Chapter XVII

Elicitation and Documentation of Non-Functional Requirements for Sociotechnical Systems

Daniel Kerkow, Fraunhofer Institute for Experimental Software Engineering, Germany
Jörg Dörr, Fraunhofer Institute for Experimental Software Engineering, Germany
Barbara Paech, University of Heidelberg, Germany
Thomas Olsson, Fraunhofer Institute for Experimental Software Engineering, Germany
Tom Koenig, Fraunhofer Institute for Experimental Software Engineering, Germany

Abstract

This chapter describes how non-functional requirements (NFR) can be elicited and documented in the context of sociotechnical systems. An approach is presented based on use cases and on quality models derived from ISO 9126, as well as general problems and challenges when working with NFR. Requirements in general and NFR in particular are subjective, have many stakeholders and are often conflicting. The approach presented includes processes for prioritizing quality attributes that are important to a specific context, eliciting NFR, and identification and analysis of...
dependencies among the NFR. The aim is to provide an experience-based approach that facilitates efficient and effective elicitation and documentation of NFR. Having a structured method that aims at providing measurable, traceable, and focused requirements rather than having ad-hoc and ambiguous ones achieves this. The approach uses use cases as the main technique, though the general principle of having a structured and experience-based process is applicable to other techniques as well.

### Introduction

Technology and the interfaces between technical devices and the persons using them are becoming a natural part of our life. The only time they are consciously thought about is when they fail to meet our expectations, for example, a person wants to send a multimedia message via a cell phone, but it takes too long and there is a time-out on the connection. The time it takes to send the multimedia message confronts the user with efficiency issues. The number of selections required to find a function confronts users with usability issues, and the need to install updates to get a new dictionary confronts them with maintainability issues. All these issues must live up to the users’ goals and expectations. If these expectations are not fulfilled, the users will be unsatisfied with the product, perhaps making it useless or even dangerous for them. While some issues only have an impact on the users’ perception of comfort, there are issues that have a more severe impact on the users or on their environment. A financial transaction, for example, is very sensitive to security issues.

The term “sociotechnical” refers, in the examples above, to the interaction of humans (the users) with a technical device during the usage of a system. This has, of course, an impact on the system development process as well as on the processes in which the software is being used. During development there are many decisions to be made with respect to the environment of the software, the software itself, and the software development process. These decisions not only depend on the users’ expectations but also on the interests of other stakeholders, such as developers or procurers. Thus it is very important for the requirements engineering activities that these expectations and interests are elicited thoroughly. In this chapter we discuss issues in the elicitation and documentation of so called **Non-functional Requirements (NFR)**, which essentially cover all constraints on how a system should achieve its functionality (Kitchenham & Pfleeger, 1996, Menasce, 2001). The ISO Standard 9126 (2001) proposes the following taxonomy:

- **Efficiency**: The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions.
- **Portability**: The capability of the software product to be transferred from one environment to another.
- **Maintainability**: The capability of the software product to be modified. Modifications may include corrections, improvements, or adaptations of the software products to changes in environment and in functional specifications.
Hardware-In-the-Loop Testing of On-Board Subsystems: Some Case Studies and Applications
www.igi-global.com/chapter/hardware-loop-testing-board-subsystems/66675?camid=4v1a

Constructive Knowledge Management Model and Information Retrieval Methods for Software Engineering
www.igi-global.com/chapter/constructive-knowledge-management-model-information/75758?camid=4v1a