Simple message delivery system

The development process of a simple message delivery add-on to be used in a work environment.

Anders Bea
910425-1815
andbea@kth.se

Aydin Heydari
821130-0192
aydinhb@kth.se
Abstract
A major problem in resource-constrained areas is the high mortality rate among children. Child birth registrations in these areas are non-existent or done using an outdated system, this makes it hard for doctors and nurses working in such areas to get an overview of the health status of children. The lack of information about the health status of children in resource-constrained areas leads to many children not receiving the vaccinations and vitamins needed in order to survive.

Using technology designed to assist the doctors and the nurses gives them the tools to be able to receive the information needed in order to take the necessary steps to prevent high child mortality rate. To improve the health care in a resource-constrained area like Uganda, the non-profit organization Shifo has developed software called “MyChild”, where the nurses in Uganda can register the health status of the children in a database. The software helps the doctors and the nurses keep track of the treatment status of the children and thereby enable them to administer the correct vaccines and vitamins, thereby increasing the children’s survival rate.

This report describes how an add-on was created for the MyChild system. The add-on was developed as a stand-alone application and was primarily designed to be used as a simple message delivery system. Enabling the functionality to deliver work related information from the administration or software related updates from the developers to the nurses, through the MyChild system.

The add-on was developed by using selective parts of the project development method, Scrum, which was adapted for the use of a smaller group consisting of two developers. With Sprint, tasks and continuous updates through meetings, the project allowed for active change in the goal outline. Using a semi structured interview method, feedback about the add-on was received and the answers from the interviews could then be analyzed, thereby inspiring further advance of the add-on.

The result from the presented work is a first version prototype that can send messages from a database to be shown in a rectangular window along with a corresponding picture. The prototype is a stand-alone application that can be modified to show the messages with different corresponding pictures and during different time durations. The main functionality of the first version prototype fulfills the requirements set by the customer, Shifo, but its implementation into the Shifo system is not presented in this thesis.

The study through interviews presented in this thesis show that the message delivery add-on could be a useful addition, not only in health care, but also in other professions. Its functionality gives the possibility for users to receive work related information without it being a distraction or a hindrance in their daily work.
**Abstract**

Ett stort problem i resursbegränsade områden är de höga dödsifforna bland barn. Registreringar av nyfödda i dessa områden är obefintliga eller gjorda med uråldrade och omoderna metoder. Det gör det svårt för läkare och sjuksköterskor att få en korrekt helhetsbild av hälsotillståndet bland barn i området som i sin tur leder många barn inte får de vaccin eller vitaminer som de behöver för att kunna överleva.

För att hjälpa läkare och sjuksköterskor förhindra dödsantalet bland barn i resursbegränsade områden skapas verktyg med hjälp av teknologin. Den ideella organisationen Shifo har därför utvecklat en mjukvara som ska förbättra hälsovården i resursbegränsade länder som Uganda. Mjukvaran heter MyChild och genom den ska sjuksköterskor i Uganda i en databas kunna registrera nyfödda barn samt ta del av barnens hälsoinformation. Genom mjukvaran kommer sjuksköterskorna ha möjligheten att se hälsotillståndet för varje barn och på så sätt kunna ge dem de vaccinationer och vitaminer som de behöver för att kunna överleva.


Applikationstilläget utvecklades genom att använda selektiva delar av arbetsmetoden för mjukvaruprojekt, Scrum, som blivit anpassade för användningen i en mindre grupp på två medlemmar. Med hjälp av ”sprints”, ”tasks” och kontinuerliga informations uppdateringar genom möten lät sig projekets mål aktivt formas utefter arbetets gång. För att bilda sig en uppfattning om användarnas åsikter om applikationstillägget och hur den kunde förbättras, gjordes analyser från svaren av semi strukturerade intervjuer.


Arbetet som presenteras i den här uppsatsen visar att meddelandetillägget kan vara ett användbart tillägg i även andra branscher än endast inom hälsovård. Funktionaliteten för applikationstillägget ger användarna möjligheten att ta emot arbetsrelaterad information utan att störas eller bli hindrade i det dagliga arbetet.
### Contents

**1 Introduction**

1.1 Background ................................................................. 1

1.2 Problem ........................................................................... 2

1.3 Purpose ............................................................................ 2

1.4 Goal, Benefits, Sustainability and Ethics .............................. 3

1.5 Methodology ..................................................................... 3

1.6 Delimitations .................................................................... 5

1.7 Outline ............................................................................. 6

**2 Theoretical Background**

2.1 Shifo ................................................................................ 7

2.2 The Shifo systems ............................................................. 7

2.2.1 The MyChild desktop .................................................. 8

2.3 Agile ............................................................................... 11

2.3.1 Individuals and Interaction .......................................... 12

2.3.2 Working software ........................................................ 12

2.3.3 Client collaboration ...................................................... 12

2.3.4 Responding to change .................................................. 13

2.3.5 Scrum .......................................................................... 13

2.3.6 Tasks .......................................................................... 14

2.3.7 The board ..................................................................... 14

2.4 Software ........................................................................... 14

2.5 Add-on ............................................................................ 15

2.6 Usability test ..................................................................... 16

2.7 Usability test analysis ....................................................... 16

**3 Method**

3.1 Agile ............................................................................... 17

3.1.1 Scrum .......................................................................... 17

3.1.2 Tasks and task board ................................................... 18

3.2 Software development tools .............................................. 19

3.2.1 Database ...................................................................... 19

3.2.2 Sublime Text ............................................................... 19

3.2.3 Jira ............................................................................ 19
3.3 Interview methods ........................................................................................................... 19
3.3.1 Structured interviews ................................................................................................. 20
3.3.2 Unstructured interviews ............................................................................................. 20
3.3.3 Semi structured interviews ......................................................................................... 20
3.3.4 Choice of interview method ....................................................................................... 20
3.4 Usability test ................................................................................................................... 21

4 The Add-on .......................................................................................................................... 22
4.1 Project steps ...................................................................................................................... 22
4.2 Message delivery ............................................................................................................ 23
4.3 Interface .......................................................................................................................... 23
4.4 Database .......................................................................................................................... 25

5 Usability test ....................................................................................................................... 28
5.1 Usability test interviews ................................................................................................ 28
5.2 Usability test answers .................................................................................................... 29
5.3 Usability test analysis ..................................................................................................... 34
  5.3.1 Method of delivery ...................................................................................................... 34
  5.3.2 Message content ......................................................................................................... 34
  5.3.2 Distraction .................................................................................................................. 34
  5.3.2 Questionnaire ............................................................................................................ 35

6 Results .................................................................................................................................. 36
6.1 Project comparisons ........................................................................................................ 36
6.2 Work methods ................................................................................................................ 36
6.3 Add-on .............................................................................................................................. 36
6.4 Usability test .................................................................................................................... 37

7 Conclusions and further improvements ............................................................................. 38
7.1 Conclusions .................................................................................................................... 38
7.2 Discussion ....................................................................................................................... 39
7.3 Evaluation ....................................................................................................................... 40
7.4 Further improvements .................................................................................................... 42
7.5 Continuing the work ...................................................................................................... 42

References ............................................................................................................................ 44

Appendix - Transcripts ......................................................................................................... 49
  Transcript 1 - Human resource consultant ........................................................................ 49
1 Introduction
The possibility of sending messages over vast distances as well as to a larger group of people simultaneously, is a function rarely praised by the modern population. When the Internet was first introduced the first function was to send messages, which previously was only possible by telephone. As email continued to thrive, work environments continued to adapt accordingly as well as the private life of the average citizen.

As technology continues to develop, new methods of sending messages are being introduced and one of these are the integrated message system in one’s direct work area. By communicating information in an integrated method such as daily messages when logging in or during less stressful periods of times one could possibly increase the staff’s motivation towards the rest of the work day. This method may also be applied in other areas, such as to improve both efficiency and understanding, by simply improving upon the content or interactivity of the messages.

In this part, the background of the thesis, namely the implementation of an integrated message delivery system that aims to inform and increase the motivation and efficiency of the staff using the developed system.

1.1 Background
Shifo, a non-profit organization, consists of a group of people fighting to improve the health care system in Uganda. Specifically they are focusing on decreasing the child mortality rate, caused to be high because of preventable diseases. These diseases are, with the modern day medicine, curable with simple vaccines that are otherwise distributed in countries with a more developed health care system.²

In order to fight the vaccination problem, Shifo decided creating a more advanced and secure vaccination registration system. Comparing the development to the previous system that consisted of a physical medium, namely paper, that can be easily lost or misplaced and was also hard to share when families moved, the new system that Shifo created, would enable the possibility of moving without concern for earlier documentation of treatment, such as vaccination. With the collective database, a record of child birth registrations at treatment facilities would be kept by default which would enable a better birth statistics for the government.³

Nurses will use the registration system whilst treating children in different areas in

³ Shifo (2014).http://shifo.org/shifosystems/ (Downloaded 2014-02-29)
Uganda. In order to facilitate the communication with the nurses, a message delivery add-on will be integrated into the registration system. The add-on will send useful work-related information to the nurses and make them aware of problems and/or diseases in the areas that they are visiting.

1.2 Problem
Today the information can be distributed through various devices, such as cellphones, tablets or computers. The delivery application can also vary in its methods, such as an email, that are delivered to individuals through a specific address or websites which present the information to anyone who enters the website. Nurses in Sweden work in a stressful environment\(^4\), where delivery of information could become distracting. Implementing a message delivery system into a work-related software would have to consider the stressful environment and avoid becoming distracting. Receiving information through the add-on might become a distraction for the nurses as they have to observant about important information and it might remove attention from their daily duties.

By implementing the system within existing software one does not only risk decreasing the usability of the already functioning program, but also a possible decrease in the motivation or efficiency. This would thereby cause a negative effect on the goal.

Introducing a message delivery system in a work environment has to be done in order to not make it a disturbance or an annoyance for the employees. The message delivery system, once finished, is to be integrated to keep the work efficiency level as high as it was pre-implementation and not make the employees less productive in their daily work duties.

How can the creation of a simple message delivery system, integrated into existing software, be useful and done without it becoming a hindrance for the employee?

1.3 Purpose
The purpose of this thesis is to explain the process of developing an integrated message delivery system that is easy to use and delivers information to multiple users. The thesis also explains the methods and tools that can be used in order to develop an integrated message delivery add-on. The presented work evaluates if the message delivery add-on is useful and how it can be developed further in order to improve its usefulness. This will give the possibility to extract the project process and build upon it for a better result.

In a work environmental sense it will improve communication and give possibility for

---

the users to receive useful information in a more direct manner. The users will be able to receive information while working with the application and will be able to read the information directly within the application. The study for continued improvement upon this system during the various development stages is to enable possible applications for projects within software development areas.

This thesis is an opportunity for further advancement within the area of integrated message delivery systems in software. The continued development of such a system will be able to determine a possible improvement upon the method of working on a software project. It will also be a way of improving upon the system for mitigating the product related information.

1.4 Goal, Benefits, Sustainability and Ethics
The goal of the thesis is to develop a prototype of an integrated message delivery system that will improve and facilitate sending information to the users. From the perspective of hospital organizations it can lead to a more effective and improved way of communication between the administrational staff and the nurses.

The add-on would aim to help the transition from a paper based system to a computerized system and also as a tool to send messages to the users in an easy and effective way. Therefore its usage will not get outdated when the transition is done because giving information to all the users of the software is something that will continue as long as the system is in use. If successful, the add-on with its software can be used in other countries in need of better systems.

Using a message delivery add-on that can give information to the user directly from the software they are using, saves up time for the user and also reduces the resources used by the computer.

The information sent through the message delivery add-on is not to contain private or sensitive information since it could invade the privacy of the user and would thereby decrease the reliability in the application and trust of the users with the administration.

The message delivery add-on is an addition that aims to make the communication within a company more direct and efficient. If implemented correctly, it can replace the need of sending e-mail to multiple users and instead give the possibility to send information directly to the employees, thereby not forcing them into using a different application in order to receive the information.

1.5 Methodology
The working method is split into four different areas. It starts with an explanation of the
theories for methods used in this thesis, continues with explanation of the methods used to develop and to test the prototype of the message delivery add-on presented in this thesis. After the methods are explained, the implementation of the message delivery add-on prototype will be presented. Finally a usability test will be made to understand the usefulness of the message delivery add-on in different professions and to find out about what improvements can be made to it. The theories consist of research the software development method used in this project, the agile software development method. It will also consist of research about interviewing methods that can be used for making a usability test. The third part consists of explaining the methods and how they could be implemented in this thesis. The fourth part is about implementing the methods, based on the knowledge that was gained through the theories, which in turn will lead to a first version prototype being produced. When the implementation is done an evaluation of the add-on will be made. Based on the feedback from the users, future considerations on improvements of the add-on will be made.

The various methods for a thesis are based upon either the inductive or deductive approach. An inductive method is a method where knowledge is generated through observation. Specific observations and measures are primarily made and from them patterns and regularities are detected. From the pattern and regularities, explore able hypothesis are made and these are used to develop some general conclusions and theories. A deductive method is a method concentrating on confirming a hypothesis. A theory about a topic is thought up primarily, the theory then gets narrowed down to a specific hypothesis. To address the hypothesis, observations are made in order to narrow down the hypothesis even further. The observations will make it possible to test the hypothesis that in turn will be able to confirm or not confirm the theory that was thought up in the beginning.

In this thesis a deductive approach will be taken. The project presented in this thesis will prove the usefulness of a message delivery system in a work environment, which in turn will lead to a more useful and appreciated add-on according to its users.

A quantitative research method is a research method where the numerical relationship between two or several measurable attributes. It is used to quantify and generalize results from a sample to the population of interest. It is used to prove, explain and

---

7 Idib.
describe bearings10.

Continuing on to the research method, the two approaches that can be used are quantitative and qualitative. Qualitative research method is a method that is characterized by trying to find out an understanding for the worldview of an individual or a group of individuals. The material that is analyzed is collected by interviews or by observation11.
A qualitative research method gives an understanding of underlying reasons and motivations, and to uncover current trends in thought and opinion.12

In the project presented in this thesis a qualitative method is used since data collected will be non-statistical, consciously chosen, without a predetermined size13 and will be used to gain an initial understanding and a sound base for future decision making14. This approach will therefore give the possibility to gain a better understanding of opinion about the add-on and its functionality from different types of users.

1.6 Delimitations
The focus on the add-on will be on its functionality and therefore the graphical part of the add-on is not prioritized this thesis. The graphical part is otherwise an important aspect that would have improved the add-on and made the user experience better.

Since the Shifo software is not fully developed yet, the message delivery add-on will be developed as a standalone version and its integration into the Shifo software will not be presented in this thesis.

The testing and evaluation of the add-on presented in this thesis will include employers from different fields and not only nurses, this will make it possible to evaluate if the add-on could be useful multiple branches of trade and not only in health care.

A common problem in software development is the delay and postponement of project deadlines. This could in turn be caused by the very project development method. By utilizing the “Agile” method, one would not only be able to improve upon the project continuously during the ongoing project but also be able to reach predetermined minor deadlines created by one as the software developer.

1.7 Outline
Chapter 2 presents information about Shifo and their application. A theoretical background is also presented in this chapter. The third chapter of the thesis contains a brief explanation of different methods that are used in this thesis and a more detailed explanation of the software developing method that is used in this project. Chapter 4 describes how the add-on was developed and the various steps that were taken in order to finish the product. The results from the interviews of the usability test are presented in chapter 5. The sixth chapter demonstrates the conclusions made from the usability test. A discussion about the project and its application is presented in chapter 7. These are the most directly associated questions related to the development and implementation of the add-on. A brief summary of the thesis and conclusions of the project is described in chapter 8.
2 Theoretical Background
This chapter will explain the background information of the project presented in this thesis. First it will give an explanation about the non-profit organization Shifo, the organization behind the idea and the developer of the software were the add-on will be used. The second part presents Shifo’s system into which the final version of the add-on would be integrated. Thereafter a software development method is presented. The fourth part will give a detailed explanation about software that delivers a user related services and how they are relevant for the project that is presented in this thesis. The chapter will give an explanation of the how an add-on can increase the usability of software in general and examples of add-ons integrated on software of today. Lastly the interview and its function, as well as the method for the analysis, are presented.

2.1 Shifo
In 2013 the nonprofit organization, Shifo, was launched by a team from KTH, Karolinska Institute and Karolinska University hospital to develop tools and methods that can improve and facilitate child birth registrations and also the follow up on child vaccinations.\textsuperscript{15} Lack of child birth registrations is a widespread problem around the world. Without the registrations, it is hard to know which child has been given what vaccine or whether or not it has been given the right number of vaccines. This leads to an increase in the child mortality rate.\textsuperscript{16} Until recently, the implementations of children registrations were based on a paper-trail which can be physically lost during transfer and hard to check up upon. Therefore, Shifo decided to step in and help implement a computer based system that will save all the information about the children and their health status on a secure database.\textsuperscript{17}

2.2 The Shifo systems
The Shifo systems are a number of free of charge applications that are specifically designed to support delivery of health care in resource -constrained settings. Part of the Shifo systems is the MyChild system. The MyChild system is designed to address issues in vaccination service delivery in underserved areas and it was developed in order to enable health personal to register and follow up vaccination status of each child and ensure that vaccination plan is completed. The MyChild system consists of MyChild desktop, MyChild Statistics and MyChild mobile. The add-on presented in this thesis will only be implemented on the MyChild desktop.\textsuperscript{18}

\textsuperscript{18} Shifo (2014). http://shifo.org/shifosystems/ (Downloaded 2014-02-29)
2.2.1 The MyChild desktop

This subchapter presents the MyChild desktop. The MyChild desktop client is software application that stores the health information of children in Uganda in a database. The software has been developed to give the nurses the correct information about the health status of the children in the area. The MyChild software stores the information of what vaccines and what vitamins a child have been given and also information about what diseases a child has. The thesis only presents a mockup of the MyChild software. The thesis does not present the real software because the MyChild software was under development at the time the thesis was written.

Picture 2.2.1a presents the main window of the software. This is the window the user sees when opening the program.

![Main window of MyChild software](image)

**Picture 2.2.1.a:** This picture represents a mockup of the main window of the MyChild software. This part of the program is used when a child is added to the database and when the user is searching for a child that is already registered in the database.

1. This part of the main window is used when a child is not yet registered and has to be added to the database. The personal information about the child is
written in the corresponding squares and the residential information is chosen in the drop down menus underneath the name forms. To register the child into the database, the “register” button is pressed.

2. The search bar is used search for already registered children. The name or the ID of the child is written in the search bar and when pressing search button, the ID, the name, the gender, the contact person and the place of residence of the child appears in the window below the search bar.

3. The third box represents where the search results are shown. When an already registered child has been searched for in the search bar, the ID, name, gender, contact person and residential area of the child will show up in this part of the window. The “start visit” button is clicked in order to see and to add information about the child.

Clicking the “start visit” button will take the user from the main window to the personal health information window. The personal health information window is presented in picture 2.2.1b.

---

<table>
<thead>
<tr>
<th>ID: 3434-123457</th>
<th>Steve Jobs</th>
<th>6 weeks</th>
<th>Set biodata</th>
<th>Sign visit</th>
</tr>
</thead>
</table>

### Personal Health Information Window

#### Child Growth

<table>
<thead>
<tr>
<th>Weight (cm)</th>
<th>Height (cm)</th>
<th>Head Circ. (cm)</th>
<th>Date</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Age at birth</th>
<th>Weight at birth</th>
<th>Height</th>
<th>Head Circ.</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>50</td>
<td>10</td>
<td>70</td>
<td>30</td>
<td>2013.01.15</td>
</tr>
</tbody>
</table>

### Immunization

<table>
<thead>
<tr>
<th>Vaccine</th>
<th>Date Given</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCG</td>
<td>2013.01.15</td>
</tr>
<tr>
<td>Polio 0</td>
<td>Give 1</td>
</tr>
<tr>
<td>Polio 1</td>
<td>Give 1</td>
</tr>
<tr>
<td>OPT-HevB-Hib 1</td>
<td>Give 1</td>
</tr>
<tr>
<td>Polio 2</td>
<td>Give (12/12/2013)</td>
</tr>
<tr>
<td>OPT-HevB-Hib 2</td>
<td>Give (12/12/2013)</td>
</tr>
<tr>
<td>Polio 3</td>
<td>Give (12/12/2013)</td>
</tr>
<tr>
<td>OPT-HevB-Hib 3</td>
<td>Give (12/12/2013)</td>
</tr>
<tr>
<td>Measles</td>
<td>Give (12/12/2013)</td>
</tr>
</tbody>
</table>

### Vitamin A & Deworming

<table>
<thead>
<tr>
<th>Vaccine</th>
<th>Date Given</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vitamin A</td>
<td>2013.01.15</td>
</tr>
<tr>
<td>Under 6 months</td>
<td>Give 1</td>
</tr>
<tr>
<td>Give 1 Already</td>
<td>Give 1</td>
</tr>
<tr>
<td>3 months</td>
<td>Give (12/12/2013)</td>
</tr>
<tr>
<td>Give 1</td>
<td>Give (12/12/2013)</td>
</tr>
<tr>
<td>6 months</td>
<td>Give (12/12/2013)</td>
</tr>
<tr>
<td>Give 1</td>
<td>Give (12/12/2013)</td>
</tr>
<tr>
<td>9 months</td>
<td>Give (12/12/2013)</td>
</tr>
</tbody>
</table>

### Picture 2.2.1b

*Picture 2.2.1b presents the personal health information window. Here information about the health status of a child can be seen and added.*

4. This part of the window shows information about the vaccines and the vitamins that a child has been given and also what vaccines and vitamins that needs to be given. In the window on the left, information about the vaccines is
stated. The information shows the age of the child of when receiving a vaccine and also if the vaccine have already been given or if it will be given in the future. The window to the right shows the information about the vitamins that a child has or will receive along with the age of the child.

5. On the top of the window information about the weight and height progress of the child is added.

6. The left part of the window shows additional information about the child, such as information about symptoms or results from the administered vaccines.

7. The “sign visit” button, highlighted in picture 2.2.1c with the number seven next to it, is pressed when a visit has ended and when the user wants to save the newly added information into the database. Pressing the “sign visit” button takes the user back to the main window presented in picture 2.2.1a.

8. Pressing the “Edit Biodata” button takes the user to a third window. The window is the Edit Biodata window and is presented in picture 2.2.1c.

*Picture 2.2.1c presents the Edit Biodata window. In this part of the software the information about the mother and father of a child is added.*

9. The Edit Biodata window shows information about the mother and father of the child. If the name, the date of birth and the residence district of the relative of the child has to be added, it is added in the biodata window presented in picture 2.2.1c. Pressing the “save” button will save the
information and pressing the “close” button will close the Edit Biodata window and take the user back to the personal health information window presented in picture 2.2.1b.

2.3 Agile
Agile is a software development method were the customer gives input and feedback during the development. The software is developed by small teams and is continuously improved and tested based on the input of the customer. Agile software development is designed for projects that are short term and is a software development method were you can build quality software faster. It is a method with less documentation and more focus on developing a fully functional software. Compared to other development methods, such as the Waterfall method where the project stages are separated into various categories, Agile focuses on separate functionalities of the project, which is called the incremental method. The categories in the Waterfall method are completed in a predefined order, although once one category reaches part completion, another is usually started for a more effective resource management. In an Agile software development method the development is done in several iterations were each iteration consist of design, development, design and testing. The agile work method includes various project methods, some of which are Extreme Programing, Dynamic software development method (DSDM) and Scrum. The first focuses on 12 steps that are derived from the best practices in development, one such step is the peer programming. The second is concentrated around three steps in the development phase, namely the pre-project, project life cycle and post-project. It results in a high quality delivery before the product is implemented, thereafter concentrating on improvement and testing. In the coming subchapter the general points of agile development are explained and lastly the Scrum method will be described in chapter 2.3.5.

24 ibid. p. 67-68.
2.3.1 Individuals and Interaction

Agile software development methods focus on effective and clear communication\(^\text{28}\). The development team is encouraged to communicate within the team about the progress and the development of their daily tasks\(^\text{29}\), but communication with stakeholders is also frequent\(^\text{30}\). Agile software development consists of tools and processes to facilitate communication, such tools are task boards where the tasks are written down and briefly explained. Based on their progress, they are put in different categories.\(^\text{31}\) A daily meeting between the development team and the project manager is a common feature in an agile software development method. In the daily meetings the project manager asks each member of the development team of their progress and each developer explains the progress that has been made with the daily task and if any problems have occurred during development.\(^\text{32}\)

2.3.2 Working software

Agile software development encourages less documentation and more towards development of working software. The development is done in frequent intervals and at the end of each interval, instead of a documentation of the progress, a small piece of working software is presented to the customer.\(^\text{33}\)

2.3.3 Client collaboration

An important part of an agile software development method is client collaboration. The customer is involved in the development throughout its duration and after each interval, the customer gives feedback about the progress made during the previous interval. The customer can sometimes alter the requirements during the development, the development team then has to adapt to the changes and meet with the new requirements of the customer.\(^\text{34}\)

---


\(^{34}\) Ibid.
2.3.4 Responding to change
Agile software development methods are designed to adapt to changes in requirements. Requirements made before the development process was initiated tend to be altered during development and with review and retrospective processes, agile software development is designed to adapt to changes based on customer feedback or business value.35

2.3.5 Scrum
Scrum is a process framework within which small teams can design and sustain complex products in order to deliver the highest value effectively36. In the Scrum framework there are four key roles, the stakeholders, the product owner, the scrum master and the scrum team. The stakeholder decides what should be built and how. The stakeholder works closely with the product owner. The product owner decides what tasks are more important in order to develop workable software and makes a priority list of all the tasks. The scrum master then has the responsibility to make sure that the priority listed tasks done by the product owner are made into workable subtasks. Lastly there is the scrum team, which consists of developers and a scrum master and their role is to develop the workable subtask into fully working software.37

The development is split into sprints where every sprint is 30 days long and by the end of every sprint working software has to be delivered. The product owner decides if the work achieved in every sprint is acceptable38. Every day there is a scrum meeting where everyone involved in the project can say what they are working on and if anyone encounters problems they can ask for help from the other team members. Once everyone have completed their tasks the project team have a meeting to agree on the next development stage and also share their opinions on the tasks that they have just completed. This cycle is repeated until a specific project is completed.39

This method of working has several advantages. Using other project development methods can take months before issues with the software is acknowledged, but the daily scrum meetings makes it possible to acknowledge issues at an early stage.40

---

2.3.6 Tasks
A task consists of the priority, expected time consumption, a title that summarizes the task and a description of the task\textsuperscript{41}. The tasks should be estimated not to exceed 12 hours. If a task is estimated to take longer than 12 hours it is recommended to split it into smaller tasks.\textsuperscript{42}

2.3.7 The board
The tasks are placed on the board in order to get an easier overview. The board is sectioned into 4 parts with which the work is organized.

- “To be done”: This category is where all the tasks that have to be done are placed. For a simpler overview of this category the organization of the importance of the task is chosen by placing the tasks higher up to represent high priority and lower down for low priority.
- “Work in process”: These are the tasks that are being worked upon. During the programming stages situations where a task can’t be completed without another task to be started or finished can be encountered. Therefore this category also contains the various tasks that are pending. The tasks that are pending would be placed further down to signify that the top most tasks are actively being worked upon.
- “To verify “: This is the category were the tasks that have to be verified again are placed. These are usually tasks were problems have been detected and now have to be verified again in order to see of the problem have been solved.
- “Done”: As the name of the category suggests, this part contains all the tasks that have been finished.

To simplify the task completion logging, the name of the person who finished the task, is written on to the back of the tasks in order for quick identification of who completed it.\textsuperscript{43}

2.4 Software
With various improvements of existing software that delivers a user related service, daily information and other everyday messages are being deployed to improve user experience at a subtle level. Looking at various applications today one can see the frequent use of message delivery systems in programs ultimately designed for other uses. Two examples of such systems are Facebook\textsuperscript{44} and Steam\textsuperscript{45}.

\textsuperscript{44} Facebook. https://www.facebook.com/. (Downloaded 2014-02.24).
The first of these examples, namely Facebook, is designed to share one’s pictures and information on a frequent basis, with an integrated message system\textsuperscript{46}. The message system enables “chatting” between users that otherwise could be handled by another dedicated application. By implementing such a system, the user frequency and experience is overall improved. Although compared to the estimated result of the project in this thesis, the difference in systems is the possibility for a discussion in the shape of a dialog. The message system developed in this thesis would implement a monolog that can be developed into a dialog through external means. An example would be face to face dialogs or even emails. The limitation is applied as a safeguard for the risk of increased cost of maintenance, such as server capabilities or consumer help desk support. By simply informing a user of general information, the user is given the choice of furthering their knowledge dependent on their personal disposition or the information importance.

The second example is the computer based game service provider Steam\textsuperscript{47}. Steam provides software that allows the user to browse, install and buy games. When initiating the software on the computer a simple daily message is shown. The message is designed to increase sale by informing the user of new games, sales or other game related information. It allows the user to browse, at least one, possible addition to their private game library\textsuperscript{48}. It presents information otherwise unnecessary and surplus, but with a clear relation to the users goal and reason for their use of the service. Comparing the example of Steam’s daily message to this project, a dissimilarity is clearly present, namely the character of the delivery. The add-on ordered by Shifo only presents a message during the active usage of the application and does not inhibit or force surplus activity from the user. By utilizing an active disabling of the message one ensures the users notability of the information but also increases the risk of decrease in user satisfaction. Since the add-on, which is being developed within this project, which in term will be contained within the otherwise standard program window, it would not intrude on or disable the user from other activities during vaccine registrations.

2.5 Add-on
Software like Microsoft Word\textsuperscript{52} or the development environment Eclipse\textsuperscript{53} facilitate add-on or plugin\textsuperscript{54}, which is a type of add-on, functionality. The user is free to add their desired function otherwise left out of the original product. With a simple install the functionality is added and the user is free to choose whether to use or remove it. A well-

\textsuperscript{46}Wiseman, J. (2008-04-06). \textit{Now we are talking.} https://www.facebook.com/notes/facebook/facebook-chat-now-were-talking/12911122130 (Downloaded 2014-02-23).
\textsuperscript{48}Ibid.
developed system may apply the functionality to enable the user to add and remove multiple add-ons, or otherwise also known as extensions or plugins, such as Microsoft Word, Eclipse or WordPress. The knowledge about add-ons in the simple message delivery system, presented in this thesis, is derived from a tutorial about Chrome application development and WordPress plugin development introduction.

The add-on presented in this thesis will work as a message delivery tool that can facilitate the communication between the users of the system. It will be possible to send information to a specific user or to a group of users. This also ensures a more localized message delivery for users within a certain hospital or location. Although the add-on is added by the developing group it will be easily ignored if it proves to become unnecessary or disruptive. It will be developed to fit the existing software in the sense that it only adds functionality and utilizes existing data for its message assortment.

2.6 Usability test
In order to test whether various functions increase or decrease the usability of the system, a usability test is performed. The usability test will be done by holding interviews, in order to learn more about the possible problems that can occur when integrating an add-on as described in this thesis. The basis for the interview is to gauge an empirical ground for the perceived basis of the add-on. In theory the interviews will possibly add new areas for research and further development. Interviews are a common practice and can be found in many various lines of work. Examples may entail companies seeking the opinions of a new product, or the police aiming to uncover the true story behind an incident. The most common area of use when discussing interviews is the academic one. Scientists who are researching subject that cannot always be uncovered through microscopes or trial and error, such as areas of history, where people may still have recollection of the actual time periods, need to use interviews to further their knowledge base.

Therefore the practice of interviews will be incorporated into this thesis and the method that will be used, will be described in the following chapter.

2.7 Usability test analysis
One method for analyzing interviews is the ideal type centered analysis. The basis for this type is to focus on various attributes, such as the interviewed subjects body language, attitude, relationship, or even experience towards the topic in question. It allows the interviewer to focus on various points of interest in the answers and ignore other parts, which seem unrelated. The source that is compared with the information gained from the interviewed subjects is the knowledge the researcher has gained.

---

57 WordPress. [http://codex.wordpress.org/Writing_a_Plugin](http://codex.wordpress.org/Writing_a_Plugin). (Downloaded 2014-02-24).
through working on the topic in question. Through investigating patterns in the answers and taking note of the possible correlation, the analysis is made.  

3 Method

This chapter describes methods that have been used within the projects, the tools for developing the add-on and how the goals of the project have been met. First an explanation will be presented on how the Agile software development method was implemented and the chapter part will continue on to one of its branch methods, Scrum. In chapter 3.2 a general explanation of the software tools used in this thesis will be given. The last parts of this chapter will present the interview methods and the usability test.

3.1 Agile

Using an Agile based project development method will make sure that the add-on presented in this thesis meets the requirements and expectations of the customer. With the continuous iterations that enable a fluent improvement of the project plan, the end result may not completely contain all of the original goal points, but instead more improved points than expected prior to the project. The continuous dialogue with the customer assures that when the message delivery add-on is developed, it will be developed towards the satisfaction of the customer, namely Shifo. The continuous involvement of the customer will assure that the message delivery add-on presented in this thesis will have the usability that the customer expects from it. The software development method used in this thesis was primarily an adaption of an agile method, namely Scrum. The reason for the use of Scrum is the project organization system, Jira, which delivers an Agile based project development solution. Further discussion of the system will be presented in chapter 3.2.3.

3.1.1 Scrum

During this project certain aspects of the Scrum method will not be used, because of the project group. Since the project only consists of two members, the roles of the Scrum master and team will be filled by the two group members and the role of the product

59 ibid. ch. 8. “Analys och Syntes”.
master will partly be filled by the customer. The choice of using Scrum in order to develop the add-on presented in this thesis is to ensure that the add-on would be finished as quickly as possible and to be able to adapt if the customer changes the requirements during the development. Using a product backlog, were tasks are prioritized, will ensure that the most important tasks of the add-on presented in this thesis will be developed first. The software will be developed in sprints, which were described in chapter 2.3.1. This approach will keep the customer, Shifo, informed about the progress and also be able to give feedback during the development of the message delivery add-on. The sprints will be consisting of up to 20 days. By having meetings, or so called Sprint planning meetings, with the group discussion will be held around the following weeks of work. The sprint planning meetings will take place by assigning tasks from the priority list to each member of the development team. Before the meeting ends, the members of the development team, thereby, know their assignment during the sprint and how to execute it. During the sprint, everyday meetings of 15 minutes will be kept in order to discuss the work done the day before and the work to be done after the meeting. Having daily meeting will help the development team to acknowledge problem in the development of a task and to inform about its progress. This will allow the members to ask for help within the developing team and also make it less probable that the problems occurring during the development of a specific task do not stand in the way of the overall development progress of the add-on that is presented in this thesis.

3.1.2 Tasks and task board
The Sprint tasks as presented in chapter 2.3.2 will be used in the same fashion as the original Scrum project method. Although only two members will work on the development and utilize the tasks, it presents the developers with the motivation and traceability by presenting the performed tasks. The task board will be used in this thesis in combination with the tasks. Using a task board when developing the add-on presented in this thesis will make the progress of the development visible to the developers and give a better overview of the development in general.
3.2 Software development tools
For the development of the software a database, based upon MySQL\(^{85}\), the programming environment Sublime Text and the project organization system Jira that was provided by the customer, Shifo, were used. The various programming languages that were used were HTTP, JavaScript, MySQL and PHP.

3.2.1 Database
The database was required for the integration into the existing system developed by Shifo. During the development of the untested add-on a separate database was created on a privately owned phpmyadmin server. The variables, of which the database consisted of, were created for their need. Through the information given by Shifo, namely that the messages can be location based and that they can also be combined with a photo, led to the choice of creating the 13 variables that are presented in chapter 4.4. Since the database was created specifically for the first version it would have to be adapted to new changes made in later versions, that aren’t covered in this thesis.

3.2.2 Sublime Text
Used for its simplicity and it’s easy to use overview of code, the tool provided a fast and efficient method of coding. Although possible better programming environments exist, this was the one most known through earlier experience in development projects and tasks.\(^{86}\)

3.2.3 Jira
Since an agile method of development was chosen a task dedicated system was required in order to keep track of the continuous stages and tasks. Through meetings and discussions with the customer, system used by the customer was suggested and integrated into the thesis work. The system is task and project orientated and is developed to support larger groups in their task handling and project development.\(^{87}\)

3.3 Interview methods
Depending on the purpose of the interview it can be done structured, semi structured or unstructured\(^{94}\). The questions for the interviews can be both open and closed, where closed questions are best suited for interviews were the answers are known in advance\(^{95}\). Since the interviews are aimed at gaining information from people who work consistently with computers, the interview can be perceived as focus group orientated\(^{96}\).

\(^{95}\) Idib. p.233.
3.3.1 Structured interviews
During structured interviews the interviewer asks predetermined questions, which in term allow the questions to be categorized. The answers are then analyzed in a quantitative manner. The structures of the interviews are similar to questionnaires. Since the interviews are structured in this fashion, all the subjects for interviewing will be asked the same questions and thereby resulting in an overall simple answer base, this may end up hindering the researcher from gaining information outside of the question guide. The limitation of focusing on specific questions is sometimes overlooked, in the goal to achieve the simpler structure of answers and responses, but most commonly a mistake made by scientists who are too engrained in their line of research.

3.3.2 Unstructured interviews
In unstructured interviews the questions are not standardized. This is useful when one wants to receive a longer explanation about the view or opinion about an added function or the design of an application. The interviewer maintains an unstructured interview by simply speaking about what the interviewed individuals talk about. This method allows the interviewed people to feel more comfortable and opens up more easily. The unstructured methods of interviews are mostly used for its explorative measures and are often used for project that are being initialized or when the topic concerns a sensitive matter.

3.3.3 Semi structured interviews
Semi structured interviews consist of both pre-defined questions and follow up questions that will be asked depending on the answers received. The interviewer will ask all the participants the same questions, although in order to receive more feedback from the participants, follow up questions will be asked until the answers give no more relevant information. This is a useful method to receive as much information as possible from a participant. The questions are based around the certain topics that the interviewer wishes to research, this method can be described as deductive.

3.3.4 Choice of interview method
Since the aim of the interviews was to be able to ask the same questions to each user but at the same time give the interviewees the opportunity to give as much information as possible, a semi structured interview method was chosen. As mentioned in sub chapter 3.2.3, a semi structured interview method gives the possibility to get an overall view.

---

99 ibid. p. 144.
about specific topics, but also personal opinions of each interviewee. This will give the possibility to get a clearer view of the functionality of the add-on that is presented in this thesis.

3.4 Usability test
The test study is performed through personal interviews with the test users about the message delivery system. Eight questions are asked after a brief explanation about the add-on and its goal function is given. The questions are decided prior to the interviews and all test users will be asked the same questions. Except for answering the questions each test user will have the possibility to add additional feedback whilst all the seven questions are being asked. The answers will be documented and analyzed when all the interviews are done. By utilizing an oral method of interview a more flexible method of information sharing is possible, enabling us to add or expand on the information given.

The aim of the test study is to get feedback of what further improvements can be made on the add-on and if there is anything in the current version that should be altered or entirely removed. Therefore, the statements on the web form were designed to ask about the main function of the add-on, which is message delivery to users, but also on functions that could be added later such as notification sounds or if the message delivery add-on should have a certain color. This would be in order not to interfere with work or be a source of annoyance for the user.

The usability test will, thereby, give an idea of what kind of information they would like to receive from the message delivery add-on. Some users may think it would be a good idea if they were to receive private messages through the message delivery add-on instead of receiving it through for example e-mail. The type of information and how often information should be sent through the message delivery add-on is an important question and sending to much information can end up making the user ignore the information or seeing the add-on as an annoyance. It could also be a useful way to get quick information from the user if the message delivery add-on was to be used to ask the users about feedback on important topics.
4 The Add-on
In this chapter the product and its multiple iterations will be presented. The first part covers the steps that where taken during the project, thereafter chapter 4.2 covers the use of the message delivery. The interface of the add-on is presented in chapter 4.3 and in the last part the database is presented and discussed.

4.1 Project steps
Once the task was given and the goal had been set the project could begin its first phase with research into the various methods mentioned in chapter 3. Scientific articles about software development, software usability and motivational software were read and taken into account for the project. Most of the scientific articles were found and downloaded from KTH Primo.

During the second phase of the project, a prototype add-on was developed and presented for further evaluation from the customer, namely Shifo. The development was performed through Sprints, where the goals consisted of single functions. The Sprints consisted out of tasks that where continuously created as the need for them arose. This allowed for a good system for keeping track of the changes and additions that were made during the phase. The method that defined our Sprints, tasks and further process methods are defined in chapter 3, namely Agile and Scrum. Once the customer’s and developer’s opinions were shared and the add-on was reviewed, the goal was adapted to incorporate all of the agreed upon points of interest.

The third phase consisted of an evaluation of the first completed version of the add-on. The evaluation was done with interviews, of which the transcriptions can be found in the appendix of this thesis. The test users consisted of individuals who are regularly using computers and software systems in their line of work. Thereafter the answers of the interviewed subjects were analyzed in detail and the feedback received, was taken into account to improve the future version of the add-on. The improvements were implemented using the same tools and methods as in phase two when the first implementation of the add-on was made, but with the reviewed plan.

The fourth and final phase of the project was the final evaluation based on the last implementation of the add-on. This evaluation proved whether the system could be implemented or if further enhancements needed to be done. By continuing the reviewing and reiteration of the project goals and plans a continuous improvement is possible. This ensures an always evolving system that strives for the best user experience.
4.2 Message delivery
The add-ons main function is to deliver a message to the end user and to distribute information that is optimally delivered through the system that is actively in use. Considering the casual method of delivering the information through such an add-on, it is to not consist of critical information that may require the user’s full attention. Examples of such non-critical messages are:

- New and edited schedule information
- News cast information about an accident that can possibly increase the workload for nurses
- The current vaccination status, which in term may increase work motivation for nurses in that area of expertise

Specific areas of interest may vary in various locations and could be decided through a form, which will be described in the next subchapter, or through face to face interaction.

Once the messages have been entered into the database, they will be presented in the message window together with the picture relative to the text. Messages with higher priority would for instance be shown more often and after a certain duration of time, possibly 10 seconds, another message is presented. The duration of time may be lengthened or shortened in order to avoid drawing the user's full attention. Messages will be shown in loops until their end date is reached or the administrator decides to remove the message from the system.

4.3 Interface
The interface that is the primary method of interaction between consumer and add-on consists merely out of two simple windows which present information. The first window, which is marked with a green square and a large letter “A” in picture 4.a, presents a picture that will improve the user experience and add content to the information from the second window by presenting a related image. This image could be a simple icon from Shifo for when new updates have been released, thereby quickly informing the user of a message associated to the software developers.
**Picture 4.a:** This picture represents the software interface that enables nurses to manage patient vaccinations. The actual graphical appearance is not the final version. The red rectangle, or the “Add-on area”, marks out the message delivery system. The green square, or area “A” marks the picture part of the message and the yellow rectangle, also marked with a “B” marks the message part.

The secondary window, which is encased in a yellow rectangle and a large letter “B” in picture 4.a, presents the user with the actual message. Although the add-ons appearance will adapt to the system developed by Shifo, a temporary use of clean text is used to present the messages. The method of presenting the information will have to be adapted to effectively not interfere with user effectiveness.

The location of both the image and message is the bottom left corner of the software application. By not being centered within the application and placed on a more remote corner a less interfering sensation is strived towards.

The add-on’s position is based upon the fact that the specific area, where the add-on is located in picture 4.a, was not being used prior to the integration. That is why the add-on can be easily removed and disregarded without affecting the users to any greater degree, if such a step would have to be taken. As previously mentioned in chapter 2, the
add-on is primarily used for its simplicity in being integrated and removed. By enabling a test and run functionality of the add-on the system becomes more modular. Whether the removal and addition of the system is to be handled by the user is up for discussion in future versions of the software’s sub function.

4.4 Database
In order to fill the windows with the updated information a database was implemented for adding and presenting new messages and images. The add-on connects to the database and reads the information relative to the current user information. The picture 4.b visualizes the message travel path. As shown in picture 4.b, the developers and local administrators can add messages into the database. The arrows represent the direction of the messages. Once entered, the message is sent to the interface that is being used by an end user and presented in the add-on area, which is more clearly presented in picture 4.a. The arrow that points from the interface to the database represents data that is specific to the user, namely location or current time.

**Picture 4.b:** This picture represents the database and how it is interacted with. Developers and local administrators add messages into the database, which then are presented in the interface. If functionality, such as deleting a message or answering a form, would be integrated, then information would be returned to the database.

The user specific information is used to avoid showing information unrelated to the active user, such as location dependent information. The database consists of 13 information containers, these are:

1. **Id:** This variable is used for software requirements and not actively used for user interaction.
2. **Priority:** By placing a priority on the message, frequency and duration can automatically be calculated by the software. An example of use would be the odd change in work schedule for nurses in a hospital. Such a message would preferably be presented frequently and over longer periods of time, since it is
often seen as important and helpful. The priority can vary from 1 to 5 which are set by the message creator. A message with the priority 1 may equal the value of a message informing what food will be served in the diner and 5 would then equal important news that directly affects the user, such as an accident or problem that affects the days or weeks work progress.

3. Start date: This variable enables the message creator to show a message after a specific date. Such a function is helpful when messages are to be prepared in advance.

4. End date: When messages become outdated, the message creator may preemptively prepare the message removal. If no date is entered the message must be removed manually.

5. Location: Adding a location variable the message may be visible in certain areas only. The current user information will be cross referenced for such occasions. The message creator will be able to select from a list of various regions where the software has users. This limits the mistakes entered through misspelling. Locations would be entered with a large scale first, down to individual hospitals. For instance would the menu show Sweden as a top choice and Karolinska Hospital as one of the lowest choices. Thereby enabling messages to be shown in entire countries or regions.

6. Message type: Since future version of the software may support message dialogs with user in the shape of forms and questionnaires, a message type variable was added for software recognition in order to present the message of form in correct format.

7. Message title: May be used as identifier for message creator or possibly presented in message window, for summarization purposes.

8. Message content: This variable contains the actual message.

9. Message questions: If the message is a form or questionnaire the questions are contained within this variable.

10. Message visual id: In order for the correct image to be presented with the message an identifier is added to the message information. This variable points to the actual image that is located on a folder on the database server.

11. Message status: This variable can be toggled in order to hide or show the message.

12. Create date: For administration purposes the creation date is added.

13. Message creator: This allows administrators to identify misuse of the message system. This value will be entered automatically through the login process for creating new messages.

In order to support forms or questionnaires, an information collector would be added to handle the submitted answers by the users. But by keeping it limited to the monolog functionality the system minimizes the use of human resources such as help desks and
the cost for server workload increase, as mentioned in chapter 2. These functionalities may become attractive in the future and lead to further development of the add-on, but by gradually implementing the systems as well as preparing the software for further expanse a more user friendly interface could possibly be developed. Once the add-on described in this thesis, namely the simple message delivery system, is to be integrated another software add-on would be added with the functionality of adding messages to the database. An administrator would use this simple secondary software add-on to enter location specific messages into the system. They would also be able to control which messages are to be shown and which are to be deleted. Although this secondary add-on is a required element for the full functionality of the primary add-on it will not be covered in this thesis.
5 Usability test

In this chapter the test study and feedback of the users testing the add-on will be presented. The chapter will first present how the test study was done and brief information about each interview. The second and last sub chapter will present the answers from the interviews and the feedback that was received from the test users.

5.1 Usability test interviews

The interviews were qualitative and semi structured, which is mentioned in chapter 3.2 of this thesis, this method was chosen in order to be able to receive as much feedback as possible from the test users. The questions asked in the interviews were dominated by the pre-defined questions and topics presented earlier in this thesis, namely chapter 3.4, but depending on the answers received, follow up questions were also asked in order to get as much useful information as possible from the test user. Four people were interviewed, all of which are presented in Table 5.1.2. These subjects were chosen for their continuous use of computer based system in their line of work. The choice of not only interviewing people working in health care was to examine the usefulness of the message delivery add-on presented in this thesis in different professions. The interviewees were all working in Stockholm, Sweden. The reason of choosing interviewees from the same area is because the functionality of the message delivery add-on presented in this thesis is in such early stages that the result will end up delivering useful information, which is applicable the same way no matter which country it will be used in. The answers from the interviews are presented in 5.2 and the transcripts of the interviews can be found in the appendix.

<table>
<thead>
<tr>
<th>Occupation</th>
<th>Place of the interview</th>
<th>Duration of the interview</th>
<th>Work environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nurse</td>
<td>Coffee shop</td>
<td>30 minutes</td>
<td>Stressful - many distractions</td>
</tr>
<tr>
<td>Computer Engineer</td>
<td>Coffee shop</td>
<td>30 minutes</td>
<td>Stressful - many distractions</td>
</tr>
<tr>
<td>Project Manager</td>
<td>Office environment</td>
<td>35 minutes</td>
<td>Stressful - many distractions</td>
</tr>
<tr>
<td>Human Resource</td>
<td>Office environment</td>
<td>32 minutes</td>
<td>Non stressful - Very few or no distractions</td>
</tr>
</tbody>
</table>

Table 5.1.2 shows the information about each interview. From the table it is possible to see that only one test user is working in a non-stressful environment while the others all work in

---

104 ibid. p. 143.
an environment where there are a lot of distractions. The table also shows that the duration of the interviews were almost the same in each interview.

The questions asked to the test users are as followed:

<table>
<thead>
<tr>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

5.2 Usability test answers

This subchapter presents the results obtained in the interviews. The answers are divided by the questions and the three topics General, Distraction and Message content that were presented in chapter 3. The answer have been divided in different tables and the answer of each interviewee to their corresponding question have been presented in the same table in order to make it easier to see the similarities and the difference of opinions between the interviewees.

5.2.1 - Topic divided Questions and Answers of the interviews

<table>
<thead>
<tr>
<th>Topic: General</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables 5.2.1.1 and 5.2.1.2 present the answers on the questions about the topic General.</td>
</tr>
</tbody>
</table>
Question | Do you find it useful to receive messages through a message delivery system?
---|---
Interviewee | Answer
1. Nurse | I think, but both yes and no. I think it could also be a bit distracting. Like if you on the system, working, and you’re writing something or looking at a patient or something, I think. And then there’s a thing popping up and telling you things. Maybe when you’re tired you tend to look at those things instead.
2. Computer engineer | Yes, I think it would be a very useful to receive information about my company or what is happening in my job environment.
3. Project Manager | Yes, definitely! It is a useful way of sending information to the users.
4. Human Resource | Yes, although it shouldn’t keep from work. Could be helpful with some types of information. The possibility of scrolling through the various messages could be good.

*Table 5.2.1.1 shows that the interviewees thought that the message delivery add-on would be useful in their daily work.*

Based on the answers presented in table 5.2.1.1 the add-on is something that can be very useful in a work environment. It is an easy and direct way of receiving information while working and to reach all the users of the company at the same time.

Question | Let’s say you have a new message and it might be important. Would you like it to notify you that there is a new message? By a sound or by start blinking?
---|---
Interviewee | Answer
1. Nurse | Only for very important messages. I can’t speak for everybody, but when you work in a hospital, in a ward, there’s so many things happening. There are people talking, there are machines beeping and so much sound. If my computer is supposed to sound and blink, I mean... No... no...
2. Computer engineer | A sound that tells you that a new message has arrived. I would also like a time stamp so I know at what time the message arrived
3. Project Manager | I would prefer to be notified by a sound, blinking would be annoying I think. Maybe instead of blinking it could change color, for example from blue to red. A sound and a change of color would be the best in my opinion.
4. Human Resource | Not through sounds, since that would end up being distracting. Possibly blinking that would only occur once. The notification could possibly only occur at important messages, such messages that required higher attention. Higher priority would thereby lead to longer duration and blinking. Low Priority messages should not have a special notification.

Table 5.2.1.2 shows the answers about whether the message delivery add-on should notify the user about a new message. The interviewees had different opinions about how to be notified about a new message but all of them thought there should be some kind of notification if the message is important.

The answers in table 5.2.1.2 shows that half the test group thought it would be best to get a clear notification such as sound or that the message window would blink when a new message is received but the other half of the test group thought it would be better with something more toned down such as the message delivery add-on blinking only one time. One test user thought it would be better to only get a notification if the message was really important and no notification at all in other cases.

**Table 5.2.1.3** presents the answers on the questions about the topic Distraction.

<table>
<thead>
<tr>
<th>Question</th>
<th>Would you like the message delivery system to have a color that makes it stand out from the rest of the interface in order to make it easier to notice?</th>
<th>Interviewee</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Nurse</td>
<td>I think it doesn’t have to be exactly the same, but it doesn’t have to be like, your box is red and the rest is white. But it could be, if the one part is white and the rest is gray.</td>
<td>Nurse</td>
<td></td>
</tr>
<tr>
<td>2. Computer engineer</td>
<td>I think the design should be different from the rest of the application, otherwise it will be hard to notice when a new message has arrived. It is important that the users get the information that has been sent and if the message delivery add-on doesn’t stand out there is a risk that the message will go unnoticed.</td>
<td>Computer Engineer</td>
<td></td>
</tr>
<tr>
<td>3. Project Manager</td>
<td>Yes, definitely, otherwise it might be ignored or forget that it exists in the application.</td>
<td>Project Manager</td>
<td></td>
</tr>
<tr>
<td>4. Human Resource</td>
<td>The design is easily ignored, therefore not very distracting. As long as one is not pressured to read all messages and only 10 - 20 messages are used it should not matter how it looks. Preferably all messages</td>
<td>Human Resource</td>
<td></td>
</tr>
</tbody>
</table>
may stand out from the rest of the software, thereby dividing the software into work and the informational message window.

Table 5.2.1.3 shows the answers from the interviewees about the color of the message delivery add-on. All four interviewees thought that the design should stand out from the rest of the software.

The results show that the next version of the add-on should focus on implementing a design that stands out a bit. The test users all thought they would ignore the add-on if it was to be similar to the rest of the program and the interviews show that it is important for the design of the add-on to be different from the rest of the application, in order to get noticed. Although they all thought the design could stand out from the rest of the software, it was a difference between how much it was to stand out in order for it to not become ignored, but on the other hand not become an annoyance. The computer engineer and the project manager thought it was to stand out a lot since they figured they would otherwise not be able to notice it. The nurse and the human resource worker thought it should stand out, but not too much since it would then be a hindrance for their work.

<table>
<thead>
<tr>
<th>Topic: Message Content</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables 5.2.1.4 and 5.2.1.5 presents the answers on the questions about the topic Message content.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question</th>
<th>What type of information would you like to see on a message delivery system? (For example: Schedule changes, news directly correlating to the work)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interviewee</td>
<td>Answer</td>
</tr>
<tr>
<td>1. Nurse</td>
<td>It has to be information that is directed to my work, I do not want any information about the price of coffee in the cafeteria.</td>
</tr>
<tr>
<td>2. Computer engineer</td>
<td>Everything that is related to the job and concerns everyone or most of the users in the company can be send through the message delivery add-on. More personal information is better to be sent through e-mail.</td>
</tr>
<tr>
<td>3. Project Manager</td>
<td>I think it could be useful, as long as it is job related information I´m happy with receiving it directly on the screen. The only problem would be if it sends information to often, as long as it is very useful information, like changes in my schedule, I would be happy with receiving it through the message window.</td>
</tr>
<tr>
<td>4. Human</td>
<td>Occupational dependence, but no jokes or fun fact. Important or</td>
</tr>
</tbody>
</table>
Table 5.2.1.4 shows that all interviewees thought it would be interesting to receive personal information such as schedule updates through the message delivery add-on as long as the information was job related. They were very clear about not wanting to receive information that would not be of any use in their work.

<table>
<thead>
<tr>
<th>Question</th>
<th>Would you like to answer questionnaires through the message delivery system?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interviewee</td>
<td>Answer</td>
</tr>
<tr>
<td>1. Nurse</td>
<td>Yeah, it could be good as long as the forms are not mandatory.</td>
</tr>
<tr>
<td>2. Computer engineer</td>
<td>No, that would not be a good idea. It will be annoying for the users if they have to fill in forms while they are working.</td>
</tr>
<tr>
<td>3. Project Manager</td>
<td>No, as a user I would not want to spend time answering forms when I’m working. I think that would only give the users reason to ignore the add-on. In a stressful environment the last thing you want is to spend time on things that is not helping you making progress on your daily work.</td>
</tr>
<tr>
<td>4. Human Resource</td>
<td>Brilliant as long as it is anonymous and voluntary.</td>
</tr>
</tbody>
</table>

Table 5.2.1.5 shows the answers about being able to answer forms through the message delivery add-on. The answers show that two of the interviewees thought it could be useful as long as it is voluntary. The other two interviewees thought it would be annoying and an hindrance in their work.

One question that gave divided answers was the questions about being able to fill out forms, as can been seen in table 5.2.1.5, half the test group thought it would be an good idea as long as it is voluntary while the other half right on thought it would only be an annoyance for the user and that it should not be implemented at all.
5.3 Usability test analysis

This chapter presents an analysis of the answers obtained in the usability test. The conclusions and our derivations from the analysis in this chapter are based on the answers that were received during the interviews and their respective coding presented in chapter 5. The theoretical method of analyzing is mentioned in an earlier chapter, namely chapter 2.6.

5.3.1 Method of delivery

The answers given around the subject of the method for message delivery, when discussing special notification, were mixed among the interviewed subjects. The nurse leaned more strongly on the opinion of the design causing distraction, whilst the others were more of the opinion that distinction between the regular system and the add-on area can be more accentuated. The method of delivery, as in notifications of important messages and the method of drawing user attention, would have to be adjusted to the targets work environment. This would be functions and variables that would have to be adjusted during add-on implementation. A calm work environment can adapt a more noticeable interface since time is not an issue. A hospital, where the environment is constantly moving and concentration may be vital as well as scarce, would require a more subtle message system that can be noticed at the nurses will when work is at its most passive state.

5.3.2 Message content

The answers given by the interviewed varied greatly, in the sense that certain types of information would be accepted whilst other must be sent through e-mail. Since the messages have the possibility of being entered by administrators and software developers, certain message content would have to be limited. Considering how nurses may work in time sensitive environments unimportant information may be frowned upon and end up decreasing work effectiveness. Rather the always displaying messages, a limitation could be applied to keep messages from being shown during stressful periods of time. It may be sufficient to show only the messages with the highest priority for longer durations, to keep stressed users from becoming distracted but also well informed.

5.3.2 Distraction

Less interaction equals less distraction. Implementing functionality, such as deleting or switching between messages, may become distraction since the user chooses to focus too much on the messages. This being said, functions that is less noticeable but with high functionality, such as hovering over the message to keep it from switching may lead to users sensing a more “frictionless” interaction with the add-on.

Implementing user specific information into the add-on would require full automation for an effective use. For example, a schedule informing message may be effective in the
sense that it shares changes in the user’s schedule. These types of messages would be simple for a supervisor to add, but would be time consuming and regarded as wasteful since an actual platform for the schedule exists. By automating the interaction between the systems the user would gain the most effective experience, but it would require greater development time for each individual system. Aside from the full automation, information that might be practical to display on the message delivery system, may require additional user informing stages, such as the supervisor delivering the schedule changes by email. Since the message delivery system is meant to minimize distraction risk, there arises a high chance of messages going by fully unnoticed by the users.

5.3.2 Questionnaire
Giving the users the possibility to be able to answer questionnaires through the message delivery system is something that can be added, but if implemented it has to be optional since some users, especially those working in a stressful environment, may find it highly disturbing to being forced to answer a questionnaire while working.
6 Results
This chapter presents the results and conclusions of the degree project and the various goals that have been reached. The first part presents the results of comparing with other projects, as well as the value of comparing with this thesis. The second part is about the work methods used in this project and their results on the project in general. Lastly the result of method for interviewing is presented.

6.1 Project comparisons
Studied method and information gather ended up presenting a clear path for the thesis progression and few deviations arose. The thesis present one out of many projects performed in one fashion put together by individual subject information. By using the acknowledged methods, more effective progress can be made, rather than reinventing the path by doing, failing and redoing once revised. By comparing with projects performed earlier, time and financial costs can be derived from them. Although this project did not entail any financial costs, the magnitude of the project and the duration of the project provide a hint to how the estimation can be done. Not only time and financial cost can be derived from this thesis, but also knowledge about the development methods taken for the completion of the first version of the add-on. Therefore the hope for this is to be able to present the knowledge that was gained and that it can be reused in similar projects such as described in this thesis. Research into various implementations of existing add-ons on systems commonly used was performed, of which is written about in chapter 2. Information gained through the research later aided in the creation of the individual project goals. The application of the research onto the project in this thesis is described in the earlier occurring chapter 3.

6.2 Work methods
Work methods, such a “Scrum” enabled a more agile method of development, although an adapted version had to be implemented, since the thesis work was performed by two individuals. Main points of focus, such as the method of tracking and managing the tasks and compiling boards to track the states of the tasks, were still used. The overview of the progress in the project presented valuable motivation and clear goal representation. For larger groups, other rules for “Scrum” exist, but were chosen not to be included in this project. Therefore this thesis presents the possibility of adapting the project work method for use in smaller project groups. Even if the smaller project groups do not deliver at the same efficiency as larger groups, project methods are necessary for a more effective development. This thesis therefore provides one interpretation that enables groups of two to develop small-scale software, such as an add-on.

6.3 Add-on
The final results of the add-on within this thesis are the first version and the noted further improvements based upon the interviews. Although the add-on was not tested in
the environment for which it is being developed, the first version delivers an insight into the possibility and usability of a software addition, such as the message delivery system. Continuing on the development track described within this thesis, the final result, namely the first prototype, would become reality and possible for integration into the software MyChild.

6.4 Usability test
Once a basis of the add-on had been developed, a study into the method of interviewing was performed. This enabled a more structured method of external opinion gathering, as well as allowing for a more varying range of add-on application information.

Receiving the information through interviews provided valuable opinions and possible further improvements. It shows that improvements not thought of before the project start, can be clear to the users. Even though the add-on only reached its first version during the project. The method of interviewing is important depending on the type of information that is sought out. This thesis shows that the semi-structured interviews can deliver information at a high quality and with benefits through analysis.
7 Conclusions and further improvements
This chapter will present the conclusions of the project, a discussion about the thesis in general and what positives and drawbacks occurred concluding the project. Finally the possibilities of future work is speculated about and discussed.

7.1 Conclusions
This thesis presents one possible method for developing the first version of a simple message delivery system. The application for registering children vaccinations, namely MyChild, was developed with the goal to help the nurses and the doctors to get a better overview of the health of children in the country.

The purpose of the thesis was to explain the process of developing an integrated message delivery system that is easy to use and delivers information to multiple users. The results presented in this thesis present a standalone version of a simple message delivery system that is yet to be integrated into software. Since the add-on wasn’t integrated the results do not incorporate the ease of use. Through the usability test and the analysis of the interviews, the interest in the simple message delivery system was discovered and verified. It thereby confirmed that the add-on would be appreciated in a work environment. This relies on how various parts of the application would be developed and configured. The software development method worked for a smaller group consisting out of two members, through interpretation of the most applicable methods in Scrum. Removing parts of the work method which cannot be fulfilled, because of the limited group size, it did not cause any larger problems. In conclusion, the parts used improved the project efficiency to a satisfied degree.

The message delivery add-on makes communication less complicated and leads to low priority information being shared quicker and without any inconvenience. The dangers of the add-on and the simple message functionality became clear through the interviews, since worry, for high priority messages being sent through the message system instead of through emails or word of mouth arose. There were clear opinions that messages of lesser importance would benefit from this system and allow its user to become informed of various topics.

Lastly, the method of creating an add-on that was not to become a hindrance to the employee was partly answered through the analysis of the interviews. By presenting the first version of the add-on and describing the situation that was possible to be applied directly to the use of such an application, further possible functionality and considerations for improvement were received. Taking the entire thesis into consideration a method for developing a system that would later be integrated was presented, with the aim of avoiding employee distraction included. Functions that would lead to high distraction would be sounds, or blinking visual stimuli. Presenting
the add-on once integrated into the Shifo system, a certain degree of graphical distinction would increase the user experience, according to the interviewed subjects. This would present the add-on as a separate function of the main application and clarify its priority.

7.2 Discussion
When working with the software, namely the add-on that is described in this thesis, a choice had to be made. Whether to create a standalone version that would later be added into the system or expanding on the existing system. The drawback in creating a standalone version is the extra work of integrating the system, since various parts may interfere with existing code. Although the integration process may be longer, the positive values in creating an add-on that is to be integrated later, is the possibility to more simply remove it if the add-on does not fulfill its designed purpose or if users experience is compromised. Since the integration is done after the functioning version of the add-on is created, there might arise a sensation of split personality in the program. The interface part that is primarily designed might clash with the later addition of a message delivery system. To inhibit problems from arising from this drawback, the risks have to be analyzed. In this thesis the risks were researched through interviews. The choice between creating a standalone version was the better alternative for this specific project, for the above mentioned benefits, but also on behalf of the customer demand.

Creating systems that are to be used in a modern society often require continued revising of the current system. By continuously improving software whilst considering the users requests for a better user experience, a better end result can be more readily delivered. The aspect of initiating a project such as described in this thesis was based upon the fact that the first version would only fulfill the developer’s wishes and demands. Since the study was done after the creation of the first functional version, more areas of improvement were added to the list defined by the developers. This method of developing ensures that the software not only lives up to the expectations of the developers, but also that a useable program is created. Although this method has its perks, it also delays the production of the system. Since possible users of the system need to be interviewed and their answers analyzed, the risk is that the system ends up becoming outdated. Delaying the development because of the responses may end up creating a program that gives the appearance of seeming fractured. This may not apply to software that is in its very early stages, since change is still feasible, which is the case in this thesis.

In order to receive as much and as useful feedback as possible from the first version of the add-on, the most relevant interview method had to be chosen. Different interview methods have their different purposes and in order to choose the right method it was
crucial to consider what the intention was with the interviews, what type of feedback was desired and what method would be most effective in order to gain it. It was important to gain knowledge on exactly how the add-on was perceived by the users, also to gain a better understanding of the user’s point of view on the message delivery add-on. The interviews were therefore done using a method to give the test user the opportunity to tell their exact opinions of the add-on and this gave the possibility to consider the improvements that according to the users were critical. This in turn gave the possibility to develop the add-on to become more effective and adapt it better to the type of environment it is used.

The project development method enabled the improvement on the goal of the project through allowing active change throughout the work duration. Although the Scrum method is designed for larger teams than two developers, parts of the work method allowed for a more focused and improved project progress. The parts that were not included have been designed to increase efficiency in groups of five to nine people, which is the required number for fulfilling all the various roles in a team, as described by Scrum. Removing these set roles, a risk of lowering both efficiency and management accuracy can arise. Although this risk may have arisen, through working with project steps which were closely related to each other and working on a smaller scale project, track ability of each change and problem became easier to manage.

External risks were largely avoided through developing a first version of a standalone application. By setting goals that did not require outside resources, except for the interviews, the minimization of the risk for delays and project hindrances relied upon the developers.

### 7.3 Evaluation

By researching the various methods for performing the project and its consecutive steps, a frictionless project development was possible. Although the stages may have been performed in shorter intervals with a greater buffer for last minute changes or corrections that corresponded with the results from the interview, the target schedule was achieved.

Creating the various tasks and deadlines proved to motivate towards completion, yet in some cases hinder increased performance in efficiency. By knowing that a deadline would be achieved a lessening of effort may have arisen and instead of finished earlier than expected the schedule was followed with precision.

Reading and researching these various grounds, such as interviewing methods, a greater understanding was to be achieved. That in turn ended up minimizing the standard problems that are commonly present at trial and error projects. Since not all the various
literature presented an exact situation that was easily adapted, steps were taken to integrate the methods and information into this specific project.

The project that has been discussed throughout the thesis is the development of the simple message delivery system. The system was developed with the actual aim at creating an add-on that would be simple to integrate into the customer’s software. Although the add-on was never integrated into the system, a first version of it was created. There are most likely many more possible methods of creating similar add-ons as described in this thesis, but the presented method still remains to be one possible answer.

As for the purpose, goals and benefits that were presented in chapter 1 the thesis resulted in the following results. The risk for decreasing effectiveness was researched through the interviews. When implementing an add-on as described in this thesis, the outcome is affected greatly by the nature of the work environment. All from the type of message to the delivery of the message could affect the user’s effectiveness in the sense that it becomes distracting. As the add-on was only developed until the first version, the results received through this interview would have to be implemented in the future versions. Therefore the solution of creating an add-on that could possibly be used in other software is not liable since a certain degree of modification would have to be implemented for the best results.

Through increase in work efficiency and possibly the improved sharing of general work-related information, the user experience would be improved. Since one of the goals for the system was to create a system that could end up increasing the acceptance of the computerized system, the result of this work cannot be fully concluded. Without implementing the message delivery system in the final software, no actual field results can be received. The only result that is presentable is the research through the qualitative interviews. Focusing on possible users in work environments who continuously utilize software, in which the add-on as described in this thesis could be implemented, an insight into the actual answer could be gauged. The insight resulted in a definite positive answer towards finalizing the add-on and new limitations and guidelines that have to be fulfilled in order to gain the best user experience.

Since the development of the first version of the add-on was to describe a method of the various stages towards the final result of possibly implementing it into existing software, the results from this development was only partly concluded. The actual integration process was never performed since only the first version was developed in order for the research into the various risks, as well as the possible beneficial opportunities, was performed through interviews. Since the thesis describes the steps taken toward the first version the goal is mostly achieved. The development aids in taking into account
multiple aspects required during development, in order to achieve a result that isn’t faulty because of poor research and preparation during the continuous project duration. The aspect of improving the information delivery as well as motivation is partly answered through the interviews, since the input describes the expected results of an add-on, as described in the thesis, being integrated into their daily work software. New limitations were thereby discovered, which could possibly cripple the future versions of the add-on if not taken into account during further development.

7.4 Further improvements
This subchapter will discuss the future improvements that can be done with the add-on, the improvements that are both essential and would make the user experience better and more effective.

In future versions of the add-on a secondary function would be added, namely forms that can be presented and filled out by the user when interest arises. The forms would be able to gather information about the user’s opinion of newly added system functionality or various message topics and quality. The questionnaires would enable a simple dialog between administrators and end users, or even developers and end users. A short and simple questionnaire may not gather critical information effectively, but may simplify the improvement of software or work conditions through a less official method of opinion transfer.

Another improvement implemented in future versions is notifications on when a new message arrives. The method of notifying is to be optional in order to not become an annoyance for the user. Two types of notifications could be implemented, notification by sound or a visual notification such as the message delivery application blinking and or changing color.

A third improvement that may be considered for future versions would be the color of the design of the add-on. The add-on is to have colors that stand out a bit from the other part of the application. This will make it easier to notice and therefore minimizing the risk of a received message goes unnoticed. It is important that the colors chosen for the design message delivery add-on do not differ in a manner that the add-on becomes a source of distraction and annoyance.

7.5 Continuing the work
In the terms of this thesis, the project could be continued by working on with the applied work method and finishing a prototype that would be integrated into the existing software. This would then be tested out in the designated location for the software, in order to gather more information about possible improvements. This would
also entail further improvement to the interview technique and the adapted Scrum method.
References

Books


Electronic Articles


Mischra, D; Mishra, A. (December 2011). *Complex software project development: agile methods*. Journal of Software Maintenance and Evolution: Research and Practice


Wiseman, J. (2008). *Now we are talking.*
https://www.facebook.com/notes/facebook/facebook-chat-now-were-talking/12811122130. (downloaded 2014-02-23)

**Websites**

This website was the official website for the Jira software that was used in tracking the project tasks.

Chrome. [https://developer.chrome.com/extensions/getstarted](https://developer.chrome.com/extensions/getstarted). (downloaded 2013-11-02). This website was used for the tutorial of creating a chrome application.

This website is the official website of Eclipse, where the main information was gathered about Eclipse in general.

This website is the official website of Facebook.

Linked to from the Scrum alliance website, this website was used for its thorough information about task boards.

This website was used for its information about agile workspace.

This website is the official website for Microsoft Word and is where general information of for the software Microsoft Word can be found.

This website discusses add-ons and plugins.

This is the official website for MySQL.
From this website information about research methods was gathered.

This website was used for its information about Scrum.

This page contains more precise information about the various parts of Scrum.

This website is the official website of Shifo, where the main information was gathered about Shifo in general.

This website discusses the software MyChild which is discussed in the thesis.


This website is the official website of Steam, where the main information was gathered about Steam in general.

This website is the official website of Sublime Text, where the main information was gathered about Sublime Text in general.

This website was used for its information about Scrum and its benefits.

Tutorialspoint. (2014).
http://www.tutorialspoint.com/management_concepts/agile_project_management.htm. (downloaded 2014-06-04). This website was used for its information about Agile.

This website was used for its information about Agile and its benefits.
WordPress. http://codex.wordpress.org/Writing_a_Plugin. (Downloaded 2014-02-24). This website was used for its information about plugins.
Appendix - Transcripts

All the people who were interviewed were chosen for their association with computer related work. Before the questions were asked all the people were given a description of the add-on and its various applications. To simplify the description a direct example of the add-on in the individuals work environment was given. In order for the transcripts to be more legible the individuals will be noted as follows. The interviewer will be noted with the letter “I” and the individual that is being questioned with the letter “Q”.

Transcript 1 - Human resource consultant

I: Do you think it would be useful to receive messages in this fashion?
Q: Definitely, it depends on the type of messages and it shouldn’t keep people from their work. But I think it could be a great idea, it should be helpful.
I: You mentioned during the introduction, that you should be able to scroll through the message. We hadn’t thought about that, but we had considered the situation where you don’t want to see a message again. Like for instance when a message has already been seen and you want to delete it. Do you think that this so called message deletion would be useful, it would be a small X in the top corner somewhere on the add-on, which when pressed, the message would disappear from their personal view. Is that something you would want, or would it hinder you?
Q: I think, depending on the type of message, like the type mentioned in the explanation, I don’t think it would be very necessary. Maybe if you had the time and energy, or maybe a button that could hide the entire add-on... Well actually, I think it could be distracting, since people might want to click away all of the messages. That would then be counterproductive.
I: That’s what we feared, when we thought of the question. So more interaction, would very simply lead to a greater risk of distraction?
Q: Yes.
I: Very well. On to the next question. When a new message is added, would you like the message window to notify you. For instance, be presented with a new color? You already have a new picture for each message, but this considers color around the actual window or maybe even a sound like a blip or beep.
Q: No sound, that would be distracting. Potentially, a blink, but only once. Otherwise it is already drawing attention through the pictures.
I: Yes, like when it would switch over. Although this is only for new messages. If you for instance already read all the other messages and this new one is added...
Q: I don't think you should have it for any random messages, but maybe for warning messages. For instance when there's been an accident. You wouldn't want a notification of every new message, but possibly at the important once. Still no sounds though, but maybe red color around the border, or a short blinking.
I: Okay. Considering the fact that I already mentioned the priorities of the messages. If you would organize it so that important messages get a higher priority and only those would blink. Would that be alright?
Q: Yes! But I still think other types of messages would be good. Like someone's birthday, but that doesn't have to be blinking and annoying.
I: Ha-ha, I see! When consider the message time duration. Messages with higher priority would be shown for longer and more frequently. Would you want the messages that have a higher priority to be presented for a longer time?
Q: Maybe this is a bit complicated, but since you mentioned a delete button. Would it be possible to have a button that pauses the rotation of the messages? This way you would be able to read the message without have to wait to the next appearance?
I: Yeah, and that would probably solve the problem with having to go back all the time.
Q: But not freezing the message for an eternity. More like add 30 seconds to the duration or something.
I: How about scrolling over the message with the mouse. Then it would pause until you remove it again?
Q: That would be a good idea, as long as you think that everyone will understand that function. For me that would be great, but maybe other wouldn't realize that such a function existed.
I: Exactly.
Q: Maybe in that respect, it would be better with a button.
I: So we already mentioned distractions, namely the main risk of our add-on is the risk that it would be distracting. We would be applying it in a work relation, a software which you would be using whilst working. The main idea of the add-on is to improve effectiveness or social awareness or even general awareness. In that aspect do you think that making this add-on standing out from the regular software, with colors or just general appearance, that that would distract you or continuously pull you gaze towards it.
Q: No, I don't think so.
I: Would it be easy enough to ignore it, even with a different color.
Q: Personally it wouldn't be too distracting. And as you said it would only be about 20 or 30 messages. They would go around all the time so you wouldn't feel pressured to read every new message all the time.
I: And considering, would it be better if it stood out, via the colors? Or would it be better to be integrated into the software, as if it were part of the software?
Q: I think the high priority messages should stand out, or rather the entire add-on could stand out. It doesn't have to be fully integrated.
I: ... Do you think that people would be more prone to distraction if the messages wouldn't be presented multiple times. Consider the situation where you get a long message, like 20 messages, so not too long, and they would read it. When they are done it disappears. With the risk that it would show up again and they would read it again and
be distracting since they would read every message over and over again without realizing it. Would that be a high risk?
Q: That's a good question, I think that that's very individual. It is hard to imagine, since for me it wouldn't be a huge distraction at all. But I think that that could possibly be a big distraction since I have seen people constantly checking their Facebook for new messages all the time. But it isn't very long messages, so you should be able to realize that it is old, quickly.
I: The combination with the same picture as before, could possibly minimize the risk or? Since you realize that it is the same message through the picture.
Q: Yeah, the pictures are always good.
I: What do you think about the pictures? Are they to big or...
Q: I think the pictures are brilliant, since you get an idea of what it is about. I don't think that people will read the messages that are just blank lines.
I: And on to content. Let's say for instance, I have already mentioned various examples of content in the explanation. What would you consider a good type of content in such a message window?
Q: As always that depends on your type of occupation and what type of software it is. Obviously I wouldn't want information that is completely irrelevant, like the joke of the day or a fun fact. I would like things that are actually important, or something that concerns me or my coworkers, like their birthday. Or maybe when there is a message about someone doing something really good or reaching a goal. That could be really motivating, always something motivating, nothing depressing.
I: That is interesting that you would mention motivation. Since the customers first idea of the add-on was that it would motivate. It is also possible that the software would be made to motivate the staff. Through motivating messages. Take the example of a nurse vaccinating a certain number of children. The friends could get a message that would inform them of their good performance, and they could give them a pat on the back.
Q: As long as it doesn't get competitive.
I: I see. We could give the message directly to the person, where it would say "Good job". But would that really be motivating since it would come from an automated system?
Q: No, personally I would rather have a message saying that the team or section did a good job and not like the person next to me did a good job.
I: Okay, but the pat on the back system would encourage the other coworkers through other people. Do you think that would become competitive?
Q: Yeah there is a risk that it could become a brag. It is potentially only a Swedish thing for people to get really angry with things like that.
I: Let us say that only the boss gets this kind of message. For instance if the staff did a lot of work today, then the boss could compliment the staff.
Q: That would be good! Since the software is job related and you think of it as you work! It kind of is like your boss telling you that this person did this! Compliment her/him! You would rather hear, you did a good job! From the boss.
I: So you mean from someone mouth?
Q: No i mean the automated system would feel like the boss. Not like a friend telling you something. But it would be good if a person who is in charge of the group could compliment instead through getting that information.
I: You mentioned competition, some companies try to inspire competition.
Q: I agree! but those companies are usually telemarketing not children vaccinations.
I: How about your line of work, human resources?
Q: We kind of already have that since you can see how many cases the other people have finished. But in our situation the cases are very different from each other so the accomplishment isn't really easy to read.
I: But don't you have a system for grading the cases.
Q: Not really, but you yourself know. I mean some competitiveness could be good. But not when it is like this.
I: No.. I see. So you would say a motivation through telling you that the team has accomplished something would be better?
Q: Yes! Or you could tell the individual that they did this much better than last month.
I: Like self-improvement?
Q: Yes, that would be more motivating since you could choose to interpret that yourself.
I: I see... Two more questions. We have information about who logs in. What would you think about the message window delivering personal message, not private! But personal message in the sense that it is mostly related to you. Like change in your schedule or maybe... staff additions directly in your area.
Q: Who would write these messages?
I: Probably you direct supervisor.
Q: As long as they don't send like thousands of messages.
I: What if we would connect it to your schedule so it would send you an automatic message, when a change has been made.
Q: That would be brilliant as long as it doesn't get misused.
I: How about if some message should probably be sent through email since they are important?
Q: Exactly, information from your supervisor shouldn't be on that message board. Actually in that case, the schedule message shouldn't be done either since you risk not noticing the message.
I: But would you think that in combination with email and message board it would work?
Q: Yes. Because the message board would work more as a reminder.
I: And in the future, if we were to continue developing new functions. We would also try to implement a type of form or questionnaire in this message window. For instance when we add new functionality, where you can now delete messages. In order to see if the users like it or not, we would create a form that would ask the user. It would show as a message and allow to user to quickly answer one question. Would that be good?
Q: Yes, I think that is great since everyone would get to choose to answer. As long as it is anonymous. It is a great idea, and if the supervisor can use it then it would be a easy way to ask the staff.
I: Good, that was all! Would you like to add anything?
Q: No, I think it is a good idea. You probably have a lot of decisions to make to decide what to include in the add-on or not. But I think it looks good. It would probably be useful in the HR.
I: Thank you for your thoughts and input.

Transcript 2 - Nurse
I: You now know what the add-on is about and we will begin with the general message and the way of delivery. Do you think it will be useful to receive messages in this type of system?
Q: I think, but both yes and no. I think it could also be a bit distracting. Like if you on the system, working, and you’re writing something or looking at a patient or something, I think. And then there’s a thing popping up and telling you things. Maybe when you’re tired you tend to look at those things instead.
I: Exactly, that’s one of the main theories in the project and a lot of our studies have been about distractions within the system. In your line of work you wouldn’t want it to be in the way.
Q: Yes, I think for me it would probably mostly be a distraction. Well if it were very plain messages, like a text, okay. But if there is a picture, then it might grasp your attention in other ways.
I: I can iterate on the functionality that I didn’t make too clear. Some messages would be shown for a certain amount of time. High priority messages would be shown for a longer time, but the add-on will automatically loop through the various messages.
Q: So, will there be a message that is presented in 5 seconds and then another is shown.
I: Yes
Q: That may be worse! Since I have my attention disorder that will force me to look.
I: The messages will be short and the picture may be a comical picture of a bosses head when an administrators message is shown. Or for instance a newspaper when news is shown. It won’t be so frequently changing so not to be to distracting. If a short viewing duration would be to distracting than a longer may be better.
Q: I am easily distracted.
I: I fully understand. If you would be able to have the oppurtunity to mark messages as read. For instance you would ha read it and then clicked a button to delete it.
Q: Like flagging it?
I: Yes, exactly like that and then it wouldn’t be shown. This would be limited to your account. Do you think this would be good or bad?
Q: Well my first thought is that... I don’t know if it would be unnecessary, but it seems very strange to be able to remove a message about something like a traffic jam. Like, come on! It’s work related.
I: But in the sense that if you read it once and then you..
Q: I understand! That seems more logical!
I: Then it would be less distracting so that you don’t have to look at it again.
Q: Then it may be good. But would the other messages pop up again? Because then you might want to clear the entire message board. You would have to delete all messages that are like two weeks old.
I: The messages are supposed to be maintained in such a fashion that you would not get a really old message. They have time frames, you would for instance only see a message for a week and then it would disappear. Messages would have the lifeline of a week. But yes it may be distracting.
Q: (Laughter)
I: You’re not the first one to say so. Next question, and this may be even more distracting. Lets say you have a new message and it might be important. Would you like it to notify you that there is a new message?
Q: Only if it’s super important. But not like if it’s for ex. “We bought a new coffee machine for the staff room.” But yes, if it is something really important, like a emergency in the emergency room or if there’s been a large accident. Then I would find it to be a great thing, since everyone would become aware of it. It depends on the importance of the message.
I: Very well. I can already hear that you are mostly concerned about the level of distraction. You have to consider that it is only displaying plain text in combination with the picture.
Q: It would be less distracting if as you said that you would have the same picture for every staff message. Then it would be less distracting, but it would be more if there were a new one for every message. For then it would feel like it was plain information and less commercial.
I: We also wondered... What do you think of the possibility of showing high priority messages over a long period of time. With that I mean for a longer period of time then compared to the information of ... lets say the food in the cafeteria today.
Q: It also depends on what a high priority message is. Like for instance “It’s time to enter you vacation form” for the time that you want your vacation. That would be good to show 3 weeks in advance.
I: Just to make clear, not for a long time. Like two weeks, but rather a long duration...
Q: Oh you mean five minutes, thats a good thing then. I think.
I: So, you would have more time to notice it.
Q: Yes, that’s a good thing. Like information about the food in the cafeteria you could always find somewhere else. But being reminded about important things that are happening, that could be good.
One of our main themes of questions is actually about distractions, so that has definitely been considered in this thesis. Do you think it would be good to give the message window a separate color or theme then the rest of the system. For us who are developing the add-on, we might not be able to adjust it sufficiently enough to melt into the rest. Do you think this would be distracting?

I: Would it be better if you could differentiate the less important parts of the window from the rest? For instance, if you would have everything green and the less important parts yellow?

Q: But that’s too strong of like...

I: Difference?

Q: Yes. If they’re in the same scale of colours, for example a darker green and a little lighter green or vice versa, but not too separate colours. It would be too distinct.

I: It would be too distracting?

Q: Only for very important messages. I can’t speak for everybody, but when you work in a hospital, in a ward, there’s so many things happening. There are people talking, there are machines beeping and so much sound. If my computer is supposed to sound and blink, I mean... No... no...

I: So, information types. What information would you like to see in such a system? I already mentioned examples like schedules.

Q: It has to be information that is directed to my work, I do not want any information about the price of coffee in the cafeteria.

I: Another example that I maybe didn’t mention actually would be schedule changes. For example if someone changes your schedule, they say “We don’t need you” or “Off tomorrow” or something like that. Maybe it’s a red day and you’re not working. You probably already got this information, for example in an email or in a meeting. But do you think it would be prudent to show this message in the message window? And this is only for you.

Q: I have a hard time picturing it because, I mean, I work red days. When you have a full-time employment, I guess you already have your schedule and it’s very seldom that they actually change it. And you have like, conversations face to face with your boss and stuff like that. So I have a hard time picturing it being useful. But, I guess, like I said earlier, a message from your boss saying “It’s time to hand in your vacation forms”, that would be super. Right now there are notes everywhere in the personnel room, so why not have it on the computer as well? And if there was someone coming in from time to time, like the dry cleaner who washes all the clothing for the patients and the staff, they
sometimes have information moments. That would be really useful, because, that’s another note among the others on the whiteboard. That I think would be a great idea.

I: Okay. That’s interesting, since it’s more nurse related. With a perspective from the inner workings. When you for instance have the described card system and various people use the same unit. We would be able to get the information of who is logging in or using the system. Therefrom we would know where they are or all related information in the database. Would you like to see personal messages that are derived from the information.

Q: All the system that I am using, for instance at the internship, all of them know that it is me who is logging in. So I can’t see a problem with that.

I: What about whether we would allow the supervisor to add new messages? Do you think that the messages could be good? Is there to big a risk that he will inform you about important information only through this system.

Q: I think that it is good to reinforce information with for instance email and this system. But I wouldn’t like the information to arrive through this system only. There is a bigger risk that you miss information this way, that’s why it is important that the information is sent via email and only use this message delivery system for reinforcement.

I: Last question, although it may not be. But in any case it’s the last that I’ve written down. One of the future functions that we would implement would be forms or questionnaires. Let’s say we add a new functionality, for instance a delete button that enable you to delete messages that we described earlier. And then we would add a form that would ask the user whether this function is good or bad.

Q: But are all these forms related to the software?

I: No, it could for instance be from your supervisor who would ask the staff questions. For instance when a new plastic glove has been bought and he or she asks whether they are crap or good.

Q: Yeah, it could be good. But would this be something the user would have to answer?

I: No, it would be fully voluntary and shown in the same fashion as the messages, with that I mean that it would only be shown for about 5 min.

Q: Yeah that could be good.

I: All our messages would as I mentioned be about 20 words as the form would be. That way it wouldn’t be too distracting. So, do you see the various functions for this system. Would you like this in your system?

Q: Yes, but you could possibly make it less distracting, but yeah why not. Us nurses sit a lot of time infront of a computer. We often sit about 10 minutes and leave, and later return to sit about 10 minutes more. So if we could get a message from our supervisor that way, why not!?

I: Okay, anything you want to add?

Q: No, I think I managed to squeeze everything in.

I: Very well, thank you for your time!
Transcript 3 - Computer engineer

I: Now I’ve told you about the add-on, let us start with the questions.

I: Do you think an add-on like this could be useful?

Q: Yes, I think it would be very useful to receive information about my company or what is happening in my job environment. The text messages should be clear and easy to read.

I: Would you like an notification that says the message have been read?

Q: Yes, that would be great. But what happens when a message is read, does it disappear or will it stay on the screen?

I: It will stay on the screen, nothing more will happen with it. Do you think it would be convenient if you could re-read a message?

Q: Yes, of course, it would be great if the messages could be listed after each other and that they had a subject field, like e-mails. That way you could click the message you would want to read again.

I: How would you like to be able to go back and reread a message, by pushing a button and click until the message returns or have the messages listed after each other, like on an e-mail application?

Q: If it is a button it should be clear that it is used in order to go through the messages. But yes, it would be a great idea! Especially since the working days are stressful and it is therefore good if you could look back at the messages whenever you have more time on your hands.

Q: The messages should also have a subject field where you can see what the message is about and also the time of when the message was sent. I also want the message to change color when it has been read, forget to say that earlier.

I: Ok, do you also want the message to have “read” written next to it when you have read it or do you think it is enough if it changes its color?

Q: Yes, but it could say “read” only if you accidentally click it and that might make me forget to read it properly.

I: Ok, so I guess it is not a good idea then?

Q: No, I think it is enough with only the message changing color.

I: Let us talk about how to get a notification about a new message, how would you like to be notified about a new message coming in.

Q: A sound that tells you that a new message has arrived. I would also like a time stamp so I know at what time the message arrived.

Q: When you open the application for the first time during a day it will not be a problem to see the message but later when you are working you need a notification that a new message has arrived. I need a clear notification because during the day I am very busy and very concentrated on my work, it will be very hard for me to notice that a new message has arrived unless I get a clear notification. A sound would be good, just like when you receive an e-mail or sms.
Q: I would like to have the possibility to be notified by both a sound and that the message add-on starts blinking.
I: Would it be enough with only a sound or should it also blink?
Q: I would like a sound and that it blinks. If there are several messages it should also have a number attached to , this makes it easier for us users to know how many messages we have received during the day or the week.
I: If we look at the design, do you think the add-on should stand from the rest of the design or would that make it a distraction for the user?
Q: I think the design should be different from the rest of the application, otherwise it will be hard to notice when a new message has arrived. It is important that the users get the information that has been sent and if the message delivery add-on doesn’t stand out there is a risk that the message will go unnoticed.
I: What kind of information do you think should be sent to the user through the message delivery add-on?
Q: Everything that is related to the job and concerns every or most of the users in the company can be send through the message delivery add-on. More personal information is better to be send through e-mail.
I: Do you think only a certain type of information should be sent or do yo think personal information also should be sent such as schedule changes or you actual schedule for the week/day?
Q: Everything that is work related would be useful . Will they still send e-mails?
I: Yes
Q: Ok, well, then I think only information that is relevant for most users should be sent, more personal information can be sent through e-mail i think.
I: Do you think it would be convenient if the user could fill in forms through the add-on?
Q: No, that would not be a good idea. It will be annoying for the users if they have to fill in forms while they are working.
I: You will not be able to write but only click on different options, for example a poll about the latest update.
Q: No, I don´t like that idea. I think it is better to use it only as an information source!
I: Ok, let us talk about the image, do you think the message window would be better without the image or do you find a relating picture with the text message the add-on more useful?
Q: The image must of course be relevant for the message but yes, of course it should be a corresponding image with the image. It could be used just like in the news when they talk about a country you can see an image of a map and the country is highlighted. The only important thing is to not make the image taking focus from the text message .
I: For example if the message is talking about an area the image could show the map of the area highlighted, would that make the image useful?
Q: Yes, exactly, that would be great!
I: Anything else you would like to add?
Q: No, I think it is a very useful application, just make sure the font is sharp and that the messages are not too long!

**Transcript 4 - Project manager**

I: What do you think about the design of the add-on?
Q: I like the design but I think you should think of making the letters as clear as possible because it should not be too much of an effort to read the messages.
I: Do you think the add-on should stand out from the rest of the application?
Q: Yes, definitely, otherwise it might be ignored or forget that it exists in the application.
I: Should the picture window be bigger?
Q: No, smaller actually. What kind of pictures will be shown?
I: It will show a picture related to the text message, for example it might show a map of the area that has been affected by an epidemic. (What do you think about having a relating picture with the message?)
Q: I think it is great having a picture related to the text message as long as the picture is not something random but something that actually makes the text message more understandable.
I: For how long should a message be shown? Should you be able to revisit a message or should they overwrite each other?
Q: You should be able to reread a message. Maybe they can show a list of the messages sent and then you can click the message you want to read.
I: From how many days back should messages be kept?
Q: It should have the messages that have been sent during the day but having the messages from several days is not that important.
I: Should it have a notification that tells if a message has been read?
Q: Yes, I think that would be useful. It would be good to know that you have already read a message.
I: Do you think it would be useful to be able to answer forms through the add-on?
Q: No, as a user I would not want to spend time answering forms when I’m working. I think that would only give the users reason to ignore the add-on. In a stressful environment the last thing you want is to spend time on things that is not helping you making progress on your daily work.
I: Should you only to be able to receive job related information through the add-on or should it also send you information that is not directly connected to your daily work such as the timetable of the cafeteria?
Q: No, not instead., I think it is better to send it through e-mail since some information have to be saved for longer than a day. Can the add-on only send information to everyone or also to some specific users?
I: No, it will be possible to send information only to one or a few users.
Q: No, then it would be a danger to not read essential information, I do not think this add-on should be used to send information that could be sent through e-mail. The only way would be to send it both through e-mail and the message delivery add-on but it would be giving the same information twice and that would only be annoying for the user.

I: Do you think the add-on could be seen as a distraction?

Q: Yes, if it is updated or new messages are coming in to often.

I: How would you like to be notified about a new message, by a sound or maybe by the message window blinking?

Q: I would prefer to be notified by a sound, blinking would be annoying I think. Maybe instead of blinking it could change colour, for example from blue to red. A sound and a change of colour would be the best in my opinion.

I: There won´t be more than 2-3 messages /day.

Q: Ok, then it would be ok if it changes color, don´t want it to blink but changing color and making a sound would be convenient. The sound notification should be optional because some people could see it as a distraction.

I: If we would think of the add-on to be used for personal use and not in a working environment, do you still think it should stand out from the rest of the design?

Q: No, then I think it would annoy me but on the other hand I would probably ignore it entirely if it does not stand out from the rest of the design.

I: Do you think it would be better if the time of when the message was sent should be displayed?

Q: Yes, I think it is important to know at what time the message have been received.

I: Should you also add the time of when the message was received?

Q: Yes, I think that would be useful information.

I: The message delivery app could have the possibility to send information to only one or some users, do you think that would be useful or should it only send information that concerns all the users? For example changes in your(schedule).

Q: I think it could be useful, as long as it is job related information I´m happy with receiving it directly on the screen. The only problem would be if it sends information to often, as long as it is very useful information, like changes in my schedule, I would be happy with receiving it through the message window.

I: Something else you think the add-on should have but that have not been brought up?

Q: Not really, adding the time on the messages is essential, that is the only thing I can think of even though we discussed it earlier.