



Software Engineering Project 2013

Chalmers Course DAT255

Post-Mortem Report

submitted by

René Niendorf

Contents

1	Introduction	check!	3
2	Practices, Tools	check!	3
2.1	Scrum	check!	3
2.2	Pivotal Tracker/ User Stories	todo! check!	4
2.3	Extreme Programming	check!	4
2.4	Git, Github	todo! check!	4
2.5	Facebook Group	check!	5
3	Team-Workflow	todo! check!	5
3.1	Work Distribution	todo! check!	5
3.2	Working Hours	todo! check!	5
4	Application Development	todo,delete,change! check!	5
5	Negative Experiences	todo! check!	6
6	Positive Experiences	todo! check!	6

1 Introduction *check!*

This report is a post-mortem report about the knowledge and experience gained during the development of an Android application in the Course 'Software Engineering Project' (DAT255) at Chalmers University of Technology. The report focusses on the cognition of the interaction and communication within our team and also reflects how we used certain practices and tools in our group which were introduced during the lectures. All these experiences are discussed from my own point of view and just reflect my personal opinion.

2 Practices, Tools *check!*

This section will comment on different practices and tools which were introduced during lectures and will provide information about how we adapted practices/ tools like Scrum, Pivotal Tracker, Extreme Programming or the version tracker Git which was mandatory by course definition.

2.1 Scrum *check!*

The practice 'Scrum' was introduced in the lecture and mainly describes the dynamic development of a product. This means that the entire product is not planned from the first to the last workload in order to achieve the product which is supposed to be shipped but consists of sprints where each sprint is defined by a beginning state, a length and a goal which should be reached by the end of the sprint. In order to be able to track what the other group members have done and are working on you meet your group in sprint meetings to exchange knowledge, ideas and problems which came during a specific sprint.

We based our project on Scrum and defined a Sprint to be 7 days long. A sprint was setup on Mondays where the group was talking about general ideas and about ideas which are still in the backlog of 'Pivotal Tracker'. These ideas were analysed and translated into proper userstories and goals for the sprint. We adapted two Scrum-Meetings within the sprint which took place on Wednesdays and Fridays.

I experienced Scrum to fit well to our project and the group. Also the chosen length of the sprint and the meetings during the week and shortly before the weekend were perfect for the own workflow. We were not always able to meet in physical appearance because of different schedules but in this case the scrum-meeting took place in our Facebook-group (Section 2.5).

We have not assigned scrum-typical roles as 'Product owner', 'Scrum master', 'Scrum team' or 'User' to one team-member but have shared all roles within the entire team. So everybody was responsible for everything which was working well because of our good communication.

2.2 Pivotal Tracker/ User Stories *todo! check!*

In this course it was mandatory to use the tool 'Pivotal Tracker' which is a strong tool if everybody is experienced with the usage of it. In our case anybody of our team has used it before and we have not used it consequently even though we were able to use it properly at the end of the project. Another point in our project was that it was always clear who has to work on which task. Hence we often forget to use Pivotal Tracker for every task respectively function we built in. In other cases we have not set up any user story for a specific function or marked a user story with 'done' much later than it originally was done and delivered. In a result the Chart which can be printed out by Pivotal Tracker does not really display the workflow of our project. The Chart shows that we accepted a bigger workload in the last sprint which reasoned by noticing that we already implementing this user story and therefor added it to this last sprint.

At the end it turned out to be a nice way of saving ideas and keeping track of the development process of our application. needs more practice to use it properly which we havent had. but its integration is easy and it makes it possible to add ideas for future featrures and makes it possible to follow the workflow

A task in Pivotal Tracker was marked with 'done' after doing tests and refactoring the code of the specific area.

2.3 Extreme Programming *check!*

We implemented ideas or parts of Extreme Programming to our project but were not following the strict definition of XP. Some practices of this methodology did not make sence in the project because of its size, the differing schedules of each member and the short development period.

For these reasons we did not use pair programming, test-driven development, open workspace, sustainable pace and several releases as they are defined by Extreme Programming. Nevertheless we were testing our code by JUnit tests (espacially for our database) and did several acceptance tests during and after coding. A sustainable pace was aimed but the worktime of each group-member was differing because of other deadlines. More details on this are explained in Section 3.2. We published only on pre-release version of 'Chalmers On The Go' but always had a functional application in our working directory which refers to the daily-build aspect of XP.

In contrast we did use following methodologys as they were defined in the lecture slides: collective code ownership, coding standards, Design improvements, customer tests (every team member was testing as a customer) and requirements in form of user stories.

2.4 Git, Github *todo! check!*

complex in the beginning but the introduction in lecture/practice lesson was useful

2.5 Facebook Group *check!*

In order to ease the communication of the group we created a group in facebook. We used this channel for unscientific questions and communication as such and used it for Scrum-Meetings when a personal meeting was not possible. If questions related to the code or bugs were found, we posted issues on Github.

3 Team-Workflow *todo! check!*

3.1 Work Distribution *todo! check!*

in first meeting we were talking about the competences of each member and assigned the tasks to them corresponding to this.

i had the feeling everybody was approximately spending the same amount of time on the project(see section Working Hours). Compared to me other team-members were more experienced in practical programming and therefore more effective in coding than me. But this was not causing a problem at all since there were different fields to work on and some party like gaining data for the database took some decent amount of time expected difficulties because of bachelor theses and different schedules haven't caused big troubles since our communication was really good. we discussed issues in our github repository and used a facebook group for more informal stuff

3.2 Working Hours *todo! check!*

planned time measurement see section Scrum

actual outcome was close to my expectations even though the amount of time spent by each person changed from week to week because of deadlines in other projects. so two teammembers had their bachelors project ending in the first half of our project and were not able to spend too much work on this project during that time but were really putting work into CHalmersOntehgo within the second half. For two other team members it was the other way round. in my case i had two deadlines in my other project during this one and was in kind of average position of the 4 other team members. in final week before handoff we managed to meet everyday and worked/coded together, which was really efficiently

4 Application Development *todo,delete,change! check!*

lot of time spent on research on google maps api
decent amount spent by anders on nav through building which was not possible to get

working at the end
database not complete because it takes a lot of time

5 Negative Experiences *todo!* *check!*

user stories need a lot of practice and need to be understood to be able to work with it in a good way.

it took really long to use them properly but they are good for bigger projects like this one. on a small project i would not use them since they take some time to handle

Working on android applications without having an android phone is a big problem

6 Positive Experiences *todo!* *check!*

group was nice and everything was carried out in discussions and we always found solution which were good for everyone

first time used scrum: good experienced, will be adapted to future projects of mine

sprint + scrum meetings were perfect

if everybody understood how to use and setup user stories: good tool as well especially for collecting ideas, good for defining workloads, changeable workhours rating in order to fit a team