The Project Report

Group 7 – Drive'n'Quiz™

29 – 05 – 2015

Contents

1. Acknowledgments

We would like to use this opportunity to thank everybody that supported us throughout the course of this project. We express our warm thanks to James, our supervisor and to Håkan and Morgan, the subject administrators, for their advices and guidance. We would like also to thank the members the other groups in this course for the great feedback during the Alpha and Beta presentations.

2. Overview

2.1. Final Report Purpose

The purpose of this report is to map the development of the project as a part of the subject Project: Systems Development, DIT524, taught at Gothenburg University.

2.2. Idea

The main motivation for the creation of our app is to create a quiz game which can be used by the driver and requires as little interaction with the phone as possible while still keeping the driver attentive and alert. We understand that using your phone in any way while driving does not promote safety, however, we also deem 'daydreaming' and tiredness to be equally unsafe. This is important since driving can be extremely dangerous. The attention of the driver needs to be focused on the situational environment which surrounds him. Through the creation of an app that requires as little attention as possible by the driver, this maximizes the time spent by the user analyzing the road conditions.

2.3. Deliverables and Outcomes

The main goal of the project is to deliver an app that will entertain the driver. Drivers should not use mobile devices whilst driving since it increases the risk of accidents. However, we are aware that we cannot control when they use their phones, therefore, through creating an app which is safer to use than standard phone applications, we will hopefully be able to entertain the driver in the safest way possible. The app will focus on increasing the driver's alertness with a quiz game, which will keep the driver entertained

as well as keep him focused and alert instead of 'daydreaming' whilst driving. The app will use a simple to understand interface as well as a large font for necessary commands whilst using text to voice to reduce time spent looking at the phone for the majority of interaction. The outcome of using the app will be to decrease the risk of falling asleep or 'daydreaming' whilst driving by using the quiz as a form of mental stimulation.

2.4. Target Audience

According to project rules our main audience is truck drivers. These types of drivers have their own characteristics. Firstly, they drive for very long distances, which means that they spend a lot of time driving. Secondly, most of their routes run not in the city, but on the highway and finally, truck drivers have to follow several important corporate rules and regulations. During the project development we need to take into account all these characteristics. Moreover, we strongly believe that our app will be interesting for car drivers as well. For example, when they drive in similar conditions or simply do not have company in the car and fear that they may be excessively tired.

2.5. Future Development

Besides being a fictional project as a part of university education, the topic covered in this project can have a practical use in real life. For real life usage, additional improvements will be required, from technical and business perspectives. For example, the app can be connected to Google maps to enhance the security aspect in relation with the actual positioning of the driver. This will also enable further specification for the questions asked which could be related to the surrounding area and therefore more interesting to the driver. Another future development can be connecting the app with various web sites for constant updates of existing facts and the addition of questions used in the quiz. Further connectivity with Facebook is another possibility for sharing scores and similar actions.

3. The Project

3.1. Design and Functionality of the Application

3.1.1. Functionality and Technology

The game is a basic knowledge quiz, which is presented in two different options depending on the driver distraction. By using AGA the app is reading the driver distraction and based on a predefined limit offers one of two possible options. The first scenario is when the distraction is lower. In this case the question is presented as text. The second scenario covers a situation when the distraction is higher. In this situation Text-to-speech is used to present the question to the user so visual interaction will be minimized. In both of the cases YES or NO possible answer are presented. The questions are having the following form: Is CITY X situated (or similar) in COUNTRY Y. The first option has a field where the question is displayed and two buttons, one for YES and one for NO. The second option has three buttons. The YES and NO buttons are the same as in the first option, and an additional button for repeating the question is added. The score will be displayed after ten questions have been answered and if the driver distraction changes under the duration of the quiz, the user will be returned to the main screen since it may not be suitable for the game to be run at that moment in time.

For programming, Java was used since this is the programming language that we have mastered the best and will therefore allow us to implement the envisioned functionalities.

3.1.2. AGA

Automotive Grade Android (AGA) is an open source software platform that enables the developers to integrate their Android application with an In-Vehicle Infotainment system. It uses standard, well-known tools to allow developers to integrate applications with the vehicle. We used AGA in our project to implement one of the safe to use while driving features. The quiz can be played in two different modes. The first mode is when the full question is displayed and the player will have to read it to be able to answer correctly. This option will be only available when the drivers estimated driver distraction is in the range of zero to one, this will simulate the equivalent of the car being at standstill or going very slowly in a traffic jam. The other option is to play while the vehicle is moving. This is still measured depending on the driver distraction, which will be in the range of two or three. This we consider as moderate driver distraction and enable the

text-to-speech to be used so the driver does not get distracted by reading the questions. Finally, if driver distraction is too high, which we deem to be four, the quiz will be unplayable and if the user still tries, he is prompted to focus on the road by a hardcoded voice message. Furthermore, if the driver resides in the driver distraction level 2 for a longer period of time, the app will output a message to the user which asks whether he would like to play the game in order to refocus attention. For this to be possible, AGA is implemented in to the application.

4. Processes and Practices Used

We decided to use the agile approach, with a SCRUM methodology that acts like a management framework for a project. It is based on backlogs, velocity evaluations, sprint cycles which last for approximately 2-4 weeks, depending on the project, and sprint meetings at the end of every cycle. There were several advantages of using agile such as the fact that it is not a plan driven model, so we were able to make a lot of changes during the process. Secondly, we could easily implement feedback from our supervisors and the subject administrators at all stages of the project. Furthermore, SCRUM was very effective partly because of the backlog we created which allowed us to see what needed to be done and prioritize accordingly and also it immensely helped with time estimations.

We implemented several SCRUM practices such as stand up meetings and sprint planning. These were helpful since the entire group met and were allowed to interact. This created a sense of security within the group so that individuals could add their own input helping to shape the project as a whole.

Another process which we used regularly was pair programming which a development technique allowing two programmers to work cooperatively on one computer. There are two roles in pair programming: coder and observer. Using pair programming was very advantageous to us since it helped us to avoid some mistakes. Furthermore, it made the process of programming much more creative, since several ideas were the result of brainstorms. Finally, using this technique we learned from each other a lot which is positive since it is the objective of this entire project.

5. Structures and Organization

5.1. Team Organization

5.1.1. Team Members

This project was done by group 7 (G7!). The group is consisted of the following members:

→ **Aleksandar Atanasov** (GitHub profile: **AlexAtanasov**)
→ **Viktor Lantz** (GitHub profile: **ViktorLantz**)
→ **Martin Kabzimalski** (GitHub profile: **MartinKabzimalski**)
→ **Darja Linkova** (GitHub profile: **Darlink**)
→ **Ilya Shabalin** (GitHub profile: **gusshail**)

During the group formation, Martin Gustavsson was as well announced as a member of G7!. However, he was not present at any of the group meetings and didn't contribute at all in the creation of the application. He will not be included in our individual evaluations and possible advantages or disadvantages that may have resulted of this situation are not taken into consideration and are perceived as irrelevant.

5.1.2. Evaluation

The majority of the time, all the team members worked together on the development of this project. In a few cases specific parts were assigned to the team members for individual completion.

Since all the members contributed equally to the development of the project, we are in agreement that all five of us take full credit and responsibility for the final result. Detailed evaluation will be individually submitted by each member via e-mail.

5.1.3. Project Communications

Communication within the group was very important for development of this project. Communication was done via e-mail, Skype and during the group meetings. Anybody in the group was able to raise questions related to the project during the meetings or state them in the online communication. There was no designated scrum master for group communications and meeting organization.

5.2. Detailed Individual Work and Time Consumption

(1d = 1 day)	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6
Aleksandar Atanasov	3d requirements analysis and vision 1d Android Studio set up 2d AGA set up	2d design DB 1d project planning documents	1d create DB and sample questions 2d create DB connector 1d build proper BackLog	1d create DB format 1d create DB connector 1d DB questions 1d toast design	1d DB questions	5d project documentation and final submission
Viktor Lantz	3d requirements analysis and vision 1d Android Studio set up 2d AGA set up			5d TextToSpeech 2d interface design / Menu activity 2d interface design / TextOnly 3d Connect AGA	2d interface design / Menu activity	5d project documentation and final submission
Martin Kabzimalski	3d requirements analysis and vision 1d Android Studio set up 2d AGA set up		0.25d state chart	5d TextToSpeech 2d interface design / TextOnly 1d DB optimization	2d interface design / Buttons UI 2d final product testing	5d project documentation and final submission
Darja Linkova	3d requirements analysis and vision 1d Android Studio set up 2d AGA set up		0.25d state chart 1d build proper BackLog	2d interface design / Menu activity 2d interface design / TextToSpeech 3d Connect AGA	2d interface design / Menu activity 2d interface design / Buttons UI 2d final product testing	5d project documentation and final submission
Ilya Shabalin	3d requirements analysis and vision 1d Android Studio set up 2d AGA set up	2d design DB	1d create DB and sample questions 2d create DB connector	2d interface design / TextToSpeech 1d DB optimization and questions 1d create DB format 1d create DB connector 1d toast design	1d DB questions	5d project documentation and final submission

5.3. Team Cohesion and Interaction Challenges

The group worked very closely together during the creation of Drive'n'Quiz. Since we were relatively new to programming in general and completely new to Android Studio it initially took some time to get started and this was our first initial challenge. We had discussions within the group whether we would use Eclipse or the newly proposed Android which created a small conflict. However, as we discussed the pros and cons of using either IDE, we made a combined decision to go with Android. After this initial conflict, we decided on regular weekly meetings where each member of the group could voice any concerns they had about the project so far.

This was extremely effective in order to keep the cohesion within the group high since we were able to speak about which direction we wanted to take the project in and everyone would discuss whether it was a viable idea.

6. Project Time Management

The project schedule was developed during the meetings of the group as well as online communication. Rough schedule was introduced rather early during one of the group meetings. It took into consideration the amount of work needed for each part as well as other obstacles such as exams or programming projects. The monitoring of the progress was done continuously during the group meetings. Project members were free to make suggestions about the schedule. All the group members attended most of the meetings. Decisions were made after group discussion by the present members. Schedule was done with reference to project milestones. Schedule was constructed with suggestions of all team members and was flexible due to occurrence of various situations, such as sickness of a group member, different work load and other activities (exams, other projects, etc.).

6.1. Timetable

The presented timetable was created in relation to the predefined milestones published in GUL.

Date	Description
23-02-2015	Sprint 1 start, 23rd of February to 9th of March
09-03-2015	Sprint 2 start, 9th of March to 23rd of March
20-03-2015	Alpha version completion deadline
23-03-2015	Sprint 3 start, 23rd of March to 20th of April
20-04-2015	Sprint 4 start, 20th of April to 4th of May
30-04-2015	Beta version completion deadline
04-05-2015	Sprint 5 start, 4th of May to 18th of May
18-05-2015	Sprint 6 start, 18th of May to 22nd of May
20-05-2015	Final version completion deadline
27-05-2015	Final report and supporting documentation completion deadline
28-05-2015	Final submissions last revision and language control
29-05-2015	Deadline for submission complete project
29-05-2015	Deadline for sending individual evaluations via email to Håkan and Morgan

6.2. Milestones

All the deadlines were met as presented in the Project plan in accordance with the subject administration as published on GUL.

Date	Description	Outcome
23-02-2015	Sprint 1 start, 23rd of February to 9th of March	On time
09-03-2015	Sprint 2 start, 9th of March to 23rd of March	On time
23-03-2015	Alpha version release	On time
23-03-2015	Sprint 3 start, 23rd of March to 20th of April	On time
20-04-2015	Sprint 4 start, 20th of April to 4th of May	On time
04-05-2015	Beta version release	On time
04-05-2015	Sprint 5 start, 4th of May to 18th of May	On time
18-05-2015	Sprint 6 start, 18th of May to 22nd of May	On time
22-05-2015	The Final Version presentation	On time
29-05-2015	Final report completion and submission into GUL	On time

7. Project Risk Management

7.1. Risk Management

As a result of our experience from the first semester, we were already aware of the possible risk that may endanger the successful completion of this project. After the project vision was created, we discussed all the potential risks that may have negative impact on the project. We identified and analyzed all the possible risks from multiple perspectives.

We identified and analyzed all the possible risks from multiple perspectives, and defined them as category of product, project and people. We analyzed the likelihood that they could happen, and the different effect levels that would impact the project and presented our strategies how to manage these risks and avoid and minimize the damage to the project in our project plan.

The described risks were continuously monitored and recorded during the progression of the project.

7.1.1. Product risks

Design risk

- **Description:** Low quality design.
- **Probability:** Low.
- **Strategy:** To collect as much as possible feedback during the Alpha and the Beta releases, keep on checking our design during the project and adjust it accordingly.
- **Outcome:** Lots of feedback was received from supervisor and also from the course managements which helped to increase design quality. Furthermore, during the Alpha and Beta presentation we got some interesting ideas from fellow class mates which helped to shape our final application.

Technical risk

- **Description:** Lack of development experience and the lack of knowledge to implement the system can affect the development.
- **Probability:** Moderate.

- **Strategy:** Improve our technical knowledge and development skills during project, consult experienced developers, share knowledge and support each other in the team.
- **Outcome:** This was a risk which we knew was unavoidable since we are learning to code whilst creating the project itself. It was, more than anything, time consuming to learn whilst working but otherwise this was a risk which we were able to work around using teamwork and internet guides.

Requirement risk

- **Description:** Requirements misunderstanding.
- **Probability:** High.
- **Strategy:** Study carefully the system requirements before the start of the project and keep track of it during the whole project.
- **Outcome:** All the group members attended the five workshops related to this project. As a result of it, we were all familiarized with the requirements and safety standards that are important for such a product. Even though this risk was graded as high, our responsible approach eliminated negative outcome.

7.1.2. Project risks

Timeline underestimate

- **Description:** No experience with Android applications development can lead to time shortage.
- **Probability:** Moderate.
- **Strategy:** Monitor the time we spend on developing the system during the whole project, and adjust tasks during the sprints accordingly.
- **Outcome:** Due to other school responsibilities, there were uncompleted tasks in the first sprints. However, as the other subjects were completed, this practice was corrected and the final result was not negatively affected. We found that as the project progressed we were able to reuse knowledge that we had gained from the previous sprint and apply it to the problems which occurred in the next.

Size underestimate

- **Description:** We might underestimate the project size, since we are inexperienced working on Android system development. The scope of the project might be larger

than we anticipated, more complex functions can be expected, which may lead to failure fulfilling the preset requirements or it may create a setback in the original time plan which we did not account for.

- **Probability:** Moderate.
- **Strategy:** Keep our design of the project within reasonable scope. Monitor the project size from the very beginning and carry out evaluations from time to time in order to assure ourselves that we will be able to complete the designated tasks.
- **Outcome:** We were relatively successful at keeping track of how large the project was becoming. Before the presentations we were always quite stressed in order to get desired functionality but this did not come as a surprise since it is usually how things are. As we approached the final presentation we had made sure to manage the tasks and this lead to a very comfortable final sprint where we only had minor alterations or additional features to implement.

Poor integration of system

- **Description:** All the components processed in the project must be compatible.
- **Probability:** Moderate.
- **Strategy:** Start the integration as early as possible to minimize the risk.
- **Outcome:** The strategy was followed and this risk was minimized.

Cooperation with other teams

- **Description:** During the creation of this project, the only cooperation with the other teams is in the informal spirit. Feedback can be received during the Alpha and Beta releases.
- **Probability:** Very Low.
- **Strategy:** Use the Alpha and Beta releases to gain ideas from the other group project and present our progress to other students for possible constructive feedback.
- **Outcome:** During the Alpha and Beta releases we received some feedback that was eventually implemented into our final product. This was very helpful and constructive since at times we were narrowly following our vision and additional possible implementations were marginalized. Getting the opportunity to see what other people are working with whilst they also give you advice on what could be improved with your own project is extremely rewarding. It is subsidizing "the real life" stakeholder-developer relationship.

7.1.3. People risks

Absence of team members

- **Description:** Team member could be absent because of illness or leave for vacation.
- **Probability:** Moderate.
- **Strategy:** We planned all the defined tasks to be processed in couples for better efficiency and as well for short term back up.
- **Outcome:** Most of the project was done without major absences, so occasional absence of some of the members didn't significantly influenced the project progress. We worked around this by also having continuous contact through different communication platforms and having the shared GitHub repository allowing access to all group members' workspaces which meant that even during absence, their work would be part of the development progress.

Conflicts

- **Description:** There can be some conflicts during the project due to different perception of the application features and internal group arguments.
- **Probability:** Low.
- **Strategy:** Discuss the outcome of the project as soon as possible and don't implement new ideas if all the members are not in agreement with it.
- **Outcome:** All the members were as professional as needed, so no serious conflicts occurred. Arguments within the group were handled immediately so that they would not develop into a larger problem which could hinder the project as a whole. Open communication additionally prevented this risk.

Low team motivation

- **Description:** Team members can have short term motivation problems that can lead to uncompleted task during the sprints.
- **Probability:** Low.
- **Strategy:** Be open about our current situation regarding the project and also the additional workload from other subjects.
- **Outcome:** All the team members were fully motivated since the first day as we thought that the idea for our app was interesting, so low motivation was not a particularly high risk for successful project completion.

7.2. Risk Management Table

Risk	Can affect	Probability	Effects	Outcome
Design risk	Product	Low	Serious	Positive
Technical risk	Product	Moderate	Serious	Positive
Requirement risk	Product	High	Catastrophic	Positive
Timeline underestimate	Project	Moderate	Serious	Positive
Project size underestimate	Project	Moderate	Serious	Positive
Poor integration of system	Project & product	Moderate	Serious	Positive
Absence of team members	Project	Moderate	Tolerable	Positive
Low team motivation	Project	Low	Tolerable	Positive

8. Strength and Lessons Learned

8.1. What Went Well in Our project

We felt, as a group, that this project went well due to a number of different reasons. Firstly, communication within the team was extremely well managed and this led to clear objectives being set both before, during and after sprints which enabled the entire group to work efficiently and cooperatively towards a common goal. This can be seen from the milestones which were met on time and the satisfactory standard to which deadline were met.

Secondly, we believed that we managed to very early on, pinpoint a topic which we all found to be compelling yet at the same time relevant and useful in real life situations. Having this clear vision set in stone early in the project lifeline allowed for easier transitions between the different development stages of the product since we knew in what direction we wanted the product to grow.

Finally, organization and the management of time and work was also well managed. We did not assign a scrum master for each sprint, however, we worked around this by having Alex as an organizer who took care of setting meetings and initiating standup meetings

and similar scrum concepts. This was beneficial for less prominent members of the group since they were able to have a clear schedule to refer to concerning meetings and also an easily approachable person to ask questions if something was unclear.

8.2. What Did Not Go Well

There are always aspects of a project which can be improved no matter how experienced or skilled the participating members may be. Therefore, it is not surprising that we, as a group, also had certain issues which could have been addressed in order to have made the project a smoother ride.

There is an obvious lack of previous programming knowledge among the members of the group which leads to the problem that we're forced to learn how to code many things whilst creating them for the first time. This has its clear disadvantage of taking a larger amount of time than, for comparison, a group which has a more experienced programmer who is able to explain how a certain function is written and then helps those less experienced when they encounter problems. This can, however, be considered positive from a certain perspective since when we're required to learn and do it by ourselves, we know how to do it in the future.

Another issue which could have gone better was attempts at interacting with AGA. AGA itself is an extremely useful application which has plenty of useful features which can be used when working with automotive signal simulations. We, however, believe that the information and instructions given concerning AGA could have been much better presented and more thoroughly demonstrated, especially since it played such a pivotal point in the project. Our interaction with AGA was very time consuming and we believe that with better guidance, more focus could have been put on other aspects of the product.

Finally, insecurities were initially high around Git and GitHub. Having a VCS implemented within a project is vital in order to allow work to be efficiently carried out simultaneously by multiple people. Since this was a new concept to us, we were slightly confused about how it worked and how we would go about incorporating it. This led to some problems early on, however, once Viktor got some help from the supervisor he was able to show the rest of the group how Git worked and from there Git became a vital tool during the development of Drive'n'Quiz.

9. Conclusion for Future Projects

This project has been extremely insightful into how to work together as a group in order to achieve a finalized quality product. Knowledge from the courses which have been studied this year have contributed to how we approached the problems encountered. We have learnt much, both individually and together, something that will affect the way in which we would approach a similar future project.

Continuous documentation of what is done or not done at the end of each sprint could have been more detailed, this would have made insight into the project from an outside perspective a great deal easier. It would also have made the writing of the final report a simpler ordeal since we would have had more 'flesh on the bone' about each activity carried out rather than having to try to remember what was done. We did do this, but it could have been done more detailed.

Earlier implementation of the VCS would be a much higher priority for a future project. We did not realize how helpful the use of Git and GitHub was and we believe that this hindered the progress of the application early on during the development of the product. Prioritizing this would therefore enable the project to faster progress towards its final state and would also relieve stress in several areas of the group since there would be less conflicts about who had the updated version. Also, it allows for clear accountability about who in the group has contributed what which is also very useful towards the end.

To conclude, we feel that the project has gone very well and surprisingly smoothly considering how new we are to programming. We are proud of what we have achieved and are pleased with the resulting product. Drive'n'Quiz is a quiz which with a few tweaks and additional implementations could be a game that could be used in a truck or car and even have the potential to save a human life which we believe to be the ultimate reward when creating an application like this.

10.References

The following sources and literature was used to develop this project:

- Software engineering; Sommerville, Ian — 9th ed., 2011, ISBN-13: 978-0-13-703515-1
- Introduction to Programming Using Java; David J. Eck - 6th ed., 2009, ISBN 13: 9781616100995
- https://developer.lindholmen.se/redmine/projects/aga/wiki
- http://developer.android.com/index.html