Post-mortem Report

Group 6:The Minions

EcoDriver

1.Introduction

We have set out to create an app that will help drivers of cars and trucks to drive more environmentally friendly. Eco Driver will help coach the driver by giving feedback while driving. In order to promote driving patterns that will reduce fuel consumption.

This type of driving called eco driving is typically learnt at a driving school with a driving instructor watching over your every move. We have tried to automate this process by reinforcing good driving behavior, lowering the barrier of entry while allowing the driver to keep his full attention on the road.

2.Processes and practices

Managing software projects effectively requires you to follow a suitable software process. Clearly there are also other ways of directing a project than just choosing a software process approach for it, but using a certain software process model can be key to a successful project outcome. When it comes to selecting a proper process for a project, software engineers should choose the type of development model that suit their way of working. Like every other software project, we also selected one development process to work with in order to enhance efficiency and productivity of the project. Thus, scrum methodology was chosen as software development process in this project.

As a team we were aware of the fact that using scrum does not mean that we will have a successful product, however that really depends very much on the people doing that. Other factors like the skills of the people involved in development effort can also greatly influence whether or not your process works. As the most prominent agile framework, scrum is widely used in software development. Implementing scrum in the project was a first-time practice for our team members. Luckily all of the group members had basic knowledge of working with scrum, since they participated in scrum workshop that arranged by the course coordinators.

For implementing scrum, the process has broken down to small pieces. Consequently, we had several incremental releases called sprints. In sprint planning, initial plan and user stories were defined, and they were divided into several tasks. Then we inserted all of them in our product

backlog. Furthermore, in each sprint tasks were time estimated by using poker-planning strategy. Additionally backlog grooming was done by looking at our backlog and we tried to prioritize tasks and user stories in each sprint. During scrum meetings each team member talked about what we have done yesterday, what we are going to do today and what problems we have had so far. At the end of each stand up meeting, we updated our burn down chart to visualize our progress during sprint. We also had sprint retrospective at the end of each sprint, which concentrate on giving feedback on the past sprint and stating good and bad thing about it, so we could share our experiences and learn from it and apply them in next sprint.

3.Techniques in project

In this project work, different techniques such as stand up meetings, pair programming, testing the product etc. have been used in order to enhance the quality of the work. Stand up meeting as one of the scrum method practices had many benefits for our project. The daily stand-up meeting helped us to always monitor, keep track of the status and the progress of the project. It was a good short recap stating the problems we had and getting help from one another. It also helped us to have a clear visualization about work progress, which was depicted, in a burn down chart.

Communication plays an important role in every project. In previous project (T1 project) most of the group members suffers from lack of communication between members, but having stand-up meeting was a good help this time since everyone's ideas were shared with one another. The only difficulty with stand up meetings was finding the right time for meeting that everyone was comfortable with, beside that we did not see any disadvantage of holding stand up meetings. We also consider using this technique in future project since we believe the best way to improve the project outcome is communication, which can be also obtained through regular daily meetings.

The other technique that has been used was pair programming. The primary goal of pair programming is to work together toward the best possible solution. Applying pair programming in our project lead to lower defects, faster delivery and produced higher quality work. Working together side by side at the same computer and solving the same problem helped us achieve faster completion times, better understanding of subject matters and to come up with new ideas for improvement. Since all the team members constantly communicated and reviewed together,

we produced higher quality work at a faster pace than either could do on their own. Therefore we found pair programming much more enjoyable and less frustrating.

Pair programming also made working together easier and at the same time created ownership and commitment to the collective code. The reason why we used pair programming was because we did not do pair programming in previous project (T1 project), and we faced with a lot of problems at the end of delivery and the time was so short to revise it. It affected the quality of the product, so we decided to change the strategy this time and tried pair programming, which we believe it helped a lot during development of the project. Despite all the benefits, pair programming had some shortcomings for us too. It was sometimes difficult to collaborate the ideas together. Besides, team members had different programming skills, abilities and background that sometimes lead to conflict between individuals. However, we want to use this technique in future projects because it is good practice to learn from one another meanwhile fault introduction can be minimized due to having someone review your work simultaneously.

Testing EcoDriver application was another technique that has been used. We used manual testing method for this project. One of the drawbacks of using this technique was that it took considerable amount of time to test our product. It was also hard to keep track of where the faults were introduced in which commit. We found it frustrating to test the application manually from time to time. Less accuracy was another downside of this method. In future project we would like to use automated testing, because it is faster, more accurate, less time consuming and easier to keep track of defects. We can also consider using other software development approach concentrating on testing such as test driven development depending on the type of project.

4.Work distribution and group dynamic

4.1.Time Management

The project had a final deadline set by the coordinators of the course. The product sets to deliver in three stages. Alpha release (23 March, 2015), Beta release (4 May, 2015), and Final release (22 May, 2105). Moreover, since scrum approach implemented for this work, the project has divided into 6 sprints meetings. Therefore, the target timeframe was based on specified time scale to fulfill all the deadlines in order to develop and launch the product. The actual works started on Feb 23, and executed in six two-week long sprints.

The time estimation for doing the project was usually 35-40 hours per week, but this has been changed due to having parallel courses, exams and holidays during the project work. Sometimes it was over 50 hours a week and sometimes less than 40 hours a week. So approximately it was 150 hours in total for each of group members. Fortunately we met all the deadlines and everything worked as we planned according to our time frame.

4.2. Work distribution and Interaction

The project group consists of 5 members. Unfortunately one of our members, Rickard Tengsand, dropped out from the Programme before the project starts. All roles and responsibilities were clearly defined at the beginning and the responsibilities were divided between the group members. The project divided into tasks, which completed in smaller groups. Every member of the group was responsible for implementing the part that was assigned to them. Moreover, all the tasks were assigned at an equal amount to all group members, so that the individual workload would be the same for all. We had very active participation from team members and the team worked hard to get the product done. The roles and responsibilities are listed as below:

Scrum master & Project manager: Marjan.
GUI design: Marjan, Jie.
Database Group: whole team members.
General application development group: Andreas, Tobias, Jie, and Kai.
Testing group: whole team members.

All of the group members contributed code to the product. It is worth to mention that the tasks were carried out in pair programming by group members. The reason for this is mentioned in section 2. In addition the outcome part done by individuals reviewed by the whole team. Usually the group discussed on the quality of it and decided whether to revise or not. Considering the fact that most of the members almost did not have prior experience of developing Android application, so the team tried to get the necessary training by learning the required skills, which included a lot of self study and research. This was a continuous process and it ran in parallel with the project. The workshops i.e. both in theory and practice, which was provided before the actual project, helped us in many good ways during implementing the project. We learned many new things especially software development in practice.

Our cooperation has started before Feb 23. Since we gather together and tried to brainstorm about the application ideas and product vision. Since we had parallel courses and group assignments along with this project, we easily got used to working with each other. Everyone opinions was heard and all of us publicly expressed his/her ideas. We did not have any major disagreement and conflict together. It is true that we sometimes had different opinions for making decisions regarding different stages of development process, but at the end we always respected each other's point of view and worked out on our differences.

5. Strengths and lessons learned

5.1.Strengths

The most successful part this project was that we worked together and supported one another from the very beginning of the project to the end. Some of the group members had some experience of Android application development, and some of us didn't. We created a Google document that everyone could put any relevant materials that would be useful for our project development, and every member could read the material that related to the tasks he or she had been assigned. However, even if some members had experience of working with Android, it was still ambiguity about how to proceed at the beginning and during the development before everything was done. Therefore every time we encountered obstacles and had difficulty to

proceed, we sat down to discuss, brought up any solutions we could think of and cleared our minds for next step which was truly valuable for us to find the proper solution to solve problems.

The first challenge to our project was to decide what kind of application we wanted to build. We did it by several ways, including creating a group on Facebook that everybody could post their ideas there, and we had casual discussion on daily meeting when we met and share our ideas together. We found it very efficient when we learnt new information from Android or AGA development, and inspired each other and made official decision on the app and main functionalities that we wanted to implement when brainstorming at sprint meetings.

Then around in the middle of the development process, as we worked on the app with some of the functions done and the others half done or undone, we were a bit confused and lost. With the help from our supervisor, we drew the class diagram of our app and E/R diagram of the database on board which greatly helped us to make clear of the whole structure of the app and applied the knowledge that we have learnt so far from our “Technical analysis and design” course.

Since the tasks were distributed to each group member, at early stage we worked quite separately on different parts, and did not realize that it would be huge workload and much difficult for later integration. At final stage, we had to integrate two relatively separate parts, GUI and GPS. We had meeting to study and adjust the database to serve both parts, and tried from different ways to merge them. Finally, we made it based on GUI and extract logical parts from GPS branch to finish integration.

5.2.Learned lessons

At the beginning of the development, we tried to plan our project according to Scrum approach which involved breaking down the user stories and further into tasks. The tasks were distributed to the team members according to what he or she was most interested in.

However, as most of the members never had experience and knowledge of Android application development, some of the assignee tasks were underestimated and it took us longer time than expected to implement. We had to search relative material to read and study a lot to build up knowledge that was needed to implement the functions we want. With the process ongoing and

as everyone concentrating on his or her own tasks, we underestimated the integration of different components. Finally we realized that we duplicated components for database, and consequently it would be quite problem to the integration part. We had two branches, one with GUI and AGA function and the other with GPS that displayed driving route, which were relatively separate when we found out and tried to merge them. It took us very long time to figure out how to integrate the two parts together.

As mentioned above, we had E/R diagram to make clear the structure of database, and cut duplicated parts and kept those that was needed to serve all the main functions of the app. We even tried every way to merge the two branches, first to merge from GPS to GUI, as we thought it would be better to keep logical and structured code in GPS branch, but it didn't work out. So finally, we decided to merge based on GUI with extracted logical parts from GPS branch.

After doing this long integration, we learned that it would be more proper if we made integration little by little, step by step during development. And it would be easier if we made small integrations all the time during development instead of at final stage, so once one feature was finished we could merge it into whole app as soon as possible. This was a tough lesson we learned from this project, and we will try to handle integration more continuously for our future software development projects.

6. **Reflections over non-process specific decisions**

We chose to use Google Maps as the provider for the map part of the app. This decision was made not necessarily after reasoning and comparing the alternatives; instead it was made since we knew that it is widely used, specifically in Android apps. We knew from prior experience that it would suit our requirements. However, under different circumstances, say if we wanted to charge for our app we would need to look closer into licenses and maybe choose an alternative map provider.

For the Track Driving part of the app where we display information about speed, rpm and fuel level from AGA we could have reduced the amount of information displayed but our friends and family we tested it out. The result of these tests showed that the user could get the relevant information in less than 2 seconds therefore we choose to keep the information as it is.

However our tests were not conducted in a real life situation, while driving a car. We tried to simulate as real of a situation as possible but we cannot with our limited test say for sure that it would have the same result in a real life situation. If they were to show that it took an unreasonable time, then we would redesign and reduce the amount of information displayed. For example remove the text field that shows the value of each signal (speed, rpm and fuel level) and only keep the gradient graphic.

7. What would you do differently in a future but similar project?

The biggest time sink that we encountered during the process was during integration of the GPS and Google Maps functionality and the part, which had GUI and AGA functionality. We worked in two different branches and we planned to integrate these two parts continuously as we finished features on each of the branch. However this did not work out as planned and we found it to be very difficult to structure the GPS and Google Maps part and we eventually wound up writing the code as we learnt more about Android and how to build these features.

This led to a situation where we ended up with two different branches, which were difficult and above all time consuming to integrate. This situation occurred mainly due to our inexperience. For future projects we will integrate more frequently to avoid a situation like this to occur, we also realized the importance of structure of the codebase to make it easier for integration.