Method of non-functional requirements balancing during service development

Larisa Globa\(^1\), Tatiana Kot\(^1\), Andrei Reverchuk\(^2\), Alexander Schill\(^3\)

\(^1\) National Technical University of Ukraine «Kyiv Polytechnic Institute», Ukraine
\(^2\) SITRONICS Telecom Solutions, Czech Republic a.s.
\(^3\) Technische Universitat Dresden, Fakultat Informatik, Deutschland

{lgloba, tkot}@its.kpi.ua, a.v.r@list.ru, Alexander.Schill@tu-dresden.de

Abstract: Today, the list of telecom services, their functionality and requirements for Service Execution Environment (SEE) are changing extremely fast. Especially when it concerns requirements for charging as they have a high influence on business. This results in the need for constant adaptation and reconfiguration of Online Charging System (OCS) used in mobile operator networks. Moreover any new functionality requested from a service can have an impact on system behavior (performance, response time, delays) which are in general non-functional requirements. Currently, this influence and reconfiguration strategies are poorly formalized and validated. Current state-of-the-art approaches are considered methodologies that can model non-functional or functional requirements but these approaches don’t take into account interaction between functional and nonfunctional requirements and collaboration between services. All these result in time and money consuming service development and testing, and cause delays during service deployment. The balancing method proposed in this paper fills this gap. It employs a well-defined workflow with predefined stages for development and deployment process for OCS. The applicability of this novel approach is described in a separate section which contains an example of GPRS service charging. A tool, based on this method will be developed, providing automation of service functionality influence on non-functional requirements and allowing to provide a target deployment model for a particular customer. The reduction of development time and thus necessary financial input has been proved based on real-world experiments.

Keywords: OCS, service deployment, non-functional requirements, requirements balancing.

1. Introduction

During service design and deployment, provided by telecom operator, using OCS [1], one important aspect should be considered. It concerns NFR\(^1\) to service provision.

There is the established fact that any system and services run on the system shall be developed not only based on functional requirements, defining software functions (inputs, behavior, outputs), but non-functional ones as well. It is very important to meet non-functional requirements in the telecom industry, especially for real time systems. Generally non-functional parameters could be classified as follows: Performance (Response Time, Throughput, Utilization, Static Volumetric); Scalability; Capacity; Availability; Reliability;

\(^1\) Non-functional requirements
Recoverability; Maintainability; Serviceability; Security; Regulatory; Manageability; Environmental; Data Integrity; Usability; Interoperability.

Non-functional requirements specify a system’s “quality characteristics” or “quality attributes”. If non-functional requirements are not considered at the designer level, then the provided service may actually be useless in practice.

Currently, NFR are not considered within the perspective of the services list, provided by Telecom Operator. The main problem is that legacy methods can design service according to NFR, but cannot model an influence of concurrency services on particular NFR because of collaboration between services.

This means that Operator has no tool that allows flexible balancing between services, run on OCS. Balancing can allow to model system behavior for a determined (requested) list of services to analyze how this configuration meets the NFR.

This paper describes a novel NFR balancing method, focusing on collaboration between functional and non-functional requirements, allowing to automate service planning stages and to reduce the time and costs for OCS adaptation in general.

The paper is structured as follows: Section 2 contains state of the art analysis of methods and approaches to considering NFR. Furthermore, NFR analysis methods are described. Section 3 introduces NFR balancing method, focusing on functional and non-functional requirements collaboration. The evaluation has been applied using a real-world scenario within a telecommunication company and it is represented in Section 4. Section 5 concludes the work with a summary and outlook on future work.

2. State of the art and non-functional testing

Errors due to omission of NFR or not properly dealing with them are among the most expensive type and most difficult to correct. Recent works [2] points out that early-phase requirements engineering should address organizational and non-functional requirements, while later-phase engineering focuses on completeness, consistency and automated verification of requirements.

There are reports [3, 4] showing that not properly dealing with NFR has led to considerable delays in the project and consequently to a significant increase of the final cost.

There are many reasons for delays and significant increasing of costs, but one of the most important reasons relies on the fact that performance was neglected during software development, leading to several changes in both hardware and software architecture, as well as in software design and code [5, 6, 7].

There could be a situation in which the system can be deactivated just after its deployment because, among other reasons, many non-functional requirements were neglected during the system development such as: reliability (vehicles location), cost (emphasis on the best price), usability (poor control of information on the screen), and performance (the system did what it was supposed to do but performance was unacceptable). As it was mentioned above, OCS shall provide all functionality to charge telecom services (GPRS, voice, sms, mms, VAS\(^2\)) using Event Charging with Unit Reservation, Session Charging with Reservation Unit, Immediate Event Charging mechanisms. Each service consumes a strictly predefined volume of system resource (memory, process time, etc.) and has influence on non-functional requirements to be supported.

\(^2\) Value added services
2.1. NFR framework

NFR are considered at the design level and there are several approaches that can help to model NFR within the scope of the developed service. NFR framework [7] is a methodology that guides the system to accommodate change with replaceable components. NFR framework is a goal-oriented and process-oriented quality approach guiding the NFR modeling. Non-functional requirements such as security, accuracy, performance and cost are used to drive the overall design process and choose design alternatives. It helps developers express NFR explicitly, deal with them systematically and use them to drive development process rationally [8]. In the NFR Framework, each NFR is called an NFR softgoal (depicted by a cloud), while each development technique to achieve the NFR is called an operationalizing softgoal or design softgoal (depicted by a dark cloud). Design rationale is represented by a claim softgoal (depicted by a dash cloud). The goal refinement can take place along the Type or the Topic. These three kinds of softgoals are connected by links to form the SIG\(^3\) that records the design consideration and shows the interdependencies among softgoals.

2.2. KAOS

Another methodology for considering NFR is KAOS [9, 10]. KAOS is a methodology for requirements engineering enabling analysts to build requirements models and to derive requirements documents from KAOS models. KAOS has been designed:

− to fit problem descriptions by allowing you to define and manipulate concepts relevant to problem description;
− to improve the problem analysis process by providing a systematic approach for discovering and structuring requirements;
− to clarify the responsibilities of all the project stakeholders;
− to let the stakeholders communicate easily and efficiently about the requirements.

KAOS is independent of the development model type: waterfall, iterative, incremental, but it also doesn’t take into account collaboration between FR\(^4\) and NFR.

The legacy software tools, for instance NFR-Assistant CASE [11], ARIS [12], don’t provide requested functionality to model nonfunctional requirements and compare their influence on functionality.

2.3. Non-functional testing

Testing of non-functional requirements is another issue. Non-functional testing [13] is concerned with the non-functional requirements and is designed to evaluate the readiness of a system according to several criteria not covered by functional testing. Non-functional testing covers:

− Load and Performance Testing;
− Ergonomics Testing;
− Stress & Volume Testing;
− Compatibility & Migration Testing;
− Data Conversion Testing;
− Security / Penetration Testing;
− Operational Readiness Testing;

\(^3\) Softgoal interdependency graph

\(^4\) Functional requirements
Installation Testing; 

It enables the measurement and comparison of the testing of non-functional attributes of software systems. The cost of catching and correcting errors related to non-functional requirements is very high and could cause full redesign of developed service (system). Testing does not have to occur once the 'code' has been delivered. It can start early with analyzing the requirements and creating test criteria of 'What' it is needed to test. The process for doing this is called the “V” model [9] (Fig. 1.).

It decomposes requirements and testing. It allows testing and coding as a parallel activity which enables the changes to occur more dynamic. NFR has a high influence on the testing process and any service that doesn’t meet NFR can cause rollback of the development process to initial phases.

![V-Model Diagram](image)

**Figure 1. V- Model**

### 3. NFR balancing method

The proposed NFR balancing method is based on creating FR and NFR collaboration model. Implementation of functional requirements is presented by listed FB\(^5\). Each of FB is responsible for a particular logical function. The proposed method includes the following main stages:
- NFR Catalogue development;
- FR decomposition;
- NFR mapping;
- FB distribution;
- Balancing;
- Target deployment model.

NFR balancing method uses NFR Catalogue, Functional Requirements to be implemented, create collaboration model between them. The main stages of the concept are represented below.

---

\(^5\) Functional Block
3.1. Catalogue of NFR

NFR are usually complex, global, conflicting and numerous. Aside from that, both software engineers and stakeholders are not used to recognizing NFR. Because of that, a knowledge base will be used to present NFR in the form of catalogues, to guide the requirements engineering through possibly needed NFR and the possible operationalizations for each NFR can be found. Thus we can operate with catalogues for performance and serviceability. These catalogues will be updated with further operationalizations to keep catalogues on NFR up to date. Such approach will facilitate future reuse of acquired knowledge on NFR elicitation.

3.2. FR decomposition

The next stage is creating the FR decomposition model. FR decomposition shall describe all services with their features’ influence on NFR. This means that each service shall be split into functional blocks. A functional block is a logical unit, responsible for providing some strictly defined functionality (for instance sending of notification, bonus system registration, etc.). What is more, services and features, they provide, will be depicted for each functional block (functional requirements).

Total distribution of functional blocks between all services, run on OCS, is represented in Table 1.

<table>
<thead>
<tr>
<th>Service</th>
<th>Functional Block</th>
<th>Functional Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service1</td>
<td>FB1.1 or FB1.2</td>
<td>FR1, FR2</td>
</tr>
<tr>
<td>Service1</td>
<td>FB2.1 and FB2.2</td>
<td>FR3, FR1</td>
</tr>
<tr>
<td>Service2</td>
<td>FB1.1</td>
<td>FR5, FR6</td>
</tr>
<tr>
<td>Service2</td>
<td>FB3</td>
<td>FR1, FR7</td>
</tr>
</tbody>
</table>

3.3. NFR mapping

Each call of FB requests a defined amount of each system resource (memory, processor time, network, etc.) and has a list of characteristics: response time, availability, etc. All of these characteristics shall be mapped to NFR from catalogue with values that specify how exact FB meets the particular NFR (it could be graded from 0 to 100 – Table 2).

<table>
<thead>
<tr>
<th>Functional block/ NFR</th>
<th>Availability</th>
<th>Performance</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB1.1</td>
<td>90</td>
<td>80</td>
<td>10</td>
</tr>
<tr>
<td>FB1.2</td>
<td>80</td>
<td>70</td>
<td>20</td>
</tr>
<tr>
<td>FB2.1</td>
<td>50</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>FB2.2</td>
<td>5</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

FB with the same first number (FB1.1, FB1.2) provides the same functionality but in different way. This means that from a functional point of view there is no difference between these two blocks. The difference is only how each FB meets the NFR.

To understand and reason about different alternatives involved in these tradeoffs between functional blocks it is required to clarify some NFR operationalizations and to negotiate which NFR should be denied or partially denied prejudicing another NFR.

To build the NFR model, it is necessary to go through every service and connect it to all needed functional blocks to cover the requested functionality.
3.4. Functional blocks distribution

Using NFR catalogues and FR decomposition, Functional blocks distribution can be realized as it is represented on Fig. 2.

Fig. 2 represents use of functional blocks by services. Influence of each connection between Service and FB on NFR is determined in Table 2. According to this, input could lead to different deployment configurations. Fig. 2 describes that FR 1 and FR 2 from table 1 can be implemented either by FB1.1 or FB1.2. The implementation way depends on the NFR specification for a particular case.

![Figure 2. Functional blocks distribution](image)

3.5. Balancing and target model

The target model would be obtained by using balancing between NFR and approaches to implementation of a particular functionality with FB. This tradeoff can be continued until target deployment configuration is received based on requested NFR. If requested NFR cannot be gained with legacy list of service, then some service should be excluded from deployment scheme. For instance, there is the Customer’s demand that service shall support the highest availability and there is no specified requirement for security and performance. Such case can be realized by the model, represented on Fig. 3. It is a simple situation and there are usually combinations of NFR in practice. Thus, a priority should be assigned to any requirement that will be considered during target model development.

![Figure 3. Target deployment model](image)
4. Charging of GPRS service

Evaluation of the proposed method is demonstrated using a real-world scenario within a telecommunication company. Charging of GPRS service at the design level, requested by Telecom Operator from OCS, is described as an example. Its FR decomposition is depicted in the Table 3.

<table>
<thead>
<tr>
<th>Service</th>
<th>Functional Block</th>
<th>Functional Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPRS</td>
<td>LBS1.1 or LBS1.2</td>
<td>Location Base Charging</td>
</tr>
<tr>
<td>GPRS</td>
<td>RF2.1 and RF2.2</td>
<td>Step Charging</td>
</tr>
<tr>
<td>GPRS</td>
<td>NB3.1 or NB 3.2 or NB3.3</td>
<td>User notification</td>
</tr>
</tbody>
</table>

Assuming that Customer takes into account availability of GPRS service and delay caused by the service as main NFR, and according to statistical data and knowledge base, all FB characteristics are estimated in the Table 4.

<table>
<thead>
<tr>
<th>Functional block/ NFR</th>
<th>Availability</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBS1.1 - location based module implemented as internal cache in OCS</td>
<td>90</td>
<td>80</td>
</tr>
<tr>
<td>LBS1.2 – using external Home Zone Billing - HZB platform</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>RF2.1 – internal Rating</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>RF2.2 – external Rating</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>NB3.1 – notification via SMS</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>NB3.2 – online notification via USSD</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>NB3.3 – offline notification via email</td>
<td>50</td>
<td>10</td>
</tr>
</tbody>
</table>

Finally, the target model for GPRS service using balancing method to get optimal deployment configuration could be created (Fig. 4). The model supposes that configuration will be applied to provide service at the highest availability with minimal delay.

5. Summary and outlook

The proposed method can be applied at both the service design and deployment stages. The method could be realized within a software tool, used for service provision software design and realization. It is also necessary to foresee the possibility of its usage during service monitoring to obtain specific statistical data. This data shall be used to evaluate how
each functional block meets a particular NFR. The method increases efficiency of development process on testing and deployment phases and allows quick system reconfiguration on customer demand. In the future, the method will be extended to consider possibly changing the NFR list and their priorities during different time periods (e.g. periods with high load, service upgrading) and also take into account changing priority between services.

References