

Construction of a project monitoring application iteratively and incrementally

Pekka Mäkiahö

pekka.makiaho@uta.fi

University of Tampere

Timo Poranen

timo.t.poranen@uta.fi

University of Tampere

Katriina Vartiainen

katriina.vartiainen@uta.fi

University of Tampere

Abstract

Developing complex software applications is a challenging task. It is hard to recognise all the needed features in advance, and therefore software development is often done iteratively and incrementally. Every iteration usually contains activities like planning, refining requirements, implementation and testing. In this paper, we describe how university student teams developed Metrics Monitoring Tool (MMT) application during the years 2014-2018. The construction process of the MMT has contained five larger development iterations so far. All the versions have been tested comprehensively with dozens of real users.

Keywords (style: Keywords)

Iterative and incremental development, metrics monitoring, application, project work

Background

“Learning by making” is a simplification of ideas behind constructionism learning theory (Papert and Harel, 1991). Constructionism underlines student-centered learning where students use information they already know to acquire more knowledge. (Alesandrini and Larson, 2002)

In computer sciences, students learn by studying different concepts and then implementing the concepts as small exercises (simple programs, algorithms, user interface designs etc.) or larger course assignments. The course projects often implement many concepts at the same time. Larger software applications can be very complex, and they can contain hundreds or thousands of minor concepts that the developer team integrates to the work as a whole.

To manage complexity, iterative, incremental and agile software development models (Sommerville, 2010) have become mainstream in software construction. The main idea behind these models is to build software piece by piece in an iterative manner. This ensures that there is no need to plan everything beforehand, but instead it is enough to plan the next development iteration carefully and to have a wider vision of the project goal. Knowledge of the development team about the software product increases, and the vision can be adjusted regularly.

Design science research methodology focuses on the development of artifacts, like software. Hevner and others (2004) have given seven principles for design science research: i) Design as an artifact, ii) Problem relevance, iii) Design evaluation, iv) Research contributions, v) Research rigor, vi) Design as a search process, and vii) Communication of research.

In this paper, we describe the incremental construction of the Metrics Monitoring Tool (MMT) application. The research follows the design science methodology principles.

Incremental development of the MMT

Metrics Monitoring Tool (MMT) is a tool designed to support project managers, project members and upper management in reporting and observing projects’ progression. Version 2.9 was introduced in the article “MMT: A tool for Observing Metrics in software Projects” (Mäkiaho et al., 2017). The main screen of the MMT is shown in Figure 1.

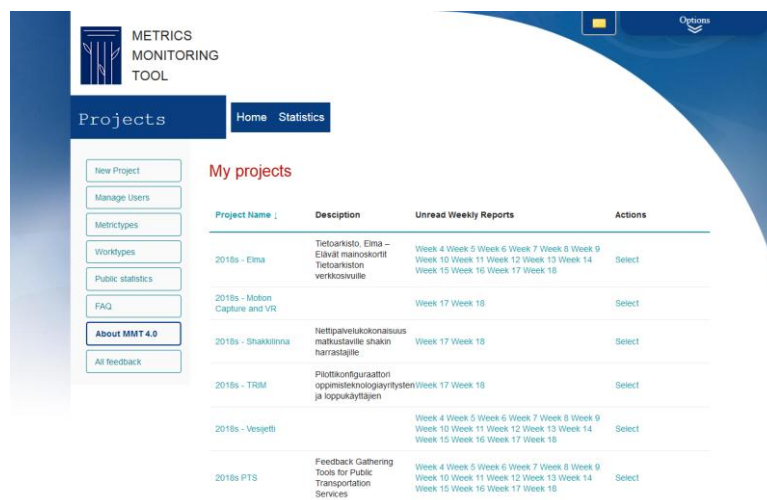


Figure 1. The main screen of the MMT

Figure 2 shows how MMT was developed incrementally and iteratively. The current version of MMT is 4.0. The tool was developed in Tampere University’s Software Project Management and Project Work courses during three academic years. The paradigm used was Design Science Research (Hevner et al., 2004). According to the Design Science principles an artefact (increment)

was first built to solve a problem. After that the success of the artefact and its design were evaluated.

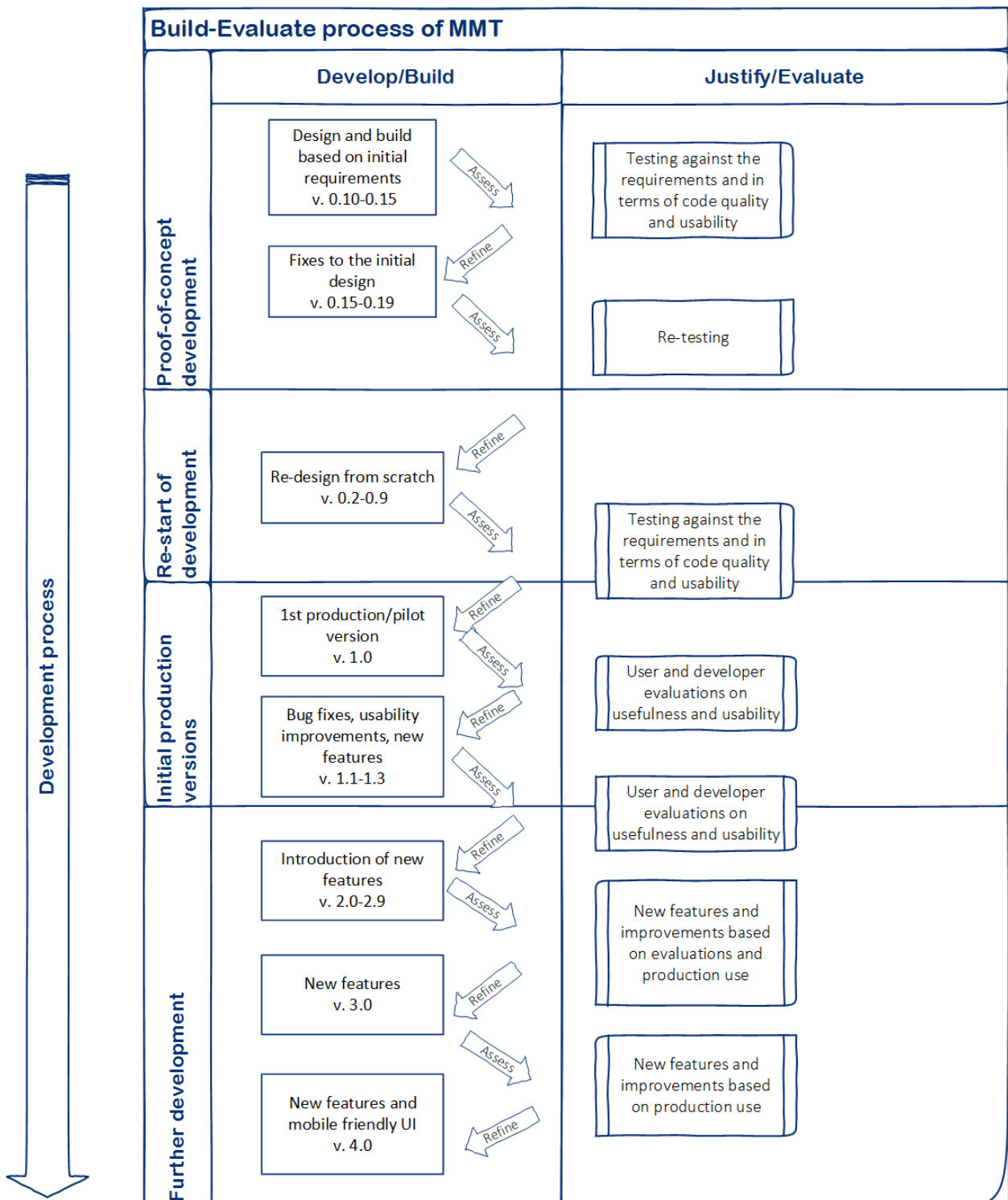


Figure 2. Development process of MMT

Version 0 - proof of concept

The first versions (0.1-0.16) were developed during the academic year 2014-2015. The client evaluated the version by reading the test report and decided that the quality of the software was

too low for taking this version to pilot use. The versions 0.17-0.19 were developed during the summer 2015 by one individual member. However, it was also evaluated that the mature of the software was not good enough even for the piloting.

Version 1 - piloting

A new project team was formed in the fall 2015. The first task of the team was to evaluate the previous version and the decision was made to begin the development from the scratch. The version 1.0 was ready by the end of the fall semester and it was piloted on the project work courses during the spring semester 2016. The new versions (1.2-1.3) were also developed in parallel so that the version 1.3 was deployed by the end of the semester.

Version 2 - production use

During the summer 2016 the experiences from the piloting were evaluated, and a single student from the team continued the development work by correcting bugs, adding some new features and increasing the usability. The main features of this version were *Logging Hours*, *Viewing Reports*, *Viewing Charts* and *Composing Reports* (Mäkiaho, et al., 2017). This version was in production use on the Project Work courses during the fall semester 2016.

Version 3 - different roles and rights

During the fall semester 2016, the developer of the version 2 continued the development as a project manager of a new team. The tool was in production use while it was evaluated and the team developed and deployed new increments. The version 3.0 was deployed by the end of the semester having new features based on different *user rights* and *roles* like project manager, member, supervisor and client.

Version 4 - mobile use and interfaces

During the production use of the version 3.0 in the spring 2017, the tool was still continuously evaluated and feedback was gathered. A single student developed new increments and the version 4.0 was deployed by the end of the semester. The version 4.0 is currently in the production use and it handles a new metric: risks. There are also interfaces to common software project tools like Trello and Slack. Moreover, the software was made mobile friendly by adding a responsive user interface. Example visualisations of the number of commits, requirements, test cases, and working hours by type of the version 4.0 are shown in Figure 3.

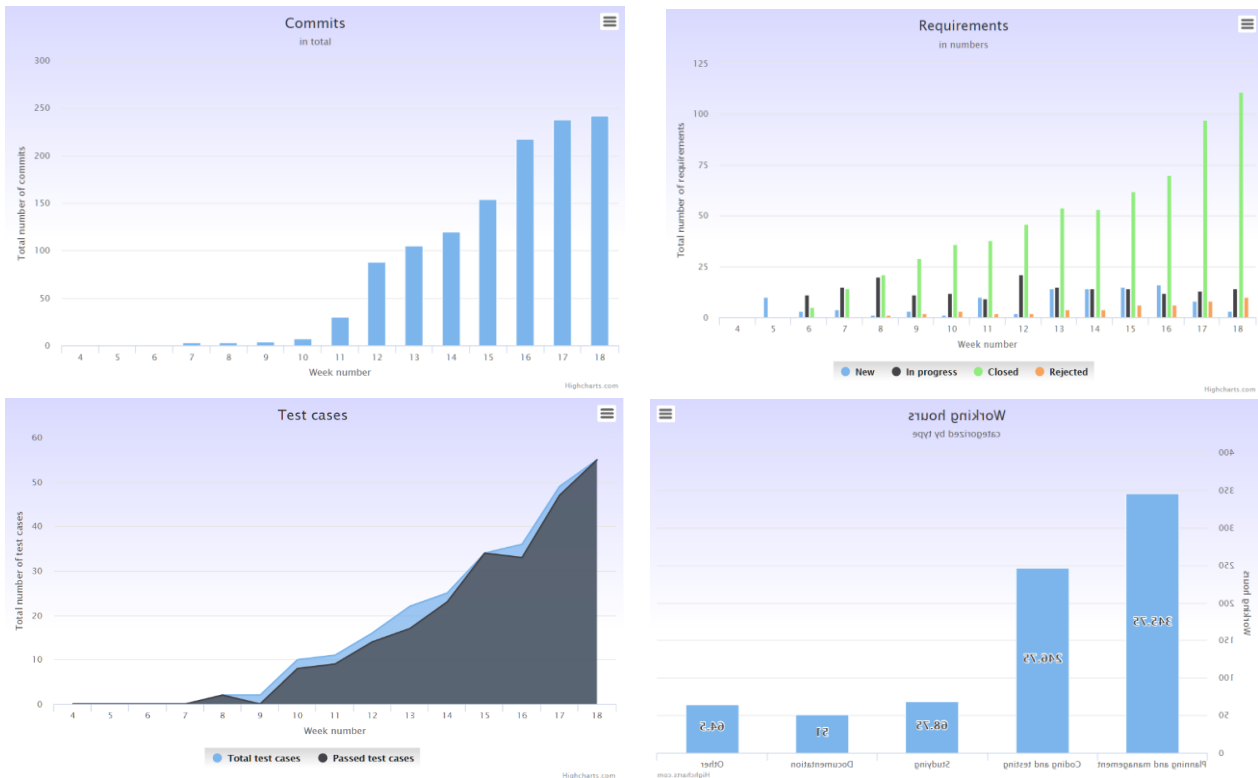


Figure 3. Example visualisations of the number of commits, requirements, test cases and working hours

Conclusions

The construction of MMT has been a long process that started in 2014. There have been roughly 20 people involved in the development of MMT. Roughly 160 students from 30 projects have so far used the application. There have also been projects’ clients and teachers using the application. Experiences on constructing and using the application have been positive among students, teachers and project clients.

MMT is a software artifact that solves project reporting and monitoring problems. Its design has been evaluated in dozens of software development projects. Future plans for further developing the tool include enhancing the user experience and adding a possibility to archive projects.

References

- Alesandrini, K. & Larson, L. (2002) Teachers bridge to constructivism. *The Clearing House*, 119-121.
- Hevner, A., March, S., Park, J., & Ram, S. (2004) Design Science in Information Systems Research. *Management Information Systems Quarterly*, 28(1), 75–105.
- Mäkiahho, P., Vartiainen, K., and Poranen, T. (2017) MMT - A Tool for Observing Metrics in Software Projects, *International Journal of Human Capital and Information Technology Professionals* 8(4), 2017.
- Papert, S. and Harel, I. (1991) *Constructionism. Situating constructionism*. Ablex Publishing Corporation, 193–206.
- Sommerville, I. (2010) *Software Engineering*, 9th edition, Pearson.