Agile Software Development

Android Prototype For The Execution of Daily Walkaround Inspections

Henric Salomonsson
Abstract

Agile Software Development

Henric Salomonsson

Keeping vehicles in good condition is key to any driver and haulage firm. By not providing simple and accessible means of reporting daily walkaround inspections, they are often done hastily or not at all, resulting in a decrease in uptime as well as road safety of the vehicles. Through close cooperation with haulage firms and development over short iterations, a prototype of an Android application and a corresponding web portal for the reporting and managing of daily walkaround inspections has been created. When letting drivers and administrators evaluate the prototype it has become clear that if done correctly, this type of support has a large chance of being positively accepted and used on a daily basis. This would provide the haulage firms with easily accessible information and statistics about their vehicles.
# Contents

1 Glossary 5

2 Introduction 6
   2.1 Objectives .............................................. 6
   2.2 Scope .................................................. 6

3 Background 7

4 Theory 8
   4.1 Daily Walkaround Inspection ...................... 8
       4.1.1 Traffic safety .................................... 8
       4.1.2 Case study ......................................... 8
   4.2 Agile development ..................................... 9
       4.2.1 What Each Iteration Must Include ............... 10
   4.3 User centered development ......................... 10
       4.3.1 Usability .......................................... 10
       4.3.2 PACT ............................................... 11
       4.3.3 Personas and Scenarios ......................... 11
       4.3.4 Usability Evaluation ............................ 12

5 Method 13
   5.1 Information gathering ............................ 13
       5.1.1 Interview .......................................... 13
       5.1.2 Questionnaire .................................... 14
       5.1.3 Observation ....................................... 14
       5.1.4 Data from secondary sources .................. 14
   5.2 Scrum ................................................ 15
       5.2.1 Roles ............................................... 15
       5.2.2 Events .............................................. 15
       5.2.3 Artifacts .......................................... 15
   5.3 Usability evaluation .............................. 16
   5.4 System ............................................... 16
       5.4.1 Android ............................................ 16
       5.4.2 Web interface .................................... 16
# List of Figures

6.1 Example of a daily walkaround inspection ........................................... 18  
6.2 The general process of an inspection .................................................. 19  
6.3 SystemOverview ................................................................................. 20  
6.4 The two options .................................................................................... 24  
6.5 The two options .................................................................................... 25  
6.6 It is clear by looking at the tabs that this is the last step of the process ................................................................. 26  
6.7 The two options .................................................................................... 27  
6.8 One chosen vehicle ............................................................................... 28  
6.9 Included information .......................................................................... 29  
6.10 Included information .......................................................................... 30  
A.1 Example of defect report ................................................................. 37  
B.1 Flow diagram ....................................................................................... 40  
C.1 Product backlog .................................................................................. 41  
C.2 Sprint backlog ....................................................................................... 42  
E.1 Two views ............................................................................................ 49  
E.2 The possibility to choose truck and trailer ......................................... 51  
E.3 Step-by-step startup ............................................................................ 53  
E.4 Button design alternatives ................................................................. 55  
E.5 Two ways to comment ......................................................................... 56  
E.6 Added reports ....................................................................................... 57  
E.7 Online or Offline version ................................................................... 59
Chapter 1

Glossary

API - Application programming interface
JDK - Java development kit
SDK - Software development kit
ADT - Android Development Tools
IP - Internet protocol
XML - Extensible markup language
Chapter 2

Introduction

2.1 Objectives

The purpose of this thesis is to develop a functional prototype of an Android application to be used for the execution and monitoring of daily walk around inspections that is performed on trucks. Through empirical evaluation with theoretical backing this report will attempt to answer the following questions:

1. How to develop a prototype that seizes the natural process of a daily walk around
2. How to develop with a user centered approach
3. How to develop using agile techniques with short iterations

2.2 Scope

The prototype will be using Android. No focus has been on researching alternatives, such as using iPhone or a web portal. Moreover, not optimization and security issues are explored at this stage. Agile development will be based on the use of Scrum.
Chapter 3

Background

The daily walkaround inspection routines of haulage firms today is lacking in several aspects. Essentially, it is up to each firm to decide when and how to perform the inspections. Oftentimes this leads to inspections not being done at all. In addition, keeping rigorous inspections requires a large amount of administration due to the quantity of paperwork it produces. Even more importantly, better inspections lead to safer vehicles, which means safer roads, and higher uptime, which means higher revenue.

By not detecting errors and defects at an early stage, making a complete service repair is very difficult and one or even two revisits are not uncommon. This could, for example, be due to limit of time, specialists not being present or spare parts not being in stock. One step in the direction of detecting as many defects as possible is to provide a structured solution of how to report and perform daily walkround inspections, as well as doing follow ups on them. With this background, the idea of an Android support device came up. By providing a simple solution on a device that the driver already has, much can be gained.

While using Android is an important part, the main focus of this master thesis is not the application itself, or the use of this specific platform. By the use of rapid prototyping (through Scrum) the application acts as a tool to present to the user with the goal of getting as much information as possible on how a solution like this could be done successfully.
Chapter 4

Theory

4.1 Daily Walkaround Inspection

4.1.1 Traffic safety

The Swedish trade organization for haulage firms; Sveriges Åkeriföretag, recognizes in their handbook (Åkeriföreningen, 2011) that daily walkaround inspections are an important part of vehicle maintenance. The recommendation is to use a checklist when performing the inspection and to report any remark found. An example of what a checklist can look like can be found in Appendix A.

A report on traffic safety within companies with heavy goods vehicles (Trivec-tor, 2003) concluded that above all, bad breaks and faulty tires were the main source for severe accidents on the roads. In addition, when this happens it usually generates negative attention by the media. According to a survey done by Åkeriförbundet one third of all haulage firms has at some point experienced faulty tires. This also has the effect that a substantial portion of all heavy vehicles are failed at the motor-vehicle inspection as a result of bad breaks. The report also states that two out of seventeen heavy goods vehicles that are involved in severe accidents had at lest one defect that could have been detected beforehand and thusly affect the outcome of the accident.

4.1.2 Case study

An interesting case study within the field of daily walk around inspections is that of the situation in England, where the inspections are prescribed by law (Federal Motor Carrier Safety Regulations 49 CFR paragraph 392.1). In Guide to maintaining roadworthiness (Vosa, 2008) general guidelines are provided. The driver of the vehicle is responsible for keeping the vehicle in drivable condition and he is also responsible for writing a remark report in case any defect is detected. The daily walk around inspection has to be made at lest once every 24 hours of both the truck and the trailer each and even if no defect has been detected, a so called Nil report has to be filed. Any reported defect need to be
stored for 15 months. An example of a checklist from a haulage firm in England can be seen in appendix A.

4.2 Agile development

The idea behind agile development first arose in the 70s as an alternative to the then (and still) widely used waterfall method. The waterfall method is based on the idea that a project should be executed in a sequential manner starting with requirements gathering moving on to design and further on step by step. This mindset is fundamentally flawed, as it has shown that projects almost always change from the original plan and it is estimated that only 9 - 16% of these projects were on-time and on-budget. (Danube Technologies 2004)

Agile development includes a variety of different methodologies that can vary widely in nature. However, at the core of the term agile there are 4 values as decided by the agile alliance called the agile manifesto (Alastair Cockburn 2002). These are:

Individually and interactions over processes and tools
People are the most important assets of the project and it is important to remember that people are different and that there are no universal processes and tools. Projects benefit from good communication and should be held as more important than for example documentation.

Working software over comprehensive documentation
While documentation is important at several stages, eg. requirements gathering, the only true measure of what has been done is the working system. Documentation used moderately in conjunction with running code will show how well a project really is doing.

Customer collaboration over contract negotiation
In a project with good collaboration there should be no "us" and "them", instead developer and customer should be considered as one. While the need for traditional artifacts such as the contract might still exist and be useful, good collaboration is generally more important and can reduce unnecessary conflicts.

Responding to change over following a plan
Flexibility is important as the project will oftentimes change as the development process goes forward. The relatively short (2 - 4 weeks) iterations of many agile methodologies helps the project to respond to change. A project plan therefore need to be adaptable to change in order to work in conjunction with the agile approach.
4.2.1 What Each Iteration Must Include

There are three general requirements for iterative work (Gulliksen and Göransson, 2002). These are:

1. Identification of necessary changes
2. A possibility to perform changes
3. A will to perform changes

More specifically, it can be described like this:

1. A thorough analysis of user requirements and context of use
2. A prototype development phase
3. A documented evaluation of prototype usability that must result in a conscious decision about changes that affects the continuing prototype development.

4.3 User centered development

4.3.1 Usability

"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO 9241-11 Guidance to usability)

This is one definition of usability that is interesting because it takes a wide approach to usability and is recognized worldwide (Gulliksen, Göransson 2002). There are several dimension to the concept of usability and some of it’s identified attributes are that it should be easy to learn, efficiency to use, easy to remember, have few errors and be subjectively appealing. The good thing about these parameters are that they are measurable and thereby comparable between different software. Efficiency to use could for example be measured by taking the time for performing certain tasks. However, there are more to good design than usability.

Though it is important for a system to be usable, good design is a complex matter with several parameters to consider. Most importantly, a system need to be accessible firstly, usable secondly and acceptable thirdly in order to be successful (David Benyon 2010).

Accessibility refers to the degree a system is accessible to different people with different physical and mental abilities and restraints. For example, a person in a wheelchair could find it hard to reach highly placed devices. It is often very hard to make a system usable by everyone, due to technical or financial
restraints, and therefore the concept of inclusive design is applicable. Inclusive design means that by identifying special characteristics of the user and weighing them by estimating how often they occur as well as the cost to fix them, all characteristics that pass a pre-decided value threshold are covered.

Separated from usability as well are acceptability. Where usability can be examined by different kinds of scientific evaluations, acceptability need to be tested in a context of use. Whether or not a technology saves sufficient enough resources (for example time or money) decides if it will be used or not. This also applies to perfectly effective and efficient technologies.

4.3.2 PACT

Pact stands for People, Activities, Context and Technology, and basically this refers to the fact that "people use technologies to undertake activities in contexts" (David Benyon 2010). Identification of these are key to successfully develop software that is usable in real life situations. The four are explained as follows:

People
Different physical and mental backgrounds lead to different design decisions. For example, some are color blind or in the extreme case, blind even. The design should obviously cater to as many people as possible, but some restrictions always have to be made.

Activities
Depending on the nature of the activity that should be performed, different aspects need to be considered. Most importantly, the overall purpose of the activity should be well defined. In addition, other factors such as complexity and safety should be thought over.

Context
Identifying where and in what social/organizational context the activity takes place is of high importance. Some devices are sensitive to dirt, while others are sensitive to bright sunlight.

Technologies
Focuses mainly on how the input and output is handled, as well as communication between different devices (should one exist).

4.3.3 Personas and Scenarios

Personas are representations of potential users of the system. A short description on who they are and what background they have will assist in visualizing the specific person. This is useful as an aid in the process of trying to identify users of the system.
Scenarios are created when the personas are put in a context, performing an activity with the help of some technology. As personas, this approach helps the developer visualize how the system could be used. However, the optimal solution is to interact with the user, but resource-wise this is very expensive. Scenarios are a cheaper method that produces a similar effect.

4.3.4 Usability Evaluation

Evaluation should be conducted regularly throughout the entire development process. While it can be easy to point out a specific problem, the challenge is oftentimes to decide how to solve it. Using a combination of some of the methods presented is a good way to provide coverage and it is preferred to involve the users as much as possible. User centered system design (Gulliksen and Göransson, 2002) describes a number of common methods.

Expert evaluation:
As the name suggest, usability experts evaluate the system. This method can detect problems that violates usability theory, but usually not domain related problems. It can be performed quickly and efficiently, but since the expert isn’t commonly a user also it provides for a limited insight into how the system performs.

Field study:
Once the system is deployed, a field study can be conducted in the normal work environment of the user. Initialization is done by separating users into user categories by letting them answer background questions. The evaluator then observes the work being done by each user using the system.

Scenario based evaluation:
By letting the user perform a task presented as a scenario, the success of the performance can be evaluated. This is useful for systems in development and goes well with agile methodology. This has been used to a large extent. Detailed descriptions of the different personas and scenarios can be found in Appendix D.
Chapter 5

Method

5.1 Information gathering

In Designing Interactive Systems (Benyon, 2010) the understanding of the requirements of a system is being handled, and several ways of reaching that enlightenment is being proposed. Before being able to design a successful system it is key to research which users that will use the system and what they want to do. It is also important to understand the shortcomings of the current system. This knowledge can then be turned into requirements that builds the foundation for the design process. However, it is very difficult to get all the correct requirements in the first step which means that research will be conducted during all phases of the development process. This notion favors agile methodologies.

While it is best to cover all the gathered requirements, this is seldom possible due to limited resources. It is therefore useful to prioritize and decide the importance of each requirement. One recognized way of doing this is by the use of a MoSCoW analysis, in which a requirement is decided to be something that the system must have (in order to function), should have, could have or want to have (but won’t have at this stage).

Several different methods of data collection and how they were used are described. A common denominator of a majority of them are that they include the user. An important insight is that the researcher himself isn’t the user of the system and therefore a human-centered approach is recommendable. While it is seldom possible to include all users, it is good to cover all the major stakeholders.

5.1.1 Interview

The first methodology described is the interview. An interview can either be unstructured, structured or anywhere in between. A strictly structured interview
uses pre-written questions. This provides the advantage of uniformity, meaning that answers can easily be compared to each other and general tendencies can be concluded. Moreover, it requires less skill than its counterpart since it limits the possibilities for discussion. On the other hand, the unstructured interview doesn’t have any pre-written questions which can sometimes be useful to avoid possible preconceptions of the interviewer. A popular method is the compromise between the two, the semi-structured interview. With pre-written questions paired with both flexibility and domain knowledge, sidetracks can be explored making the interview more in depth.

5.1.2 Questionnaire

If no direct contact with the user is possible, an alternative is to use a questionnaire. In addition, it is possible to reach out to many more people than for example with an interview where a form of direct connection is required (be it physical location or by telephone). The downside is that it is very work-intensive to construct and requires careful design. With less than about 10 people to conduct the research on, it is seldom efficient to use as a replacement for using interviews.

These methods are directed at one individual at a time, but it can also be important to see how a group of people reason and take to the task. A common method is focus groups where a group of people react to question presented and in turn to each others responses. Another method is using brainstorming.

However, questionnaires were not used because of the limited number of users that could be reached.

5.1.3 Observation

In some cases it is difficult to explain how a task is being performed in real life and therefore it could be useful to observe it instead. This way the researcher can quickly get an understanding of the task, though it is important to be mindful of the fact that the process can’t be generalized over one observation. Moreover, while being a resource-economic method, it comes with some drawbacks. Most notably, people that know they are observed may change their behavior. It is important to make the person feel confident to reduce this effect. Another problem is observer bias. For example a biased comment from the observer might change the behavior of the individual being observed.

5.1.4 Data from secondary sources

As mentioned before, a human-centered approach is necessary in to get a comprehensive insight to requirements, but this should be supplemented with research on existing reports and studies. There’s also the possibility of similar work that has been done.
5.2 Scrum

In The Scrum Guide (Schwaber and Sutherland, 2011) Scrum is described as a lightweight and simple framework that can be used to support complex product development.

5.2.1 Roles

A Scrum team should consist of a Product Owner, a Development Team and a Scrum Master. The product owner is responsible for the product backlog and making the development team understand it. The Scrum master has a coaching and leading role for the team. In order to fit these roles into a team of two the Product Owner and Development Team were defined as one role.

5.2.2 Events

Regular time-boxed events are used in order to minimize the need for other meetings. The first of these are the sprints which is the Scrum term for iterations. Each sprint should include a sprint planning meeting at the start as well as a sprint review meeting and sprint retrospective at the end. The time-box for a sprint were set to two weeks. This was enough time to present a new version of the prototype, and still be able include a large number of sprints. In addition, Daily Scrum should be held every day where the following three questions should be asked:

1. What have I done since yesterday?
2. What will I do today?
3. What prevents me from performing my work as efficiently as possible?

5.2.3 Artifacts

Central to Scrum are a number of artifacts. The product backlog is first set up and includes all the functionality that should be implemented. Each sprint a number of items, called stories, are taken from the product backlog and put into the sprint backlog, the second artifact. The sprint backlog includes everything that should be done during a specific sprint. A burndown chart can also be used to illustrate how much work that is left. This artifact were left out in favour of a more traditional breakpoint chart. Examples of a product and sprint backlog that was used can be seen in Appendix C.

The requirements were gathered from interviewing each of the stakeholders and put down as user stories. From the gathered requirements a product backlog was set up. Functional requirements were not included in the product backlog. Instead a velocity was set up that decided the percentage of the time of the sprint that should be spent on items from the product backlog and that should be spent on other items (such as meetings and thesis report writing).
5.3 Usability evaluation

At the end of each sprint, a usability evaluation were conducted and documented (the result can be seen in appendix E). By doing this, the last requirement of what each iteration must include has been satisfied. The most common approach for evaluation was the scenario based. The reason for this was that it goes well with the agile process and uses the potential of the prototype to a maximum. With each iteration providing new functionality or design to the prototype, based in last iterations evaluation, the prototype is constantly evolving according to the will of the users in conjunction with the will of engineers at Scania.

The prototype were also evaluated by using measurable evaluation metrics such as effectiveness (is it quick to use) and learnability (is it easy to learn).

5.4 System

5.4.1 Android

In order to demonstrate the prototype to the user, the Android platform were used. As such, the use of Android were a mean to reach the conclusions in this report and not the purpose of the study. Details about Android can be found in Appendix B.

5.4.2 Web interface

The web interface was made as a basic homepage that could send/receive data to and from the Android device as well store and display the received data.
Chapter 6

Result

6.1 Daily Walkaround Inspection

The routines for daily walkaround inspections are different in each haulage firm and are, in a strict sense, being done sporadically at best. This conclusion has been drawn after visits to three different types of haulage firms around the area of Södertälje. A common mentality among drivers are that they are in touch with their vehicle and instantly knows if something is wrong. No written checklist (see appendix A for example of a checklist) are handed in. However, the administrative personnel are requesting this and are positive towards that added structure that the application would provide. It was also detected that there exists different types of inspections, not just the daily walkaround. Examples include weekly inspections and even scheduled inventories.

The performance of a daily walkaround inspection at Transportlabbet were observed and after that, the following illustration was made:
As can be noted, more general categories (engine, front side, left side etc) can be derived from the items to check. By mapping these categories in the correct order to the technology used for reporting the inspection, the nature of the process will be kept.

A general process for performing an inspection was also identified. This process consists of the minimum amounts of steps needed to perform a walkaround. The bent arrow illustrates that this process is done repeatedly.
6.2 PACT

6.2.1 People

The main group consists of truck drivers and their administrators. Ages and attitude towards technology vary widely and not everyone can be said to be familiar with smartphones, though everyone is more or less familiar with the use of mobile phones. Other than that, the user group is fairly homogeneous and consists of truck drivers. Motivation could be provided either by being mindful of having a safe vehicle or by showing administration that inspections are being made. Most people are used to performing the activities supported by the technology, but not the technology itself. Culture, or "this is how we do it here" will be a strong factor. Physical constraints that need to be considered are: large hands, far sightedness (problems with reading small text) and color blindness.

6.2.2 Activities

The different types of activities can be separated into two main groups: activities performed on the Android and activities performed on the web interface.
All activities have in common the characteristic that they are non complex, fairly quick and should be supported by an intuitive interface. They are carried out alone.

**Android: Perform different types of inspections**
This activity is current mostly performed by the use of a paper report that is written and handed in to the office, but the exact process varies from company to company. The purpose of this activity is to make sure that the truck (and trailer) are in good condition and to show (to the administration) that regular inspections are being made. The activity should be done every day and as such it is important that everything can be done quickly (around 5 minutes). It takes place mainly during the start and/or end of the work day. The activity is safety critical in extreme cases, where failing to report a critical defect could lead to a potential hazard. It is important that the design follows the natural flow of the walk around so that it doesn’t come across as interfering with an already learned task. All input is being made by the use of the touch screen.

**Android: Send remark**
The purpose of this activity is to quickly be able to send a remark without having to go through the process of making an entire inspection. This activity isn’t done regularly, only when something happens unexpected happens.

**Interface: Monitor vehicle conditions**
This activity is currently being done by checking the paper reports that the drivers hand in. The purpose of this activity is to support forward planning
and to make sure that all vehicles are roadworthy. If something is wrong with a vehicle, the administration need to be informed as quickly as possible. By receiving information about defects as early as possible, it is possible to inform the workshop before truck service as a preemptive measure.

**Interface: Monitor that inspections are being made**
This activity is currently being done by checking the paper reports that the drivers hand in. The purpose of this activity is to make sure that inspections are being made often enough and as a guarantee that the driver has done the inspections as required.

**Interface: Administer (add/remove/edit) drivers, vehicles and types of inspections**
The purpose of this activity is to set up content of the Android application to support the specific company. This is not done regularly, but need to be done before the Android phone can be ready to use.

### 6.2.3 Contexts

Locating the device is the first consideration. The physical condition consists of anything the weather can provide, with everything from rain and snow to bright sunlight. The environment is often very dirty and working gloves are sometimes used. The activities can be performed either at the firm of haulage or somewhere along the road on a long haul. Stress could be a factor. It is sometimes hard to see the importance of performing the walk around, especially for doing it every day. Organizationally, each company has their own rules and regulations, but there are also laws concerning the subject. The activities should support and strengthen the check up on this. Scania and their service workshops are also interested in knowing if any defects are being reported in order to be able to plan more efficiently (service appointments, stock purchases etc).

### 6.2.4 Technologies

A small amount of data has to be entered quickly, on the Android this is done with the help of a touch screen. It has to be obvious at every stage what to do next in order to complete an inspection. The communication between the Android and the computer need to used wireless communication and the output from the Android need to be handles by a server. On the server, a database need to be set up where both the interface and the phone can get information.

### 6.3 Scrum

Scrum has been a central part of development and the prototype is a result of the agile work being done using Scrum and the outcome of each iteration can be seen in Appendix E. The iterations has been described with a summary, an
analysis, an evaluation and a lessons learned.

Over the course of the thesis work, several insights into how to work in an agile way were gathered. The advantages that was noticed were:

1. It was interesting to see that the enthusiasm of the stakeholders increased by each time new material was presented and by showing progress continuously with regular intervals, they developed a confidence in the product and were able to provide thoughtful insights into it.

2. Each scrum demonstration also meant having a presentation of the results which often led to discussion and insights into what was really achieved. Demonstrating the product as early as possible, even though it is far from finished, is a good way to detect and adjust error before they are irreversible.

3. Communication is a key aspect of Scrum and this means that the team shouldn’t be too small in order for it to provide a significant productivity gain. However, even a team with as little as two members can work efficiently using the framework.

4. Many of the advantages of agile methodology has one thing in common; they are related to keeping close contact with the user during the entire development process. This was one of the biggest advantages of working agile that was noticed.

Working agile also presents a number of obstacles and pitfalls. Those that came up were:

1. Scrum is a lightweight framework but even so it is not always the work flow itself that has to be adjusted, but also the framework itself. The key rules to what an agile process and iterations must include should be fulfilled, but other than that it is a question of what can be applied to the specific project.

2. Writing stories for the sprint backlog that are descriptive is difficult but any extra effort put into this pays off during the sprint. It is also important to not allow any confusion over how the result should be demonstrated and when it is done, the so called definition of done. The Scrum master and the development team might have different opinions of what the result should be otherwise.

3. One of the advantages of using short sprints are that different branches could be explored in order to conclude what’s best for the specific case. It is however not the easiest thing to conclude that the result of a sprint was unusable and that it should be discarded. By keeping a mindset that values the lesson learned and that the insights gained are important enough, the value in both good and bad results can be seen.
4. It will require a couple of sprints in order for things to get set, so the variables such as the length of each sprint should be flexible and adjusted according to what fits best to perform the work at hand. This only applies to the beginning of the development process though. After that continuity is preferable.

5. Having many iterations means more demonstrations and more feedback, but care has to be taken not to make the iterations too short. There should be enough time to be able to develop something new. Two weeks proved to fit this kind of assignment well.

6.4 Design choices

These choices were concluded from the usability evaluations from each iteration shown in appendix E. The reviewers of the alternatives are put into the categories: Transportlabbet, Other haulage firms, Engineers at Scania, Other stakeholders at Scania.

6.4.1 List vs Writing

There are two different alternatives for presenting choices being made, the first one that lets the user pick alternatives from a scrollable list that is retrieved from the server and the second one that instead lets the user type in the alternative (preferably with auto correction). From a software developer point of view, both have their advantages and disadvantages. The list is easy to interact with but could get cumbersome if there are many names in the list. Typing allows the activity to look clean and simple, but having to use the virtual keyboard has shown to be a tricky process if you’re not an experienced Android user.
These alternatives has been shown to administrative staff at two haulage firms; TransportLabbet and Ahréns as well as engineers at Scania. Ahréns gave the most definitive answer. Since many older drivers are dyslectic and have trouble writing sms, the use of the virtual keyboard should not be mandatory at any stage of the application. TransportLabbet were more ambiguous and could see disadvantages of both. A combination of both alternatives where an optional search box is added to the list is another alternative, though it provide slightly more complexity.

After studying several other users use the virtual keyboard, it is clear that there certainly arise issues with using it. The first issue is knowing where to touch to make the keyboard appear (in this case the edit box), secondly numbers and special characters are hard to locate and lastly it is not obvious how to hide the keyboard when finished typing. It is also important to point out that this (choosing driver or vehicle) probably isn’t something that will happen very often. It is likely that the name is picked once and the vehicle only when the drivers changes vehicle. Going with the most simple alternative is often the best and in this case it is the list approach that is recommended.

6.4.2 Number of buttons of each element in the checklist

The argument is whether or not each element should include a button for each function available (making a remark/measure, take a photo, verify what’s im-
important to check) or a two button approach where one button is for ok’ing the item and the other is a so called plusmenu (as shown in the figure) button that leads to a menu that holds the functions mentioned above. All the reviewers were in favor of the two button approach and indicated that more buttons would provide confusion. Engineers at Scania also pointed out that the plusmenu would also need to keep a simple design with few alternatives, meaning that several menus are preferred over one menu that includes everything.

6.4.3 Use of tabs and the number of tabs

Tabs has the added benefit that it not only provides a way to switch between view, it also displays where in the process the user is and how many steps it is left. Engineers at Scania pointed out that they provide a clear flow to the application. The approach using three tabs has been well received and it has been commented that using a greater number of tabs would be bad because each tab shouldn’t get any smaller (which would be case should another tab be added) and complexity would be raised.
6.4.4 Types of activities

The name activity was chosen over inspection (as in the daily walkaround inspection) since all of the haulage firms asked pointed out that the ability to report a remark/measure without having to perform an entire walkaround was necessary. The term activity is chosen to include different types of inspections and other forms of special tasks, such as the one mentioned. The ability to send a notification can also be done here, or report weight.

6.4.5 Adjustable parameters

Important questions include what should be adjustable in the portal and sent to the application and what should be pre-determined. From the result of visiting different haulage firms it quickly became clear that inspections of vehicles vary widely, partly from firm to firm, but also because there are many types of different vehicles that each require customized attention. For these reasons the functionality to design what an inspection should include has been added to the portal. It is possible to name it, add which items to include and specify what’s important to check and ordinary remarks/ measures. The reason for adding ordinary remarks/ measures is that, as mentioned in the section ”List vs Writing” the use of keyboard should be avoided, and by covering ordinary remarks/ measures most reports should be done by a few simple screen presses.
Furthermore, there are also two other things coming from the portal; drivers and vehicles.

6.4.6 Critical and non-critical remarks

A way of stating whether or not a reported remark is critical or not is asked for by haulage firms. Which remarks that are critical has to be decided by each individual haulage firm. A three choices reporting standard has been proposed, where green is ok, yellow is non-critical remark and red is critical remark (The demonstrator only includes green and red).

(a) Reporting a critical remark
(b) A checklist item with a remark

Figure 6.7: The two options

6.4.7 Existing remarks

By showing the existing remarks of the chosen vehicle, the driver doesn’t run the risk of driving a vehicle with defect he is not aware of. This also ensures that the same defect doesn’t have to be reported more than once. Figure 6.6b illustrates an existing remark. Another alternative that, when used with moderation, has been appreciated are a warning box that pops up once the application is started and/or when vehicle is changed, that makes the user aware that there are existing remarks on the chosen vehicle.
6.4.8 Vehicle setup

It was stated early by Transportlabbet that it should be possible to make one inspection that covers both the truck and the trailer. Other haulage firms has said that it would be good to chose the vehicle setup by the use of pictures (of for example a truck with a connected trailer). Engineers at Scania has asked for a method of adding a number of vehicles to perform the inspection on. No viable solution of merging the checklists of two or more vehicles has been come up with so the demonstrator can only handle one vehicle at a time.

![Diagram](image)

Figure 6.8: One chosen vehicle

6.4.9 Include photo

All reviewers agree that including a photo is a big advantage when using the Android device. The question is where to introduce the photo functionality in it. It has been debated if the possibility to add a photo to each remark is useful or if photos are so rare that adding one photo before sending in the report is enough. No definite conclusion has been made. The demonstrator includes the ability to add a photo at the last step of the process.

6.4.10 Information needed

The start page could include a lot of information, for example not only truck registration number but also it’s type, it’s color and so forth. What has been made clear by all reviewers is that a way to uniquely identify a truck or trailer
has to exist. Because of this the start page includes the function to choose a vehicle from a list in which the administration has added uniquely identifiable vehicles (registration number is one example, but other firm-specific identification numbers could exist). The start page also include the function to choose name. By sending a report, the chosen driver signs that the inspection has been performed according to regulations. The third and last piece of information choose able in the start page is activity. This is necessary when building the checklist in the second tab. Different activities provide different checklists. No reviewer has stated that any other information is necessary.

![Start page with options to choose vehicle, name, and activity](image)

**Figure 6.9: Included information**

### 6.4.11 Use of checklist

The question has been asked whether or not the checklist serves a purpose at all. The alternative would be letting the user report if everything is ok without any specific items and, in the case of a defect, adding it by writing. This option was not chosen since the checklist has proven to provide several advantages. The first being that it is clear what an inspection includes and secondly this also means that the items in the checklist are standard of the haulage firm and as such can easily be used in for example statistics. It is also possible to write common defects and measures for each item, preventing unnecessary use of the keyboard.
6.4.12 Included functionality

Several additional functionalities to what is included in the prototype has been suggested during the course of the research. With everything from being able to send notes between the application and the portal to displaying data about when the next service is. Several engineers at Scania has stressed the danger of making an application too general, being able to do several things at an ok level but nothing really good. The prototype includes the basic functionality that makes up a walkaround inspection.

6.4.13 Auto-complete button

The use of a button that automatically finished the walkaround inspection has been debated back and forth between haulage firms. Transportlabbet were at first positive to the idea, but later changed their minds, fearing that it would be too easy to cheat on an inspection by using it. With further support from Scania engineers the button is not included in the prototype version, but there still exist a certain amount on uncertainty about this decision.

6.4.14 Help function

During early sprints a help function were added to the prototype. By pushing a button, a pop-up window displaying information on how to use the application were shown. It was noted that the application is small and relatively intuitive
and that after the first few users, the help function would only be in the way. Better would be to include the help function as an external source.

6.4.15 Physical restraints

The PACT-analysis identified several restraints that the design of the application need to consider, such as large hands and farsightedness. When it comes to hand size, it has been commented by users with large hands that the buttons are big enough to press without effort. Moreover, the text is easily readable to a person with good eyesight, but the font size need to be increasable in other cases. As a consequence of this, all text has been created using scale-independent pixels, which means that the text will scale by the user’s font size preferences.

6.4.16 Feedback

Not just for this application, but for software design in general, it is important that the user gets feedback whenever an action is taken. This has been implemented into the application. Whenever a setting is changed a textbox displaying the change appears. Whenever a report is added the addition can be seen in the report list and a remark also generates the error icon in the checklist.
Chapter 7

Conclusion

This section provides conclusions to the questions asked in the objective based on the result.

1. How to develop a prototype that seizes the natural process of a daily walk around

   It is useful to break down the process into smaller parts that are technology independent (Identification of parts can be seen in figure 6.1). By then mapping these to the flow of execution of the prototype, the natural process can be kept intact.

2. How to develop with a user centered approach

   User centered development is closely related to agile methodology. Meeting with the user regularly, preferably each sprint as a part of its evaluation, made the user feel part of the development process, thus providing more insightful commentary.

3. How to develop using agile techniques with short iterations

   Avoiding ambiguity, especially concerning sprint backlog items, by making proper documentation is closely related to the success factor of the iteration. Keeping an open mindset and taking advantage of the short iterations by exploring different solutions is another key factor.
Chapter 8

Future Work

The natural next step would be to let several drivers use the application in their daily work as an advancement or replacement for their current daily walkaround inspections. This kind of review went just outside of the scope of this thesis. The application itself supports everything necessary to perform a basic inspection, but currently need to be connected to the same local network that the server is on. In the future a public server could allow the application to run anywhere with wifi and/or 3G reception, and therefore be testable in a real live situation.

The idea of, instead of a checklist, having a graphical representation of the truck came up. By for example tapping the wheel options to remark could present itself. Due to lack of time, this branch wasn’t explored any further, but could be something for a future research. The ever important goal of keeping the application simple would be hard, yet reachable, task.

Functionality that has been discussed but not implemented at this point: The main functionality is the ability to perform an inspection on both the truck and trailer at once, and thus merging their separate checklist. This is something that the haulage firm has asked for. It would also be interesting to look into whether or not the application should be specifically for inspection, or if it should include other functionalities.
A lot of time has been put into making the prototype functional. This has both advantages and drawbacks. The prototype could have been much more like a prototype in nature, only for displaying functionality. With this approach a wider array of design approaches could have been tested. However, the use of a fully functional prototype that has been developed in close cooperation with the user has allowed the user to actually perform an inspection and from that experience, discuss what’s good and what’s not.

When performing a small amount of case studies, it is always important to ask the question whether or not the result achieved are representative for its segment. Three haulage firms and several stakeholders from different branches at Scania were included in the study. Their answers have been fairly unanimous, the idea as such (using an application to perform inspections) is very good, but are probably a couple of years away from being implementable. With the exception of Transportlabbatet, that are very adaptable to change, there are no reason to believe anything other than that the other haulage firms are typical of their business and good representatives of it.

Working in close cooperation with the user has been key to the result that is presented in this thesis. With the office being a closed environment where mainly engineers operate, actually being at the haulage firms has provided new ideas and insights at every step. Making the user part of the development process has a twofold advantage since it makes the product itself better and at the same time gives the user the feeling of being part of the development process of a product that ultimately will be used by them. One of the biggest challenges of this type of work is that answers will often vary from each user and one of the indications that the project is going in the right direction ought to be when the responses become more and more unanimous.

Agile methodology has been a central part, but the inherited artifacts from Scrum both caused clarity and confusion. The idea of having a planning meeting where the details of the next sprint are printed as items on the sprint back-
log, provide a foundation for structured work during the sprint. However, while these items were clear at the time of writing their meaning were often lost over the course of the sprint. It is therefore very important to always write down clear descriptions of each item, even though they might seem obvious.
Bibliography


## Appendix A

### Example of defect report

![Defect Report Form](image)

**Figure A.1:** Example of defect report
Appendix B

Android

B.1 Theoretical background

The android developers guide (Google, 2011) provide an introduction to android programming. The programming language of use is Java with the possibility (recommended) of using xml for user interface programming. With the help of the Android SDK the is compiled to a .apk file that can be installed on the smartphone.

Each application can be built with the help of four different components. Namely:

Activities:
When an activity is created it is granted a window to fill with content. An activity typically fills a screen but it is possible to have multiple activities sharing a screen. One activity has to be declared as main and as consequence being the first activity to start when then application starts. In addition, while many activities commonly form an application, they are independent of each other. This allows another application to use, for example an existing camera activity and thus reducing redundant work.

Services:
A service runs in the background (and therefor has no user interface) and is used for long running tasks that aren’t supposed to block the use of the application. One example of this is playing an mp3 file.

Content Provider:
The content provider provides several ways to store and retrieve persistent data for applications. Examples of storage units are the file system or an SQLite database.

Broadcast receiver:
Responds to system-wide announcements. For example low battery.
Activities, services and broadcast receiver can be started by the use of intents. An intent is a asynchronous message that defines what type of action to perform. It can either be used by directly naming the component to start or by the use of an intent filter. When using intent filters the system itself finds the most suitable action to perform.

**Android Manifest**

The manifest file is used to identify which components that the application includes, which means that they have to be declared in the file. The manifest also handles permissions of the application (such as Internet access), the minimum API level requirement and hardware and software requirements. This is important because of the fact that Android runs on several different kind of devices. Screen size and density may differ and though the application is by default able to run on all sizes and densities, but certain layouts may look different and should be handled accordingly. Furthermore, the platform version could also differ with newer APIs including functionality not supported by earlier versions.

**Application resources**

Beside the source code, an Android application also includes resource files. These include the defined XML layouts and media files included in the package. The Android SDK assigns an ID to each resource so that they can be accessed from the code. By providing alternative resources the application can use the most appropriate according to device configuration.

**B.2 Practical advices**

In order to develop for Android, JDK and Android SDK, that includes all the necessary components such as libraries and the useful emulator, is needed. The SDK also need to be complemented with at least one platform (a platform equals a version of Android). In Accordance with google recommendations Eclipse IDE were used with the ADT plugin.

For the Android phone, an LG p500 were used. It was chosen because it was one of the cheapest phones that used a relatively old version (2.1). Since an Android application is forward compatible, this would mean that the application should work on a large amount of different devices.

Strings:
All strings should be kept in a separate xml file as a resource. The most obvious advantage of this is that an xml file for each supported language can easily be created and the device configuration decides which to read from. However, no localization has been done at this stage. Swedish were chosen as the language of the demonstrator since it should only be presented to Swedes with unknown English language skills. Furthermore, the key advantage for using this approach
for this specific project setup is that when transferring the project between computers with different default character sets, special characters are kept unaltered.

Shared variables and functions:
The documentation of the application class states "Base class for those who need to maintain global application state. You can provide your own implementation by specifying its name in your AndroidManifest.xml's application tag, which will cause that class to be instantiated for you when the process for your application/package is created". Implemented as a singleton this class has two desirable qualities: it exists as long as the application is running and only one instance of it exists. These qualities makes it a viable choice for storing global variables and functions. However, to avoid direct manipulation of variables so called getters and setters were used.

Communication: The system were setup using the client-server model, where all components were connected to the same network.

Class diagram
The flow diagram shows how each activities are connected to the others. Wrapping all activities are the ApplicationController class (which is not an activity, but instead holds all shared variables and classes) which is meant to illustrate that all other classes include an instance of this class. Two special activities exist; the Welcome class which is the main class (point of entry) and the Host-Tab which contains the tabs and a frame that holds one of the three different activities depending on which tab that is selected.

Figure B.1: Flow diagram
Scrum artifacts

This section includes examples of two of the most important Scrum artifacts, the product backlog and the sprint backlog.

C.1 Product backlog

![Figure C.1: Product backlog](image-url)
### C.2 Sprint backlog

<table>
<thead>
<tr>
<th>#</th>
<th>Prize Dibehaving</th>
<th>How to draw</th>
<th>Status</th>
<th>Draw File</th>
<th>Amount</th>
<th>Adept</th>
<th>App 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>220</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>221</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>222</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>223</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>224</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>225</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>226</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>227</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>228</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>229</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>230</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>231</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>232</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>233</td>
<td>15.42gpris på hotellet</td>
<td>You start</td>
<td>Accepted</td>
<td>Red</td>
<td>2014-10-24</td>
<td>70</td>
<td></td>
</tr>
</tbody>
</table>

**Figure C.2: Sprint backlog**
Appendix D

Personas and Scenarios

D.1 Personas

D.1.1 John
1. Age 22
2. Has an Android phone
3. Recently employed
4. Recently got his driver’s license for trucks
5. Eager to follow company standards

D.1.2 Olle
1. Age 35
2. Is neither for nor against new technology
3. Long haul driver
4. Does walk around inspections before and after a long haul

D.1.3 Sven
1. Age 59
2. Rarely uses computers
3. Has been employed for 30 years
4. Only does occasional informal daily walk around inspection (i.e. doesn’t hand in a paper report).
D.1.4 Bengt

1. Age 43
2. Manager
3. Works at the office
4. Tries to convey the importance of doing regular inspections to his employees.

D.2 Scenarios

D.2.1 First daily walkaround

PACT

People: Young, used to technology, recently employed, recently got his driver’s license.

Activity: Perform a daily walk around inspection.

Context: Outside, at the firm of haulage. Normal external conditions (weather etc.). The truck is not connected to the trailer.

Technology: Android phone.

Rationale

This scenario attempts to describe how a walk around inspection is being done when using the device for the first time.

1. John is in his mid twenties and it is his first week at We Drive Goods AB. He has just been assigned a new truck and is about to embark on his first drive.

2. Since he is fresh out of driving school it is natural for him to perform a walk around on his vehicle before starting to drive.

3. By his manager, he has been given a paper containing notes on how a walk around is being made at We Drive Goods AB. It describes where the mobile device used is being found and information he need to know in order to perform an inspection with it.

4. John walks to his truck and gets in. Once there he finds the phone in the glove compartment as instructed. He starts the application and is presented with the start screen.
5. Since he is used to using different types of technology, he doesn’t use the help function at this stage and starts to fill in his name as well as truck and trailer registration numbers. He also chooses the type of inspection he wants to do. Now he is ready to start performing the chosen inspection.

6. After scrolling through the list for a while he has a clear view of what’s to be checked and turns on the lights and indicators before he gets out of the truck.

7. When performing the walk around inspection he follows the order of the checklist and ok’s the items he finds to be ok.

8. He notices that the front left indicator isn’t working and therefore writes a remark on that item in the list.

9. The rest of the walk around continues without finding him finding any other things to remark.

10. The trailer is now connected and the same inspection procedure is being repeated in it. He finds nothing to remark.

11. He then sends the report, finishing the walk around inspection.

12. Since John knows that it is illegal to drive without working indicators and he also knows how to change the light bulb, he fixes the indicator and reports that he’s done so by the use of the phone.

13. He is now ready to start the drive.

D.2.2 Experienced driver using the Android device for the first time

PACT

People: Old, not used to new technology, has his own way of doing the inspection.

Activity: Perform a daily walk around inspection.

Context: Outside, at the firm of haulage. Raining. The truck is connected to the trailer.

Technology: Android phone.

Rationale

This scenario attempts to describe how a walk around is being done by an experienced driver using the device for the first time.

1. Sven always drives the same truck and knows it from within and without.
2. Since it is required by the company to use the Android device for performing a daily walk around, he grabs a copy of the instructions and locates the device.

3. His process of doing a daily walk around inspection is to quickly walk around the truck and trailer and to make mental notes of anything he finds to remark on.

4. He then gets in the truck and starts using the Android device in order to report his inspection. With the help of the instructions he starts the application. He notices a button that says help and pushes it. The help function is now displayed and guides him through the process of reporting an inspection.

5. When the report is sent he is ready to start driving.

D.2.3 Send a remark from the road side

PACT
People: Middle aged, long haul driver
Activity: Send remark
Context: Outside, away from the firm of haulage. Driving on the road.
Technology: Android phone.

Rationale
This scenario attempts to describe how a remark is being sent when something breaks down while on a long haul.

1. Olle is currently undertaking a drive bound for Hamburg
2. Close to Malmö a stone chip suddenly cracks his front window. He quickly pulls over.
3. After inspecting the damage he fetches his Android device and sends a report. He makes the decision that the damage doesn’t affect his driving performance in a major way and continues to drive.
4. The manager notices the remark that Olle sent, but since he didn’t report it as urgent he doesn’t look for a service workshop along the way to Hamburg. Instead he reports it to the local workshop that will deal with the damage when the truck gets back to the firm.
D.2.4  Look at the status of inspections and vehicles

PACT

People: Middle aged, Manager.

Activity: Monitor vehicle conditions, monitor that inspections are being made.

Context: At the office.

Technology: Android phone.

Rationale

This scenario attempts to describe how the manager can get information about the inspections that has been made on his vehicles.

1. Bengt is at his desk at his office. At the end of each day he always checks the web portal for inspections to see that his drivers perform inspections as they should as well as make sure that there are no new reports of defects on his vehicles.

2. He starts the web browser and logs in to the portal. Firstly, he looks at the table containing reported inspections and see that all his drivers has reported as they should.

3. Secondly, he looks at the table containing remarks. He notices there’s one new item reported. One of his trucks has a damaged rear window.

4. He chooses the subject vehicle and is presented with a table containing all reported remarks on it. It turns out that this specific vehicle also has other reported remarks.

5. Bengt decides that a service appointment is need and sends the remarks to a service workshop. The workshop will send back a confirmation once they have found a suitable time to perform the service.
Appendix E

Iterations

E.0.5  Sprint init

Summary
Setup phase.

Analysis
The first three weeks were named init, since it was before the necessary components that constitutes a sprint could be done. Instead this sprint focused on initial tasks such as setting up the scope for the project, installing the Android development environment, getting to know Scania standards and starting to meet with Transportlabbet. With the exception of this first sprint, as well as the final sprint, all sprints had to include the same components. These components were decided during this sprint and consisted of the following:
Each iteration must include an identification of necessary changes (represented by the user stories chosen from the product backlog), a prototype development phase and a written prototype usability evaluation.

Evaluation
No usability evaluation were performed, instead general design and usability has been discussed.

Lesson learned
Several important theoretical foundations were gathered at this stage, including agile methodology (Scrum specifically) and Android programming guidelines.

E.0.6  Sprint 1

Summary
The first prototype were developed. This also meant that it was developed without any feedback from any user and only theoretical knowledge on how a workaround is being made.
Analysis
Some of the most important basic building blocks that the prototype has come to rely on during the course of its development were decided, in collaboration with Transportlabbet, at this stage.

The tab approach was formed at this stage, as well as the use of a checklist which was a direct translation of the current paper based version of many haulage firms. It also became clear that two way communication between the Android device and the web portal would be necessary in order to make the application useful to several different types of vehicles. The demonstrator were mainly prototype based at this time, meaning that there were little to no functionality implemented but rather a graphical interface to represent the general thought. One item worth mentioning is the inclusion of a auto-complete button below the checklist with the text "No other defects were found". This was a request from Transportlabbet at the first stage and was meant to simplify the process of performing a walkaround inspection. They later came to the conclusion that it would be too easy to perform a walkaround with this button and as a consequence it was removed.

The web portal were introduced with the use of existing style sheets that were already in use on the fleet management portal. The reason for this was that it was stressed by Scania that a similar look-and-feel of the two were key when making people understand that the Android application was a part of a bigger system.

Figure E.1: Two views
Evaluation
The prototype was demonstrated using a series of pictures that each represented a possible screen of the prototype. By simulating a walk around the pictures were presented according to the actions of the test person. For example, if the button "Start walkaround" in the Start activity were pushed, the picture containing the Walk around activity were shown. This was done in order to get an indication if the interface did satisfy a few defined usage criteria. These criteria were:

1. It should be easy to learn
2. It should be possible to make a complete walk around
3. The appearance should be subjectively appealing

The user were able to figure out how the application worked without interruption, but did complain that due to the fact that no option to enter fix existed, a complete walk around couldn’t be done. Moreover, the user were pleased with the general look and feel of the prototype.

Lesson learned
The users must know how to get the application, start it and then perform the check. This can be done by the use of an external (to the application) written instructions as well as an internal pop-up dialogs that display information. The prototype must be demonstrated using an Android phone in order to make a full usability evaluation (as opposed to using a powerpoint presentation).

E.0.7 Sprint 2
Summary
Where sprint 1 focused on adding functionality and design to the prototype, this sprint focused on the important communication between the device and the portal.

Analysis
The Android now receives information about drivers, vehicles, walkarounds and walkaround items from a database. When received, they are in the json format and naturally parsed accordingly. A string in json format is now created when the button "finish" is pushed. This string contains information about which driver that did the inspection, which vehicle it was done on and which type of inspection that was done. In addition, it includes any remarks that the user entered when performing the inspection. It is now possible to specify if you want to do the inspection on the truck, the trailer or both.
Figure E.2: The possibility to choose truck and trailer

While not implemented at this stage, the inspection list should look different depending on the selection made. The checklist must be completed in order to be able to proceed. If not, a dialog box will present information that the list is not completed.

**Evaluation**
The prototype was demonstrated using an LG-P500 smartphone. At this update, mostly functionality and communication to back end has been implemented, which means that the UI was about the same as was presented in Sprint 1. The prototype was evaluated for interactivity and the following questions were asked:

1. Are all the touch objects big enough?
2. Is the font size readable?
3. Does anything block the interactivity?

The user had fairly big hands and found the touch objects to be easy to interact with. The text was readable, but he did mention that people with reduced eyesight probably would experience problems. At this stage, the prototype did contain bugs that interrupted a smooth user experience.

**Lesson learned**
Scrum definition of done is something that might look self-explanatory at first
sight, but is, as it turns out, an item that need careful consideration. For this sprint the desired functionality were implemented, but they included several bugs. By the use of different interpretations this can be decided as done or not done.

E.0.8 Sprint 3

Summary
The last sprint provided functionality that covered many parts of an ordinary daily walkaround inspection. However, they include several bugs as well as a UI that isn’t very nice.

Analysis
No functionality has been added to the Android application at this stage. However, a basic step by step startup has been added to support users with little to no experience with Android technology.
In general, focus has been on the UI, making it as intuitive as possible as well as pleasing to the eye.

The web interface has been provided with administrative pages for customization of drivers, vehicles and inspections of the firm of haulage.

**Evaluation**
The prototype was demonstrated both using a powerpoint presentation and the LG-P500 smartphone. The general look and feel of the application had been majorly updated during this sprint. During the presentation, 3 questions were asked:

1. Does the application react (on touch event) as expected?
2. Is the interface intuitive?
3. Does the application appeal to all users? (i.e. various degree of Android experience)

It became clear that the application handiness lacked at several points, including the step by step startup, the yes/no button for choosing vehicle and the ”plus” icon in the checklist. Further development while keeping a close contact with the user is key at this stage. Furthermore, a significant challenge is how to inform the user how to minimize the virtual keyboard. Some form of external documentation could be an option. The terminology used in the application and the web interface is not unanimous which could provide for confusion.

Lesson learned
The sprint has highlighted the complexities of user interface design. Further work will be put into making the application as intuitive as possible. The importance of packaging and a well thought out introduction of the application has been stressed with the mentioning that other technological innovations has taken over a year to accept once introduced.

E.0.9 Sprint 4

Summary
Instead of adding more functionality and tweaking the existing UI, it was also important to decide if the current setup was the desired one or if something else was better. Previously problems with making people explain what they wanted had been hard.

Analysis
The use of different design alternatives were a major focus during this sprint. The figures below illustrate two alternatives presented. The choice to be made concerns whether or not each list item should have a specific button for making remarks (a) or a general button for several things, including making remarks (b).
This approach has proven to be the most successful thus far when it comes to making the interviewees think creatively and critically. Future sprints will see more of this method.

In addition, the ability to take a photo and attach it to the activity has now been implemented.

**Evaluation**
The user were presented with a scenario that described a simplified version of a daily walk around inspection and then tried to execute it with the help of the Android application. The scenario was: “Perform a daily walkaround inspection on your truck, license number ABC 123. While doing the inspection you find that one of the front indicators are nonfunctional. This is a critical defect and must be fixed immediately” Of the two test users, both were able to finish the inspection with only minor difficulties. Most notable was that the functionality of the check-box is not obvious for an inexperienced user, though this is something they learned quickly.

**Lesson learned**
One step in the direction of making people think critically and creatively when demonstrating something. This is something that is very difficult, but the use of different alternatives was helpful.
E.0.10  Sprint 5

Summary
Version 1.0 of the application is very near completion. For the first time, the application should now (in theory) be able to cover all parts of a daily walkaround. However, each design choice isn’t fully motivated yet and this is something that will be the focus for the remaining time of the thesis project.

Analysis It is now possible to comment on a remark, a measure or making an extra remark not related to any item in the checklist. This remark is shown in the portal as a comment to the inspection and not a separate remark.

![Figure E.5: Two ways to comment](image)

(a) Remark  (b) Extra remark

The web portal now includes functionality to customize each item of an inspection. There are three things that can be added or removed: what to check, common remarks and common measures.

Evaluation
A driver was asked to perform a daily walkaround as it is performed normally. The application now supports making a full walkaround on one vehicle at a time (preferable the application should handle two for truck and trailer etc). According to the driver it was very easy to get started up until the point where the checklist was presented. It took a while for him to realize that the checkbox was for ok-ing the item and the plus button for remarks. In addition, some
help were needed in order for him to make an actual remark. Once he got these first things right he expressed that he thought that the application were easy to interact with and that he would use it if he were given a phone.

**Lesson learned**

It is always important to take notes. Both the process of actually writing it down and the fact the there exists something to read helps keeping details in memory. This became clear at the mid sprint tuning, where it was obvious that the meaning of some sprint items had been forgotten.

**E.0.11 Sprint 6**

**Summary**

Version 1.0 is presented during this sprint, which means that it is now possible to perform an entire daily walk around with the help of the Android device. Though already some flaws has been detected, both with the new design and the lack of some needed functionality. One thing that need to addressed is to remove the need for a server connection in order to be able to use the application and that the chosen name and vehicle is saved.

**Analysis**

One of the biggest update to date has been implemented, which is the functionality of adding and removing remarks and measures. The application also displays current remarks/measures that has been reported on the chosen vehicle.

![Figure E.6: Added reports](image-url)

Figure E.6: Added reports
Evaluation
The application has been shown to several drivers this sprint (around 10) and they've all gotten to speak their thought on it. The new design includes pop ups for remarks/measures and this is something that hasn’t been received very well. As always it is hard to make people critical and it was the thought of one driver that to truly test whether or not this application could work, it needs to be used by drivers in their daily work for some time. It was also mentioned again that there should always be ways to complete a task without using the virtual keyboard in order to support any type of user (difficulties with technology, dyslexic etc).

Lesson learned
It gets harder and harder to predict a complete sprint backlog as the project grows and the project time nears its end. Things get more complex and some items become urgent. It is therefore important to do more analysis before predicting anything.

E.0.12  Sprint 7

Summary
There is a "finished" version of the application that will be shown to several stakeholders. If they are satisfied, then the application should be done. However, much of the work on the report remains and work done during the next sprint will be important. If some issues arise the project could be delayed for a couple of weeks.

Analysis
The application previously required an up and running web server and a connection, but now there is an "offline" alternative so that it is possible to demonstrate the application without the entire system running.
Other than that, only different tweaks has been added to the application.

**Evaluation**

The application was shown to several stakeholders at Scania as part of a presentation of it. While they didn’t get to try the application in a regular way, they were presented with the full functionality of it and were able to comment on the process. No question marks or thoughts regarding the design were spoken which has to be seen as a positive thing. They did appreciate the similarities in look and feel of the portal and the existing FMP. One raised the question whether or not the checklist was useful at all which is an interesting question to consider.

**Lesson learned**

It is clear that when it comes to adding functionality to an application, it is much more complex than imagined. It sounds more obvious than it is, but it is important to make sure that the important parts are covered and only then move on to other things. A MoSCoW analysis was made at the start in order to rank the importance of each functionality and it should have been used much more strict. Making the core functionality satisfactory at the first stage and only then move on to the next stage of functionality.

Several things came out of the feedback meeting after the presentation at Scania. It is not only the information itself that is important, but also how it is expressed. It is good to start with the positive aspects. Important to prepare several things for a preparation and not only the presentation itself. For ex-
ample, prepare for different questions that are likely to come up: what is good with the solution, what is not so good, what does other people think etc.