Development of handheld mobile applications for the public sector in Android and iOS using agile Kanban process tool

Fredrik Bergström
Gustav Engvall

2011-06-10
Development of handheld mobile applications for the public sector in Android and iOS using agile Kanban process tool

Examensarbete utfört i medieteknik
vid Tekniska högskolan vid
Linköpings universitet

Fredrik Bergström
Gustav Engvall

Examinator Camilla Forsell

Norrköping 2011-06-10
Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under försiktiga omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© Fredrik Bergström, Gustav Engvall
Abstract

The innovation progress for municipalities is currently going slowly. Today most of the communication, decision-making, delegation and service of process are still handled via letter mail or phone calls. Companies and municipalities are starting to comprehend this and see opportunities. With the technology that exists today it is possible to speed up the processes when citizens and municipalities communicate with each other. By migrating from the old way of applying and filing different matters by letter mail to so-called digital e-services, the citizen can apply, complement and get info on how their matter is going directly on the web. About AB works with optimizing information handling for municipalities and has recently made it possible for municipalities to eliminate their tedious way of handling matters by traceable web communication. They have a vision to take this to a new level of easement to the municipalities and their citizens by offering these services in mobile phones as so-called apps. One of these contemplated services is to file error reports. The service will offer the citizen to send in a report with attached meta data such as photo and the location of the regarding matter. This will be done by the mobile phones built in hardware in form of camera and GPS. This master’s thesis will describe the work to develop a prototype for filing an error reports in mobile phones that support Android and iOS operating systems.

The goal for this thesis work is to develop this prototype by means of a process tool called Kanban and investigate if the tool fits the situation properly. The situation is that two persons will work in parallel with slightly different projects. Also the quality and usability of the resulting prototype applications is asserted with qualitative evaluations that results in higher awareness of the defects and suggestions on how to reform and improve the application in a usability point of view.

The actual implementation work was planned according to a slightly loose Kanban paradigm. Kanban is all about improvements along the way and the implementation phase is divided into smaller phases with sub goals and recurring process evaluations. Suggestions of improvement regarding the work processes is a result of this. A Kanban board that helps visualize the workflow is used to help team members and project owners to get a good overview of current situations. Several adjustments were made throughout the project and the so-called lead times, the time it takes to complete a task, was shortened in average from the beginning of the project compared to the end of the project.

Each phase in the implementation part of the thesis work resulted in different functionalities in the application. A phase was two weeks long and ended with a demo were the new functionality was demonstrated. Prior to each demo session everyone at the company where the thesis work was performed was invited. The participants in these sessions were encouraged to give feedback on what they saw to help increase the quality of the applications. The prototype applications resulted in a more easy accessible service to citizens. By cutting out the process steps that previously needed to been done on a computer the
application contributes to the possibility of an increased democratization grade for the citizen.

The investigation regarding if the Kanban process tool was a suitable aid in the implementation part of the project resulted in that it was suitable for the specific situation. By measuring both lead times and other activities that did not concern the implementation work and plot them in a diagram explained why some weeks did not have as high productivity as others.

To assert the awareness and quality of the developed applications in terms of usability a so-called heuristic evaluation was performed. This qualitative evaluation resulted in a set of usability problems. These problems were all graded into different priority to be solved and some was given suggestions on how to be solved. The main purpose with this evaluation was to find the problems and document them to possible future programmers if the prototype will go into production. The result of this evaluation has increased the usability of the application and consequently the quality.
Acknowledgement

This report is the result of a master’s thesis in Media Technology and Engineering at the Department of Science and Technology, Linköping University. The thesis project has been performed at the company Abou AB in Stockholm and we would like to thank people making this thesis possible. Many thanks to the company Abou and all employees, especially our supervisors Malin Sjölin and Stefan Hellström for all support during this project. Many thanks to the employees at Upplands Väsby municipality for all help with the applications. We would like to express our appreciation to our supervisor and examiner Camilla Forsell at Linköping University.

Fredrik Bergström & Gustav Engvall

Stockholm 06/14/11
# Contents

1 Introduction 9
   1.1 Objective ................................... 10
   1.2 Method ..................................... 10
   1.3 Delimitations ............................... 10

2 Background 11
   2.1 Related work .................................. 14
   2.2 Lean .......................................... 14
   2.3 Scrum .......................................... 15
   2.4 Kanban ........................................ 17
   2.5 Usability ...................................... 19
      2.5.1 Heuristic evaluation ....................... 20

3 Method description 21
   3.1 Question formulation .......................... 21
   3.2 Problem definition ............................ 22

4 Result 23
   4.1 The process tool .............................. 23
      4.1.1 Kanban in a two person project .......... 23
      4.1.2 Adapting the project to Kanban .......... 25
      4.1.3 Initial estimates and changes .......... 28
   4.2 The prototype ................................. 31
      4.2.1 Phase 1 - GUI and navigation .......... 31
      4.2.2 Phase 2 - Photo and GIS ................. 32
      4.2.3 Phase 3 - JSON .......................... 33
      4.2.4 Phase 4 - Usability evaluation .......... 33
      4.2.5 How to use the prototype application .. 33
   4.3 Graphical user interface in a usability approach 37
      4.3.1 Heuristic evaluations ..................... 37
      4.3.2 Result of heuristic evaluation of Android application 42
      4.3.3 Result from heuristic evaluation of iOS application 43

5 Discussion 45
   5.1 The process tool ............................. 45
   5.2 Usability discussion for Android application 46
   5.3 Usability discussion for iOS application ........ 47
   5.4 Usability discussion and comparison between the platforms 47
List of Figures

1. This process chart depicts how it often looked before the e-service portal existed. ........................................ 12
2. This process chart depicts how the e-service portal significantly cut down on the process steps. ......................... 13
3. This is an example of how a Scrum board can look like. Figure is taken from [25] ................................................. 16
4. This is an example of how a Burndown chart can look like. Figure is taken from [25] ................................................. 17
5. This is an example of how a Kanban board can look like. Figure is taken from [21] ................................................. 18
6. This is an example of a Kanban board from the project. This board consists of prioritized and color-coded activities in two parallel projects (the rows Android and iOS). The “Duties” row is for activities that may or may not concern the actual implementation work but still is urgent. ........................................ 24
This figure depicts the re-occurring happenings in an implementation phase in the project. Each phase is two weeks long and ends with a demo session where the functionality that has been implemented is demonstrated. Reconciliation sessions are to check if the project goes according to plan and on these sessions team members and tutors participate. In the retrospective meeting that occur in the middle of the phase decisions on adjustment regarding the work are made.

This figure depicts the lead-times in green and urgent duties in red for the Android project. The x-axis represents the week in the project starting with week one and so on and the y-axis represents the number of finished activities or duties that week. A pattern can be seen that when productivity is high in the project the duties are low and vice versa, with few exceptions.

This figure depicts the lead-times in green and urgent duties in red for the iOS project. The x-axis represents the week in the project starting with week one and so on and the y-axis represents the number of finished activities or duties that week. A pattern can be seen that when productivity is high in the project the duties are low and vice versa, with few exceptions.

This figure depicts the first Kanban board v. 1.0 that was created in the project. The WIP limits are not based on anything in particular other than what the team members think is adequate. These parameters are to be adjusted in the future.

This figure depicts the second Kanban board v. 1.1 that was created in the project. Adjustments were made regarding the “Implementation” and “Test” columns to fit the xCode IDE for the iOS project better. Also the WIP limits were decreased.

This figure depicts the final Kanban board v. 1.2 that was created in the project. Adjustments were made regarding the “Implementation” and “Test” columns. This was done because not all activities needed to be tested.

This figure depicts the initial time schedule that was created before the project started.

This figure depicts the final time schedule. The time schedule has been updated continuously throughout the project.
This figure describes how the applications communicate with the web services, both in the test environment that has been utilized within the project and how it will communicate in the future if the applications goes into production. (a) This figure described the test environment. The arrows represent the data flow. The application posts a report object to the web service. The web service stores the report in a database. The application in turn fetches data from the database with a so-called “get” call. (b) This figure describes the environment that the applications will communicate with in a future production. The arrows represent the data flow. The application posts a report object to the web service. The web service stores the report in a database and also creates a LEX object, which basically is a mark-up structured file. LEXTalk is a program that the municipality owns and the administrators work with. This program creates real PDF reports and stores them in a database. The application fetches data from the database with a so-called “get” call.

Start page. This is how the start page looks like in the both applications. (a) iOS. (1) List of recently filed reports from other citizens. (2) Button that takes the user back to the start page if not there already. (3) Button that takes the user to the actual filing of the report. (4) Button that takes the user to a pages with Frequently Asked Questions. (b) Android. (1) Button that takes the user to the actual filing of the report. (2) List of recently filed reports from other citizens.

Error report form page. This is where the user can specify everything regarding the report. (a) iOS. (1) Category text field. (2) Description text field. (3) Address text field. (4) Address number text field. (5) Fetch address via GPS button. (6) Attach photo button. (7-8) Feedback selection buttons. (9) Next button. Takes the user to the next page. (b) Android. (1) Category text field. (2) Description text field. (3) Address text field. (4) Address number text field. (5) Fetch address via GPS button. (6) Attach photo button. (7-8) Feedback selection buttons. (9) Next button. Takes the user to the next page. This button becomes visible if the user scrolls down the page.

Personal record page. This is where the user can specify the personal information that the administrators need to be able to return feedback on the matter. (a) iOS. (1) First name text field. (2) Last name text field. (3) Email text field. (4) Phone number text field. (5-6) Feedback way selection. (7) Next button. Takes the user to the next page. (b) Android. (1-2) Feedback way selection. (3) First name text field. (4) Last name text field. (5) Email text field. (6) Phone number text field. (7) Next button. Takes the user to the next page.
Summary page. This is where all user specified information is summarized to be validated. (a) iOS. (1) Send button. If this button is pressed the report is sent in to the administrators in the municipality. (b) Android. (1) Send button. This button is not visible until the user scrolls down further on the page. If this button is pressed the report is sent in to the administrators in the municipality.


Different severity ratings of usability problems for the Android prototype. The x-axis represents the different severity ratings and the y-axis represents how many usability problems that were found for each severity rating in the evaluation.

Different Severity ratings of usability problems for the iOS prototype. The x-axis represents the different severity ratings and the y-axis represents how many usability problems that were found for each severity rating in the evaluation.
1 Introduction

Today a common municipality supplies and administrates over 900 different services. Some completely analogous by paper forms and other in a slightly more efficient way where certain matters are available to apply for in a digital form. A matter can be everything from applying for a building permit to apply for a parking spot in your neighbourhood. A common name that will be used in this report for the phenomena to have the possibility to apply for things digitally is e-service. To apply and handle matters can be a tiring process where it is a lot of time wasted due to substandard directives on how to handle these matters and also to many steps in the process. There are two point of views in this case to handle matters, partly the administrative point of view which are the people actually working with handling the matters, and partly the citizens point of view which are the citizen who apply something to the municipality and they only notice that the process is time consuming. Tendencies are starting to show regarding a migration towards a more digital environment for the handling of matter at the municipalities. There are so many different matters and not all of them are suitable to handle digitally. There are matters that are applied for more seldom than others and therefore are not suitable to migrate from an economic point of view since the cost to develop such an e-service can not be compared to the value of the municipality having it in their supply of e-services. On the other hand there are services that are applied for frequently and in high quantities and these are ideal for migrating into the digital world. There are also differences between matters who are simpler to perform and the ones that need complementary data. Building permits are good examples of such matters. If the citizen who plans to build a house and applies for a building permit forgets to attach correct blue prints then the administrator needs to send a complementary notice usually by letter post. If the citizen miss to fill something important out in the next try then the process repeats it self again. This easily turns into a long and prolix communication stream between the citizen and the administrator.

One possible way of cutting the waiting time out is to migrate to communicating via email and applying for things online in a digital way. This approach is starting to be more and more common in the municipalities at the same time they get more and more aware of the defects in their handling of matters. The process of filing and applying for different e-services still has some disadvantages. The citizen cannot apply or file stuff on the fly, they still have to seek up a computer or wait until they get home to perform this task.

About AB is a company that works with migrating processes to a digital world for municipalities and they have a vision that it will be possible for the citizen to apply for things directly using their smartphones. Smartphones indeed open new door regarding what can be done and makes it possible in the terms of applying or filing matters to the municipality.
1.1 Objective

The objective of this Master thesis is to develop two prototypes of mobile phone applications. The applications will support Android and iOS mobile operating systems and they will perform the task of filing error reports to the municipality via a web service. Is it possible to increase the usability of the application by performing qualitative evaluations and consequently increase the quality to make it as an real product in the future. To ensure the quality of the prototypes certain process tools and evaluation technique are used.

1.2 Method

This is a pilot project and a completely new product will be developed. Therefore room for miscalculations foremost in the planning will not come as a surprise. The implementation phase of the thesis work is performed in a Scrum and mostly Kanban influenced kind of way. These paradigms are further described in section 2. The corner stones of the workflow are from Scrum and tweaked towards a Kanban way of working. For example all the strict rules has been removed from the Scrum approach. The visualization of the workflow has been considered from the Kanban approach and also the limitation associated with parallel activities. The work and the prototype in particular will have focus on usability and therefore a heuristic evaluation of the user interface of the prototype will be conducted after the implementation phase. The purpose of this evaluation is to find and prioritize usability problems. The prioritization will be done with a five-grade severity rating and if the severity rating is high enough then the problem need to be solved or proposed to be solved in a future work.

1.3 Delimitations

Two prototype applications to smartphones will be developed. The applications is available for the following devices: iPhone, iPad, Android phones and tablets with support for Android. The two different applications will perform the same service and have the same graphical user interface with exceptions for the look and feel of the different platforms such as default buttons or characteristic looking lists.

The publication of the applications isn’t included in the thesis work because it can be a unpredictable process and it can easily pace over the time span that the thesis work is included in. Its foremost depending on applying the application to the review processes that Apple handles. Also no performance requirements will be considered concerning loading times.
2 Background

Upplands Väsby municipality has a portal called “digitala rummet” (digital room) on their web page. This is a place where citizens, companies, and unions can apply and seek information about different matters. The user can check status, look up the time it takes to manage a specific matter or basically get a more thorough insight in the handling, some e-services also demand electronic identification. This is to verify that the person is who s/he claims s/he is. An example of an e-service is to file error reports in the nearby surrounding that can concern vandalism, scribbling, out of service street lights and so on.

The service of filing error report in particular is today done like this: A user finds a malfunction or error in his or hers municipality. If the user decides to report this s/he has to visit and fill in the form on the web page [16]. This obviously demands that the user has access to a computer. If it is the first time a user must perform an error report on the webpage, then it can be difficult to find the specific e-service. When the user find the service s/he has to fill in no more than 15 entries to finally send in the matter to the persons responsible to process this report. The process is still more efficient in a time consuming point of view compared to send a letter but can still be made more accessible and faster to utilize by the user. Before the e-service portal, it often looked like in figure 1.
Figure 1: This process chart depicts how it often looked before the e-service portal existed.

This process should be a simple thing that easily could be done in these small gaps of time when a person otherwise doesn’t do anything meaningful. It should also be possible to file the report instantly when the person finds the malfunction rather than waiting until s/he find a computer. The e-services that exist today cut down on the process steps significantly (figure2) and also contribute in bringing the municipality closer to the citizen. But the accessibility for this e-service is still somewhat restricted. As described earlier the person in question still has to find a computer and use this to visit the web service. The first step towards bringing the municipality closer to the citizen has been made but there are more to come.
Next step in the progress is to bring the municipality even closer and more accessible to the citizen. Barack Obama mentioned the open government philosophy [18] where business of government and state administration should be opened at all levels to effective public scrutiny and oversight. Open government is composed of four cornerstones: web 2.0, net generation, social networking and wikinomics [18]. Web 2.0 can be seen as an active platform and not just a website. Social networking is when things are built of many collaborators such as Wikipedia. Wikinomics is how mass collaboration changes everything with networked business models such as iTunes. The last part, which defines the open government manifest, is the net generation. Nowadays, those who use open government, web 2.0 or social network have lived a life with IT. Today, those people are between 13-30 years old. The age differences may grow in the future since more and more citizens are learning how the information technology works. For this purpose, if the public could affect the environment in the municipality with a smartphone, then matters and errors may be corrected in an early stage.

Opengov.se [10] is a Swedish website where public data are available for anyone. Examples of existing data are the financial service, government offices and medical products agencies. By using this data, private persons and companies can build applications for different purposes that will be helpful to many.
The accessibility issue can easily be solved quite obvious with today’s technology that the smartphones provide and more precise with mobile phone applications that the citizen can download free from App Store or Android Market. The company that facilitates support and resources to make this thesis possible is a company named Abou AB. Abou is specialized in e-government and works as a strategic partner to many of Sweden’s biggest municipalities and authorities. Abou is a company that develop innovative solutions based on lean methodology, which simplifies the process of continuous improvement and development. The company helps other companies and public sector with faster and simpler internal processes backed by clear and visual solutions that eliminate unnecessary administration. Abou is a team of business consultants, developers and designers.

2.1 Related work

There are similar applications and services that manage similar tasks as the error report e-service for Uppland Väsbys municipality. Citizens Connect [1] is a mobile application for Android and iPhone that manage errors such as potholes, graffiti, streetlights, etc. One downside with this application is that it isn’t compatible with iPad or tablets. This application primary user is the citizen of Boston, MA, USA. By a glance at the feature called “Recent reports” in this application that shows reports from other users from all of Boston an average of 4 reports are filed every hour during the day. This means that there is a need for such an application and very much so according to these numbers. Another e-service named FixMyStreet.com [4] manage potholes in the streets and other problems in the UK. The public can specify a problem on a map, enter details regarding the problem and send it. This web site also provides a small amount of statistics in form of how many reports that have been solved past month. These statistics also tells that this service is well used by people throughout the UK with an average of more than 1000 reports a week. The closest thing to a interactive mobile application for municipalities is "Svenska Kommuner" and can be downloaded from this web page [14]. The application is only an informative application. It works like this: Municipalities in Sweden can join this application and thereby gets to upload information about the municipality in question. Mostly it consists of a news flow and information about the weather and distances to points of interest. The lack of interactivity in applications for mobile devices that helps the user to perform meaningful tasks open ways to refine and develop a new kind of application.

2.2 Lean

Lean production is a philosophy about managing resources. The purpose of lean thinking is to eliminate unnecessary resources called waste in the production process and make time from raw material to finished product as short as possible. The basic aim is to make more value for the end customer. Lean was invented
by Toyota for their massive factories. Toyota production system (TPS) [28] is a resource effective process that continually reduces waste. However, lean is not all about mass production in factories; lean has in recent years been very useful in healthcare [33] by i.e. reducing waiting times for patients. Lean has the potential to reduce human misery and increase human happiness by doing more with less, while providing meaningful work [20]. Lean IT is another lean approach and it is an extension of lean manufacturing. One major similarity between IT and manufacturing is that the manufacturing function manufactures goods of value to customers and the IT function “manufactures” business of value to customers. Typical wastes and it’s business outcomes in IT organizations are: poor customer service, miscommunication, lost revenues and increased capital expenses [32].

2.3 Scrum

Scrum is a framework for agile system development [19]. The word originally comes from the games of rugby and is referring to the moment or sequence when the ball comes into play. In system development, Scrum focuses on the people in the project, not the technology [19]. Scrum manages changeable requirements in a more efficient way. This increases the motivation and communication between clients and project members. Companies such as IBM, Microsoft, Xerox and Adobe have implemented Scrum in their way to work. There are three kinds of people involved within the Scrum process: product owners who prioritize requirements in the project, scrum team members who develop the requirements and a Scrum master who support and coaches the team.

Before a project starts, the product owners prioritize requirements and sort them by highest priority in a product backlog. While the project is running, new requirements will be added to product backlog together with new priorities. Every sprint begins with a sprint planning meeting where the Scrum master, the Scrum team and the product owner are planning the upcoming sprint. These meetings are depending on three variables: scope, estimation and importance [25]. The product owners present the scope and the importance of different stories while the team estimates the time of each story. A story here is a task or a function. Each requirement plus priority and time estimation will be gathered in a sprint backlog sorted on the priority from the product owner. A sprint is usually a time between two and four weeks and is ended with a sprint retrospective, which is a meeting where experiences and lessons are discussed. The purpose of sprint retrospective is to raise the knowledge within the team and let the product owner test the software made so far, this is usually called demo review.
Figure 3: This is an example of how a Scrum board can look like. Figure is taken from [25]

Usually during a sprint, every day begins with a “daily scrum” meeting where each member describe how their work is going and which stories that will be handled next. If a team member has any issues or problems, then those can be discussed during the meeting. To visualize the change of state of the stories throughout the sprints, a task board or a scrum board is often used. An example is shown in figure 3. Every sprint has also a burndown chart, which basically describes how the work is going. The days can be seen in the x-axis and the story points can be seen in the y-axis. The aim with the burndown chart is to “burn” every story before the last day of the sprint. As it can be seen in Figure 4 on page 17, if the chart follows the line it means that there are just enough stories added into the sprint. If not, there are probably too much or too little stories added into the sprint.
2.4 Kanban

Kanban is like Scrum, a lean approach to agile software development and a part of the lean thinking. The word Kanban is Japanese and means “visual card” and was invented by Toyota, which used this process tool for the visual and physical signalling system that ties together the whole Lean Production system [2]. In 2004, David Andersson [2] dived further into Lean and invented a more direct implementation of Lean thinking into software development. However, Kanban in software development can be divided into three core parts:

1. Visualize the workflow - split the whole work up into pieces and write them on cards, put every card on a kanban board, see figure 5. With the cards and the board it is easy to visualize the workflow.

2. Limit work in process (WIP) - compared with the scrum board, there are often three columns on the board: the “To Do” column, the “On going” column and the “Done” column. The Kanban board is almost the same as the Scrum board except from a limit or number in the column and that is all. For example, if the prescribed number is two then it can only be a maximum of two stories in progress simultaneously.

3. Measure the lead time - or measure the average lead time, sometimes also-called “life cycle”[21]. Lead time is simply the time it takes to complete
one story or activity. This will hopefully optimize the process to make lead time as short and predictable as possible.

Figure 5: This is an example of how a Kanban board can look like. Figure is taken from [21]

When it comes to the second core part, the WIP in Kanban is limited per workflow state in contrast to WIP in the Scrum, which is limited per unit of time. To clarify this further, both Kanban and Scrum limit WIP, but in different ways. Scrum teams usually measure velocity - how many activities get done per iteration. Once the team knows their velocity, which becomes their WIP limit. A team that has an average velocity of 10 will usually pull in no more than 10 activities to a sprint [21]. And this is the difference in WIP between the two paradigms.

Usually, too low limits will result in bad productivity and people without any work. On the other hand, too high limits result in bad lead-times and tasks or activities without anyone implementing them. If there are many stories allowed by the limit in the ongoing process and if an initiated story cannot be finished, then it is possible to start with a new story. However, there are a lot of methods on how to limit WIP. Christopher M. Shinkle [30] talks about how to apply the Dreyfus model of skill acquisition into the Kanban system. He suggests three ways to set queue sizes:

1. Start every limit at one, add tokens on at a time until the person is busy.
2. Start every limit at an arbitrary large value, subtract tokens one at a time until flow is observed.
3. Create a value stream map and measure the time-on-task distribution of each activity.

Another advantage by using Kanban as a process tool is that product owners can change requirements in a product backlog at any time during the sprint.
time. In Scrum, requirements are added in the sprint planning meeting and cannot be changed during the sprint process. Leadership is given in Scrum but optional in Kanban, leadership can therefore not be seen as important. Marko Ikonen [22] talks about how waste can be reduced with the right leadership even in self-organized teams of Kanban projects. Kanban is all about avoiding waste as much as possible because wasted time is wasted money. Waste in IT can be overproduction, e.g. implementing functionalities that the customer has not asked for but still are quite useful. Unused creativity is another category of waste in IT and can be e.g. lack of processes in a company to share knowledge. Recognizing waste and thereby minimizing its impact in projects can save resources and accelerate lead-time [22]. According to the article, waste seems to appear in lack of communication or when tasks are switched between project members.

2.5 Usability

Usability is the learnability and ease of use of for instance a website or a software application. To determine how the application works for none experts, a usability approach is done to improve the functionality on the GUI on the device. The primary notion of usability is to design an object or product with the users psychology and physiology in mind. Things to keep in mind when studying usability are e.g. to see how long time it takes to accomplish a particular task, if the operation can be learned by observing the object and if it is satisfying to use the object. According to Nielsen and Shneiderman, the concept of usability can be divided into five core parts or questions [7]:

- Learnability - how easy is it to learn as a beginner?
- Memorability - how easy is it for a user to remember what he/she has done?
- Errors - how many errors do users make?
- Efficiency - Once users have learned the design, how quickly can they perform tasks?
- Satisfaction - How pleasant is it to use the design?

For the purpose to determine how good the usability is for a mobile application, feedback can be retrieved in different ways. Usually, there are two main ways to retrieve feedback in a usability purpose. One method is called empirical evaluation and is a method within software testing where the purpose is to evaluate a product by testing the software on persons who are mentioned to use the product. Moreover, there are different ways to determine the usability when using usability testing such as hallway testing, remote testing and interviews. Another method to test the usability of a user interface is to perform an analytical evaluation and here it is common with usability inspections [17], which
is a generic name for a set of methods that are all based on having usability experts, rather than users, inspect a user interface. Typically, usability inspection aims for finding usability problems in the design and can be performed early in the usability engineering life cycle [17]. Common usability inspection methods are pluralistic walkthrough, cognitive walkthrough, heuristic estimation and heuristic evaluation. In this project, the heuristic evaluation method is used.

2.5.1 Heuristic evaluation

The heuristic usability method is a part of the usability inspection method where the purpose is to find problems in the GUI. In its original form usability experts are performing the evaluation but it can be modified to involve users as well, often under the supervision of experts. Often, usability problems that are discovered within heuristic evaluation are categorized using a five-grade scale, this is called severity ratings. Why heuristic evaluation is so powerful is that the evaluations can be done early in the development stage and it gives fast feedback so that problems can be solved in the next implementation iteration. There are a lot of sets of rules also called heuristic sets but the most common and most-used are Nielsens heuristics [15] and these are also used in this project among others. The evaluator (expert or user) reviews the GUI and if any usability problems occur then the specific problem is matched to one or more heuristics that the problem is in conflict with. This is often called that the problem breaks or violates the heuristic.
3 Method description

There are many ways of working in a project. Since prototypes for iPhone, iPad, Android mobile phones and Android tablets will be developed in this thesis work and the purpose for this is to improve the pipeline of filing reports to the municipality the ideal approach for this is to use something called Action Research Method (ARM). ARM\(^\text{1}\) is basically to develop a prototype and simultaneously study the process. Since the thesis is of a problem solving nature the ARM will be used throughout the thesis because it is a valuable support. The steps that will be followed are:

1. Observe a situation and identify the problem that will be solved in this report. This will be defined in section 4.

2. Develop a proposal to a solution. The proposal is a prototype of a mobile phone application, which has been described earlier. The question is if the prototype will make a difference and make the pipeline of filing reports easier?

3. Perform the solution. The development and the actual work will be done according to Kanban. The question is if Kanban is an ideal way to work in this specific situation where two person will develop almost the same prototype but on different platforms. Kanban press for pair programming and knowledge sharing. This can be a downside but the upsides like the visualization of the workflow outweigh the downside.

4. Evaluate the solution by observing it in its context. This will be done by heuristic evaluation of the prototype in its natural environment and with real users.

5. Analyse and reflect how it worked out. This will be described in section 5.

3.1 Question formulation

The conditions for this thesis work is that two person develop slightly the same prototype but in separate environments, iOS and Android. Mentioned earlier are process tools such as Scrum and Kanban. Scrum has certain rules that does not fit this situation. Is Kanban a suitable process tool for this thesis work? The “cons” are that pair programming can’t be applied fully and also knowledge sharing can be difficult. The “pros” are primary the concept of visualize the workflow, finding bottlenecks in the flow easy and so on. The development of the two different platforms will officially be separated into two different projects but will still be visualized on the same Kanban board. This will probably increase the clearness even more across the projects. Will this new way of filing

\(^{1}\text{Aktionsforskningsmetodik in Swedish.}
reports be more efficient that the old way where the user had to first seek up a computer and then file the report?

### 3.2 Problem definition

Until now, the management process of matters within a municipality has been like a huge cobweb. By matter management system means the whole system that the municipality administrates. It can be matter such as building permits to error reports. The system should manage all processes from the first registration to filing the documents in the end either in some kind of electronic archive or a real one. Trivial matters takes too long time and they are experienced to be encapsulated in the bureaucracy. Abou AB has recently taken the matter management system further into the digital world. They have converted a significant amount of services from the old ways of dealing with paper forms to completely handle the whole process digitally. And by doing this, more problems and more important, opportunities will reveal them selves. In the struggle to make services more efficient new techniques and new products are required.
4 Result

This section first handles the result derived from working with the process tool, the resulting prototype, and the result from the usability inspection. The aim of the thesis work is to produce mobile application prototypes for Android and iPhone, the applications is implemented in Java with Android SDK v2.3 and Objective-C with Cocoa framework v2.0. The company who acts as the constituent has plans in adopting the agile process tool Kanban into their workflow. The thesis work will therefore follow these methods guidelines. The aim is to find the optimal parameters in how to work as efficient as possible with only two persons using the Kanban paradigm.

4.1 The process tool

This sub-section will describe the results from working with the process tool Kanban and also the improvements and adjustments made during the implementation phase.

4.1.1 Kanban in a two person project

A two-person team using a Kanban process tool performs this project. Usually, a team consists of five to nine individuals working on the same project. In bigger organizations there are often many teams working on the specific project and such kind of teams are called cross-functional self-organized teams [21]. In this case, the applications should work on two different platforms and therefore the stories can be divided into two sub-projects one for each platform.
The visual workflow in the Kanban board is split into two separate rows where each row belong to a certain sub-project. A decision to divide projects into rows was made and also to make a separate row for things that concern the thesis progress. The row with blue post-its is used for the iOS platform project and the row with green post-its is used for the Android platform project. The pink post-its in the bottom row are used for the duties that can appear whenever and has to be performed. A duty can be a meeting, a workshop or a preparation before a story that must be dealt with before the usual implementation work can continue. A just as important thing to have on the Kanban board is the embodiment of breaks, which in this case is the “Coffee Break” cell. To keep concentration up and focus on work it is important to take break once in a while and also to make them a part of the work. The second core part in the Kanban software development method is to limit the work in progress (WIP). In this project, when a story is inserted into the workflow and before the activity can be labeled “Done” which means that the activity is implemented correctly and simply is done, it has to pass a certain amount of columns. The initial columns that was decided to be on the Kanban board was:

The first column is “Ready for implementation” which means that the activity just before it was inserted into the workflow it had the highest priority in the product backlog. And also when the activity has been inserted into the workflow the product owner has no more authority over it. It is now completely up to the programmer to finish the activity.

The second column is “Implementation” and and this column is simply what it
sounded like, the implementation part of the pipeline. Here the actual code is
written.

The third column is “Test” and also this column has a self-explanatory name.
This column consists of the activities that are pending to be tested or are
involved in testing for the moment.

The fourth column is “Done” and the activity is inserted into this column when
it has passed the prior columns. This column is recurrently emptied each week.
Each state is assigned with a limit that tells how many stories that can be
performed at the same time. The initial limits for each column can be derived
from equation 1 below.

\[
L = 2n - 1
\]  

(1)

Here \( L \) is the limit and \( n \) is the amount of people in the team. In Scrum, priori-
tization is always done by sorting the product backlog [21] and the prioritization
are often set for the next coming sprint. In Kanban priority can be set at any
time during the project or the team can choose not to use priority at all. How-
ever, it is important to define how to prioritize. In Kanban, every team needs
some kind of decision rule for which item to pull first. As it can be seen in figure
6, this project uses three smaller post-its to prioritize the stories, which makes
it easier for the product owners to change the priorities.

In Scrum, daily meetings, often called “daily scrum” are preferable. These every
day meetings are held at the same time and at the same place, often within 15
minutes. In Kanban, daily meetings are optional but are common in the most
software development projects. Because the team in this project contains of only
two persons, the daily meetings won’t be necessary. The decisions and updates
on how the work progress will automatically be exchanged in a natural way.
A week begins with a retrospective meeting where priorities are discussed and
changed to the Kanban board are made if necessary. Feedback meetings with
the product owners occur one time every week where updates and feedback are
discussed 7.

4.1.2 Adapting the project to Kanban

In Scrum, each sprint contains an amount of prior tasks that has to be done
before the end of the sprint. Instead of sprints in this Kanban project, each
iteration can be seen as a phase and each phase ends with a demo. This is
depicted in figure 7. In each demo, persons that are involved in the project
are able to test and give feedback on the product. This is a way of getting
fast feedback on the product and changes can easily be made by adding new
activities based on the feedback in the product backlog. The third core part in
the Kanban software development is to measure the lead-time which basically
means to measure the average time to complete one item, sometimes called
“cycle time” [2]. In this project the lead-time is the time between starting to
implement the activity and reaching the “Done” column on the Kanban board.
Figure 7: This figure depicts the re-occurring happenings in an implementation phase in the project. Each phase is two weeks long and ends with a demo session where the functionality that has been implemented is demonstrated. Reconciliation sessions are to check if the project goes according to plan and on these sessions team members and tutors participate. In the retrospective meeting that occur in the middle of the phase decisions on adjustment regarding the work are made.

The lead-times has been gathered in figure 8 and 9. The green and the grey line represents the activities from the backlog that has been finished in that specific week and the red lines represents the daily duties that had to be done. The x-axis is the week and the y-axis is the amount of duties or activities finished. It can clearly be seen that the activities and duties correlate quite hard in the negative way, which mean that if the specific amount of finished activities that week is low then an explanation for this is that the programmer has had many duties to perform before he can return to the activities from the backlog.
Figure 8: This figure depicts the lead-times in green and urgent duties in red for the Android project. The x-axis represents the week in the project starting with week one and so on and the y-axis represents the number of finished activities or duties that week. A pattern can be seen that when productivity is high in the project the duties are low and vice versa, with few exceptions.

Figure 9: This figure depicts the lead-times in green and urgent duties in red for the iOS project. The x-axis represents the week in the project starting with week one and so on and the y-axis represents the number of finished activities or duties that week. A pattern can be seen that when productivity is high in the project the duties are low and vice versa, with few exceptions.
4.1.3 Initial estimates and changes

The initial WIP parameters were set to four in all columns. This decision was made by assumption that the work will be more efficient and if the programmer gets stuck s/he can take on a new activity and have 4 different activities parallel or pending. The hypothesis is that this will create an environment where the programmer never can be out of work and always know what to do next. The initial parameters are depicted in figure 10.

![Figure 10: This figure depicts the first Kanban board v. 1.0 that was created in the project. The WIP limits are not based on anything in particular other than what the team members think is adequate. These parameters are to be adjusted in the future.](image)

The first changes to the Kanban board were to merge the implementation and the testing columns for the iOS project. This decision was made on the grounds that testing can easily be included in the xCode project as a separate target. The initial WIP limit for this merged column was set to four. xCode is a set of tools developed by Apple and is basically for developing software for iOS and Mac OS X. The WIP limits was also reduced to two in the implementation and test columns for the Android project. The reduction of the limits was done because the activities that didn’t got finished started to stack up and bottleneck started to visualize in the Kanban board. This forces the programmer to deal with the problems concerning the activity and finish them before starting with a new one. The changes is depicted in figure 11.
After noticing that not all activities needed to be tested or even could not be tested in a natural or intuitive way the same change that was made to the iOS project was made to the Android project as well. That is to merge the implementation and test column together as one. This decision was made after determining that it worked well in the iOS project and also that not all activities needed to be tested as well. The weekly meetings or retrospectives were initially in the project performed without exceptions. As the project moved on the team members noticed that this was only a waste of time since the two team members always sat near each other in the office and worked close together. In a Lean point of view all that is waste should be eliminated. If changes in the future need to be made it could easily be decided in more informal way like in the lunchroom or on a coffee break. The final Kanban board is depicted in figure 12.

Figure 11: This figure depicts the second Kanban board v. 1.1 that was created in the project. Adjustments were made regarding the “Implementation” and “Test” columns to fit the xCode IDE for the iOS project better. Also the WIP limits were decreased.

<table>
<thead>
<tr>
<th>v 1.1</th>
<th>Backlog</th>
<th>Ready for implementation (4)</th>
<th>Implementation (2)</th>
<th>Test (2)</th>
<th>Done (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Daily duties</td>
<td>Future duties</td>
<td>Urgent duties (2)</td>
<td>Performing duties (4)</td>
<td>Done</td>
<td>Coffee break</td>
</tr>
</tbody>
</table>

Figure 12: This figure depicts the final Kanban board v. 1.2 that was created in the project. Adjustments were made regarding the “Implementation” and “Test” columns. This was done because not all activities needed to be tested.

The time schedule of the implementation has been altered several times during the phase. This is a typical way or working in agile software development. Figure 13 depicts the initial plan with milestones.
Figure 13: This figure depicts the initial time schedule that was created before the project started.

After each milestone was completed a new updated version of the schedule was created. Since this is a pilot project and creating a new prototype can be hard to plan the final schedule which is depicted in figure 14 was quite different than the initial one. During the week flow, many stories, which where planned to take weeks to finish where instead done in a couple of days. This happened when the implementation of camera and GUI where done in the same iteration.

Figure 14: This figure depicts the final time schedule. The time schedule has been updated continuously throughout the project.
4.2 The prototype

In this section the different phases of the implementation work are described more thoroughly. There are four main phases and each of them deal with separate things. Figure 15 describes the communication between the application and the web service, both in the test environment where the application has been tested during the project and how it will communicate in the future if it will go into production.

![Diagram](image)

Figure 15: This figure describes how the applications communicate with the web services, both in the test environment that has been utilized within the project and how it will communicate in the future if the applications goes into production. (a) This figure described the test environment. The arrows represent the data flow. The application posts a report object to the web service. The web service stores the report in a database. The application in turn fetches data from the database with a so-called “get” call. (b) This figure describes the environment that the applications will communicate with in a future production. The arrows represent the data flow. The application posts a report object to the web service. The web service stores the report in a database and also creates a LEX object, which basically is a mark-up structured file. LEXTalk is a program that the municipality owns and the administrators work with. This program creates real PDF reports and stores them in a database. The application fetches data from the database with a so-called “get” call.

4.2.1 Phase 1 - GUI and navigation

The GUI is built upon a survey that was made with people working on Upplands Väsby municipality and aims to emulate their web page e-service for error messages [3]. The application is designed for understanding users, having a sense of
people's capabilities and limitations [24]. However, the start page of the error report application has a list of earlier filed matters from other users. The list consists of the earlier matters category and the address. As it can be seen in the figure 16, the start page is designed as a typical first page that greets the user. This page also contains the main Upplands Väsby municipality logo at the top. The application is designed for users that utilize the existing e-service at the web, therefore it is important that the design is made for the same target group and gives a recognizable feeling. The implementation of all objects such as text fields and radio buttons and how they should behave is implemented first. In section 4.2.4 the GUI is based on the results from the evaluation in a usability point of view.

**GUI - iOS** The GUI in general for iOS is designed to show separate designs depending on if the device is an iPhone or an iPad. There are only two different sizes for screens to iOS devices so this solution is suitable. This is also a conscious choice that has its roots in the usability paradigm. Since the screen sizes on these two devices differ greatly and obviously can contain different amount of information it’s logical to tweak the different GUI’s to fit the screens better. Also this makes it possible for future addition of more information to the screen.

**GUI - Android** The application for Android has another solution. Since an Android mobile device can have a screen size between 2.7 and 10.1 inches, the device basically expands all objects in the GUI to fit properly on the screen. Since there is a great number of different sizes and screens that are compatible with Android, this solution fits better than making separate GUI’s for all possible sizes of screens. It would be a tedious process.

**4.2.2 Phase 2 - Photo and GIS**

These two functionalities are implemented in the same phase because that the time for both of these phases where greatly miss estimated. The time it took to implement these two functionalities turned out to be less than half. This is because of lack of experience in such specific assignments and therefore it is natural to make the estimates to big to be on the safe side. The photo functionality makes it possibility to attach a photo either from the photo library of the device or take a photo using the built in camera. This is if the device has a camera. Some devices such as tablets, early models of iPad and iPod does not have this hardware support but must still be able to use the prototype application.

The GIS (Geographical Information System) functionality means that the application makes use of the device built in hardware support for finding out the current location of the device. This information is used in conjunction with a method called reverse geocoding and also Google’s map service [5, 9]. The reverse geocoding method is basically the conversion of coordinates to human readable information about that location the coordinates consider. The reverse
geocoder in turn uses Google services to provide location data. This is if the
device has a camera. Some devices such as Android tablets, early models of
iPads and iPods do not have this hardware support but must still be able to use
the prototype application.

4.2.3 Phase 3 - JSON

JSON (JavaScript Object Notation) is basically a more lightweight mark-up lan-
guage than e.g. XML (Extensive Markup Language). It has been notoriously
popular to use in solutions for mobile devices. This is because it’s an independ-
ent language with parsing possibilities available for mostly all programming
languages. A JSON object is a collection of simple structured arrays and data.
This functionality makes it possible for the application to talk with a so-called
RESTful web service. This web service can receive http-requests with JSON
objects as its body. To summarize this up it is basically the request that tells
the web service to store the data that the JSON object represents. The func-
tonality implemented is both to send JSON objects as requests to this service
and also to fetch JSON objects from a certain URL.

SBJSON

SBJSON is a third party framework that is used in the iOS application. Un-
like Android that has this built into the IDE (Integrated Development Environ-
ment), this framework has to be downloaded and integrated manually from[8].
The framework implements a so-called JSON parser and also a generator of
JSON objects for Objective-C.

4.2.4 Phase 4 - Usability evaluation

During this phase a heuristic evaluation has been performed. This evaluation
has the purpose of finding problems with the user interface. If it is a usability
problem it has to violate a certain preselected so-called heuristic. A heuristic
is basically a rule and several heuristics are used to make sure of the quality of
the application. These heuristics are derived from [6] and are mostly based on
Nielsens ten usability heurstics [27].

4.2.5 How to use the prototype application

The start page depicted in figure 16 is the first page that the user sees when
starting the application. The objects on this view are a welcome text, and a list
of earlier filed reports, and a way of navigating to other pages. The purpose of
the list is so that the user can check if the error s/he has found is not already
processed by the administrators that work for the municipality.
Error report form view depicted in figure 17, the user can apply all info regarding the error. The numbered list below appertains to the numbers in figure 17.

1. First the user has to specify a category from a pre-set list of categories.
2. Second the user has to specify a description in free text of the error.
3. Third the user has to specify the address either explicitly in text or by using the GPS functionality that converts the coordinates where the device currently is located into a human readable address.
4. Fourth the user has the possibility to attach a photo of the error that the user has found. This can either bee done by taking a photo with the
devices camera or by browsing the photo album for earlier taken pictures if the user isn’t on the location where the error itself is located.

5. Last the user has to choose if the user wants any feedback on the matter. This is a choice that either leads to the next view called personal records view or if the user chooses not to get feedback he will be sent to a summary view.

Figure 17: Error report form page. This is where the user can specify everything regarding the report. (a) iOS. (1) Category text field. (2) Description text field. (3) Address text field. (4) Address number text field. (5) Fetch address via GPS button. (6) Attach photo button. (7-8) Feedback selection buttons. (9) Next button. Takes the user to the next page. (b) Android. (1) Category text field. (2) Description text field. (3) Address text field. (4) Address number text field. (5) Fetch address via GPS button. (6) Attach photo button. (7-8) Feedback selection buttons. (9) Next button. Takes the user to the next page. This button becomes visible if the user scrolls down the page.

Personal view in figure 18 is for users that has chosen to get feedback on the matter they wants to file. Here the user specifies his or her name and depending on what kind of feedback way the want, they can choose between e-mail and telephone. There is also a way to navigate to next view.
Figure 18: Personal record page. This is where the user can specify the personal information that the administrators need to be able to return feedback on the matter. (a) iOS. (1) First name text field. (2) Last name text field. (3) Email text field. (4) Phone number text field. (5-6) Feedback way selection. (7) Next button. Takes the user to the next page. (b) Android. (1-2) Feedback way selection. (3) First name text field. (4) Last name text field. (5) Email text field. (6) Phone number text field. (7) Next button. Takes the user to the next page.

Summary view in figure 19 is where the user can check if the information s/he has specified is correct. All the input from the earlier pages is summarized here. The user has the choice to go back and change the input or to send in the report.
Figure 19: Summary page. This is where all user specified information is summarized to be validated. (a) iOS. (1) Send button. If this button is pressed the report is sent in to the administrators in the municipality. (b) Android. (1) Send button. This button is not visible until the user scrolls down further on the page. If this button is pressed the report is sent in to the administrators in the municipality.

4.3 Graphical user interface in a usability approach

This section describes the results from the usability study in form of statistics and comparisons between the different applications.

4.3.1 Heuristic evaluations

The purpose for this evaluation is to increase the usability of the system at the same time as defining system bugs that can be both time consuming and expensive to find later in a possible production project of this prototype. The evaluation is not a systematic way of solving usability problems, only to define them.

It is important to define the usability problems, partly so they can be solved in a better way directly and partly to make persons that are going to implement solutions to these problems in the future aware of them. Theses persons are
possibly not the same persons as the ones who performed the evaluation and if the results are documented properly this person is spared to do more extensive evaluation work. It is often obvious how to solve the problems that are the result of the heuristic evaluation.

The conditions for the actual heuristic evaluation are described below. First a set of heuristics or rules of thumb are defined based on Nielsons Ten Heuristics [27]. Each heuristic comes with a set of questions that will help the observer to check if the problem violates the heuristic. The heuristics and associated questions used in this evaluation are:

1. Visibility of system status
   - Is status feedback provided continuously (e.g. progress indicators or messages)?
   - Are warning messages displayed for long enough?

2. Match between system and real world
   - Are the words, phrases and concepts used familiar to the user?
   - Does the task sequence parallel the user’s work processes?
   - Is information presented in a simple, natural and logical order?
   - Is the use of metaphors easily understandable by the user?

3. User control and freedom
   - Are facilities provided to "undo" (or "cancel") and "redo" actions?
   - Are there clearly marked exits (for when the user finds themselves somewhere unexpected)?

4. Consistency and standards
   - Is the use of terminology, controls, graphics and menus consistent throughout the system?
   - Is there a consistent look and feel to the system interface?
   - Is there consistency between data entry and data display?
   - Have ambiguous phrases/actions been avoided?

5. Error prevention
   - Is a selection method provided (e.g. from a list) as an alternative to direct entry of information?
   - Is user confirmation required before carrying out a potentially ‘dangerous’ action (e.g. deleting something)?
   - Does the phone ensure work (e.g. unsent text messages) is not lost either by user or system error?
• Does the phone prevent calls being accidentally made?

6. Recognition rather than recall

• Are help and instructions visible or easily accessible when needed?
• Is the relationship between controls and their actions obvious?
• Is it possible to search for information (e.g. a phone number) rather than entering the information directly?
• Is the functionality of the buttons on the device obvious from their labels?
• Are input formats (e.g. dates or lengths of names) indicated?

7. Flexibility and efficiency of use

• Does the phone allow for a range of user expertise?
• Does the phone guide novice users sufficiently?
• Is it possible for expert users to use shortcuts and to tailor frequent actions?
• Is it possible to access and re-use a recent history of instructions (e.g. recently called numbers)?
• Does the phone allow for a range of user goals (e.g. calling, text messaging, playing games) and interaction styles?

• Aesthetic and minimalist design
  – Is the design simple, intuitive, easy to learn and pleasing?
  – Is the phone free from irrelevant, unnecessary and distracting information?
  – Are icons clear and buttons labelled and is the use of graphic controls obvious?
  – Is the information displayed at any one time kept to a minimum?
  – Is the phone easy to remember how to use?

• Help users recover from errors
  – Do error messages describe problems sufficiently, assist in their diagnosis and suggest ways of recovery in a constructive way?

• Help and documentation
  – Is help clear and direct and simply expressed in plain English, free from jargon and buzzwords?

• Navigation
– Is navigational feedback provided (e.g. showing a user’s current and initial states, where they’ve been and what options they have for where to go)?

- Use of modes
  – Does the phone use different modes appropriately and effectively?

- Structure of information
  – Is there a hierarchical organisation of information from general to specific?
  – Are related pieces of information clustered together?
  – Is the length of a piece of text appropriate to the display size and interaction device?
  – Has the number of screens required per task been minimized (by minimizing use of white space and careful use of menus)

- Enjoyment
  – Is the phone fun to use?

The evaluation in this thesis work is done with four participants. Each participant does the evaluation with identical scenarios for the iOS application and the Android application. The participant is first familiarized with the heuristics and some general questions regarding usability to be kept in mind throughout the evaluation. The participant is also encouraged to keep a dialog about what s/he experiences. The evaluation starts with the observer giving the participant a scenario that in theory could be taken from a real life situation. The scenario is built as a task that the participant is encouraged to perform accordingly. Depending on how long a scenario takes to perform, a set of at least two scenarios and at most five is presented to the participant. The maximum time that the evaluation must not exceed is one hour. The observer is documenting all problems that occur and after that session with the participant is completed the problems are evaluated and associated with one or more of the usability heuristics that are violated. The result of the evaluation is documents containing tables of usability problems. Each problem is documented with properties such as the problem occurred in the system, which heuristic or which heuristics that got violated, the participants experience level, the impact of the problem, and the frequency of the problem. All these properties weighted together generates a severity rating. This rating is a number between zero and four [12]. The different severity rating that are used are described in table 1.
<table>
<thead>
<tr>
<th>Severity ratings</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>I do not agree this is a usability problem at all</td>
</tr>
<tr>
<td>1</td>
<td>Cosmetic problem only: need to be solved unless extra time is available in the project</td>
</tr>
<tr>
<td>2</td>
<td>Minor usability problem: solve this should be given low priority</td>
</tr>
<tr>
<td>3</td>
<td>Major usability problem: important to solve so should be given high priority</td>
</tr>
<tr>
<td>4</td>
<td>Usability catastrophe: imperative to solve before product can be released</td>
</tr>
</tbody>
</table>

Table 1: The severity ratings in this table are used to prioritize the usability problems found by the heuristic evaluation. The ratings are taken from [12].

Figure 20 lists all heuristics and how often they got violated according to the participants that performed the evaluation. Note that the same problem can possible have been found by several participants. Consequently the participants finds problem in the system and experts that observe them document and connect each problem with heuristics and rate them. The actual results of the evaluation is documented and attached in Appendix I and Appendix II but a summary of them is presented in subsections 4.3.2 and 4.3.3.

![Figure 20: The different heuristics and how often they are violated. The x-axis represents the different heuristics and the y-axis represents how many times each heuristic is violated. The different heuristics are: 1. Structure and information 2. Navigation 3. Match between system and real world 4. Visibility and system status 5. Recognition rather than recall 6. User control and freedom 7. Error prevention 8. Consistency and standards 9. Help user recover from errors 10. Aesthetic and minimal design 11. Flexibility and efficiency to use](image-url)
4.3.2 Result of heuristic evaluation of Android application

The results from the usability evaluation are depicted in figure 21. The most common usability problem in terms of severity ratings is problem with severity rating three and the rating ranked second is problem with severity rating one. Regarding the heuristics, as it can be seen in figure 21 most of the heuristics that are violated is the heuristic of “match between system and real world”. By reviewing the problems, which had the highest severity ratings, three out of four participants thought that the GPS button was non-functional. The application coordinates where updated when a new position was found, without that the user knew about it. Another problem with the same severity rating was that the camera crashes when using it more than once in the same application run. In this scenario, three out of four participants found this problem as important to solve but also easy to overcome since the camera button is optional to use. This problem belongs to the error prevention heuristic, which can be seen in figure 21. Another problem that had the severity rating of three was the “send” button in the last page in the application. Three out of four participants notified this as a problem. The problem was that the screen freezes when pressing the “send” button. By reviewing the first page in the application, all participants thought that it was to many items in the list, this has a connection to the heuristic of “match between system and real world”. A usual scenario is when a user search after a specific error in the list, then it takes too long time to find the this error by searching through this list. However, the severity rating in this case was set to two, which basically means that it is a minor usability problem. This problem can therefore be solved when needed. In total the users found 28 problems and 17 of them was unique. The users didn’t find any problems in the most severe rating. In the next highest category the users found 7 different problems and the two most common was found by 3 of 4 users.
4.3.3 Result from heuristic evaluation of iOS application

Most of the problems obtained from the evaluation were on the not so severe scale or more precisely in the category 2: Minor usability problem which can be seen in figure 22. This should be given a low priority to solve and will be performed when the programmer has time. The most severe problems that was given the highest rating four was three different problems. The first and most obvious was that the users personal information was sent in if the user prior to a anonymous report filed another one which contained personal information. This problem three out of four participants found and the fourth participant that did not, found another one that was slightly different but occurred on an earlier page. The next and not so obvious one was that the personal information was still in the text fields when the user wants to file a new report. This problem was experienced by two out of four participant. The highest rating, four, only one out of four participants found. This problem was a system crash that occurred when the user tapped between views rapidly multiple times. The next highest usability problem rating is three and the participants found 7 different problems in this rating category. The fist and most obvious was that if a number contains more that one digit it is not visible in the text field. Three out of four participants found this problem. The second problem that three out of four participants found was the error message given when the GPS functionality does not work is given in English and is very cryptic to the participant. In total

![Severity Ratings](image)

Figure 21: Different severity ratings of usability problems for the Android prototype. The x-axis represents the different severity ratings and the y-axis represents how many usability problems that were found for each severity rating in the evaluation.
the participants found 32 problems together and 20 of them were unique. What causes these problems and how some of them can be solved will be proposed and discussed in the following chapter.

Figure 22: Different Severity ratings of usability problems for the iOS prototype. The x-axis represents the different severity ratings and the y-axis represents how many usability problems that were found for each severity rating in the evaluation.
5 Discussion

We have seen that e-services bring the citizen closer to their municipalities. The lead-time from when an activity comes onto the board until it has been finished has been reduced significantly. We think this is because of the continuous communication between the team members and if improvements were necessary they were made. According to what Obama mentioned about an open government [18], we can see that if public data are available to be used for free then private persons and companies would be able to build services with them. Internet is the modern world’s public space and through that, governments have now opportunities to better understand people’s concern. By looking on the existing public data that for example opengov.se [10] provides, it is possible to receive data from the public sector in Sweden. Recently developed mobile applications such as STHLM traveling [13] are built upon public data in Sweden. Since public data becomes more accessible and more applications are built upon that data, we therefore believe that the democratization grade for the citizen will increase more in the future. By developing e-services for mobile devices we believe that the connection between citizen and municipality will occur more frequently. Since a mobile device is easier to carry than say a laptop and the mobile communication network is growing rapidly, it will be easier to get access to the municipality and information regarding it. Also to download the applications are easy and free for citizens.

5.1 The process tool

Kanban method is intuitive to understand and gives rather free hands to the developers to do their work [33]. However, nobody has evaluated the Kanban method on a two-person project before and therefore, most parts within this approach could be seen as a first attempt. Things estimated to be useful from the process tool in future projects are milestones such as in figure 7. Ikonen et al. [26] compile a research framework, which contains nine literature-based aspects of project work and the expected influences Kanban has to them. Much of these expected influences are recognized in this project. For example, visualization, which is important to get an overview, is solved by a Kanban board. Another example is the problem solving issue. Unexpected problems are found quickly on the Kanban board and usually take the form of bottlenecks. One more example is the feedback. Feedback was given continuously in the demo sessions, which resulted in fast redesigns. Finally the team members really tried to embrace the method Kanban, by actually trying to find changes in the Kanban board that would lead to improvement of the workflow. This has been made and described in section 4.1 and clearly improvements have been done. By looking at the figure 8 and figure 9 we can see that the productivity strongly correlates to how many duties that has occurred in the specific week. To separate the work in this way makes it more understandable why some things are more time consuming. This clear picture can motivate the team in rationalizing their work.
and duties that isn’t a part of the project.

So, is Kanban an ideal way for a small two-man team to work, project based? Since the tool is so flexible and basically gives the team members the freedom to change the work processes that doesn’t fit we would say, yes. Kanban is a more suitable way than Scrum anyhow because Kanban does not have as many rules than Scrum.

In retrospective, we can see that no long time waste ever occurred since raised problems where resolved early in the process. By always using limits of two in the Kanban board we can see that it is difficult to achieve any waste in the process. The risk is that the work can come to a total stop if no solution the the problems is found. Compared to Scrum, if a story can’t be finished then it is easy to shift to another story and continuing with that one. This can lead to waste because activities never become finished. Another difference between Scrum and Kanban is when adding stories to the board. In Scrum, when making the sprint backlog, all stories for the up coming sprint are added to the board, waiting to be done. Compared to the Kanban process, stories can be added no matter what day it is in the development phase. A lesson from this project is that managing stories has been an advantage compared to the Scrum way because stories where continuously popping up on the Kanban board. Instead of having to wait to start a specific story then it can be added directly onto the Kanban board. Especially many duties where added into the board since this project contained a lot of unforeseen meetings.

We would absolutely suggest adopting the Kanban way to similar projects like this one and even more strongly if the team contains of more than two persons. The most valuable help we got from Kanban was the visual part and the Kanban board, which gave a good overview of both projects at the same time. We also think that the visual part can contribute to a more insight for people that are not members of the development team such as mentors and project owners.

5.2 Usability discussion for Android application

The most common problem or the most commonly found problem by the users for the Android application was to use the GPS functionality, the camera functionality, the absence of interactivity with the list on the first page and the absence of the progress indicators. Many of these detected errors seem to be related to the heuristic “visibility of system status”. Also, several usability problems with low severity rating seemed to be related to the language and phrases in the application. Most of those problems that were pointed out with only cosmetic defects are quite hard to find solutions for. A usability problem described as “a feeling that something is wrong” is harder to find the solution to rather that the more severe usability problems where the solution usually is more obvious. One specific usability problem that three out of four participants found in the evaluation was that it took long time to find the geographical location when operating inside. This general usability problem occurred on several places and the user reacted on that the data was presented too slowly.
Roto et al. [31] say that users normally shift their visual attention from a system context after four to eight seconds. Suggestions to solve this problem could be to either have some kind of time-outs that kills the loading times if they are depending on band connectivity to the Internet or if the loading process take too long, have a process indicator that visualize that the system is idle. This is to ensure the user that there are not any problems with the system. A rule of thumb for loading views in general is that it is not appropriate to exceed 10 seconds [11].

5.3 Usability discussion for iOS application

By looking at the results from the heuristic evaluation of the iOS device and comparing them to the results from the evaluation of the Android device the conclusion is that the problems are quite similar. The most problems that occurred also seemed to correlate with insufficient or unclear feedback from the system to the user. Aside from the severe problems that were obvious and that well-aware of before the heuristic evaluation such as system crashes other usability problems seems to concern language and miss use of correct or terms and phrases. We can also see that the most severe usability problems were found by three out of four respectively two out of four participants, and the ones with lower severity rating and is not as important was not found by as many participants. That is, it was not as common that many users found that same usability problems in the low severity rating categories.

By looking at the graph in figure 20 there were two heuristics that was overrepresented. The first one and the most common was “Match between system and real world”. This heuristic can be summed up as that the context of the system should be presented in a logical and simple way with words and phrases that are familiar to the participant. The second most violated heuristic was “Recognition rather than recall”. This heuristic is quite similar to the prior. This heuristic says that the system should provide a logic that helps the user to understand what is happening and what will happen if the user acts in a certain way. This can be that the user already knows by by intuition from previous knowledge, what will happen if s/he taps a button. Why these two heuristics are overrepresented can be explained by that the logic of a system is a hard thing to estimate from no guidelines to presuppose from. If there already exist a foundation it is more likely to get the logic more right. Most of the usability problems that occurred have obvious and easy to implement solutions.

5.4 Usability discussion and comparison between the platforms

The evaluations did result in tables of usability problems that are documented in Appendix I and Appendix II. These will with great benefit serve as blueprints to enhance the future product. It is also very easy to measure the usability enhancements in the future by simply checking if the usability problems in the
tables have been solved. So, will this prototype make a difference and make the pipeline of filing an error report easier? We claim that only by getting rid of the need to have a computer to file the report which was the case on the prior e-service solution and making the filing possible anywhere with a mobile phone it is more easy. In a usability point of view the heuristic evaluation also has contributed to a higher awareness of the problems with the system. This generates a good foundation to improve the system even more in the future. One possible source of error in the heuristic evaluation can be that the participants assignment was to evaluate two almost identical systems. Let’s say that the participant starts with evaluating the iOS application, theoretically speaking when the participant performs the second evaluation s/he must remember what kind of problems to expect. This can also be an explanation that the result where similar and it is possible that problems that were found in one evaluation was transferred to the next automatically. This is both a good and a bad thing. It is good because more problems are found in total but if we want to investigate more specific how many problems a certain user find independently from each other then it is a bad thing and the results cannot be used because the participant has been influenced by external conditions.

The results from the usability inspection of the Android application have some differences regarding the GUI comparing to the iOS application. This is depicted in figure 21 for Android and figure 22 for iOS. However, the aim with the evaluation was to find similar problems in the applications since the design of the GUI was the same. Compared to the iOS application where two unique problems of severity rating four was found, no problems of severity rating four was found in the Android application.
6 Future work

The applications are developed with the possibility to expand them in the future. A vision is to make it possible to do more services than filing error reports in the application. Such services can be room booking or to file complaints about your accommodation. Not all services are suitable for the application. Services that are not very commonly used are not suitable for the application since the application is basically an extension to the services on the web page. Services that are not commonly used does not have a need to be simplified and more available. If the service is performed seldom then there will be more work to implement it than the users experienced value. It is not worth it basically. This application is also made for a specific municipality and this can with benefit be changed to a general version that other municipalities also can use. If a reusable code base can be implemented then it would not be necessary to make a specific application the new municipality only minor adjustments to get the application up and running for the specific municipality. Also this application could be adapted to other markets also like the housing market where filing of error reports is a common thing to do.
References


Part I

Heuristic evaluation result tables
for iOS application
## A Table 1 - Result from participant 1

<table>
<thead>
<tr>
<th>Place of Occurrence</th>
<th>Problem description</th>
<th>Heuristics violated</th>
<th>Specific user consideration</th>
<th>Severity rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report view</td>
<td>The user reacts on the text on the button that makes it possible to either take photo with camera or fetch photo from library. The text says “Hämta bild”, it give the illusion that its only possible to fetch from library.</td>
<td>- Match between system and real life - Recognition rather than recall. - Aesthetic and minimalistic design</td>
<td>The user can overcome this by tapping the button and get a new opportunity to make a choice. This time two additional buttons appear where the user can choose both to take photo with camera or fetch from library</td>
<td>2 Minor usability problem</td>
</tr>
<tr>
<td>Report view, personal records view</td>
<td>All information from the previous report is still in all text fields. Should be deleted when the report is filed.</td>
<td>- Error prevention</td>
<td>This is nothing the user can overcome by them selves.</td>
<td>4 usability catastrophe</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>Impact</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>non-experienced</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report view</td>
<td>The user can’t see the whole number when editing the text field where the user should specify street number. The user thinks that he/she maybe has specified wrong or missed the first digit if the number contains more than one digit.</td>
<td>Recognition rather than recall</td>
</tr>
</tbody>
</table>

| Report view | The error message given when the GPS functionality didn’t work was unclear to the user. | - Visibility of system status - Help user recover from errors | non-experienced | The user can overcome this problem by tapping the “OK” button on the error message. The result is that the user did understand that something went wrong and not what went wrong. | Every now and then when the Google map service decides not to respond. | 3 Major usability problem |

| Report view, personal records view | The application crashes if the user is flipping between views rapidly and frequently. | - Error prevention | non-experienced | The user can start the application over again. | Every time the user flips between views more than estimated four or five times. | 4 usability catastrophe |
### Table 2 - Result from participant 2

<table>
<thead>
<tr>
<th>Place of Occurrence</th>
<th>Problem description</th>
<th>Heuristics violated</th>
<th>Specific user consideration</th>
<th>Severity rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error report view</td>
<td>The error message is ambiguous and should be in Swedish.</td>
<td>- Match between system and real life - Consistency and standards</td>
<td>non-experienced Since it’s the error message code that Google responds with its hard to understand what have happened and overcome this problem.</td>
<td>3 Major usability problem</td>
</tr>
<tr>
<td>Report view</td>
<td>If the user specifies the street number with more than one digit the whole number will not be visible and therefore the user can think that he/she has specified the number wrong.</td>
<td>- Visibility of system status</td>
<td>non-experienced The user can easily misinterpret what he/she has specified in the text field and this can be hard to overcome every time the street-number contains more that only one digit.</td>
<td>3 Major usability problem</td>
</tr>
<tr>
<td>Report view</td>
<td>When the user is attaching a photo of the error the text in the buttons is in English but everything else is in Swedish in the system</td>
<td>- Match between system and real life</td>
<td>non-experienced</td>
<td>If the user isn’t familiar with the English language which can be the case it’s impossible to overcome the problem.</td>
</tr>
<tr>
<td>Report view</td>
<td>The user envisaged that it was possible to simply tap a cell in the category picker rather that first scrolling to the selection and then tapping the “Done” button</td>
<td>- Match between system and real life</td>
<td>non-experienced</td>
<td>This is simple for the user to overcome since the “Done” button is just next to the selected cell in the picker. After the user understands that it’s not possible to choose category by tapping the cell he/she almost always realize that it’s the “Done” button that is the real submit action.</td>
</tr>
<tr>
<td>Personal record view</td>
<td>There are two separate text fields for input of feedback way. The user either wants feedback via mail and via phone not both. This should be either that one or the other text field gets disabled dependent on what the user picks to be his/hers feedback way.</td>
<td>- Match between system and real life (Metaphors aren’t easy to understand) - Recognition rather than recall(The functionality isn’t obvious).</td>
<td>non-experienced</td>
<td>It’s simple for users to only apply for one feedback way but it’s still possible to apply for both which shouldn’t be possible.</td>
</tr>
<tr>
<td>Summary view</td>
<td>The describing text is always visible even when the user has choose to be anonymous.</td>
<td>- Aesthetic and minimalism</td>
<td>non-experienced</td>
<td>The user can’t overcome this at all, only ignore this aesthetic problem.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Summary view</td>
<td>The feedback alert view gives the same message depending on if the user has choose to be anonymous or if he/she wants feedback. It can benefit greatly if the messages are different.</td>
<td>- Match between system and real life</td>
<td>non-experienced</td>
<td>The user can’t overcome this at all</td>
</tr>
<tr>
<td>Summary view</td>
<td>If the user wants to file a new report after another report has been filed and more precise the user first wanted to file a report with feedback and then wants to file a report anonymously the person records from the prior report occurs on the summary view.</td>
<td>- Match between system and real world</td>
<td>non-experienced</td>
<td>This is nothing the user can overcome by them selves.</td>
</tr>
<tr>
<td>Start page, report view, personal record view and summary view</td>
<td>The progress indicator says “Konfigurerar data” on several places. This text should be more describing on what happens, not so general.</td>
<td>- Match between system and real world - Visibility of system status</td>
<td>non-experienced</td>
<td>This is nothing the user can overcome and doesn’t give the user enough feedback on what happens with the system</td>
</tr>
<tr>
<td>Start page</td>
<td>The information that the table on the start page consists of is loaded before the view is visible.</td>
<td>- Visibility of system status.</td>
<td>non-experienced</td>
<td>The user can’t overcome this or bypass this in any way. He/she has to wait until loading is done. The user can misinterpret this to a crash of the loading is time consuming.</td>
</tr>
<tr>
<td>Error report view</td>
<td>Too much text that describes what the text fields should contain.</td>
<td>- Aesthetic and minimalistic design</td>
<td>non-experienced</td>
<td>The user do understand what to write in each text field so the problem is easily overcome but the information describing this can be more minimalistic.</td>
</tr>
</tbody>
</table>
## C  Table 3 - Result from participant 3

<table>
<thead>
<tr>
<th>Place of Occurrence</th>
<th>Problem description</th>
<th>Heuristics violated</th>
<th>Specific user consideration</th>
<th>Severity rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start view</td>
<td>The progress indicator that is displayed first thing in the application indicating that the application is loading disappeared to quickly so that the user didn’t understand the meaning if it.</td>
<td>- Visibility of system status</td>
<td>the user did overcome this with just a reminder that something happened but didn’t understand what.</td>
<td>1 Cosmetic problem only</td>
</tr>
<tr>
<td>Start view</td>
<td>The user think that it will be possible to tap the different cells in the table. When the user taps a cell it indicates that something will happen and the cell gets blue. But nothing happens.</td>
<td>- Recognition rather than recall</td>
<td>The user can easily overcome this problem by simply avoid tapping the cells in the table.</td>
<td>2 Minor usability problem</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Report view</th>
<th>The user can’t see the whole number when editing the text field where the user should specify street number. The user thinks that he/she may have specified wrong or missed the first digit if the number contains more than one digit.</th>
<th>- Recognition rather than recall</th>
<th>The user can overcome this by deleting and specifying the number again more carefully. The number is visible also when the user stops editing the text field and then the user also can confirm that the number was correctly specified.</th>
<th>Every time the user is located or wants to file a report on an address with number containing two or more digits.</th>
<th>3 Major usability problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report view</td>
<td>The user reacts on the text on the button that makes it possible to either take photo with camera or fetch photo from library. The text says “Hämta bild”, it gives the illusion that its only possible to fetch from library.</td>
<td>- Match between system and real life - Recognition rather than recall. - Aesthetic and minimalistic design</td>
<td>The user can overcome this by tapping the button and get a new opportunity to make a choice. This time two additional buttons appear where the user can choose both to take photo with camera or fetch from library.</td>
<td>The first time the user will attach a photo to the report. The next time the user has learnt that the functionality is behind this button.</td>
<td>2 Minor usability problem</td>
</tr>
<tr>
<td>Start view</td>
<td>The user felt that the Done buttons on the keyboard in both the report form and the personal records form wasn’t intuitive enough. He wanted the button to have the functionality to send the user to the next text field rather than just minimizing it.</td>
<td>- Matching system and real world non-experienced</td>
<td>This is nothing the user can overcome by themselves.</td>
<td>Every time a user will file more than one report.</td>
<td>4 Usability catastrophes</td>
</tr>
<tr>
<td>Report view, personal records view</td>
<td>The user can overcome this easily by just minimizing the keyboard and tapping the next text field or simply ignore to minimize the keyboard and only tapping the next text field.</td>
<td>- Recognition rather than recall non-experienced</td>
<td>Every time a text field will be filled in.</td>
<td>1 Cosmetic problem only</td>
<td></td>
</tr>
<tr>
<td>Report view, personal records view</td>
<td>All information from the previous report is still in all text fields. Should be deleted when the report is filed.</td>
<td>- Error prevention</td>
<td>non-experienced</td>
<td>This is nothing the user can overcome by them selves.</td>
<td>Every time a user wants to file more than one report.</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------</td>
<td>-------------------</td>
<td>-----------------</td>
<td>------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>Report view</td>
<td>The error message given when the GPS functionality didn’t work was unclear to the user.</td>
<td>- Visibility of system status - Help user recover from errors</td>
<td>non-experienced</td>
<td>The user can overcome this problem by tapping the “OK” button on the error message. The result is that the user did understand that something went wrong and not what went wrong.</td>
<td>Every now and then when the Google map service decides not to respond.</td>
</tr>
<tr>
<td>Start view</td>
<td>The user react on that the list doesn’t show the latest report when the user is redirected back to the first view, if the user has scrolled down the list earlier. The list should be shown with its topmost cell every time the user goes back to the view.</td>
<td>- Match between system and real world.</td>
<td>non-experienced</td>
<td>The user can scroll to the topmost cell by them selves and get confirmation that their last report was filed correctly.</td>
<td>Every time a user files a report with the prerequisite that the user has scrolled down the table on the first view earlier.</td>
</tr>
</tbody>
</table>
### Table 4 - Result from participant 4

<table>
<thead>
<tr>
<th>Place of Occurrence</th>
<th>Problem description</th>
<th>Heuristics violated</th>
<th>Specific user consideration</th>
<th>Severity rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error report view</td>
<td>When the user choose category he thought the choice was made when he touched the text in the picker not by clicking the submit button.</td>
<td>-Recognition rather that recall.</td>
<td>The user can easily overcome this problem by using the button that says “Done” and is located next to the picker.</td>
<td>Only the first time the user performs the task in the system. Next time the user have to pick a category it has learnt that it has to use the “Done” button.</td>
</tr>
<tr>
<td>Start page</td>
<td>The user think that it will be possible to tap the different cells in the table. When the user taps a cell it indicates that something will happen and the cell gets blue. But nothing happens.</td>
<td>- Recognition rather that recall</td>
<td>The user can easily overcome this problem by simply avoid tapping the cells in the table.</td>
<td>This happens the first time the user uses the system. After the first time the user encounter this problem it learns that the list isn’t interactive.</td>
</tr>
<tr>
<td>Error report view</td>
<td>When the user choose category he thought the choice was made when he touched the text in the picker not by clicking the submit button.</td>
<td>-Recognition rather that recall.</td>
<td>non-experienced</td>
<td>The user can easily overcome this problem by using the button that says “Done” and is located next to the picker.</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>---------------------------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Start page</td>
<td>The user think that it will be possible to tap the different cells in the table. When the user taps a cell it indicates that something will happen and the cell gets blue. But nothing happens.</td>
<td>-Recognition rather that recall</td>
<td>non-experienced</td>
<td>The user can easily overcome this problem by simply avoid tapping the cells in the table.</td>
</tr>
<tr>
<td>Summary view</td>
<td>If the user wants to file a new report after another report has been filed and more precise the user first wanted to file a report with feedback and then wants to file a report anonymously the person records from the prior report occurs on the summary view.</td>
<td>non-experienced</td>
<td>This is nothing the user can overcome by them selves.</td>
<td>Every time a user will file more than one report.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Error report view, personal record view</td>
<td>Misses an asterisk near the text fields that are mandatory</td>
<td>non-experienced</td>
<td>People will often forget to complete the mandatory fields.</td>
<td>Every time the user tries to leave a specific text field empty that is mandatory.</td>
</tr>
<tr>
<td>Report view</td>
<td>When the user is attaching a photo of the error the text in the buttons is in English but everything else is in Swedish in the system</td>
<td>non-experienced</td>
<td>If the user isn’t familiar with the English language which can be the case it’s impossible to overcome the problem.</td>
<td>Happens every time the user wants to attach a photo to the report.</td>
</tr>
<tr>
<td>Summary view</td>
<td>The describing text is always visible even when the user has choose to be anonymous.</td>
<td>- Aesthetic and minimalism</td>
<td>non-experienced</td>
<td>The user can’t overcome this at all.</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>----------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Summary view</td>
<td>The feedback alert view gives the same message depending on if the user has choose to be anonymous or if he/she wants feedback. It can benefit greatly if the messages are different.</td>
<td>- Match between system and real life</td>
<td>non-experienced</td>
<td>The user can’t overcome this at all</td>
</tr>
<tr>
<td>Error report view, personal record view</td>
<td>The predefined text in the text field isn’t intuitive enough. It should be more of a hint on what the text field should contain. When tapping the text field the first time the hint should disappear but if the user writes something in the text field this text should not disappear if the user taps the text field a second time.</td>
<td>- Flexibility and efficiency to use - Recognition rather than recall - Help and documentation</td>
<td>non-experienced</td>
<td>The user can’t overcome this at all</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Start page</td>
<td>unnecessary loading time when flipping between tabs. More precise when flipping between start page and all other tabs.</td>
<td>- Visibility of system status - Consistency and standards (Ambiguous action)</td>
<td>non-experienced</td>
<td>The user can’t overcome this at all</td>
</tr>
</tbody>
</table>
Part II

Heuristic evaluation result tables for Android application
## E Table 1 - Result from participant 1

<table>
<thead>
<tr>
<th>Place of Occurrence</th>
<th>Problem description</th>
<th>Heuristics violated</th>
<th>Specific user consideration</th>
<th>Severity rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start page</td>
<td>Too many items in the list, hard to find a specific matter in a short time.</td>
<td>Match between system and real world.</td>
<td>The user can overcome this problem by searching in the first 10-20 items in the list.</td>
<td>2 Minor usability problem.</td>
</tr>
<tr>
<td>Service request page</td>
<td>Take too long time to retrieve address from the GPS button.</td>
<td>Visibility of system status</td>
<td>It is possible to overcome this problem by not using the GPS button.</td>
<td>1 Cosmetic problem.</td>
</tr>
<tr>
<td>Service request page</td>
<td>Take too long time to retrieve address from the GPS button.</td>
<td>User control and freedom</td>
<td>It is possible to overcome this problem by not using the GPS button.</td>
<td>1 Cosmetic problem.</td>
</tr>
<tr>
<td>Service request page</td>
<td>Misunderstanding in the language. This appears in the radio button “I don’t want any feedback”</td>
<td>Match between system and real world</td>
<td>This is possible to overcome by learning what the words is meaning.</td>
<td>1 Cosmetic problem</td>
</tr>
<tr>
<td>Service request page</td>
<td>GPS button. The GPS function seems to update when new coordinates are found.</td>
<td>Consistency and standards</td>
<td>This is possible to overcome when not using the GPS button</td>
<td>3 Major usability problem.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>Impact</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>un-experienced</td>
<td>The user can overcome this problem by searching in the first 10-20 items in the list.</td>
<td>Every time the user search after a specific item in the list.</td>
</tr>
<tr>
<td>un-experienced</td>
<td>It is possible to overcome this problem by not using the GPS button.</td>
<td>Every time when the GPS button is pressed.</td>
</tr>
<tr>
<td>un-experienced</td>
<td>It is possible to overcome this problem by not using the GPS button.</td>
<td>Every time when the GPS button is pressed.</td>
</tr>
<tr>
<td>un-experienced</td>
<td>This is possible to overcome by learning what the words is meaning.</td>
<td>Depends how the user perceive the sentence.</td>
</tr>
<tr>
<td>un-experienced</td>
<td>This is possible to overcome when not using the GPS button</td>
<td>This happens rarely.</td>
</tr>
<tr>
<td>Service request page</td>
<td>It is possible to compose letters instead of numbers in the street number text area.</td>
<td>Recognition rather than recall</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Summary page</td>
<td>The finish message dialog shows incomprehensible information to the user</td>
<td>Visibility of system status</td>
</tr>
<tr>
<td>Service request page</td>
<td>It is hard to know which fields that are mandatory.</td>
<td>Match between user and real world</td>
</tr>
<tr>
<td>Service request page</td>
<td>No feedback when street cannot be found</td>
<td>Error prevention</td>
</tr>
<tr>
<td>Service request page</td>
<td>Hard to see different between categories in the spinner</td>
<td>Match between system and real world</td>
</tr>
</tbody>
</table>
### Table 2 - Result from participant 2

<table>
<thead>
<tr>
<th>Place of Occurrence</th>
<th>Problem description</th>
<th>Heuristics violated</th>
<th>Specific user consideration</th>
<th>Severity rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>First page</td>
<td>Too many items in the list, it takes too long time to find the error in the list</td>
<td>Match between system and real world</td>
<td>The user can overcome this problem by searching in the first 10-20 items in the list</td>
<td>Every time a user search after an existing error</td>
</tr>
<tr>
<td>Service request page</td>
<td>The GPS button is lying below the edit text areas for street and street nr</td>
<td>Match between system and real world</td>
<td>The user can easily overcome this problem by pressing the GPS button before editing the edit text areas</td>
<td>Only the first time. The user is able to learn this problem</td>
</tr>
<tr>
<td>Summary page</td>
<td>Feedback labels becomes visible in the summary page when users choose “no feedback”</td>
<td>Consistency and standards</td>
<td>The user can overcome this problem by not looking at this labels</td>
<td>This problem is visible every time the user choose to be anonymous</td>
</tr>
<tr>
<td>Service request page</td>
<td>When selecting address via GPS button: wrong address is then shown in the list at the first page after sending this error</td>
<td>Error prevention</td>
<td>The user can overcome this problem by not using the GPS button</td>
<td>This problem is not common</td>
</tr>
<tr>
<td>Summary page</td>
<td>No feedback when pressing “send” button</td>
<td>Recognition rather than recall</td>
<td>experienced</td>
<td>The user can overcome this problem by pressing the number button in the phone keyboard</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------</td>
<td>-------------------------------</td>
<td>-------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Service request page</td>
<td>Too harsh words in message dialogs</td>
<td>Match between system and real world</td>
<td>experienced</td>
<td>The user can learn to handle this worlds</td>
</tr>
<tr>
<td>Feedback page</td>
<td>It is possible to compose letters instead of numbers in phone text area</td>
<td>Recognition rather than recall</td>
<td>experienced</td>
<td>The user can overcome this problem by pressing the number button the phones keyboard</td>
</tr>
<tr>
<td>Service request page</td>
<td>The application crashes when using the phones camera</td>
<td>Error prevention</td>
<td>experienced</td>
<td>The user can overcome this problem by not using the camera</td>
</tr>
</tbody>
</table>
### Table 3 - Result from participant 3

<table>
<thead>
<tr>
<th>Place of Occurrence</th>
<th>Problem description</th>
<th>Heuristics violated</th>
<th>Specific user consideration</th>
<th>Severity rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>First page</td>
<td>Cannot see the number of a specific street in the list</td>
<td>Match between system and real world</td>
<td>un-experienced: the user can overcome this problem by not search after a specific street number in the list</td>
<td>Every time the user see the list</td>
</tr>
<tr>
<td>First page</td>
<td>Cannot click on an item in the list</td>
<td>Navigation</td>
<td>un-experienced: The user can overcome this problem by not selecting in the list</td>
<td>Every time the user sees the list</td>
</tr>
<tr>
<td>Service request page</td>
<td>The application crashes when using the android in built camera</td>
<td>Error prevention</td>
<td>un-experienced: The user can overcome this problem by not using the camera</td>
<td>This happens rarely</td>
</tr>
<tr>
<td>Summary page</td>
<td>The user accidentally sent three error reports to the web service</td>
<td>Visibility of system status</td>
<td>un-experienced: This could be overcome if status feedback was presented e.g. progress indicator</td>
<td>This happens often</td>
</tr>
<tr>
<td>Service request page</td>
<td>No timer or feedback on the progress bar when pressing GPS button</td>
<td>User control and freedom</td>
<td>un-experienced: The user can overcome this problem if the back button works or with a timer.</td>
<td>This happens rarely</td>
</tr>
</tbody>
</table>
### Table 4 - Result from participant 4

<table>
<thead>
<tr>
<th>Place of Occurrence</th>
<th>Problem description</th>
<th>Heuristics violated</th>
<th>Specific user consideration</th>
<th>Severity rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>First page</td>
<td>It is difficult to know if the Address is already reported</td>
<td>Structure of information</td>
<td>un-experienced</td>
<td>The user can overcome this problem by not looking after a specific street number in the list</td>
</tr>
<tr>
<td>First page</td>
<td>The items in the list are not clickable</td>
<td>Navigation</td>
<td>un-experienced</td>
<td>It is easy to learn this problem</td>
</tr>
<tr>
<td>Service request page</td>
<td>Misunderstanding word “Hämta bild” recognition rather than recall</td>
<td>un-experienced</td>
<td>It is easy to learn this problem</td>
<td>Seem to happen often</td>
</tr>
<tr>
<td>Service request page</td>
<td>The built in camera crash when using it more than once in the same run</td>
<td>Error prevention</td>
<td>un-experienced</td>
<td>The user can easily overcome this problem by not selecting the camera button</td>
</tr>
<tr>
<td>Feedback page</td>
<td>The Edit texts values phone and email can be null</td>
<td>Help users recover from errors</td>
<td>un-experienced</td>
<td>The user can overcome this problem by typing either phone or email</td>
</tr>
</tbody>
</table>