(2), 99-115. DOI= http://dx.doi.org/10.1109/TSE.2003.1178048