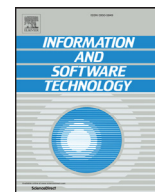




Contents lists available at ScienceDirect

## Information and Software Technology

journal homepage: [www.elsevier.com/locate/infosof](http://www.elsevier.com/locate/infosof)

# An industry experience report on managing product quality requirements in a large organization

Parastoo Mohagheghi<sup>a,\*</sup>, Mario Ek Aparicio<sup>b</sup>

<sup>a</sup> Norwegian Labour and Welfare Administration, Oslo, Norway

<sup>b</sup> Miles Oslo AS, Oslo, Norway

## ARTICLE INFO

### Article history:

Received 3 May 2016

Revised 3 February 2017

Accepted 4 April 2017

Available online xxx

### Keywords:

Product quality

Non-functional requirements

Agile

Experience report

Maintenance

Technical debt

## ABSTRACT

**Context:** Eliciting, managing and implementing product quality requirements (in-short quality requirements) in a large organization can be challenging when many stakeholders are involved and projects run in parallel; sometimes with varying priorities with regards to quality. In this case from a public organization in Norway, the separation between business units and the IT-department and the legacy burden are additional factors that increase the complexity of requirement management.

**Objective:** This paper presents results and experiences from three years long work with quality requirements, starting from ad-hoc handling of quality requirements in separate projects to systematic work across projects with reusable sets of requirements and processes.

**Method:** We present how quality requirements are captured and classified, as well as changes to the agile software development process as a consequence of increasing focus on product quality.

**Results:** The ISO/IEC-25010:2011 standard is tailored for better context fit and is supported by concrete requirements and a methodology that covers the life cycle of software products in both greenfield and brownfield projects. In addition, the organization had to examine the current state of existing IT-capabilities in order to establish a quality baseline for future development, and develop shared vision and roadmaps for product quality.

**Conclusions:** In our experience, stakeholders prefer an iterative and lightweight approach in eliciting and refining quality requirements. The classification model and requirement lists are used as guidelines in requirement workshops. The developed terminology, updated templates and processes are reusable in projects and generalizable to different contexts, and are well adopted by the IT and business units.

© 2017 Elsevier B.V. All rights reserved.

## 1. Introduction

The term *Non-Functional Requirements (NFRs)* has been broadly used in the industry to cover the concerns other than functionality of an application or service. These concerns may cover how well an application or service performs and is experienced by users (such as its availability, usability, reliability and security), compliance with rules and IT-standards, or compliance with development constraints such as time, cost, resources, development processes and tools. Others have used the terms *product quality requirements* or *quality requirements* to cover the above concerns. In this paper, we use the term “quality requirements” for concerns other than functionality, while development constraints (such as time, budget and technical environment) are out of the scope of the work.

Quality requirements can be challenging to implement for numerous reasons [22]: First of all, unless developers and analysts proactively elicit quality requirements, the stakeholders might well believe them to be implicitly understood. Second, quality requirements tend to exhibit tradeoffs that must be carefully negotiated and resolved. Finally, quality requirements are often harder to measure and track than their functional counterparts.

This paper presents findings from the work to manage quality requirements efficiently across various organizational units and projects in a large public organization; i.e. The Norwegian Labour and Welfare Administration (NAV). In a large organization such as NAV, multiple projects are often running in parallel, a large number of stakeholders are involved and there are inter-relations between the results; thus requiring extensive communication and making tradeoffs when conflicts in goals or plans are detected. Huge effort is spent over time to elicit, anchor the “ownership”, and benchmark the level of performance. The success of work with quality requirements lies in involving stakeholders in a progressive way, developing sustainable requirement lists and processes that

\* Corresponding author.

E-mail addresses: [parastoo.mohagheghi@nav.no](mailto:parastoo.mohagheghi@nav.no) (P. Mohagheghi), [mario.aparicio@gmail.com](mailto:mario.aparicio@gmail.com) (M.E. Aparicio).

are reusable across projects and organizational units, an incremental improvement of the status of quality management support, shared responsibility for the results and open communication that enables learning.

This paper is an industry experience report that describes the context, problems that encountered during development, discussion of the methods that were developed and how we mean they solved the problems, and what lessons we learnt. By discussing the state of the art, we try to put the methodology and lessons learned in a broad context and make them useful for other practitioners as well as researchers.

The remainder of this paper is organized as follow: the context and motivation of the work is presented in Section 2 while Section 3 presents state of the art. In Section 4, sources and types of quality requirements are discussed and Section 5 presents processes developed for managing quality requirements. These processes, the classification model and requirements baselines are elements of the methodology for managing quality requirements. Section 6 discusses the impact of the work on practice and includes examples from different perspectives; thus evaluating and validating the methodology. In Section 7, we present experiences from workshops with requirement owners. The contributions of this work relative to the state of the art, the validity of the results and key lessons from our work are discussed in Section 8. Finally, Section 9 presents conclusions and directions for future work.

## 2. Background and motivation

NAV administers a large proportion of welfare benefits and social security schemes in Norway. These include almost 50 types of benefits that cover different stages of life of citizens such as parental leave, unemployment benefit, sickness benefit, disability pension and retirement pension. With over 2.8 million users, NAV pays out one-third of the Norwegian national budget in benefits. Over 400 IT applications provide services to citizens as well as to 19,000 employees of NAV who manage the benefits. NAV emerged in 2006 as a result of a merger of three organizations responsible for different benefits; resulting in a large organization with a burden of legacy IT systems from different domains and built over time using multiple technologies. Many new online services have been developed since 2006. However, the IT landscape still includes many legacy systems, the oldest from 1978.

With the vision of offering more user-friendly and available services, NAV is heavily modernizing its services and IT-portfolio, moving from paper-based processes with manual processing to self-service and online services where most of the processing should happen automatically by codifying rules to check eligibility and enable disbursements. The IT-department in NAV with over 570 employees manages IT-services and applications across the organization. A *service* in this context refers to what is offered to end users; external as well as internal. An *application* is a component or system that offers the service or is used in the process of offering the service. In the remainder of this paper we use the term *IT-capability* to generally cover services and applications. The directorate in Oslo administers NAV and the organizational units in the directorate (called business units in this paper) are responsible for defining, prioritizing and financing changes to be done to IT-capabilities using a portfolio management process. When participating in development, they have the role of *product owners* [35] and are responsible for refining requirements, prioritizing items in project backlogs and accepting the results.

New development and major updates are mostly handled by initiating projects. NAV's internal resources (from business units as well as the IT-department) manage the projects, specify requirements and are responsible for quality assurance and acceptance of the resulting software products. They also participate in

software development in some specific roles and have the overall responsibility for operation and maintenance. The practice by now has been to charge external contractors with most software development and technical maintenance based on contracts with NAV. This is however changing gradually and NAV is taking more responsibility for development and maintenance of IT-capabilities.

NAV's IT-department manages the overall software architecture and processes for development and maintenance. Since 2012, the development process is an agile methodology, which is an adaptation of Scrum [35]. Some extra activities and roles are added for using agile on a large scale and integrating it with other processes such as the portfolio management process. Examples of extra activities are managing a product backlog across projects and evaluating consequences of proposed changes by the central software architecture team.

NAV started a large modernization project in 2012 with the goal of supporting new political reforms as well as offering more self-services, improved management of benefits and more automation. Business units specified the new IT-capabilities that were later assigned to different development tracks and contractors. Quality requirements were, however, only listed in the initial project specification document and were not part of epics or user stories. Recognizing the need to specify the qualities of services, the modernization project asked development and maintenance teams for input and gradually collected a list of quality requirements. This list gathered over 1000 requirements classified using the ISO/IEC 25010:2011 standard [26]. The high number of requirements was by itself a challenge. Additionally, many requirements were not adequately specified. We refer here to the definitions of a SMART requirement [17]:

1. *Specific*: the quality requirement is clear and unambiguous.
2. *Measurable*: the quality requirement is measurable so that it is possible to know if it has been delivered.
3. *Relevant*: the quality requirement is relevant for the types of IT-capabilities the organization has and the desired qualities of these in the future.
4. *Attainable*: the quality requirement is realistic and attainable.
5. *Time-bound*: it should be a target date for the quality requirement or in our case often a road map for improving the state.

Many quality requirements in the list were not measurable and were not assigned to deliverables and release plans. Some quality requirements would drive cost and complexity excessively. In addition, business units meant that they were not adequately involved in the elicitation process and were especially concerned that compliance requirements were missing. A public organization must adhere to a host of laws and regulations such as privacy requirements and public archiving. Finally, the list mixed quality requirements for IT-capabilities to be delivered (product quality requirements) with other types of quality requirements such as process quality requirements.

Following the agile development process, functional requirements are defined in *epics* and *user stories* and define *what* an IT-capability is expected to do. Quality requirements on the other hand describe *how well* those functions are accomplished. User stories are not part of Scrum but are one technique among many to specify product backlog items. NAV's agile development process did not originally include explicit guidelines on how quality requirements should be addressed.

With the recognition that the modernization project needed support for managing quality requirements, the IT-department started a small project to improve the organization's maturity on quality requirements; called here for the *QR-project*. The QR-project started in November 2012 and was initially planned to be a short task force with focus on the elicitation and specification

Download English Version:

<https://daneshyari.com/en/article/4972301>

Download Persian Version:

<https://daneshyari.com/article/4972301>

[Daneshyari.com](https://daneshyari.com)