

Responsible for this document:

Emma Albertz
Linnéa Claesson

PUSS154219 v0.1 October 7, 2015

TEAM 2

Project Final Report

Authors of this document:

Emma Albertz
Linnéa Claesson

Contents

1	References	1
2	Introduction	1
3	Project Metrics	1
4	Project Evaluation	1
4.1	Evaluation of Project Metrics	1
4.2	Evaluation of Performance and ... of the Team	1
5	Suggestions for Improvement	5

1 References

No references.

2 Introduction

This document aims to evaluate and analyse the team's continuous work and result of the delivered system.

Någon form av introduktion, där vi bland annat skriver om att överlag så är alla deltagare väldigt nöjda, väl fungerande projekt etc.

3 Project Metrics

4 Project Evaluation

4.1 Evaluation of Project Metrics

4.2 Evaluation of Performance and ... of the Team

As part of the project evaluation, a questionnaire was sent out for all team members to fill out. The replies to this questionnaire are used to evaluate the team's work during this project.

Overall, the general consensus has been very positive and everyone seems very satisfied with both their own and the team's contribution to the project.

The team has during the work been divided into four subgroups with different responsibilities:

Project Managers Responsible for the entire project, planning and making sure everything is delivered on time and up to standard.

System Architects Responsible for the technical part of the project and also communications between the developers and testers.

Developers The developers have been responsible for the coding of the application.

Test Group The testers are responsible for testing the developed system.

4.2.1 Communication

One thing in particular that has worked very well throughout the project is the communication within the team, within each subgroup and also between the subgroups of the team. The developers have reported that there has been exceptional communications between them and the system architects, who have been a great help and contribution in the developers success.

A big contribution to this has been the project group meetings, held once a week and which everyone has been expected to attend. The meetings have been held at the same time and place each week, to avoid confusion and people missing meetings due to unclear scheduling.

At these meetings information to the team was given from the project managers and each subgroup has updated the rest of the team on their current status and progress. The questionnaire and the attendance showed that these meetings were appreciated and a great help in keeping track of what needed to be done and when. Who would be responsible for what part of the project, how issues should be handled and deadlines were set at these meetings.

The group meetings were also a place for discussion and gave an opportunity to ask the other team members questions and raise concerns. It also helped keep the project managers updated in how the work was going in each subgroup and making sure deadlines could be met.

Meeting protocols were written each time and posted afterwards both on the team's joint Drive and Git repository, to be available for later reference and/or read by those who might have missed a meeting.

The subgroups have also had their own internal group meetings whenever they needed, to discuss topics not needed to be addressed at the larger group meetings. These topics include e.g. division of assignments and internal deadlines.

Information that could not wait to be brought up at the project meeting has been sent out by email. A Facebook group has also been used as a means for fast but secondary communications.

There was a discussion at the beginning of the project on whether or not to use Piazza as a means of communication within the group. Since it was not available from the start of the project other alternatives were used instead and by the time Piazza was working, these alternatives had already been established. There has been a difference of opinion within the group whether or not Piazza should have been used during the project and should be evaluated for future projects.

Communications with the customer and experts have not been as well functioning as communications within the team. Especially the testers have reported that they could have avoided some, in hindsight, simple issues by contacting the experts at an earlier stage of the project. The project managers and/or the system architects could also have had a better dialogue with the customer to continuously make sure that the system being developed was in fact the system ordered. The acceptance test has still not taken place at the time when this report is being written, but two formal reviews have been held with the customer prior to the acceptance test to make sure that the right system is being developed.

4.2.2 Time Planning

The scheduling and time planning within the group have received very positive response from the group. The project managers put a lot of time into working out a reasonable schedule that would also make sure that the project was delivered prior to external commitments the team members had at the end of the time period scheduled for this project. This means that the deadlines sometimes were relatively tight, but the team has been motivated to meet them due to it being to their personal advantage. At the time of this report being written, the deadlines have all been successfully met.

The group meetings have been a help in keeping the members of the team updated on upcoming deadlines and providing a chance for the project managers to make sure that they could be met.

The Weekly Schedule and Gantt Chart produced by the project managers could have been referred to more often throughout the project. Overall though, the team members have stated that they have known what and when they needed to do things. It has been stated that the pace of the project has been high, but that it has been good so the project can be finished before the external commitments, mentioned earlier.

The developers did not have as much work in the beginning as the others, but this changed once they started implementing the code and they had to put in a couple of weekends to finish. This seems to have worked fine, even though there were some who could not attend these group sessions.

The testers were sometimes dependent on the system architects to finish their report before the testers could finish their own, which added a bit of pressure at the final stages of the writing of the reports. A slightly earlier deadline was then set for the system architects than the testers,

but they were not given much extra time either since this would affect the system architects negatively.

4.2.3 Work Distribution

Responsible person(s) for each task have been decided at the group meetings. Then division of the larger tasks into smaller have been done internally within each subgroup, sometimes in collaboration with other subgroups (such as system architects and developers). This system has worked very well for this project. The project managers have always known who to talk to about each specific task to make sure everything is coming along as expected and that person can in turn make sure everyone within his or her subgroup is making the progress they should.

The responsibilities between the two project managers were easily shared. They had prior knowledges that complemented each other very well and therefore found natural ways to divide the tasks between them.

The same arrangements were made within the system architects group, where some had more technical skill sets and therefore took on greater responsibilities in those areas. Others had more prior experience in working with this kind of project where a lot of reports were to be produced at a certain standard and could therefore take on more responsibility there.

Even though one of the system architects have been system leader, the organisation within their group has been very flat. They have met often to make sure everyone is on the same page with everything and they have all contributed greatly to this project.

It was a bit unclear in the beginning how large a responsibility the system architects should have. This could (and should) have been evaluated better and made clear from the beginning, to avoid confusion. Overall though, they have known what needed to be done and no major issues arose due to this.

The developers have worked in pairs with their assignments. Most of them had no previous experience with Android development. Working in pairs really helped overcoming something that at first might have seemed very difficult, but still making each member responsible for a part of the project. The division of what each pair should do came as a suggestion from the system architect group, who had prior experience with Android development and knew approximately how much time each task should take. There was one part of the project that was slightly larger than the others and the pair working on it had some difficulties, but the system architects stepped in and offered help so they could finish on time.

Overall the distribution of work among the developers has worked fine. Some of them have not seemed to prioritise this project and instead go and work on other things (everyone in the team has other commitments). This has slowed down the development of the product, which could have been finished at an earlier stage had not some members postponed their tasks to the last minute.

The work among the testers has also worked very well. There has been some inequalities in work load due to the fact only a couple of people within the group had previous knowledge of LaTeX and GitHub. This meant that they had to pull a larger weight in fixing problems for the entire group.

4.2.4 Knowledge

Both among the project managers and the system architects, the members had different sets of knowledge when coming into this project. Some had more technical skills, such as prior experience with the tools used (GitHub, LaTeX and Android Studio) whereas others had more experience with this kind of administrative work and report writing. In both groups this worked very well, since they complemented each other and made the groups more knowledgeable as

whole than each individual. Everyone could make great contributions and bring value to the work.

Not many among the developers had previously worked with Android development, LaTeX and/or GitHub. Since the external training that was provided came far too late in the process, the system architects had to step in and take a large part in training the developers in these areas. The external training also only covered Android development and not the other tools used for this project.

During the development of the product, the developers often chose to meet up and work together, even though the task was divided into smaller ones among them. This meant they could collaborate and help each other with problems. In addition to this, they could keep an open dialogue and everyone knew at what stage everyone else was at.

Som issues arose from the fact that not many among the developers had any previous experience with programming in a larger group. People would push code to git that could not be compiled, leading to problems for the rest of the group. A lot of time was wasted in dealing with these situations.

Within the test group there was a large difference in prior working experience with the tools they used (LaTeX and GitHub). This led to some doing a lot more work than the others when it came to solving problems generated by the tools they used, since they were the only ones who knew how to do it.

4.2.5 Technical Issues

Some technical issues were encountered during the work on this project, most of which were introduced outside of the group's control.

Bugs were discovered in the back end product, which the group had no control over or ability to fix and therefore had to work around.

Android Studio was not properly installed and available on the computers provided for this project. This meant that the developers had to use their own computers and install Android Studio on them. This caused issues for some members of the group who only owned stationary computers and therefore could not work together with the rest of the group. Some installation issues were also encountered but eventually fixed, but a lot of unnecessary time was spent on this process that otherwise could have been spent on development of the product.

The team was also supposed to use Piazza as a means of communications, both within the group and with the experts and customers. Piazza was not made available from the beginning of the project and therefore other tools were established before the team could start using Piazza and its advantages were then lost.

E-puss was also not available from the beginning, which caused some discomfort but no lasting consequences.

4.2.6 Reports and Final Product

When the questionnaire was sent out and this report written before the product was completely finished. Estimations have therefore been made with respect to how satisfied the team will be with the final product. A lot of confidence was shown throughout the replies of the questionnaire, that the developers will be able to present a application that reflects the team's efforts and hard work.

The system architects have made sure that the Software Top Level Design Document (STLDD) has been followed. As always, things could always have been implemented differently but at the time of the writing of this report, everything looks good and the system architects have faith in

the developers ability to deliver the product ordered, based on the competence and hard work they already have put into it.

The project managers are very satisfied with the quality of the documents so far delivered and believe without a doubt that with the help of the system architects and testers, the developers will be able to finish the application on schedule and to a satisfactory result.

Most of the documents are based on templates provided by the customer, to make sure they follow a certain format and standard. This has been really helpful, since it has been very clear from the beginning what the customer has expected.

The system architects are satisfied with their work and results. They have put a lot of effort into this project and have made sure that the developers have had everything they needed throughout the development of the product.

Since there has been a very high level of competence among the system architects, the system leader would have liked to see an even higher level on the documents they delivered. The time and fast pace have been limiting factors for this, though.

There has been some dependence between some of the documents the team has delivered. It has been up to the system architects to manage this and make sure that everyone concerned has gotten the information they need to update their documents according to the changes made in another. A lot of work has been put into this and the other groups have seemed satisfied with the result.

The developers are pleased with their contribution to the product. The main issues they have encountered in the development stage have been due to the back end and sensor not working properly. This has been out of their control, but a list of bugs in the back end has been delivered to the customer.

4.2.7 Other

Samarbetet mellan grupperna har fungerat jättebra!

Strukturen som sattes upp i början har följts och fungerat bra.

Kodat samtidigt som STLDD skrevs, fungerade sådär i och med att när STLDD ändrades var man tvungen att ändra i koden också.

Brister i problemrapportshanteringen. TG har behövt ändra i SVVS utifrån ändringar i SRS, men har inte alltid fått den informationen. Har inte heller haft möjlighet att kommentera hur lång tid ändringarna i SVVS eller SVVI kommer ta, blev bättre.

5 Suggestions for Improvement

PG och/eller SG kunde tagit större ansvar för kundkontakten. Administrationen har varit omfattande och kunde kanske reducerats.

Mer iterativ process (enklare om inte alla hade haft annat utöver projektet). Mer gemensam samsyn på hur en del dokument ser ut och höja prestationskraven.

Utvärdera kunskapsnivåerna, UG kunde skaffat sig information om AS mycket tidigare.

I början av utvecklingen hade vissa svårt att komma igång för att de inte visste hur de skulle använda klasser som de inte själv skrev. Detta kan bero på att arbetet delades upp och kodningen började innan STLDD granskades så att när dokumentet ändrades var inte alla med på exakt vilka uppgifter som tillhörde de olika klasserna. Ett sätt att förhindra detta kan vara att låta UG börja med att göra ett litet system för att få känsla av vilka klasser som behövs och vad de behöver göra samtidigt som STLDD skrivs, så att UG kan lämna synpunkter på designen innan utveckling av den slutliga produkten börjar. På det sättet börjar UG:s arbete tidigare samtidigt som det kan bli lättare att komma igång när man börjar med ett mindre system.

UG kunde ha större ansvar för STLDD. Något kort om hur man kan skriva ett program som kommunicerar med databasen (hämta sensorvärden etc). Kunde ingått i en labb.

Gruppledarna kunde haft ett kort möte efter varje gruppmöte, för att snacka ihop sig lite mer.

Schemalagda tider i datorsal för att jobba med projektet.

Tydlig kommunicerad tidslinje i början när allt ska vara klart.

Öka kommunikationen med kund och experter.

UG had a general low pre-knowledge, skulle tagit större ansvar själva enligt SG. Nu fick SG se till att de fick kunskapen de behövde.

"Education" should be more extensive and held at an earlier stage of the project. SG had to go in with their knowledge, would have been a major issue if they hadn't had it.

Oklart med MVD och backend, hade underlättat om det fanns mer och tydligare information tidigare om detta. (Läs igenom Technical System Description innan utvecklingen).

PG mer delaktiga i slutprodukten.