

Non-functional Requirements in Large-Scale Agile Development

Parastoo Mohagheghi

The Norwegian Labour and Welfare Service (NAV)
Parastoo.Mohagheghi@nav.no

Abstract. The abstract discusses challenges related to managing non-functional requirements in a large-scale agile project.

Keywords: Agile, Scrum, non-functional requirements, software architecture

Extended Abstract

The Norwegian Labour and Welfare service (NAV) administer a large proportion of the most important welfare benefits and social security schemes in Norwegian society. These include for example unemployment benefit, sickness benefit, disability pension and retirement pension. With over 2.8 million users, NAV pays out nearly one-third of the national budget in benefits. NAV has 19 000 employees where 470 of them work with developing and maintaining IT systems, consisting of more than 300 systems and 12 core systems.

With the vision of offering user-friendly and available services, NAV is heavily modernizing its services, moving from paper-based processes with manual processing to self-service, online services where most of the processing should happen automatically. NAV Business Strategy for 2012-2020 identifies the main challenges NAV must address in order to achieve its social mission. The strategy covers five priority areas and one of them is “Active users”; i.e. providing good self-service solutions that allow as many users as possible to help themselves. As the first step, NAV invested \$500 million over four years to deliver its new pension program. It is the only system in the world that integrates the citizens’ private and public pension benefits, gathers users’ information and can generate a decision letter and enable pension payment in seconds. The system received the Computerworld honors in 2012 for best human services.

Like many other organizations, NAV has moved from traditional, plan-based development methods to iterative and agile development (in this case Scrum). The reasons are many; among them:

- The welfare services are managed through several laws and regulations. These are subject to frequent changes and the changes should be reflected in IT systems continuously and rapidly. IT systems should be flexible so that

changes are incorporated rapidly and effectively. Frequent releases are therefore required.

- IT systems are owned by the business side and are used by both external users and the employees of NAV. Users demand active participation in the development and opportunities for feedback. Business value should be demonstrated more efficiently. Agile methods promise user involvement and early demonstration of value and running software.
- The risks of large and complex projects are easier to mitigate with agile methods. It should be possible to change the delivery plans or the priority of services to be developed.

However, there are challenges in moving to the agile methods and this presentation will focus on the challenge of managing non-functional requirements. While dividing functionality to user stories is in the scope of agile methods, the view is more complex for non-functional requirements:

- Combining project management stages with the life cycle of agile projects is a challenge. Large projects require a phase of inception that defines the scope, high-level requirements (lots of them are non-functional requirements), high-level software architecture, estimates of costs and a main delivery plan. Most agile methods do not cover the stage of inception well.
- Large and complex systems also require heavy integration and acceptance testing (especially for non-functional requirements) that should be integrated in the life cycle of agile projects.
- Software architecture is crucial for managing non-functional requirements such as availability, capacity, scalability, and resource utilization. A view of software architecture also allows division of work between teams and is thus important for coordination among agile teams. The architecture work should be managed more centrally while it should be integrated with the work of self-organized teams and will be evolved during iterations.
- Non-functional requirements have typically several stakeholders and are negotiable. Involving customers to clear the fuzzy or negotiable requirements is a process which is out of the scope of a single iteration.
- Some non-functional requirements may be assigned to single iterations while others are system wide and cut cross functional boundaries. There is a need to decide which cycle should manage or verify a requirement and how.

This presentation will discuss the above challenges, what the state of practice is and what questions remain to be answered.