

# The potential of crowd sourcing applications in organizational context – A railroad case study



**logica**  
be brilliant together

**UNIVERSITY OF TWENTE.**



**Author**

Name: D. (Durk) Boersma BSc  
Phone number: + 31 6 41194330  
E-mail address: durk@boersma.nl  
Institution: University of Twente  
Study: Business and IT  
Student number: 0137294  
Address: Erwtenland 5  
9205ER Drachten  
The Netherlands

**University of Twente**

First supervisor: Dr.Ir. J.M. (Hans) Moonen  
Faculty: School of Management and Governance  
E-mail address: hans.moonen@utwente.nl  
Second supervisor: Dr. N. (Klaas) Sikkel  
Faculty: Electrical Engineering, Mathematics and Computer Science  
E-mail address: k.sikkel@utwente.nl  
Address: Drienerlolaan 5  
7522NB Enschede  
The Netherlands

**Logica Netherlands**

Company mentor: Roland Dijkstra  
E-mail address: roland.dijkstra@logica.com  
Company mentor: Margriet Meerholz  
Email address: margriet.meerholz@logica.com  
Address: Eemsgolaan 1  
9727DW Groningen  
The Netherlands

# MANAGEMENT SUMMARY

## Motivation

Since the introduction of Web 2.0, the characteristics of it become more and more visible in our everyday life. Moreover, these characteristics also become visible in organizational contexts, often labelled Enterprise 2.0. This phenomenon makes organizations more connected, both internally and externally. Organizational leaders claim to benefit from Enterprise 2.0. One of the techniques that makes this possible is crowd sourcing, which is the use of large undefined groups or customers within the business process. In this thesis, we research the potential of using crowd sourcing applications in organizational context. Since NS currently encounters problems, and customers notice these problems, NS is an interesting context to conduct a case study.

## Results

We performed a literature study to find the most important characteristics of crowd sourcing applications, the best practices and success factors. Based on this literature study we developed a complete and validated set of requirements for a crowd sourcing application for NS for feedback on the passenger information in the train. To test and validate our ideas we created a prototype, a feedback system for passenger information in the train. This prototype application was tested by 63 passengers in the train and a questionnaire was conducted. From the test and questionnaire we drew the following conclusions:

1. There seems to be enough support from passengers to make in-train crowd sourcing a success.
2. It is crucial for the success of such a crowd sourcing application that complaints and suggestions made by passengers are acted upon and some form of feedback is given to these passengers.
3. There are different channels to implement in-train crowd sourcing: portal website, mobile application, or both. Results indicate that the portal website alone is not enough.
4. Publicity and incentives are key factors in the success of the crowd sourcing application.
5. In-train crowd sourcing can be implemented as stand-alone application or part of a social media environment.
6. Previous results by van der Wees and Moonen are confirmed:
  - The most valuable data for NS is a better understanding of who is travelling their trains and at what trajectories. Within the train [...], feedback on the sentiment of the journey, the attractiveness of the train environment, and the feeling of safety are most sought after.
  - Most people clarified that if they felt that it was useful, they would likely keep using the application.
  - Customers are most likely to adopt applications that give them the feeling they contribute to something worthwhile. [...] Applications closer to the core business are more likely to get adopted.

Since our research shows that passengers have the willingness to participate in a crowd sourcing application, our advice is to implement such a crowd sourcing application for use in the train. However, we advise to wait with the implementation until there is a near-to-perfect back-end for the application. Results indicate the willingness to participate, but if NS does perform action, the willingness and number of contributions will decrease. Our final advice is to implement this crowd sourcing idea as part of a bigger whole, the social media environment.

## **Consequences**

All of this could contribute to NS having a better brand image to the outside world and an improvement in customer satisfaction. However, certain preconditions need to be fulfilled for this to come true. If passengers do not know how to find the application, it is not effective. Worse, if passengers do find it but realize their comments are not noticed by NS, it will backfire and reduce customer satisfaction. In order to avoid these risks, NS should wait with the implementation until there is a good process for processing and solving the passengers' comments. Additionally, NS should choose the right channel for implementing such an application: mobile, portal website or both. Finally, it is important that NS finds a good fit between the back-end process and the number of responses; the more responses, the better the back-end needs to be.

# ACKNOWLEDGEMENTS

This document contains my master thesis, the final document for my master Business and Information Technology at the University of Twente. It describes the results of my research on the potential of crowd sourcing applications in organizational context, which I carried out at Logica. The last few months have proven to be challenging months, both personally and organizationally. The IT-services sector is going through challenging times, something that was certainly noticeable at Logica. Finishing this research project means a lot to me and could not have been possible without the help of many people.

I would like to start by thanking my supervisors from the university of Twente, Hans Moonen and Klaas Sikkel. Hans, thank you for introducing me into the worlds of Logica and NS. You were there when appointments had to be made at NS, when conversations were cancelled and results were dissatisfying. With your guidance and enthusiasm, we found a research approach that worked out. Klaas, thanks for your continuous feedback and sharp criticism, your feedback and insights were always valuable.

I would also like to thank my fellow colleagues and supervisors at the Logica office in Groningen. My fellow colleagues provided a pleasant atmosphere to work on this project and the informal conversations at the coffee machine brought me new insights, hints and feedback. The supervisors, in order of appearance, are thanked for their enthusiasm and quick reasoning: Margriet Meerholz, Jacob Mulder, Paul Zondervan, Roland Dijkstra and Arjan Stoter. A special note goes out to Erik Verhoeve, who provided me with great new insights on functional designs and always took the time to give me feedback.

With NS being the subject of my case study, I would like to thank them for the participation and openness I needed for my project. I would like to thank Herbert van Buitenen and Bram Hansma in person, because they really helped me with validating my results. Furthermore I would like to thank Erik Boshoeve and Piet de Roo for their criticism along the way.

Of course, this acknowledgement would not be complete without thanking my parents and family, who supported me throughout my entire study and encouraged my activities. Thanks to the loving support and faith of my parents, I was able to finish my study and make it a great time to look back on. Anna and Folkert, thanks for your criticism and feedback on the draft versions of this thesis. My final acknowledgement goes out to Janneke, she has been loving and patient throughout my graduation.

I hope you will enjoy reading my master thesis about the potential of crowd sourcing applications in organizational context. If you have any questions, please feel free to contact me.

Durk Boersma

Drachten, August 2012

# TABLE OF CONTENT

<b>Management summary</b>	<b>4</b>
<b>Acknowledgements</b>	<b>6</b>
<b>List of Tables</b>	<b>9</b>
<b>List of Figures</b>	<b>10</b>
<b>1. Introduction</b>	<b>11</b>
1.1 Motivation	11
1.2 Environment	11
1.3 Problem statement	12
1.4 Research approach	13
1.5 Contribution	15
1.6 Outline	15
<b>2. Problem Investigation: Literature study</b>	<b>17</b>
2.1 Enterprise 2.0	17
2.2 Crowd sourcing	18
2.3 Examples of crowd sourcing in everyday life	19
2.4 Crowd sourcing in organizational social media context	20
2.5 Crowd sourcing in public transportation	21
2.6 State-of-the-art and best practices in the field	23
2.7 Literature on testing and validation methods	26
<b>3. Solution Design: In-train crowd sourcing</b>	<b>28</b>
3.1 Solution design initiation	28
3.2 Introduction of the Dutch railway system and OBIS	28
3.3 Designing the solutions	29
3.4 Introduction and goals of the stakeholders	31
3.5 Requirements elicitation for the reporting system	32
3.6 Requirements elicitation for the feedback system	35
3.7 Functional design for both solutions	36
3.8 Technical design	37
<b>4. Solution Validation: Questionnaire</b>	<b>38</b>
4.1 Testing and validation platforms	38
4.2 Prototyping the feedback system	38
4.3 Method of validation	40

4.4	Developing the questionnaire	40
4.5	Execution of the validation research	43
<b>5.</b>	<b>Results from the questionnaire</b>	<b>45</b>
5.1	Results from the general questions	45
5.2	Results on scientific models	48
5.3	General considerations	51
<b>6.</b>	<b>Solution validation: Expert interview</b>	<b>53</b>
6.1	Expert interview at NS	53
6.2	Validation questions from the regulative cycle	56
<b>7.</b>	<b>Conclusion</b>	<b>59</b>
7.1	Results	59
7.2	Reflection in scientific context	60
7.3	Reflection in the context of Dutch Railways	61
7.4	Reflection in the context of Logica	65
7.5	Challenges and future research	66
	<b>References</b>	<b>68</b>
	<b>Appendix A: Collective intelligence genes</b>	<b>71</b>
	<b>Appendix B: Functional design</b>	<b>72</b>
B.1	Requirements elicitation process	72
B.2	Use case modelling	72
B.3	Use case matrix	72
B.4	Prioritization	73
B.5	Functional use case specifications	75
B.6	Use case specification for the feedback system	81
	<b>Appendix C: Technical design</b>	<b>84</b>
C.1	Technical use case A1: Submit contribution	84
C.2	Technical use case B1: Submit contribution	87
C.3	Implementation of the non-functional requirements	90
	<b>Appendix D: Prototyping the feedback system</b>	<b>92</b>
D.1	Deployment of the prototype	92
D.2	Testing the application	93



# LIST OF TABLES

Table 1.1: Regulative cycle throughout the thesis.....	16
Table 2.1: Key managerial challenges and Enterprise 2.0 applications .....	18
Table 2.2: Overview of the genes .....	23
Table 2.3: Variations of the how gene for crowds .....	24
Table 2.4: Example of a gnome.....	25
Table 3.1: Example of an empty gnome for crowd sourcing applications.....	30
Table 3.2: Gnome for the reporting system .....	31
Table 3.3: Gnome for the feedback system.....	31
Table 3.4: Overview of the non-functional requirements for the RSTC.....	33
Table 3.5: Overview of the functional requirements for RSTC .....	35
Table 3.6: Functional Requirement for FSPI.....	35
Table 3.7: Use case list for the reporting system for train compartments.....	37
Table 4.1: General questions.....	41
Table 4.2: revising the organization genes for crowd sourcing.....	41
Table 4.3: Questions on the organization genes model.....	42
Table 4.4: Questions on the four challenges model.....	43
Table 4.5: Research data for July 6.....	44
Table 4.6: Research data for July 10 .....	44
Table 4.7: Research data for July 11 .....	44
Table A.1: Overview of the collective intelligence genes model.....	71
Table B.1: Use case matrix.....	73
Table B.2: Prioritization matrix for the non-functional requirements using the MoSCoW-method .....	74
Table B.3: Prioritization matrix for the functional requirements using the MoSCoW-model .....	75
Table B.4: RSTC functional use case 1 .....	77
Table B.5: RSTC functional use case 2 .....	78
Table B.6: RSTC functional use case 3 .....	79
Table B.7: RSTC functional use case 4 .....	79
Table B.8: RSTC functional use case 5 .....	80
Table B.9: RSTC functional use case 6 .....	81
Table B.10: RSTC functional use case 7 .....	81
Table B.11: FSPI functional use case 1 .....	83
Table C.1: RSTC technical use case 1 .....	85
Table C.2: Structure of the feedback specification for the reporting system .....	86
Table C.3: Database scheme for the reporting system.....	86
Table C.4: FSPI technical use case 1 .....	88
Table C.5: Structure of the feedback specification for the feedback system .....	89
Table C.6: Database scheme for the feedback system .....	90
Table C.7: Overview of the Must-have non-functional requirements .....	90

# LIST OF FIGURES

- Figure 1.1: The regulative cycle by Wieringa .....13
- Figure 2.1: Social environment as proposed by Divol .....20
- Figure 2.2: Organizational genes for crowd sourcing .....23
- Figure 2.3: Conceptual framework for testing an validation .....27
- Figure 4.1: Screenshot of the portal website .....39
- Figure 4.2: Screenshot of the application prototype .....39
- Figure 5.1: Venn diagram of the results on Q1 .....45
- Figure 5.2: Results on Q2 .....46
- Figure 5.3: Results on Q3 .....46
- Figure 5.4: Results on Q4 .....47
- Figure 5.5: Results on Q5 .....47
- Figure 5.6: Results on Q6 .....48
- Figure 5.7: Results on Q9 .....49
- Figure 5.8: Results on Q10 .....50
- Figure 5.9: Number of responses per hour of the day .....52

# 1. INTRODUCTION

*This chapter is meant as an introduction to this master thesis. It discusses the research objectives and approach, the motivations for doing it and the contributions made to the scientific field. Furthermore, this chapter gives an outline to the rest of this thesis.*

## 1.1 Motivation

Since the introduction and growth of Web 2.0, which makes online collaboration easier, more and more organizations try to use the knowledge and ideas of customers and outsiders in their business activities. These people are often labelled 'the crowd' and using the crowd for a specific reason is called crowd sourcing (Howe, 2006a; Howe, 2006b). Results thus far are varying, but examples like YouTube (2012a) and Threadless (Brabham, 2010) are an immense success. During the Logica innovation event Transform11, which took place on May 17<sup>th</sup> 2011, Logica colleague Hans Moonen presented the idea of using the crowd at Nederlandse Spoorwegen (Logica, 2012a). Over one million passengers travel by train every day (NS, 2012b), but there is little information known from the passengers in the train. Since so many people travel by train on a regular basis, these people are very close to the services of NS and could have great ideas or comments on these services. In other words, this crowd of passengers could be of great value for NS.

At the same time, NS is modernizing its intercity trains by introducing the On-Board Information System (OBIS). This information system, which is implemented in the renovated intercity trains, is currently used for two applications. The first application is to provide passengers a Wi-Fi internet connection. To use this Wi-Fi internet connection, people in the train can use their personal laptop, tablet or phone to visit the portal website of the train and accept the terms. The second application is to provide passengers with travelling information, which is presented on screens in the train and on the portal-site. Both applications are implemented to provide a better travelling experience to the passengers in the train. Since the OBIS has overcapacity, new applications could be added in the future.

## 1.2 Environment

Several organizations are involved in this research project. In this section these organizations and the role they play will be introduced.

### 1.2.1 Logica

Logica provides business and technology services and has 45.000 employees worldwide. The company offers business services, system integration and outsourcing services to customers all over the world. Logica is basically divided into three separate business pillars: Business Consulting, Professional Services and Outsourcing Services. Business Consulting has a focus on advising customers regarding the design, transformation and implementation of IT systems and services. Professional services has a focus on the delivery, support and design of new IT systems. Outsourcing Services is the pillar that gains the most revenue on a yearly basis. The main tasks within this sub organization are Application Management, Infrastructure Management and Business Process Outsourcing.

This graduation project is executed within the Working Tomorrow program at Logica. This program was initiated to give students the opportunity to study and do research in an innovative company. Students are working on

agent technology, multi-touch technology, augmented reality, smart grids, smart meters and mobile solutions. Approximately 120 students from Higher Professional Education and University graduate annually.

### 1.2.2 University of Twente

The University of Twente is one of three universities of technology in the Netherlands. The university was founded in 1961 and is the only university in the Netherlands which has a campus. In 2011 almost 9500 students were registered at the University of Twente. The School of Management and Governance (SMG) offers the Business Information Technology Master's program in association with the School of Electrical Engineering, Mathematics and Computer Science (EEMCS). IT is of great significance in virtually every sector – financial services, hospitals, transport or telecommunications. For businesses it is crucial to be able to deploy new technology successfully. The Business Information Technology program is a combination of computer science and business administration. Students are educated to be professionals who can bridge the gap between those two fields. The development and implementation of systems in businesses forms the heart of Business Information Technology, with a focus on both technological and human aspects.

### 1.2.3 Nederlandse Spoorwegen

The NS is a Dutch railway organization. Each day, NS has to take care of the transportation of over one million people. This is done using 4.800 train rides daily. NS also delivers additional services, like bus transportation, in the door-to-door travel chain. In addition, the management of over 400 train stations in the Netherlands is done by NS and together with several partners they try to create dynamic transportation hubs at busy junctions in the Netherlands (NS, 2012a; NS, 2012b). The NS, more specifically the department NS Reizigers, provides the context for this research project.

## 1.3 Problem statement

According to Howe (2006a; 2006b) first results of scientific research show that crowd sourcing is a relatively cheap and easy to use mechanism with high potential for organizations. In this research project we want to explore the possibilities of the crowd in a real organization. This leads to the following problem statement:

*“Explore the potential of crowd sourcing in organizational context, by introducing a crowd sourcing application and conducting a case study.”*

Brabham (2010) claims that both a technical platform and a crowd have to be present for crowd sourcing to be successful. Since NS can provide both the technical platform (OBIS) and the crowd (passengers), NS seems to be a good context for a case study.

In the 2011 annual report, the NS state in their mission that they want to move more and more passengers safely, on time and comfortably via attractive stations (NS, 2012). Information on customer satisfaction shows that in some of the areas this mission has been accomplished. However, some other areas lag behind, meaning that the customer satisfaction in these areas is not growing. The image of NS is a hot topic in the media lately, and not always in a positive sense. From a business point of view, improving the customer satisfaction should be one of the key goals. The fact that NS has no clear image of what is going on in the minds of passengers seems to indicate the reason why customer does not improve. Next to that, passengers cannot share their opinion with NS easily.

## 1.4 Research approach

The objective of this research project is to find out what the possibilities of crowd sourcing are for NS. The research approach that has been chosen for this project is the regulative cycle by Roel Wieringa (2009) based on van Strien (1997). The same research approach has also been termed engineering cycle (Wieringa and Heerkens, 2006; Wieringa et al., 2006). This research model can be considered as an iterative cycle with four stages, which is depicted in Figure 1.1. The first iteration of the cycle starts off with the problem investigation. Subsequently, a phase of solution design occurs. The third phase is used for the validation of the design and the fourth phase is used to implement the solution. After the first iteration, the second iteration will use the same phases. The iterations can be continued until the stakeholders are satisfied. Since this research project has a limited amount of time that can be used, only one iteration of the regulative cycle will be conducted and described in this thesis. In the following four sections, sections 1.4.1 through 1.4.4, the four phases of the regulative cycle will be described. For each of the four phases the main goal of the phase is given. Moreover, we provide a description of how each phase will be operationalized in this research project.

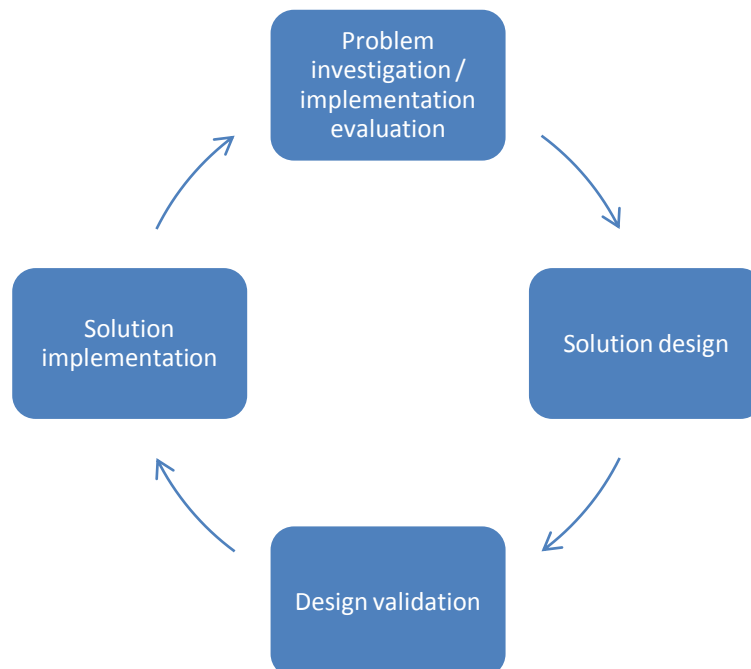


Figure 1.1: The regulative cycle by Wieringa

### 1.4.1 Problem investigation

In the first phase, the goal is to find information about and understanding of the given problem, to describe the problem and to explain it. According to Wieringa it is useful to distinguish four non-exclusive reasons for investigating practical problems. Each of these reasons implies different activities in the problem investigation process. The following section contains a short description of the reasons that Wieringa identifies.

In **problem-driven investigation** the stakeholders experience some kind of problem, which has to be analyzed before solving it. An example of such a problem could be a supermarket encountering problems in supplying different stores in different locations. **Goal-driven investigation** describes the situation where a problem may not be experienced, but reasons to change something do occur. This situation occurs for instance when the government implements a new law, by which organizations need to change in order to comply. **Solution-driven**

**investigation** can be considered as the situation in which new technology can be used to solve problems. With **impact-driven investigation** the focus of the research is not on future solutions, but rather past actions. This means that this type of research is suitable to evaluate and analyze the implementation of new system.

From the problem statement in section 1.3 and the description of the different reasons to investigate practical problems, it becomes clear that this research project needs solution-driven investigation. According to Wieringa (2009), this means that “problem investigation in this case would start with an investigation of the properties of the new technology; and solution design would be an exploration of ways in which it could be used to achieve new goals”. Therefore, the problem investigation needs to contain information on the new technologies to be used.

This project will use the problem investigation phase to perform literature research. This literature study will be focussed on the new technology to be used, starting with the application of Web 2.0 in organizational settings. From here we will focus more on specific areas. Also, we will use this phase to study literature needed for the design validation phase, hence validation methods and practices.

#### 1.4.2 Solution design

The solution design phase can be used to design a solution. In other words, we will design a solution that solves the problem as stated in section 1.3. Wieringa points out that solutions might be designed that will turn out to be useless after implementation, or even make things worse. In addition, it should be noted that a solution is always designed to some extent. Often solutions are not completely specified before validation or implementation.

In this research project, the solution design phase will be used to introduce the context of this research and to explore the possibilities of the technologies found in the problem investigation phase. Once the possibilities are clear, we gather requirements, make a functional design and a technical design.

#### 1.4.3 Design validation

The design validation phase will be used to validate the solution that has been designed in the previous phase. According to Wieringa (2009) three kinds of validity can be analyzed, in which the first validity has two subcategories:

- **Internal validity:** Would this solution design, implemented in this context, satisfy the criteria identified in the problem investigation phase?
  - **Causal question:** In this specific problem domain, would this solution have the intended effects?
  - **Value question:** Do the effects satisfy stakeholders?
- **Trade-offs:** How would slightly different designs, implemented in the same specific context, satisfy the criteria?
- **External validity:** Would the design, implemented in a slightly different context, also satisfy the criteria?  
This kind of validation is also known as sensitivity analysis.

During this research project, the design validation phase will be used to validate the design developed in the solution design phase. This will be done with help of colleagues at Logica, other students in the Working Tomorrow program and a brainstorm with two representative train passengers. Furthermore, we will use the literature from the problem investigation phase as a basis in which we hope to find a suitable validation method. Additionally, we want to answer the validity questions as stated in the regulative cycle by Wieringa (2009).

#### 1.4.4 **Solution implementation**

This phase of the research approach will be used to implement a prototype of a crowd sourcing application. In his research paper, Wieringa (2009) is of the opinion that “what constitutes an implementation depends on what solution has been designed”. If, for example, the goal of some project is to bring a new home appliance to the consumer market, and a process to achieve this has been designed, then the implementation is to execute that process. If on the other hand the goal was to test this new home appliance, and a series of validation tests is designed, then the implementation is the execution of those tests according to the plans. In this specific research context, the implementation phase is like the first example. However, since such an implementation typically has to be performed by NS, this is something which is not part of this project.

### 1.5 **Contribution**

In this section the relevance of answering the research question will be pointed out, both the scientific relevance and the managerial relevance.

#### 1.5.1 **Scientific relevance**

Due to the use of information technology and the diversity in the sourcing of workforce, working environments have changed over the last years (Vijaykumar, 2010). In traditional situations, organizations hired people to perform tasks inside the organization. Nowadays, people from all over the world can perform tasks for organizations, be it dependently or independently of each other. Although a lot of new and innovative things can be done using crowds, we discovered only few publications concerned with public transportation. With the results of this research effort, we hope to contribute new information about the potential of crowd sourcing applications in organizational context, specifically public transportation.

#### 1.5.2 **Managerial relevance**

From a managerial point of view, the result of this research can be useful for organizations. First insights (Howe, 2006a) tell that the use of crowds is relatively cheap and easy to implement. If organizations can benefit from the idea of using the crowd, this could be a relatively cheap and easy way to improve their services. The scientific publications (Doan et al., 2011; Malone et al., 2009) that have been found on the design and implementation of crowd sourcing systems can help organizations to design and implement better and more efficient crowd sourcing systems. In the context of Logica, the result from this assignment can also be relevant. From a business point of view, Logica is always looking for innovative ideas and solutions. Since the use of crowds in public transportation is something which is relatively new and innovative, it is something which fits their business ideas.

### 1.6 **Outline**

The outline of this thesis is inspired by the regulative cycle as described in the research approach section. This means that we describe each phase in a different chapter, using two chapters for the last phase. Table 1.1, on the following page, shows in which chapter we start a specific phase.

---

Topic	Chapter
Problem investigation	2
Solution design	3
Design validation	4-6

**Table 1.1: Regulative cycle throughout the thesis**

**Chapter 2 – Problem investigation: literature study** describes the phase of problem investigation in which we perform literature research on the scientific fields of Web 2.0 in organizations and testing and validation methods.

**Chapter 3 – Solution design: in-train crowd sourcing** introduces the context in which this research project will be placed. Also, we will present and design a solution to the problem statement.

**Chapter 4 – Solution validation: Questionnaire** presents the validation methods that will be used in this research context and describes how these methods will be developed.

**Chapter 5 – Results from the questionnaire** describes how the questionnaire from the solution validation phase is executed and what results are gained from the questionnaire.

**Chapter 6 – Solution implementation: Expert interview** present the second part of the implementation phase, the expert interviews with stakeholders from NS.

**Chapter 7 – Discussion** is meant to close this research project and discuss the results. Moreover, we reflect on future challenges for the stakeholders in that participated in this research project.

In addition to the chapters stated in the overview above, four appendices are included at the end of this thesis. The first of these appendices is a supplement on one of the scientific models stated in chapter two. The second of these appendices is a supplement on the functional design in chapter three. The third appendix is a supplement on the technical design, which is also stated in chapter three. The last appendix is a more detailed description of the prototyping phase, which is described in section 4.2.



## 2. PROBLEM INVESTIGATION: LITERATURE STUDY

*In this chapter, an introduction is given to the scientific field of user contributions to organizations. The chapter starts with an introduction and a definition of crowd sourcing. In the following section some examples of crowd sourcing in everyday life are presented. In section 2.3 the examples of crowd sourcing are focused on the area of public transportation, since that is the field of interest for this thesis. In section 2.4, the state-of-the art in this field is described. The last section of this chapter presents scientific literature on the possibilities for testing and experimentation platforms.*

### 2.1 Enterprise 2.0

In the introduction we already presented the phenomenon of Web 2.0. It is commonly known that the internet is always moving. The bursting of the dot-com bubble in 2001 marked a turning point for the internet as we knew it. Although some people thought the internet was on its return, others saw new possibilities for the web to emerge. This emerging trend was later on termed Web 2.0. The exact definition is hard to give; the core principles are commonly stated. One of these overviews is given by Alexander and Levine (2008), from which the most important characteristics are stated below:

- **User-centred design:** This means that web designs are created in ways that fulfil the user's needs and give the user the freedom and possibilities to customize parts of the design.
- **Crowd sourcing:** In a Web 2.0 environment, every contribution is useful and many contributions together form one big system. In practice this means that users of a certain Web 2.0 environment contribute to that environment.
- **Web as platform:** In earlier days, the content and look of a web service was heavily dependent on the operating system and plug-ins that users had installed. In Web 2.0 the services do not require client downloads, nor are the services operating system dependent.
- **Collaboration:** Web 2.0 services give great possibilities for collaboration. A good example of such collaboration is Wikipedia, which is considered more accurate and complete than Encyclopaedia Britannica, but is built only by user contributions.

All these characteristics together define Web 2.0. But more interesting in this context is the use of Web 2.0 in organizations. The use of Web 2.0 in organizations, termed Enterprise 2.0, is first discovered in an article in MIT Sloan Management Review by Andrew McAfee (2006). In this article six Enterprise 2.0 technologies are presented, which converge to the original four characteristics by Alexander and Levine (2008): Search, Links, Authoring, Tags, Extensions and Signals. This research is further extended by a KPMG Enterprise 2.0 research by Matuszak (2007), which presents applications of Enterprise 2.0 in the four key managerial challenges of the twenty-first century. Table 2.1 gives an overview of the four key managerial challenges and the applications of Enterprise 2.0 in these challenges, as derived from Matuszak (2007).

Knowledge sharing and management	Problem solving	Innovation	Collaboration
Collaborative document production	"War room" for fast-changing situations	Broadcast search	Customer relations
All-purpose "team ware"	Sharing computing power and brainpower	Crowd sourcing	Transformative ideas
Knowledge management	Answering questions	Expressing collective judgment	
Knowledge broadcast			

**Table 2.1: Key managerial challenges and Enterprise 2.0 applications**

One of the results of using Enterprise 2.0 in organizational context is that a different kind of company emerges: the networked company. According to (Chui et al., 2009), two different kinds of networked organizations can be distinguished: the internally networked organization and the externally networked organization. Internally networking organizations can be recognized by the internal integration of Web 2.0 tools among employees, and externally networked organization by the use of technologies to strengthen the ties with customers and business partners. In other words, by being in touch with your customers (external networking) and sharing this information in the right way (internal networking) organizations can benefit from Enterprise 2.0. The same research (Chui et al., 2009) shows that 69% of the respondents measure business benefits like "more innovative products and services, more effective marketing, better access to knowledge, lower cost of doing business and higher revenues".

One of the thriving characteristics for external networking, as stated in table 2.1, is crowd sourcing. We will explore this concept in more detail in the following section.

## 2.2 Crowd sourcing

Traditionally, organizations hire people to do work for them. Those people can be hired in different ways, but the main characteristic is that the people who perform the jobs are in the organization. But as years progressed and innovation came along, other models for sourcing labour were found (Vijaykumar, 2010). Due to the technological, social and economic growth in the early ninety's, many economies were opening up and organizations outsourced work to markets where work would cost less. In the middle of the last decade, the sourcing of labour became more open and the term open source emerged. This kind of sourcing was mostly used in the IT industry. In this kind of sourcing, people could participate in work and the source became available for free. The newest idea on sourcing, which might in part be due to the emerging Web 2.0 and social media, is to involve consumers in the labour process. This means that people, and also communities, from outside the organization can have influence on some of the organization's processes. In a sense, this could be seen as voluntary user contributions to the organization, since most of the time the reward for participation is not that big.

Each day, millions of people are involved in voluntary user contributions, especially on the web. With examples as Wikipedia (<http://www.wikipedia.org>), which is built completely by voluntary user contributions, the concept is not to be ignored. A taxonomy for voluntary user contributions was provided by Cook (2008), which distinguishes

between active and passive voluntary user contributions. In the passive concept, users contribute without even knowing they do so. An example of passive contributions can be found at Google Search ([www.google.com](http://www.google.com)), which registers the search query for building even better search engines (Cook, 2008). Active contributions can be seen as the user intentionally contributing to something, for example multimedia content or ratings. According to Borst (2010), new types of organizations have come into existence to organize voluntary online activities. In her PhD-thesis she differentiates between three types of organizations: online communities, open source software development communities and crowd sourcing. An online community is defined as (O'Mahony et al., 2007) "a social group with a shared basis of authority. Such a social group consists of people sharing common interests and needs. The specific characteristic of online communities is that members primarily interact via online communication media instead of face-to-face contacts". The second type of organization, open source communities, was formerly exclusively used by groups of voluntary software developers. However in recent years the term has been used more broadly, also applied by people in other activities than software development. Although the activities performed differ from group to group, the groups are still led by one or more characteristics from the open source software development community, namely licensing regimes, shared tools and internet-enabled communication (Borst, 2010). The third term, crowd sourcing, is coined by Jeff Howe, when describing a new business model for harnessing creative solutions from within distributed networks of people. "Crucial to crowd sourcing is the use of an open call format and a large network of potential labourers" (Howe, 2006). A big difference between online communities and crowd sourcing is the social connection between the participants, which is not required for crowd sourcing.

As pointed out in the previous section, crowd sourcing is the use of groups of people (crowds) to harness creative solutions. Organizations can crowd source certain jobs that otherwise would have been done by people within the organization. This concept comes with both advantages and disadvantages (Borst, 2010). On the one hand, it costs organizations little time and money to use the crowd for certain jobs. Moreover, payments are only granted by result and are most of the time even omitted. Organizations using crowd sourcing can tap into a wider range of knowledge, talent and ideas than they would from within the organization. Customers are the closest to the product or service, using their knowledge provides first-hand insights in customer desires. On the other hand, crowds can sometimes feel exploited, doing creative work for (almost) nothing (Wu et al., 2007; Lampel et al., 2007). Another downside is that the organization that wants to use crowd sourcing, needs to provide both a technical and a social environment to make crowd sourcing happen. Often the good solutions that are shared are accompanied by a lot of noise, so filtering has to take place. A last, and maybe most interesting downside of crowd sourcing, is the incentive for people to join. Since these crowds do not get paid well, or do not get paid at all, the key to participation is the contributor's self-selection to assist with a task (Lakhani et al., 2007). The most leading work on incentives and rewards for crowd sourcing is done by Borst (2010).

## **2.3 Examples of crowd sourcing in everyday life**

The rise of Web 2.0 led to more and more initiatives in which crowd sourcing and collaboration are recognizable. In the following paragraphs some of the most well known examples will be explained. The first and probably best-known example of a crowd sourcing system is YouTube. The website, which attracts 800 million unique users every month (YouTube, 2012), is based on video content. This content is exclusively contributed by users of the YouTube website. Every minute, users contribute 48 hours of video, which means that nearly 8 years of content is added every day. Over 3 billion videos are viewed every day. The astonishing statistics of the usage of the website say it all; YouTube is a great example of a crowd sourcing system.

A second example of a system in which the crowd is used, is Amazons Mechanical Turk (MTurk) platform. This platform is named after a chess-playing machine invented by Wolfgang von Kempelen in 1770 (Kittur et al., 2008). The system can be seen as a marketplace in which users can post and complete small tasks, sometimes referred to as micro-tasks. Users of the platform can post tasks that they want to be completed and offer users who provide a solution a (monetary) reward (Kittur et al., 2008). People involved in the MTurk platform, often referred to as Turkers, complete tasks that typically require little time and effort in return for a very small amount of money.

A third example, which is quite successful, is Threadless. This American company is known for its website on which customers can buy custom t-shirts. Each month, every registered user can submit designs for new t-shirts. Users who do not take part in the design contest can rate the design of t-shirts that other users have posted. At the end of each month, a number of the highest rated t-shirt designs is then produced and sold exclusively in the web shop at Threadless.com. The winning designers are rewarded with cash and Threadless gift certificates (Brabham, 2010).

## 2.4 Crowd sourcing in organizational social media context

As the examples in the previous section show, crowd sourcing can be applied in a stand-alone kind of fashion. However, the application of crowd sourcing in business context can also be part of a bigger whole, which is shown by a recent McKinsey research (Divol et al., 2012). In this publication, the researchers state that “Companies invest millions of dollars in social media, with little understanding of how it influences consumers to favor their brand or buy their products. Without knowing how social media affects behavior, companies run the

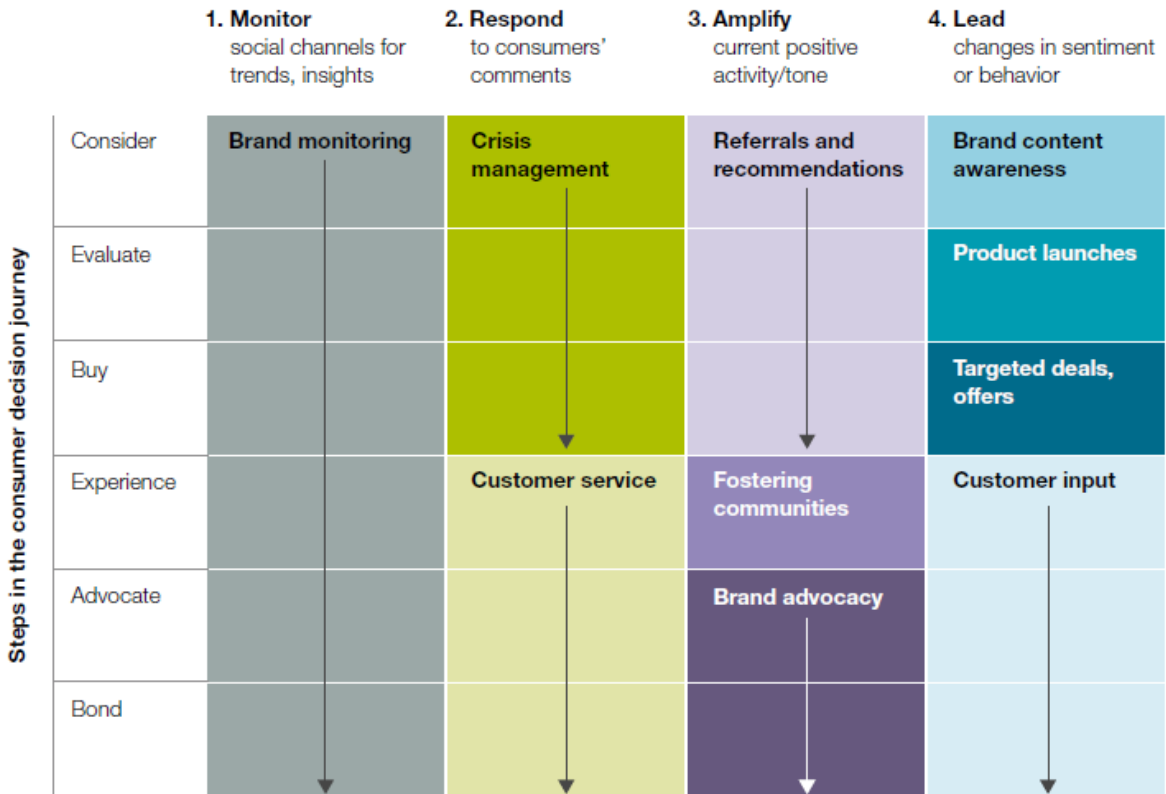


Figure 2.1: Social environment as proposed by Divol

risk of aiming it at the wrong targets, wasting time and money on ineffective efforts, and generally failing to harness its potential.” A model, containing four stages, is presented in this research, which should help organizations to harness the potential of their social media efforts in a more successful way. The model is depicted in figure 2.1, in the following paragraphs we will explain the model in more detail.

The model tries to mix the consumer decision journey with four stages in organizational context. The first stage is to monitor what is going, in other words, knowing what is said online about your products and services. Although this monitoring phase can generate valuable insights, it is only the start. The second phase is to respond what is going on online. According to Divol et al. (2012), this can be done either in a crisis management kind of way, or in a way to enhance the customer service. The third step of the social environment model is to amplify the positive things that are said about the organization. This means that the core concepts of the marketing campaigns should invite customers into “an experience that they can choose to extend by joining a conversation with the brand, product, fellow users and other enthusiasts”. The final stage in this model is to lead consumers toward long-term behavioral changes.

In total, the model contains four stages and each stage contains six phases from the consumer decision journey. This means that there are 24 different steps in this model. Within some of these steps, the use of crowd sourcing applications could stimulate the outcome of the step. Since more and more organizations try to involve social media platform into their business activities, this research provides a serious model to keep in mind when investigating the potential of crowd sourcing applications in organizational context.

## **2.5 Crowd sourcing in public transportation**

Although the possibilities of using crowd sourcing systems seem almost endless, the area of public transportation is a field that is underexposed. A search query using both Google Scholar and Scopus, results in only a few articles published on crowd sourcing and public transportation. In these articles, roughly three themes emerge, which will be exemplified in the following subsections.

### **2.5.1 The potential of in-train crowd sourcing**

The first and probably most interesting publication on crowd sourcing and public transportation is by van der Wees and Moonen (2011). This research, on the potential of in-train crowd sourcing, describes the possibilities of crowd sourcing applications in the Dutch intercity trains. These applications should be considered web-based and passengers can access them with their own laptop, tablet or smart phone. Based on a small number of expert interviews, a list of possible crowd sourcing applications was generated. Based on the number of times an application was mentioned, the adoption potential and the relevance, three crowd sourcing applications were chosen to be examined in more detail. The first application is a chat-application. Using this application, passengers can chat with other people in the train or the specific train compartment. The second application is a meeting service, in which people can meet other people with a shared interest. The third option is a sentiment measurement tool. With this tool people can share their opinion (positive, neutral or negative) on their current travel with NS, potentially with a small text explaining why they have that certain opinion.

For each of the three ideas, a set of screenshots was developed. These screenshots were then showed in a real train and passengers were asked for feedback on the application. The surveys showed that people liked the idea of each of the three applications, but would probably not use the first and the second. The only application that people would like to use, and also continue to use, is the sentiment measurement tool. One of the most interesting conclusions van der Wees and Moonen (2011) drew from this research, is that “the adoption of

applications with more serious characteristics, or closer to the core rail-transport process, seem to have a better chance of adoption as applications with a more recreational background”.

### **2.5.2 Real time travelling information using crowd sourcing**

The second theme which emerges in the combined field of crowd sourcing and public transportation, is real time information on buses or trains using the crowd. In this kind of situation, described by Steinfeld et al. (2011), passengers can function as sensors within the transportation system. This means that passengers with a smart phone can use the sensors and abilities of their smart phone to gather data on their current travel. This information can be categorized as either continuous or event-based. The continuous information gathered can range from GPS-location, speed and direction to temperature and noise. This data is then sent to the public transportation company, where it can be aggregated or processed. The GPS-location, speed and direction could, for example, be used to create real-time maps of transportation vehicles to measure the traffic load or to give estimates at stops. Temperature and noise could be indicators for the comfort of the trip and how crowded the vehicle is.

Event based information is implemented in this example by a simple ‘report’ button in the app on the passenger’s smart phone. With this button, passengers can file a report to a specific experience they had during their trip. This could, for example, mean that passengers can report a broken wheelchair lift, a broken seat or a positive experience. This information is then sent along with the passenger’s location, date and time to the public transportation company. The public transportation company can then take appropriate actions. Both the continuous and event-based information have been implemented in a smart phone application, which was tested during a pilot. Since the pilot was mostly focused on defects in the software, the results presented in this publication cannot be used in this context.

### **2.5.3 Public participation in public transportation planning**

The third theme that emerges in the combined field of crowd sourcing and public transportation is the use of public participation. This theme is described by Brabham et al. and Brabham. According to Brabham (2009) involving citizens in planning processes “helps ensure a plan that will be more widely accepted by its future users” and “the crowd sourcing model would be perfect for this”. In the context of public participation, this means that passengers can be used in the planning processes of public transportation. An example of such an effort is presented by Brabham et al. (2009) in their publication about the next stop design project. In this project the collective creativity of a crowd is harnessed and used for designing new bus stops. Bus passengers from all over the world were asked to participate in a specially implemented crowd sourcing system named Next Stop Design ([www.nextstopdesign.com](http://www.nextstopdesign.com)).

Since the bus stop is the first point of contact between the passenger and the bus service, this makes a good focus point for the research according to Brabham et al. (2009). “The spacing, location, design and operation of bus stops significantly influence transit system performance and customer satisfaction”. In the first month 338 users were registered and 47 bus stop designs were submitted. The preliminary results of this effort are clear; using the crowd sourcing model for public participation in public transportation seems effective in a number of ways. First, the online bus stop design contest involves citizens more deeply. Second, the volume of participation is much higher than in traditional public participation projects. As a third and last advantage, the geographical diversity is much higher since people from all over the world did participate.

## 2.6 State-of-the-art and best practices in the field

In this section we will present two of the most used and mostly cited researches in the field of crowd sourcing.

### 2.6.1 Organizational genes for crowd sourcing systems

In several different contexts, crowd sourcing systems have been implemented to make use of the crowd. Scientific research shows different ways to implement such a system. One of the most cited researches on this topic is the organizational genes model for crowd sourcing systems by Malone (2009). In MIT's Centre for Collective Intelligence, over 250 examples of crowd sourcing systems are gathered. From these examples, Malone et al. identified a small set of building blocks for crowd sourcing systems. These building blocks can be classified using two pairs of related questions: (1) Who is performing the task? Why are they doing it? (2) What is being accomplished? How is it being done? This leads to the framework depicted in figure 2.2.

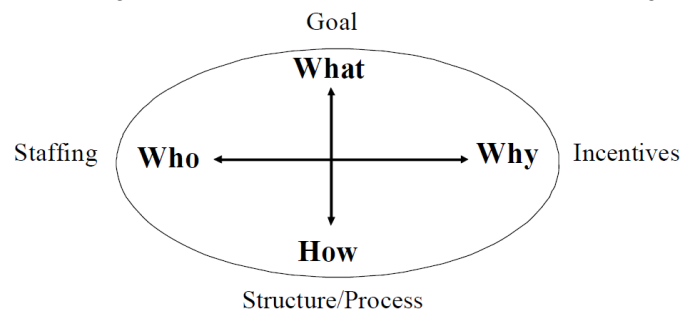


Figure 2.2: Organizational genes for crowd sourcing

To provide a comprehensive classification of the different types of building blocks (or “genes”), all four questions that are stated above need to be answered. The answers to each of the questions are predefined and an overview of these answers is given in table 2.2.

Who?	Why?	What?	How?
Hierarchy	Money	Create	- Collection (independent)
			- Collaboration (dependent)
Crowd	Love	Decide	- Individual decisions (independent)
			- Group decisions (dependent)
Glory			

Table 2.2: Overview of the genes

The answer on the who-question can either be hierarchy or the crowd. If the answer is hierarchy, this means that a situation applies in which “someone in authority assigns a particular person or group of people to perform the task” (Malone et al., 2009). The task can be assigned to people within the organization, but also people outside the organization. In the crowd gene, tasks can typically be undertaken by anyone in large group of people, who chooses to do so. This means that this one person does not need to be assigned by someone in a position of authority.



The answer on the why-question, the incentive to participate, can be money, love or glory. Money is obviously the easiest of these answers. The reward does not necessarily need to be cash money, but could also be loyalty points, discount coupons or other money-related rewards. The incentives categorized love, are a bit more difficult to describe. According to Malone et al. (2009) the love gene can take several forms: “people can be motivated by their intrinsic enjoyment of an activity, by the opportunities it provides to socialize with others, or because it makes them feel like they are contributing to a cause larger than themselves”. The last important motivator is the glory gene. Glory can be seen as recognition by others. This recognition could for example be realized by earning experience points or by ranking people on a scale or scoreboard.

The what-question can be answered with create or decide. In the create gene the users obviously generate something new. In the decide gene users evaluate and select alternatives. The last question, how is it being done, is typically answered by describing the organizational structure and processes. Since in this research we are interested in using the crowd, we will only look at the how gene where the crowd does a create or decide task. A key determinant in the possible answers is whether the crowd does this create or decide task independently of each other or whether there are strong dependencies between the separate contributions. This determinant makes that there are four possible how genes for the crowd, as depicted in table 2.3.

	<b>Independent</b>	<b>Dependent</b>
<b>Create</b>	Collection	Collaboration
<b>Decide</b>	Individual decisions	Group decisions

**Table 2.3: Variations of the how gene for crowds**

The collection gene occurs when users contribute independently of each other. This results in a collection of contributions. A good example of such a gene is YouTube, which has a collection of videos. The collaboration gene can be best described as multiple users contributing to create something. Additionally, there are important dependencies between the contributions. A good example of such a gene is multiple users contributing to one article on Wikipedia (although Wikipedia as a whole qualifies as collection). The individual decisions gene occurs when members of the crowd make a decision, which can be informed by crowd input, but does not need to be identical for every user in the crowd. An example of such a gene is the individual user of YouTube who decides for himself which video to view, but can be influenced by recommendations or rankings from other users. The last gene, the group decisions, can be described as multiple inputs that are assembled to generate a decision that holds for the crowd as a whole. A good example of such a gene is the voting process for t-shirt designs at Threadless.

By making use of the four questions (what, who, why and how) and their possible answers, a complete system can be formed, which Malone et al. (2009) call a gnome. A gnome is a schematic view on the crowd sourcing system under consideration and can be used to analyze or validate the design of such systems. An example of a gnome is shown in table 2.4. The example is based on the open source operating system called Linux, which is a form of crowd sourcing the development of an operating system.



Example	What	Who	Why	How
Linux	Create: New software modules	Crowd	Love Money Glory	Collaboration
	Decide: Which modules warrant inclusion in next release	Linus Torvalds and his lieutenants	Love Glory	Hierarchy

**Table 2.4: Example of a gnome**

As can be seen in the example, the Linux project makes use of crowds in two different ways; for creation and decision. Each gene in the gnome has certain preconditions under which it can be successful. For example, for the decision, the Linux Project uses hierarchy as a how-gene. Collaboration could also have been chosen, but then every developer would have a say in the process, which cannot be coordinated. The research of Malone et al. (2009) therefore gives an overview in which situations certain genes are useful. This overview can be found in appendix A of this thesis.

**2.6.2 Key challenges for crowd sourcing systems**

Another widely used research is the key challenges model for crowd sourcing systems by Doan et al. (2011). They conducted research to provide a global picture of crowd sourcing systems on the web. In the research, four key challenges are stated that should help in creating a crowd sourcing system on the web.

**Challenge 1: How to recruit and retain users?**

Since the crowds form the basis for crowd sourcing, recruiting and retaining users is crucial for the success of any crowd sourcing system. According to Doan et al. (2009), there are five ways to recruit users for a crowd sourcing systems; require users to make contributions, pay users to make contributions, ask for volunteers, make users pay for the service and piggyback on the user traces of a well-established system. Note that this last solution is more applicable for systems in which user contribution is passive.

Once the users are recruited, they have to be retained so contributions keep on being posted. In the research by Doan et al., some of the most popular encouragement and retention schemes are mentioned. First of them is instant gratification. Also, enjoyable experiences or necessary services could be used for retention. Establish, measure, and show fame/trust/reputation is a third solution for retaining users. The forth solution is to create competitions and the last solution concerns ownership situations.

**Challenge 2: What contributions can users make?**

The type of contribution users can make in a crowd sourcing system strongly depends on the complexity of the system itself. In simple systems users can for instance evaluate, review, rate or tag. However in more complex systems, users can do more cognitively complex contributions. Concerning the type of contribution a user can make, four important factors should be considered:

- How cognitively demanding are the tasks?
- What should be the impact of a contribution?

- What about machine contributions?
- The user interface should make it easy for users to contribute.

The last factor is highly non-trivial. An example in this case could be users contributing to a knowledge base. If some user wants to contribute the piece of domain knowledge “no current living person is born before 1850”, how can this be contributed in an easy way? Natural language would be easy for the contributor, but hard for the system. The other way around, formal language is difficult for the contributor and easier for the system.

### **Challenge 3: How to combine user contributions?**

Contributions by separate users can provide useful and valuable information, but in some cases the information becomes even more useful and valuable if the contributions are combined. When contributions need to be combined, there are basically two ways of merging them. The first is manually, thus by humans. The second is automatically, with an algorithm or a function. A difficult aspect of merging comes into consideration if not only humans can contribute, but also machines.

### **Challenge 4: How to evaluate users and their contributions?**

Crowd sourcing systems often have to deal with malicious users and noisy contributions. There are basically three methods to mitigate these side effects: block, detect and deter. First, users can be blocked based on their previous contributions or other kinds of whereabouts. Second, a variety of techniques can be used to detect malicious users and contributions, like monitoring, flagging and enlisting. Third, users can be deterred by the threat of punishment. Despite of all the effort that is put into stopping malicious users and contributions, they will never disappear completely. Therefore, the crowd sourcing system should find a way to undo the contributions. In systems where contributions are merged, this is much harder than in systems where contributions are loosely or not at all merged.

## **2.7 Literature on testing and validation methods**

The last scientific research field that we will review in this graduation project is the field of testing and validation methods. Since public authorities and consumers not only influence, but also more and more contribute to the innovation process of organizations, it becomes increasingly apparent that innovations need to be validated in different ways and earlier than in traditional settings. Therefore, organizations are looking for new facilities to test and experiment with their innovative ideas and ICT developments. This means that validation and experimenting will happen in experimental settings, where the context will imitate the user’s real life and where the user can interact and exchange views with the organization for optimal innovation introduction.

In a 2005 research on testing and experimentation platforms (TEPs), Ballon, Pierson and Delaere (2005) gave an overview of the European Practices in that field. A typology of six different types of TEPs is given, which will be replicated below:

- Prototyping platform: a design and development facility used prior to mass production and resulting in the first proof-of-concept of a new technology, product or service.
- Test bed: a standardized laboratory environment used for testing new technologies, products and services and protected from the hazards of testing in a live or production environment.
- Field trial: a test of technical and other aspects of a new technology, product or service in a limited, but real-life environment.

- Living lab: an experimentation environment in which technology is given shape in real life contexts and in which users and end-users are considered 'co-producers'.
- Market pilot: a pilot project in which new products or services that are considered to be rather mature are released to a certain number of end-users in order to obtain marketing data or to make final adjustments before the commercial launch.
- Societal pilot: a pilot project in which the introduction of new products and services into a real-life environment is intended to result in societal innovation.

Ballon et al. (2005) moulded these six TEPs in a conceptual framework based on three characteristics. The first characteristic is the technology readiness. This characteristic is concerned with the maturity of the technology or application, ranging from really immature to almost market ready. The second characteristic is a scale that is concerned with the focus, ranging from testing technology or application to a focus on the design aspects. The third characteristic is the differentiation in the degree of openness, ranging from in-house activities to open innovation platforms.

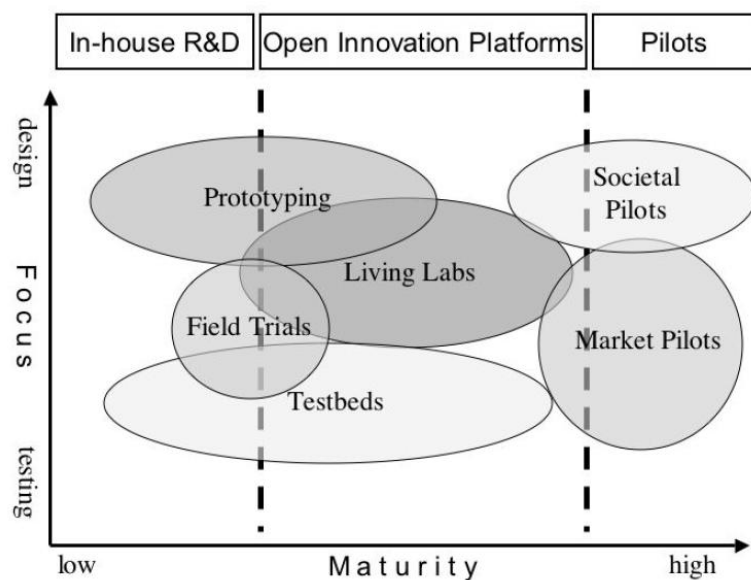


Figure 2.3: Conceptual framework for testing an validation

### **3. SOLUTION DESIGN: IN-TRAIN CROWD SOURCING**

*In this chapter we start with the solution design phase. This phase is typically used for designing solutions to the problem as stated in the problem statement. We start this chapter with an initiation of the design phase, followed by an introduction to the context in which the solutions has to be designed. The rest of the chapter presents the solutions as proposed and specifies them with a functional and technical design.*

#### **3.1 Solution design initiation**

In chapter one we already stated that this phase is meant to design a solution to the problem as stated in the problem statement. The solution we design in this chapter has to be related to the scientific fields we described in the problem investigation phase. This means that the solution to be designed needs to focus on crowd sourcing technologies. We will use this phase to introduce the context in which the solution has to fit, find out which problems play a role in this context and design a crowd sourcing solution in this context.

#### **3.2 Introduction of the Dutch railway system and OBIS**

The Dutch railways (NS) is a railway company in the Netherlands, which is completely owned by the Dutch government. The NS makes use of the railway network in the Netherlands, which is the busiest of Europe (Ramaekers et al., 2009). Currently, NS has several activities regarding passenger transport. These activities can be broadly divided into intercity services, regional train services, national high-speed trains and international high-speed trains. Besides NS a small number of other companies are offering particular regional train services. The intercity services are the most important for this research project. The trains that are used in this intercity service are specially designed for long distance intercity services, the so-called intercity material (ICM). These trains consist of 3 (ICM-0, ICM-1 and ICM-2) or 4 (ICM-3 and ICM-4) compartments per train and can move approximately 153 and 187 passengers. These intercity trains are known for their characteristic 'Boeing-like' front and back. Also, they were characterized by the fact that it was possible to walk from one compartment to the other during the travel. In Dutch they are therefore nicknamed 'koploper' (Koch, 2009).

In November 2006, the Dutch Railways started renovating their intercity trains (ICM), which originated from 1977. This operation started off with the trains that have 3 compartments. In April 2007 the first renovated intercity train was delivered, nicknamed ICMm. These renovated intercity trains can be recognized by a completely new interior, which includes approximately 13% more seats. A more clear division between the first and second class is made, using different seat colours. Moreover, the renovated intercity trains are equipped with improved facilities as air-conditioning, toilets for disabled people and digital passenger information. The renovation was also a good moment to introduce two features that NS considered earlier on; the introduction of internet onboard the train and a platform for implementation of new applications in the train. This last thing was mainly driven by different people within the organization having good ideas for new applications in the train. Both new ideas, the internet and a platform for new applications, were implemented by means of a new information system, the On-Board Information System (OBIS).

In the current OBIS, two of these applications are running, of which the first is the connection to the internet. This connection is implemented by sim cards in the train, which make a connection to the third generation (3G) network of T-Mobile (2012). Each compartment has its own connection point, which works similar to a standard Wi-Fi network at home. By offering internet in the train, the ability to work or play is offered to the passenger.

Besides the fact that travel time can now be converted into working time, the subjective travel time is decreasing, meaning that passengers experience shorter travel times. This should all lead to a higher customer satisfaction. The second application that is running on the OBIS platform has to do with passenger information. This application is characterized by four different systems:

- Information displays in the compartments
- Indication of the destination on the outer side of the train
- Direct passenger information on the train's portal website
- Automated voice announcements

The information displays, two in each compartment, are used to display real-time passenger information like the next station, estimated time of arrival, connections, breakdowns et cetera. The outer side of the train contains big displays where the destination of the train is shown, which in Dutch is called 'buitenbestemmingsaanduiding' (BBA). The passenger information that is shown to the passenger on the screens in the compartment can also be checked on the train's portal website. This portal website functions as a kind of homepage for the train, and is also used for passengers to connect to the internet. Other than showing passenger information, the portal also contains some advertisements. The automated announcements system should inform the passenger of the upcoming stations and other passenger information, but due to technical errors this system is not working yet.

Lately, NS decided that the current portal website in the train is not sufficient anymore. This means that in the forthcoming period the portal website will be redeveloped. This redevelopment is a great opportunity to add new applications to the portal website of the train. Since this research is about crowd sourcing, we will look at different possibilities for interaction with the crowd. Currently, NS is already in contact with the crowd, via Twitter. NS has a specialized team of employees, the so called Twitter-desk, which monitors what is being said about NS on the web and tries to respond to these messages. In the following section, we present two ideas of interacting with the crowd, but then in a specified context, namely the train.

### 3.3 Designing the solutions

Before finding crowd sourcing solutions, which could increase the travelling experience, the areas where passengers are most unhappy about should be found. A first place to find these areas is in the customer satisfaction questionnaires by the Dutch Railways. In the figure 3.1 (NS, 2010a), it can be seen that the overall score of the customer satisfaction has improved over the years. Roughly speaking there are three areas in which the score is behind on the overall score, namely:

- Trains travelling on time
- Providing information during disruptions and delays
- Cleanliness of the trains and the station

Since the scope of this research is to use crowd sourcing applications in the train, the first area for improvements is outside the scope of this research. The second and the third area can be used as inspiration for new crowd sourcing applications in the train. To give a unified description of each idea, the organizational genes model for crowd sourcing systems by Malone (2009) will be used. This means that for both crowd sourcing ideas the questions of 'what?', 'who?', 'why?' and 'how?' need to be answered. Table 3.1, presented below, gives an example of an empty gnome; the table itself thus represents the method to describe the crowd sourcing ideas.

Klantwaardering NS over het 3e kwartaal 2010							
	2005	2006	2007	2008	2009	2010	2010
	3e kw	3e kw	3e kw	3e kw	3e kw	2e kw	3e kw
<b>Algemeen oordeel</b>	68%	71%	74%	78%	79%	76%	78%
Op tijd rijden	42%	44%	51%	59%	60%	54%	58%
Beschikbaarheid zitplaats	80%	80%	79%	81%	81%	77%	80%
Informatie bij ontregeling	48%	49%	51%	54%	55%	55%	55%
Sociale veiligheid trein en station	72%	75%	76%	78%	79%	78%	79%
Reinheid treininterieur en station	54%	54%	55%	54%	57%	51%	57%
Klantgerichtheid personeel	58%	57%	60%	63%	65%	63%	63%

Figure 3.1: Customer satisfaction over the years

<i>Example</i>	<i>What</i>	<i>Who</i>	<i>Why</i>	<i>How</i>

Table 3.1: Example of an empty gnome for crowd sourcing applications

In answering these questions, guidance can be found in the overview that is presented in Appendix A. In this overview Malone et al. give suggestions on what genes could be useful in specific situations.

### 3.3.1 Reporting system for train compartments

The first idea to use the crowd in providing a better travelling experience is to implement a system in which passengers can report dirty trains or broken things in the train. Since the passengers are close to the trains, they are the first to notice things that are broken or not clean. Additionally, their travelling experience is directly affected by the cleanliness and tidiness of the train. If passengers in the train notice that something is dirty or broken, for example the toilet or a seat, they can report that via a system that is integrated in the portal in the train. Workers from the Dutch Railways can then fix the problem in an earlier stage, thereby improving the travelling experience for the passengers. After the problem is fixed, the reporting passenger could receive feedback and maybe a reward for his report. Next to implementation on the portal in the train, this application could also be implemented into the travel planner app for smart phones. If this idea is applied, the app could also be used to report dirty or broken things on the platform or in the station.

To describe the reporting system, the four questions that are stated in the previous section should be answered. The 'what'-question can be answered by either create or decide. In this case it should be clear that the right answer is create, because passengers can create new reports for broken or dirty things. The 'who'-question can be answered by the crowd since the preconditions in Appendix A, resources are distributed and the tasks can be cut up in little pieces, are met. The 'why'-question is a difficult one to answer in this stage, but Malone et al. state that "appealing to love and glory, rather than money, can often reduce costs and providing glory and money can often influence group's direction and speed". The 'how'-question, together with the answer of create, leads to a decision between collection and collaboration. The difference between these two is that collection is used for

activities that can be divided and collaboration is used for activities that cannot be divided. In this case the collection gene is most applicable. This results in the following gnome:

Example	What	Who	Why	How
Reporting system	Create: New reports	Crowd	Money, love and/or glory	Collection

Table 3.2: Gnome for the reporting system

Note that the description of the system above is only a short introduction of the idea. In the functional and technical designs, a more detailed description of how such a system would look and work is given.

3.3.2 Feedback system for passenger information

As shown in Figure 3.1 another area that passengers are not happy about, is the passenger information during disruptions. The idea behind the feedback system for passenger information is that people can give feedback on different kinds of passenger information. This could be, for example, passenger information during disruptions, the outside destination display (BBA) or displays in the compartment. This feedback should be gathered and processed by someone in the train or at the back office. The feedback can also be checked by the system under attention. In the same portal website in the train, NS can also show recent feedback comments and ask if this problem still occurs or if other people encounter the same problem.

To give a unified description of this crowd sourcing application, again the gnome theory by Malone (2009) is used. Since the answers are no different than for the first solution, they result in the following gnome:

Example	What	Who	Why	How
Reporting system	Create: New reports	Crowd	Money, love and/or glory	Collection

Table 3.3: Gnome for the feedback system

3.4 Introduction and goals of the stakeholders

In the previous section we described two solutions; the reporting system for train compartments and the feedback system for passenger information. This section gives an overview of the different stakeholders that are involved in both solutions. For each stakeholder, the goals that they want to achieve with the solutions under construction are described. Throughout this chapter the concept of general status of the train is used several times. With the general status of the train, we mean how clean and tidy the train is, if there are any broken things in the train and also the maintenance state of the train.

3.4.1 Product owner

The product owner for both solutions is Piet de Roo, sales manager for Logica at NS. From a professional point of view he is interested in the possibilities of implementing crowd sourcing applications in the Dutch intercity trains.



The intended solutions have to be developed as a demo for the NS to show the power and possibilities of crowd sourcing in the train.

### **3.4.2 User of the system**

The intended users of the solutions are the people that are travelling in the Dutch intercity trains. Each day, over one million people travel by train. It turns out that these people together use 600.000 internet sessions per month. This means that the portal website in the train is viewed over 600.000 times per month. The people that view this portal website are the intended users. With the solutions, passengers can report problems or points of attention in the train. If these points can be fixed quicker, the travelling experience for these passengers increases.

### **3.4.3 Customer of the system**

The customer of the solutions is NS, more specifically the management of the department of NS Reizigers. They keep track of customer satisfaction over the last few years. It has become clear that some areas score lower on customer satisfaction than others, as can be seen in Figure 3.1. These same categories do not improve (much) over the recent years. One of these areas is the general status of the train compartments; another is the passenger information during disruptions. Solutions as proposed in this chapter will function as a tool to improve the customer satisfaction in these areas.

### **3.4.4 Administrators of the system**

Often the customer of a system is not the same entity as the people who are going to work with it. In this case, the NS is the customer for the intended solutions. But information that comes in via the intended solutions needs to be processed; otherwise the information would be useless. It is not plausible that the people at NS, who decide to implement the intended solutions, are the same people who are going to process the information. The people in the organization, who are working with the information, will be termed workers throughout this project.

### **3.4.5 Realization team**

The fifth group of stakeholders is the team that has to provide the realization of the solutions. This team needs to build the functional design and implement a demo of the solutions. Also, this team is responsible for validating and testing the solutions.

## **3.5 Requirements elicitation for the reporting system**

In this section we present the requirements elicitation process for the reporting system for train compartments. The requirements elicitation process is meant to state the functional and non-functional requirements. (Sommerville et al., 1998; Chung et al., 2009). The following subsections present both types of requirements for the reporting system. The elicitation process we used during this phase is brainstorming. During the brainstorming process different iterations have been performed, under supervision of the product owner and two experts from Logica. We started with a set of initial requirements from a brainstorm session and discussed these requirements with an information analyst from Logica. This initial set of requirements was then refined and extended, by means of brainstorming sessions with a second Logica expert. The second expert, an requirements engineering with fifteen years of experience, was used nine sessions of an hour each. During the process, we validated the requirements two times with the product owner. The final set of requirements is presented in this thesis.



### 3.5.1 Non-functional requirements for the reporting system

Non-functional requirements describe how the system should be; they can be viewed as quality demands. An overview of the non-functional requirements can be found in the table below. Each requirement is stated with a number, a name and a short description. Where needed, the requirements are also provided with details. All numbers for requirements for this system also include the letter A, to show that these requirements are part of the reporting system for train compartments.

Number	Requirement	Description	Details
NFA01	Availability	The system has to be available during certain hours	(1)
NFA02	Usability	The system needs a simple (frame-like) and user-friendly user interface for the users	
NFA03	Decomposition	There should be a clear decomposition between the front-end of the system and the back-end of the system	
NFA04	Connectivity	The system needs an internet connection to communicate between user interface and back-end	(2)
NFA05	Platform independence	The front-end of the system should function the same on every user platform	(3)
NFA06	Failure management	The system needs to be available also whenever there is no connection between the train and the 'shore-side'	(4)
NFA07	Accessibility	There should be a possibility to access the system in each train compartment or train via the internet	
NFA08	Accuracy	Contributions in each train compartment are real-time	
NFA09	Extensibility	There needs to be a possibility to extend the system in the future with new features	
NFA10	Privacy	The privacy of users of the system is preserved at all time, unless the user indicates otherwise	
NFA11	Scalability	The system should be scalable, horizontally as well as vertically	(5)
NFA12	Security	The contribution that is submitted by the user should be stored safely	

**Table 3.4: Overview of the non-functional requirements for the RSTC**

Whenever a number in parentheses is presented in the table above, the corresponding number in the section below explains that respective requirement in more detail.

- (1) Most intercity trains are operational between 05:00 and 02:00 the next day. This means that the application should have the same availability as the On-Board Information System (OBIS), which is already in the train.
- (2) The OBIS already offers passengers in the train an internet connection. The same internet connection is also used for communication between trains and the back-office, so it can also be used for the reporting system for train compartments.
- (3) The fact that the front end should function the same way on every platform means that it does not matter if the user is using the application on a smart phone, tablet or notebook. In each case the way the application functions should be the same.
- (4) In the jargon that is used within the Dutch Railways, the offices and train stations are termed 'shore-side', analogous to shipping.
- (5) Scalability can be defined as either horizontally or vertically. To scale horizontally (scale out) means to add more nodes to the current system, for example adding a new server to a server park. To scale vertically (scale up) means to add resources to a single node, for example adding more memory or disk space to the current servers.

### 3.5.2 Functional requirements for the reporting system

Functional requirements are descriptions of the desired behaviour of individual functions of the system (Agile Alliance, 2001a; 2001b). Each requirement is stated in the table below, Table 3.5, and is provided with a number, name and description. Where needed the requirement is also provided with details. The requirements in the table are stated from different perspectives, like user, conductor and system. Although a conductor can also be classified as an employee of NS, the conductor has a special characteristic, namely that he is present at the train.

Number	Requirement	Description	Details
FRA01	Contribution	As a user, I want to contribute on the general status of the train to improve my experience in the train	
FRA02	User overview	As a user, I want an overview of the contributions that have been made this ride to check if my contribution has already been made before	
FRA03	Worker overview	As a NS-Worker, I want the possibility to view the contributions in a certain compartment to steer on this information	
FRA04	Conductor overview	As a conductor, I want the possibility to view the contributions in a certain compartment to check if they are real and if further work needs to be done	
FRA05	Status update	As a conductor, I want the possibility to change the status of contribution to show that certain problems have been solved	

FRA06	Conductor notification	As a system, I want to alert a conductor on certain contributions so that he is in the right place quickly	
FRA07	Detection and filtering	As a system, I want to detect and filter malicious contributions because they do not contribute	(1)
FRA08	Tracking	As a system, I want to register which conductor changed the status of certain contributions to ask for further details if necessary	
FRA09	Positioning	As a system, I want to detect from which train compartment a contribution has been made	(2)

**Table 3.5: Overview of the functional requirements for RSTC**

Whenever a number in parentheses is presented in the table above, the corresponding number in the section below explains some details on the respective requirement.

- (1) There is a possibility that users of the system contribute malicious things on purpose. The system should be capable of filtering out such contributions. This can for example be done by setting a certain threshold that needs to be exceeded. If the number of times a contribution is posted is below the threshold, the contribution is not taken seriously.
- (2) This should be done in part to ensure that people who contribute to the system are really in the train. On the other hand this prevents people from contributing on a train compartment in which they are not physically present.

### 3.6 Requirements elicitation for the feedback system

Since each non-functional requirement for the reporting system for train compartments (RSTC) also holds for the feedback system for passenger information (FSPI), the non-functional requirements can be mapped one-on-one. This means that the requirements from table 3.4 can also be used as non-functional requirements for this section by replacing each A in the number by a B.

Since only the description for the first functional requirement is fundamentally different from the first functional requirement from the other solution, this is the only functional requirement that is stated. The other functional requirements, FRA02 – FRA09, can be translated one-on-one to FRB02 – FRB09 for this solution.

Number	Requirement	Description	Details
FRB01	Contribution	As a user, I want to contribute on the passenger information in the train to improve my experience in the train	

**Table 3.6: Functional Requirement for FSPI**

### 3.7 Functional design for both solutions

This section is meant to lay down the functional design for both the reporting system and the feedback system. The requirements are transformed into use case specifications, which define the interaction between actors and the system. Since the requirements for both systems are alike, the use case models apply to both systems.

The main part of this section consists of Unified Modelling Language (UML) Use Cases. A use case is a list of steps, typically defining interactions between an actor and the system. The steps are typically taken to pursue a goal. Actors in this case can be humans, but also other systems. Use cases are expressed in the UML. UML is a standardized general-purpose modelling language, which is mostly used in the field of object-oriented software engineering (Fowler et al., 2000). In order to come to the use cases, first an overview of the use cases and their coherence is presented. The overview is called a use case diagram. Following the use case diagram, a use case list is given, which is a table that contains the use cases in more details. The use case list shows for example which requirements form the basis for a certain use case. Generally, the use case diagram is used to create an overview of the use cases, while the use case list is used to provide more insight.

The figure below shows the seven use cases that have been derived from the functional requirements. It also shows how the use cases are connected with stakeholders and other use cases. When an arrow is labelled with extends, it means that the next use case can sometimes follow the previous use case. If an arrow is labelled with uses, it means that the use case is always followed by the next use case.

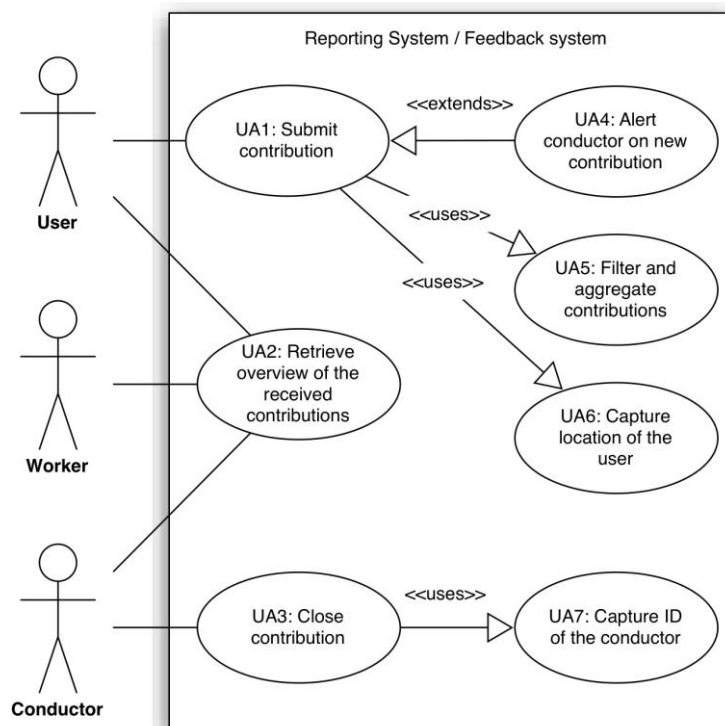


Figure 3.2: Use case diagram for the RSTC

Table 3.7 below contains a use case list. Since the use cases apply to both systems, each A could be replaced for a B.

Use case	Requirements	Primary actor	Secondary actors
UA1: Submit contribution details	FRA01	User	System
UA2: Retrieve overview of the received contributions	FRA02, FRA03, FRA04	System	User, Conductor, Worker
UA3: Change status of a contribution	FRA05	Conductor	System
UA4: Alert conductor on new contribution	FRA06	System	Conductor
UA5: Filter and aggregate contributions	FRA07	System	None
UA6: Capture the location of the user	FRA09	System	None
UA7: Capture ID of the conductor	FRA08	System	Conductor

**Table 3.7: Use case list for the reporting system for train compartments**

The next step in developing a functional design is to check whether all requirements are covered in the use cases as stated in the table above. This is done by creating a use case matrix. Then the requirements are prioritized using the MoSCoW-method, because of the time constraint on this project. Then for each of the use cases a functional specification is stated. These three steps can be found in Appendix B.

### 3.8 Technical design

The technical design is meant to describe how the intended behaviour in the functional design is realized. From the prioritization process in Appendix B, we concluded that only one functional requirement could be implemented during this research project. This means that only one use case will be implemented, indicating that only one technical specification is needed. The technical specification of both the reporting system for train compartments and the feedback system for passenger information are presented in Appendix C. This appendix also explicates the technical design for the non-functional use cases. This means that for each non-functional requirement there is a statement on how the application should comply with that specific requirement.

Throughout the functional and technical design process, we already concluded that both applications show a remarkable resemblance. The only things that are fundamentally different in both applications are the information that is filled in and the way it needs to be integrated in the business processes of NS. That is the reason why we only develop one of the two systems in more detail, the feedback system for passenger information.

## 4. SOLUTION VALIDATION: QUESTIONNAIRE

*In this chapter we start with the solution validation phase. In the first section of the chapter we use the testing and experimentation platforms model for finding a suitable validation technique. In the rest of the chapter we describe how we use this technique and perform the validation.*

### 4.1 Testing and validation platforms

To find a suitable validation method for this research project, we use the Testing and Experimentation Platforms model (Ballon et al., 2005). Therefore we should analyze the characteristics as stated in chapter two; maturity of technology, focus and differentiation in degree of openness. The first characteristic, the maturity of the technology, can be considered low. There are examples of crowd sourcing applications, but there are only little examples of crowd sourcing applications in the train. The second characteristic, the focus, can range from testing to design. In this graduation project, the focus rests mainly on design, since there is no known implementation of a system alike in the same context. The third characteristic, the differentiation in the degree of openness, has three different answering options. Since there are no pilots and open innovation platforms involved in the design, the focus will be placed on the in-house Research and Development. This means that the testing and evaluation platform that fits with this research is prototyping. According to Budde et al. (1992) prototyping can be defined as creating a demo of a new system. Prototyping is essential for clarifying information requirements. The design of a system (functional specification) must be finalized before the system can be built. While analytically-oriented people may have a clear picture of requirements, others may not.

The development of the prototype, which is a process on its own, is presented in the following section. This process is guided with the regulative cycle, which is also used as guidance for this research project. This means that we start a new iteration of the cycle here, which can be seen as a sub cycle of the initial cycle started in the beginning of this thesis.

### 4.2 Prototyping the feedback system

Since the main goal of this prototype is to use it for evaluation and validation, we implement the regulative cycle in a more simplistic way in this section. Considering we already have a clear problem, the absence of a prototype, and we already designed a crowd sourcing application in detail, the first two phases of the regulative cycle are not presented in detail. We used HTML, CSS and PHP to develop a working crowd sourcing application.

The application is part of the already existing portal website which is currently running in Dutch intercity trains. A screenshot of this portal website can be found in figure 4.1. The left side of the page, which is coloured pink, is used to connect to the free internet in the train. The second frame on the left is used for frequently asked questions (FAQ). The yellow frame on the right is used to display the real-time passenger information. The applications to be developed should have at least the same look and feel and should also fit right into the existing portal website. That is why we have chosen to insert the applications to be developed right underneath the passenger information. This seems the only logical place where such an application could be placed. For the implementation of the prototypes the same screenshot as below has been used as background for the application. This means that the original look and feel of the portal website remains intact, and the other functions of the portal website cannot be used in the prototypes. Within this background, an active area is created of 617 pixels wide and 458 pixels high, the exact same width as the yellow frame used for the passenger information.

This active area is used to display the content of the applications to be developed. The placement of the active area is realized with help of HTML and CSS.

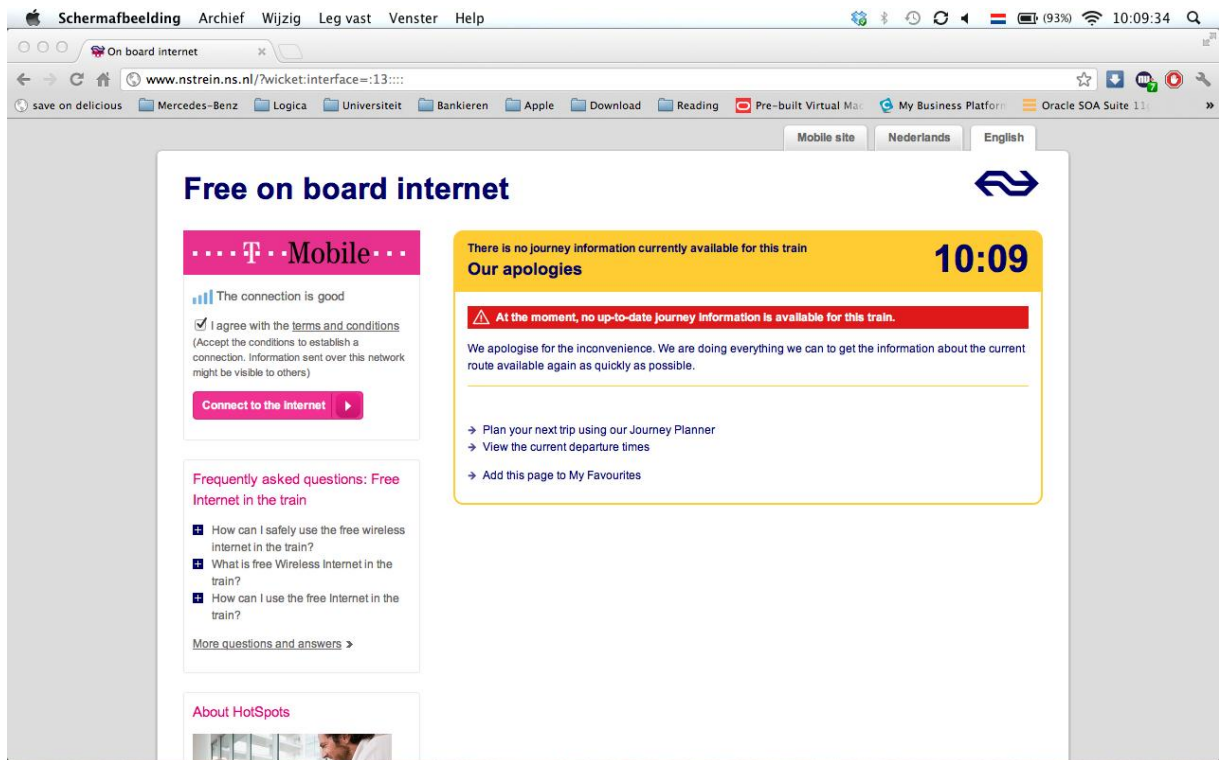


Figure 4.1: Screenshot of the portal website

For the prototype we have chosen to develop the feedback system for passenger information. This choice has been made for several reasons, the most important being the customer satisfaction, which is lowest in this area. As mentioned in the technical design, several PHP pages should be displayed in the frame as presented in the previous section. The pages all have a common look and feel, which is presented in figure 4.2.

A screenshot of a feedback form titled 'Deel uw feedback met NS'. The form is contained within a yellow-bordered box. It starts with a paragraph: 'Als reiziger staat u dicht op de serviceverlening van NS. Wij willen graag uw feedback ontvangen over de treinreis. Vul alstublieft het onderstaande formulier in. Met uw feedback kunnen wij de diensten van NS nog beter maken!'. Below this is a section titled 'Gegevens' with two input fields for 'Naam' and 'Email'. The next section is titled 'Wat voor feedback wilt u melden?' and contains three radio button options: 'Klacht', 'Compliment', and 'Overig'. At the bottom of the form is a 'Verder' button.

Figure 4.2: Screenshot of the application prototype



After we finished the problem investigation and solution design phases, the design was validated. We did not put as much effort in the validation process as Wieringa (2006) describes. This is because our most important goal for application is to be suitable to take it into the train and let passengers evaluate it. The validation for this prototype has been done with help of student from the Logica Working Tomorrow programme and two representative train passengers that were selected for this purpose. After finishing the validation of the prototype application, the application was deployed for testing and validation purposes. This process would be termed solution implementation in the regulative cycle. A description of this process can be found in Appendix D.1. With the applications being deployed, some tests have been performed on the prototype to find out if the prototype complies with the requirements as stated in the functional and technical design. More information on the testing process can be found in Appendix D.2. With the implementation of the prototype application we conclude the sub cycle in the original regulative cycle.

### 4.3 Method of validation

The validation process for the application is done by means of a questionnaire. The easiest way to do a questionnaire these days is online. After a short study into the world of online questionnaire tools, the conclusion was made that Qualtrics (<https://qtrial.qualtrics.com>) would be an easy and sufficient tool for this purpose. The advantages of Qualtrics are that it is free to use and the results are stored and presented automatically. Another advantage, something that for example Google Docs (<http://docs.google.com>) does not have, is the ability to show a status bar and add text areas and images that can be customized. The only downside is that there is a limit on the number of respondents that will be saved, namely 250. Since this is enough for this context, we decide to construct a questionnaire utilizing Qualtrics.

When conducting a questionnaire, two things are vital for the success of the questionnaire and the reliability of the outcome. The first is the environment in which the questionnaire is held. Before the respondents can fill in the questionnaire, they first need to play around with the application. This can be done in the most natural environment, hence the train, or outside the train. For this questionnaire we have chosen to conduct the questionnaire in the train. The second thing that is important is the number of respondents that will get. For this questionnaire to be sufficient, we aim at 50 respondents, at least, to fill in the complete questionnaire.

### 4.4 Developing the questionnaire

In this section we present the process of developing the questionnaire. The basis of the questionnaire is formed by the two scientific frameworks that are introduced in chapter two. This means that with the intended questionnaire we cannot only validate the prototype, but also validate the scientific frameworks. Before we state the questions that are derived from the frameworks, we start with a general introduction into the questionnaire.

#### 4.4.1 General

This section is used to introduce the research to the respondents. Next to that, this section could also be used to ask general questions. Since there is no reason, a priori, to believe that the questionnaire is biased, there is no real need for general questions. The only two things that could be useful to register are the timestamp on which the respondent finished the questionnaire and the experience passengers have with smart phones, tablets or laptops in the train. The first thing to register, the timestamp, is realized by a built-in functionality of Qualtrics. The second thing, experience with mobile devices in the train, leads to the following questions:



Number	Question	Type
Q1	Which of the following devices do you own? Possible answers: Smart phone, tablet, laptop	Multiple choice
Q2	How often do you use the following devices (smart phone, tablet, laptop) in the train?	Matrix
Q3	When using your device in the train, how often do you use the internet?	Matrix
Q4	Would the application be useful for you, NS and other passengers?	Matrix
Q5	How would you rate the following characteristics (interface, usability) of the application?	Matrix

**Table 4.1: General questions**

#### 4.4.2 Organizational genes for crowd sourcing

In section 2.4.1. we described the organizational genes model for crowd sourcing systems, by Malone et al. (2009). We use the questionnaire as described in this chapter to validate the model itself and the use of it in this graduation project. The model as proposed in chapter two concentrates on four questions. These four questions will be restated in table 4.2 below, to analyze if it is useful to insert questions about it in the questionnaire.

Who?	Why?	What?	How?
Hierarchy	Money	Create	- Collection (independent)
			- Collaboration (dependent)
Crowd	Love	Decide	- Individual decisions (independent)
			- Group decisions (dependent)
Glory			

**Table 4.2: revising the organization genes for crowd sourcing**

Since we already concluded in the second chapter that the who-question could be answered with crowd, it is not useful to state any questions about this. The what-question is also not interesting, since passengers can only create and not decide. Furthermore, since the contributions are independent of each other, the how-question can be answered with collection and is therefore not interesting to include in the questionnaire. The only area in this model that we can validate is the why-question. We start off with a number of open-ended questions about the incentives to use the application, followed by a number of multiple-choice questions. By first asking about their own ideas and then presenting other ideas, we try to prevent a biased view. This leads to the following questions:

Number	Question	Type
Q6	Do you know how to provide feedback to NS in the current situation?	Yes/No
Q7	Why would you want to use this application?	Open-ended

Q8	How could NS make you use this application?	Open-ended
Q9	What would motivate you to use the application? Possible answers: discount on train tickets, the idea that you can help NS, if your progress is ranked on a score board, other	Multiple Choice
Q10	Which statement suits you best? Possible answers: (1) the service of NS needs to be improved, to help them I would use this application, (2) If the NS wants feedback from me, they should offer me something for it, (3) Only if NS forced me, I would use this applications, (4) If my friends would use the application, I would also use it, (5) I would use this application to express my frustration, (6) None of the statements fits me	Multiple Choice

**Table 4.3: Questions on the organization genes model**

Questions 9 and 10 in the table on the previous page need some explanation. The first three possible answers on question 9 are directly related to the incentives as stated by Malone; money, love and glory. The last option, other, is added to validate the incentives by Malone. If people give the fourth answer, they also have to provide a textual answer. These answers can be analyzed, some of these answers might be categorized according to Malone, but it might also be that passengers add new incentives. The answers on question 10 also relate to the incentives by Malone, two answers per incentive.

#### 4.4.3 Four challenges model

In the four challenges model by Doan et al. (2009), as mentioned in chapter two, four challenges are stated that should be analyzed in every crowd sourcing project. It seems legitimate to use these four key challenges in the validation process also. Therefore, we will analyze each of the four challenges and see if any questions concerning these challenges can be asked.

The first challenge as mentioned by Doan is how to recruit and retain users. According to Doan there are five ways to recruit users; require users to make contributions, pay users to make contributions, ask for volunteers, make users pay for the service and piggyback on the user traces of a well-established system. The second challenge as mentioned in chapter two is what contribution users can make. The third challenge in the model by Doan is how to combine user contributions. In this prototype application we did not consider the possibility to combine user contributions. The fourth challenge that Doan states is how to evaluate users and their contributions. It seems that the third and fourth challenges are more focused on the back-end of the system, which makes the challenges not applicable in this stage of the project. The second challenge also does not seem applicable, since the kind of contributions passengers can perform are more or less scripted. The only challenge that seems applicable in this stage is therefore the first.

A part of this challenge is thus overlapping with the why-question in the organizational genes model, since we already asked about the incentives to participate there (questions 7-10). In these questions, we skip the last two methods Doan proposes because they do not apply here. The second part of this challenge, how to retain users, is something that has not yet been asked for. However, since there can be a large difference between what passengers say about keep using the application and really keep using the system, it is a difficult area to do research on. It is quite obvious that people will only keep using the application if the NS really does something with the information that is derived from it, so there needs to be a good back-end process. This is something that

should be designed at NS and is therefore not suitable for this research. However, we can ask about the NS keeping passengers updated on their feedback. This could motivate people to keep posting feedback. This leads to the following questions:

Number	Question	Type
Q11	Do you want NS to keep you updated on the feedback you posted?	Yes/No
Q12	If NS kept you updated, how would this affect your motivation and usage?	Scale

**Table 4.4: Questions on the four challenges model**

The questionnaire is ended with an open-ended question in which people can give general comments on the application or the questionnaire. This makes up a total of 13 questions, which are divided over two pages, questions 1-8 on the first page and questions 9-13 on the second page.

## 4.5 Execution of the validation research

To get representative results, the questionnaire has been conducted in a real-life setting over multiple days. To overcome the problem of creating a bias, we chose to conduct the questionnaire on a Tuesday, Wednesday and Friday. Tuesdays in the Netherlands are normal working days, so a lot of people travel in the morning and afternoon to and from work. The same goes for Wednesday. On Fridays a significantly higher number of people are travelling by train. This can be explained by students travelling from their university towards home for the weekend. In total we spent 20 hours of validation research in the train.

During this validation research, we asked people who were active with their device to participate in our research. This means we only have respondents who were actively using their device, which is not completely representative. However, research shows that 81% of the people in the Netherlands owns a laptop, 52% owns a smart phones and 1,7 millions tablets are sold last year (CBS, 2012). These figures reflect that a large part of the Dutch population owns one or more of these gadgets. Together with the fact that the percentage of higher educated people and students in the train, which have higher penetration of these gadgets, this indicates that our sample is relatively representative.

Tables 4.5, 4.6 and 4.7 show for July 6, July 10 and July 11, which specific routes have been used, what kind of trains were used for this route and the number of compartments we visited during the route. For each route we counted the number of laptops and tablets that passengers were working on, behind that we state the estimated percentage of passengers active with a device. The results are presented in tables 4.5, 4.6 and 4.7.

From station	To station	Laptops / Tablets	# Responses
Schiphol Airport (ICMm-4)	Amersfoort	6 (10%)	3
Amersfoort (ICMm-4)	Zwolle	7 (10%)	7
Zwolle (ICMm-3)	Leeuwarden	3 (5%)	6

Leeuwarden (Spurt)	Groningen	5	0
Groningen (VIRM-8)	Zwolle	14 (10%)	9
Zwolle (VIRM-6)	Amersfoort	0	0
Amersfoort (ICMm-4)	Amsterdam C.	5 (5%)	3

**Table 4.5: Research data for July 6**

In table 4.5, we have to note that the route between Leeuwarden and Groningen is operated by another railway company. There is no internet in these trains and the questionnaires could not be conducted. The route between Zwolle and Amersfoort was operated by a double-decker train in which the internet was not yet implemented.

From station	To station	Laptops / Tablets	# Responses
Groningen (ICMm-4)	Zwolle	22 (10%)	9
Zwolle (ICMm-4)	Utrecht	6 (5%)	1
Utrecht (VIRM-8)	Leiden	3 (< 5%)	1
Leiden (VIRM-8)	Rotterdam	0	0
Rotterdam (VIRM-8)	Amsterdam C.	7 (5%)	3
Amsterdam C. (VIRM-8)	Utrecht	8 (5%)	1
Utrecht (ICMm-4)	Zwolle	13 (10%)	2
Zwolle (ICMm-4)	Groningen	26 (15%)	2

**Table 4.6: Research data for July 10**

The number of responses on the route between Zwolle and Groningen is relatively low because of the failing internet connection at that moment.

From station	To station	Laptops / Tablets	# Responses
Groningen (ICMm-4)	Zwolle	13 (10%)	7
Zwolle (ICMm-4)	Leeuwarden	18 (10%)	9

**Table 4.7: Research data for July 11**

In total 63 respondents have filled in the questionnaire, creating 158 records in the database. This means that every respondent on average filed 2 feedback. Since the ip-addresses for all respondents are the same, we cannot conclude that every passenger tested the system before filling in the questionnaire, but there are strong indications that this is the case. The results that are gained from this dataset are presented in the next chapter.

# 5. RESULTS FROM THE QUESTIONNAIRE

After we constructed a questionnaire in the previous chapter, this chapter presents the results gained from the questionnaire. The results are presented in the same format as the questionnaire, meaning that we start off with the results for general questions, followed by the organizational genes model. We end the results chapter with the four challenges model and some general remarks we encountered during this validation research.

## 5.1 Results from the general questions

In the questionnaire, five general questions were asked. The results per question are presented in this section. The first question, which of the stated devices a respondent owns, is answered in the following way:

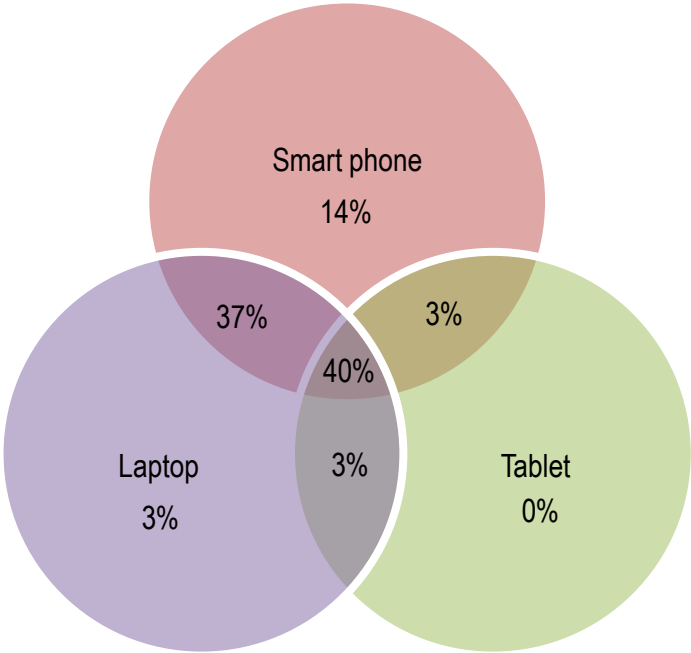
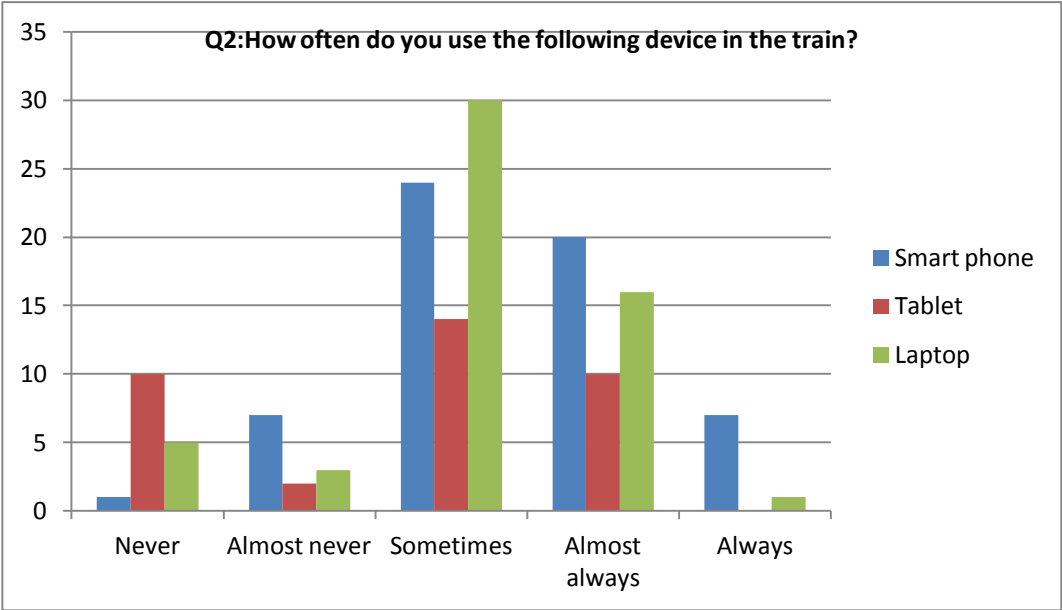


Figure 5.1: Venn diagram of the results on Q1

The vast majority of respondents in the train own a smart phone (94%), which is no surprise these days. It is more interesting to see that more than 77% of the respondents own a laptop as well. Just under half of the respondents own a tablet. All four respondents who state that they do not have a smart phone, do have a tablet. Most of them are using a more simplistic mobile phone along with a tablet, often provided by their organization.

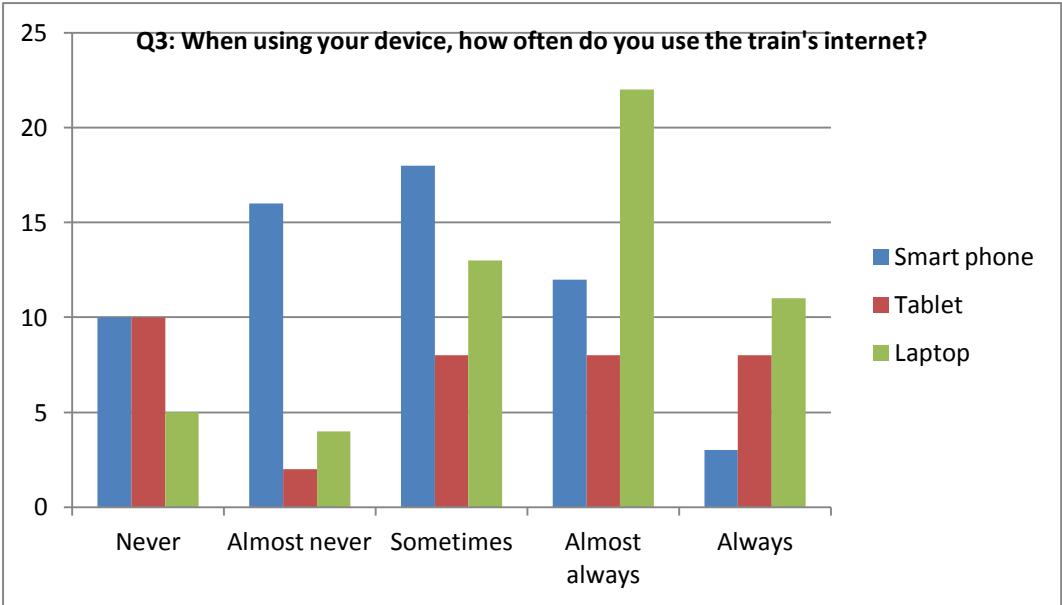
The second question is about how often passengers use their devices in the train. The results on this question are presented in figure 5.2.



**Figure 5.2: Results on Q2**

The use of most of the devices seems to be on a regular basis, no real surprises can be detected in the results. The only thing that could be interesting is that the smart phone seems to be the device that is used most in the train.

The third question in this category is about how often passengers use the on-board internet when they are using their device in the train. The results of this question are presented in figure 5.3.



**Figure 5.3: Results on Q3**

The results show clearly that smart phones are less used for internet connection in the train than other devices, which means they do not visit the portal website. This might be because most smart phones have a data plan and

therefore internet themselves. Laptops are considered the devices most used for internet in the train. Note that this is a snapshot view of a particular moment, we cannot conclude anything about future shifts in this pattern.

The fourth question in this category is focused on the application itself; would the application be useful for you, NS and other passengers? Results are shown in figure 5.4.

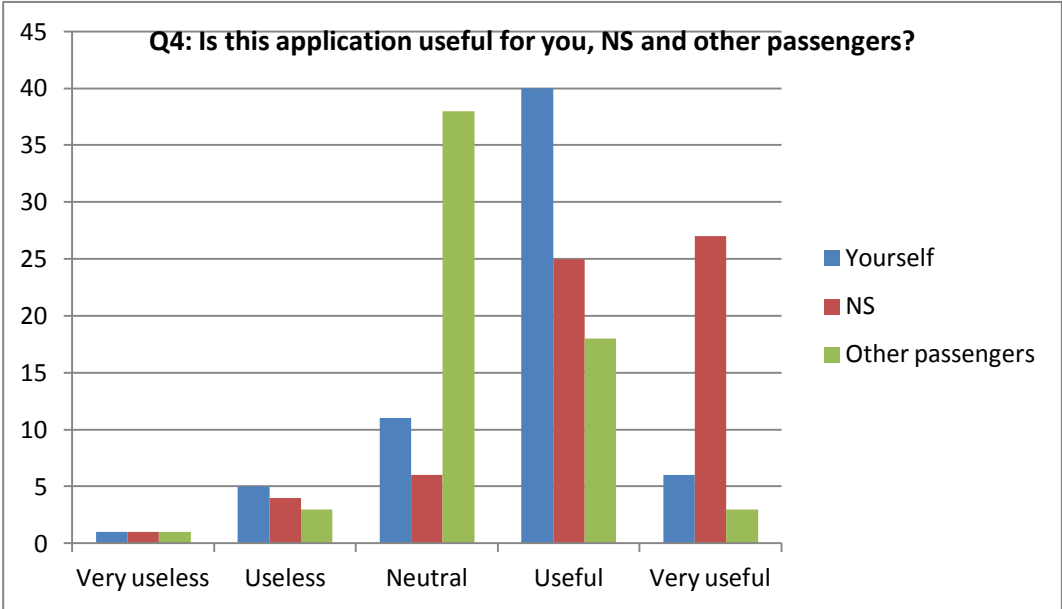


Figure 5.4: Results on Q4

The most interesting trend that we can derive from these results is that passengers seem to find the application useful. The extent to which the application seems useful is different per group. The usefulness for NS is valued higher than the usefulness for passengers, but the usefulness for other passengers is valued below that. It seems that passengers value the application as useful for NS, for themselves, but do not value the usefulness for other passengers the same as for themselves.

The fifth question, which completes this category of general questions, is about rating the interface and the ease of use of the application. The results on this question are presented in figure 5.5.

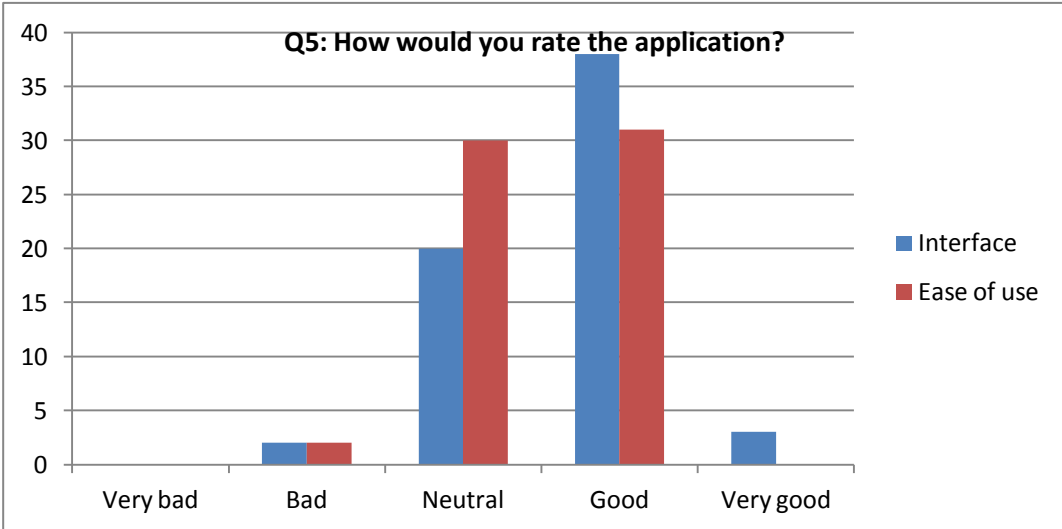


Figure 5.5: Results on Q5

The results in figure 5.5 show that passengers are quite happy about the interface and the ease of use of the application, although the interface is valued higher than the ease of use. This might be because the interface is a direct copy of the NS style.

## 5.2 Results on scientific models

The second category of questions in the questionnaire was concerned with the organizational genes model. The results from these questions are presented in this section. The first question in this category is meant to show whether or not passengers know the current procedures for giving feedback to NS. The result of this question is shown in figure 5.6.

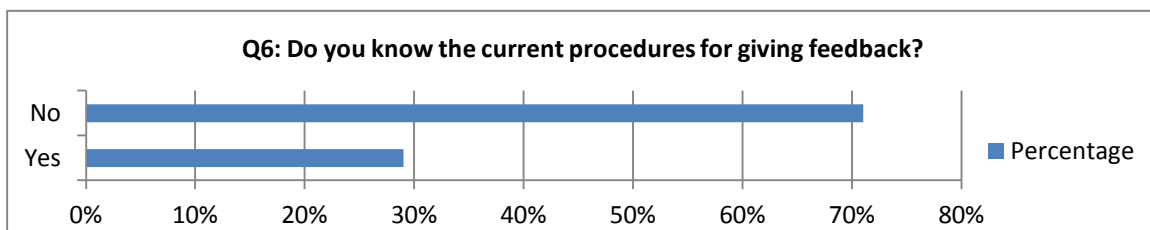


Figure 5.6: Results on Q6

The results show that only 29% of the respondents indicate that they know the current procedure to share their feedback with NS. This is a somewhat shockingly low number. If more people were acquainted with the procedure, more feedback would be available and NS would understand more about what is going on in passenger's heads.

The second question in this category is about the incentive passengers have to use the application; why would you want to use this application? We first asked an open-ended question, to prevent that passengers were inspired by our answering options. We analyzed the answers and categorized them. There are roughly five trends that can be recognized in the answers on this question:

- (1) To provide feedback. The first trend is the most logical; people want to use the application to share their feedback and experiences with NS. This trend holds for almost a third of the respondents.
- (2) Frustration. The second trend is that people want to use the application to write down their frustration and annoyances. This trend holds for approximately a quarter of the respondents.
- (3) Boredom. Some passengers also state that they would like the application in moment when they are using their device and are bored. This trend can be recognized by approximately 10% of the respondents.
- (4) The compensation. Some passengers state that they only want to use the application if there is a good compensation for the use. This trend holds for approximately 10% of the respondents.
- (5) Not use the application. The last trend is that passengers state that they do not want to use the application at all. The most frequent reason for this is that they think NS would not listen to them or would not take their feedback serious.

The third question, how could NS make you use this application, is about the compensation that NS can provide for making the use of the application higher. For this question we did the same as for the previous, analyze the answers and categorize them. We found three recognizable trends:



- (1) Use the feedback. The first trend is about passengers that state that NS should do something with the feedback, provide good solutions. About one 35% to 40% of the respondents answer with something that can be categorized in this trend.
- (2) Compensation. Another big trend, which can be recognized by approximately a quarter of the respondents, is that they would be motivated to use the application by the compensation offered. In this answer we did not look at the form of compensation, but just at the fact that the compensation would motivate the passengers to use the application.
- (3) Publicity. This category recalls the fact that only 29% of the passengers know about the current procedure for providing feedback to NS. About 20% of the respondents therefore think that they would use the application if NS would spend more publicity to the application.

The fourth question in this category is about the incentives to use the feedback application. As baseline for this question we used the three incentives as stated by Malone (2009); money, love and glory. We added an extra category, namely other. The first result that we present is the number of answers in each category, which is shown by the blue bars in figure 5.7.

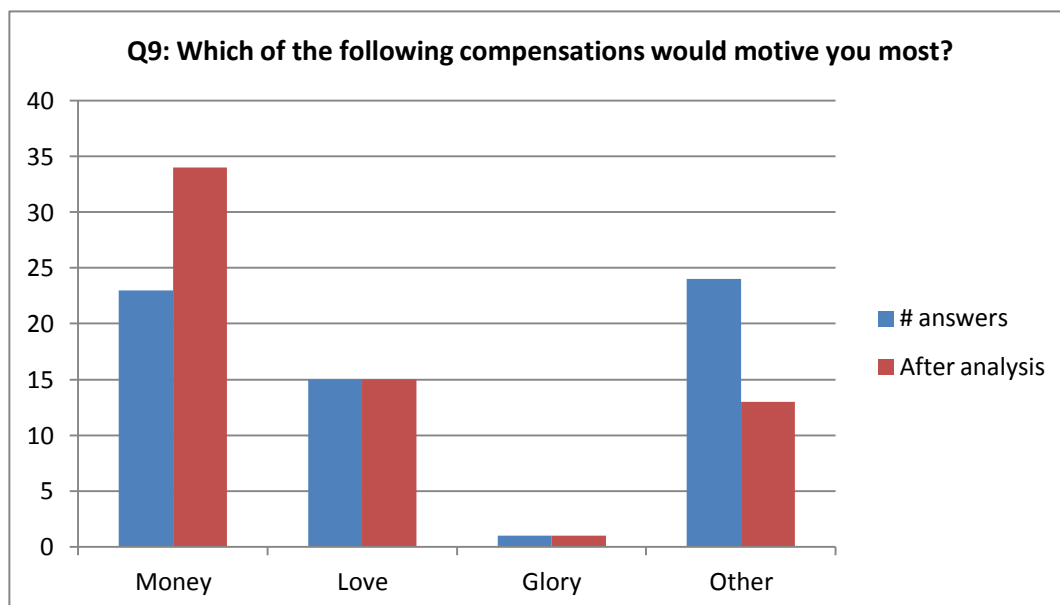


Figure 5.7: Results on Q9

The answers labelled as 'other' are analyzed in more detail. It turns out that from these answers, some can be categorized as money, love or glory. Out of the 24 answers, 11 can be labelled 'money', five respondents state that they do not want any compensation and another five state that NS should do something with their feedback as compensation. Two respondents state that they want to contribute out of frustration, so their compensation would be to write their frustrating comments down. One respondent states that he would like faster internet with a guaranteed speed as compensation for the feedback post. It is interesting to see that glory does not seem to be an important incentive in this context. Money appeals more than love, especially when we also include the answers originally labelled as other. The results after analyzing the answers labelled as 'other', are presented by the red bars in figure 5.7.

The fifth question in this category provides five statements from which the respondents have to choose. Statements one, three and five are derived from the incentives as presented by Malone (2009). Statements two

and four are derived from a brainstorm with representative passengers up front. The last statements can be used if none of the other statements fit. The results of this question are presented in figure 5.8. The statements are as follows:

- (1) The service of NS needs to be improved, to help them I would use this application
- (2) I would use the application to express my frustration
- (3) If the NS wants feedback from me, they should offer me something for it
- (4) Only when NS forced me, would I use the application
- (5) If my friends use the application, I would also use it
- (6) None of the statements above fit me

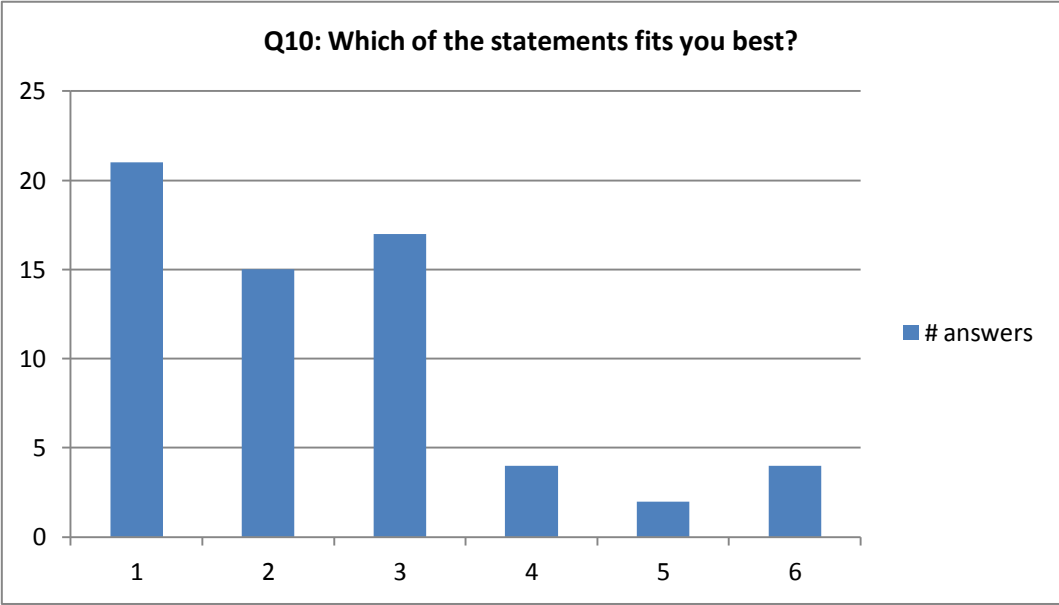


Figure 5.8: Results on Q10

Interesting in these results is that the highest fraction of respondents state that statement 1 fits them best, followed by statement 3. Statement 1 can be translated into the love-incentive by Malone (2009); statement 3 can be translated into the money-incentive. Although the preference here is opposite to the previous question, it confirms the fact that money and love seem to appeal more than glory, which is statement 5. Another category that scores high is the frustration incentive, which is not presented by Malone. Noteworthy is the fact that more people indicate frustration as a motive in this question than in the previous question, question 9. It could be that respondents rather check an existing answer than think of a new one. However, this phenomenon indicates that frustration is a motive not to be ignored.

The last two questions in this section of scientific models are derived from the Doan (2009) four challenges model. The focus in these questions is on whether or not passengers would be happy if NS keeps them informed and how this would affect their use of the system. The results are presented in figures 5.9 and 5.10 on the following page.

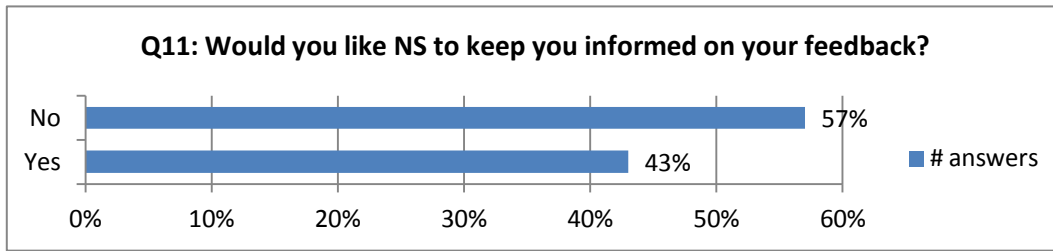


Figure 5.9: Results on Q11

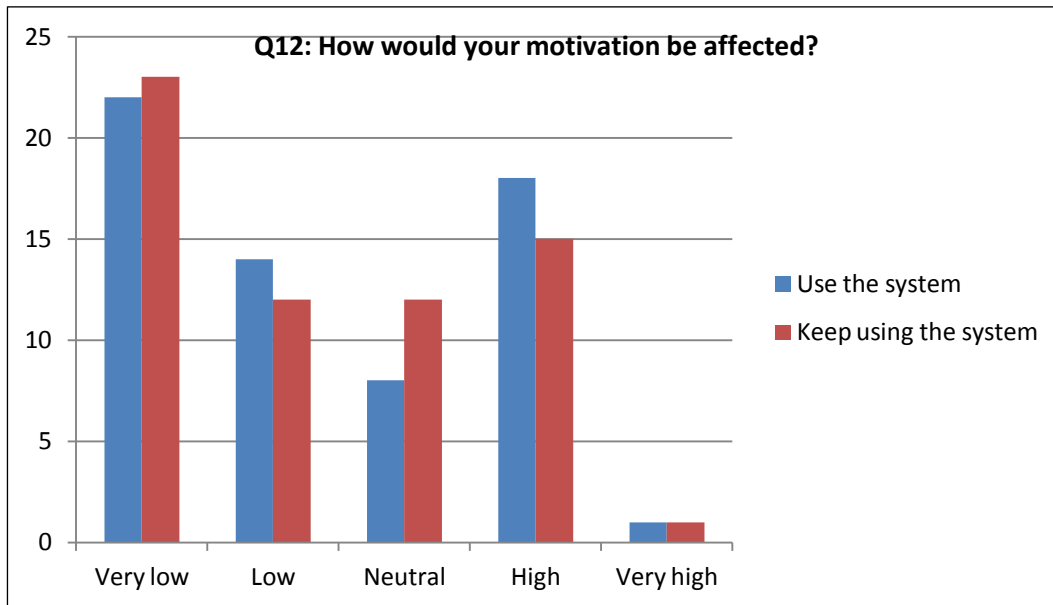


Figure 5.10: Results on Q12

It is quite interesting to note that a small majority of respondents does not prefer to be informed on their feedback and how it is processed. This trend can also be seen in how passengers think this would affect their motivation to use the application and keep using it. The majority is either very lowly motivated or highly motivated. These numbers seem to be in accordance, the passengers who do like be kept informed are highly motivated. The passengers who do not want to be kept informed are very lowly motivated.

### 5.3 General considerations

In this section we present some general consideration that we encountered during this validation research. We also present the answers to the final question of the questionnaire: do you want to share anything about this research or this questionnaire?

During our three-day validation research in the train, 158 new entries were made in the database. This means that, on average, each respondent filed more than two feedback posts. What is interesting about the use of different devices in the train is that the use of laptops seem to be concentrated more in the first class, while tablets are more used by younger people. However the number of trains equipped with internet is quite large these days, it occurred several times during our validation research that the train we planned was not equipped with internet. This meant that we had to wait for the next train, or change the route we had planned.

Since for each route we counted the number of laptops and tablets that were active, we can say something about these numbers and about the use of the devices. Since people can turn their device on and off at any moment, the numbers are not statistically significant, but just give an indication. There seems to be a higher number of active devices in the first part of the morning and in the second part of the afternoon. Both parts of the day are rush hours, typically moments that passengers travel from and to work. The same trend can be slightly recognized in the number of responses that were gained in specific hours. Figure 5.11 presents the number of responses divided over the day.

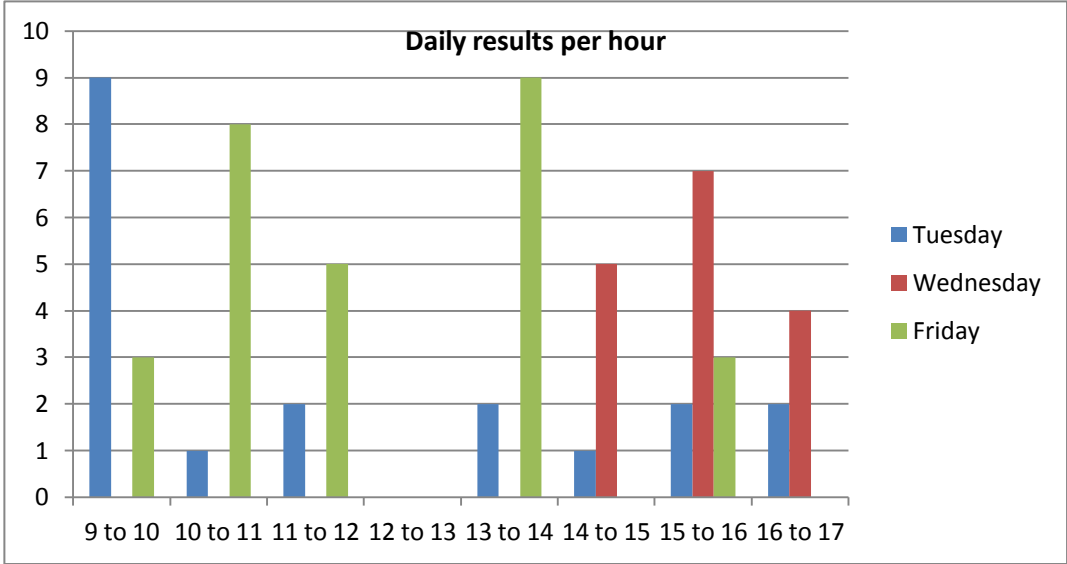


Figure 5.9: Number of responses per hour of the day

Another thing we encountered during our validation research, is the speed of the internet in the train. The speed seems to have a relation to the number of passengers that use the internet, which is quite logical. However, since we recognized the trend that during peaks the internet is used more heavily, the internet is slower during peak hours. Notwithstanding the fact that the application and the questionnaire were relatively lightweight applications, sometimes it took more than a minute to load them.

In the final paragraphs of this validation research chapter, we want to end with the answers on the final open-ended question in the questionnaire. In this questions we asked people if they wanted to share any comments on the research or the questionnaire. Fifteen passengers took the liberty of commenting on this question. Seven had a compliment about the research. Five of them posted a comment related to mobile phones. Although our research shows that smart phones a often used in the train, these comments and our research also show that the majority of these users do not use the free internet connection in the train. This means that passengers using their own internet connection do not need to connect via the portal. This means the crowd sourcing application on the train’s portal website will not be shown to them either, so those passengers cannot contribute. We could solve this problem by implementing the crowd sourcing application not only on the portal website, but also including it in the mobile app from NS, Reinsplanner Xtra, or an app specially designed for this purpose.

Three passengers commented on the features that are provided by the application and suggested features for the future. One of these features is to also include feedback on how clean the toilets, windows and compartments are. These features are already described in chapter three. One last passenger commented the following on success factors of such an application: “One of the biggest success factors is whether the NS takes these feedback messages seriously and tries to solve the problems mentioned in it.”

## 6. SOLUTION VALIDATION: EXPERT INTERVIEW

*In the previous two chapters, we presented a questionnaire for validation research (including a basic prototype) – see chapter four - and questionnaire's results in chapter five. In this chapter we will extend our validation research with an expert interview. This expert interview, along with the results from the questionnaire, is used to answer the validation questions by Wieringa as stated in section 1.4. The first section presents the outcome of the expert interview, the last section reflects on the original validation questions.*

### 6.1 Expert interview at NS

As soon as the results from the questionnaire were clear, we arranged an expert interview with Bram Hansma (passenger information business consultant) and Herbert van Buitenen (product manager passenger information) from NS. We had two main reasons to conduct such an interview. First, we sought reflection from NS on the research performed and described in this thesis. Second, we aimed to get a more complete answer on the validation questions as stated in sections 1.4 and 6.3. We decided to conduct the interview in an open manner, meaning that we did not develop a script up front. The first step in the expert interview was to present the research we did during the last few months. Concluding this fifteen-minute presentation of the research and the results, we posed three main questions on which we wanted the NS professionals to reflect:

- (1) What do you think of the research and its applicability?
- (2) What are things you would change in the intended solution?
- (3) What do you think of the results gained with the questionnaire?

We found the two experts at NS, both in an area that connects with our research, in a previous meeting with Erik Boshoeve. The first expert is product manager for passenger information. He is responsible for the passenger information in the “thuis & onderweg” domain. As such, he is the product owner of the NS Xtra passenger information smart phone application and the Ns website (e.g. the passenger information part of that website). Second expert is a passenger information business consultant for NS. This means that he acts as a linkage between NS's demand and supply organization. We invited both professionals for an interview at NS headquarter in Utrecht. In the following sections, we present the headlines from this expert interview.

The first impression of both the business consultant and the product manager were positive. Both concluded that the research was innovative and that NS could certainly benefit from organizing a crowd sourcing initiative. However, the question would be in what area and to what specific purpose NS should deploy crowd sourcing initiatives. According to the product manager, his primary focus is not on retrieving feedback. Although our research is out of his primary scope, he is interested in the possibilities and applications. The business consultant however, has a much higher need for feedback, especially in the context of processes and systems. It appears difficult to get high quality feedback with the current tools. He is therefore more interested in the solution we presented. However, both gentlemen see the advantages of crowd sourcing application in their specific domains.

The most interesting result gained from the questionnaire is, according to the product manager, that the smart phone is the most used device in the train. Combined with the result that shows that most of the smart phone users in the train almost never use the train's internet connection, this indicates that most of the smart phone users never see the train's portal website. The business consultant underlines this. Together they argue that this problem, combined with the fact that not every train has an internet connection, provides three possible

approaches to use during the implementation of crowd sourcing initiatives; mobile, portal website or both. The first of these channels, the mobile channel, is the domain of the product manager. Since the business consultant is linked with passenger information in general, all three channels apply to him. Both professionals agree that the mobile channel would be much easier for the situations where there is no internet in the train. People can then use a crowd sourcing application on their own smart phone, using their own mobile connection. This also means that the application of crowd sourcing initiatives is no longer limited to in-train settings. With a mobile phone application, passengers could also be used in settings before and after the travel, on stations, and so on. On the other hand, using the train's portal website also has certain advantages. Since every train has a local installation of the software that runs the portal website, different trains can be implemented in different ways. This means that each train can provide a different application or a specified application. It also means that the front end of the application, installed on the local version of the portal website, can identify from which train a user is making a contribution, including further details about the train's geographical position and information on the trajectory the train is travelling. In fact, since there are multiple Wi-Fi access point antennas in the train, the local installation of the portal website should even be able to detect from which compartment a contribution is made. The third option, to use both channels simultaneously, is also considered a possibility by both the product manager and the business consultant. We can consider the channel to be used within a future implementation as one of the main takeaways from our expert interviews.

A second takeaway we collected throughout the expert interview, is the specification of 'the crowd'. Both the product manager and the business consultant are unsure about what they would specify as the crowd. They state that more contributions, resulting in more data, is theoretically better for the organization. But more data also means that the back-end of the system, the place where the information is processed, needs to be better organized. The back-end can be seen as a system in which the feedback messages are received. The messages need to be processed by the back-end system quickly and automated. The processing of feedback can include filtering, grouping, aggregation and judging. The processing stops with the automated follow up, the processed feedback message is sent to the right person or system. The product manager adds to this, that the internal processes within NS should be in order before such a crowd sourcing initiative could be implemented. Asking whether this is currently the case, both professionals agree that in some areas the internal processes are in order, but in other areas this is certainly less the case. Reflecting more on the specification of the crowd, we conclude that there are two imaginary scales that are inversely proportional. The bigger the crowd is, the better the back-end needs to be. The reverse is also true, the smaller the crowd the less perfect the back-end can be.

This leads us to the question what the intended crowd for a crowd sourcing initiative at NS should be. Given the fact that the internal process is not in order in all places, it would not be wise to start with an initiative focused at all passengers at once. This would imply that the back-end needs to be perfect directly from the start, which is not possible since the internal processes are not in order in all places. On the other hand, it would also not be wise to start with just a few people. This would imply that only a small amount of feedback would be submitted, which means the effort needed for implementation outweighs the gain that can be derived. But there are a number of other options on the spectrum between the whole crowd and just a few people. The first of these options is the crowd of so-called signal passengers (in Dutch: signaalreizigers). Signal passengers are NS employees that report problems during their own travel. This means that they report their experience in the train, ranging from broken seats and dirty toilets to failing displays or unfriendly conductors. Signal passengers contact the Medewerker Contact Centre (MCC) to submit their feedback to a colleague. This colleague from the MCC will send the feedback to the right person in the organization. The MCC functions as a back office, which means that there is little automation or formalization. The business consultant recognizes a nice tool in the intended crowd

sourcing initiative. He states that with such a tool, the signal passengers can provide their feedback in a faster, easier and more uniform way. A second option, which is kind of similar to the first, is using NS employees as the crowd. This means that every NS employee in the train, can provide feedback or share their experience with the headquarter using a crowd sourcing application. There already is a less formalized feedback mechanism for employees, but the advantages of utilizing crowd sourcing for this would be to formalize and streamline the process and have a dedicated back office. The final option is to arrange a special focus group for the use of a crowd sourcing application. In the current situation there are already some existing focus groups, for example for filling in questionnaires or counting passengers. An extra focus group could be created for providing feedback about the passenger information in the train. Another important aspect about the crowd is the publicity that is given to the application, so that passengers know about the application. Results from our questionnaire indicate that only 29% of the passengers know current methods to share their feedback with NS. Our experts agree that this is not a high number and publicity should be a focus point when implementing a crowd sourcing initiative.

After the experts reflected on the possible channels and the composition and specification of the crowd, we asked them what a good use of such a crowd sourcing application would be. As stated before, the product manager does not have a big need for such an application, but the business consultant does. He states that it is not wise to start with the audio announcements in the train, because that would reveal information on how well conductors are doing their job. Neither would it be a good idea to ask people about the service the conductor provides. Therefore the crowd sourcing application is best used for measuring details on a system, not on human performance. Measuring human performance would also give valuable insights, but legislation and labor unions prevent organizations from doing this. A crowd sourcing tool could be a nice addition to already existing monitoring tools that NS has. This means that, for example, asking for feedback about the information on the displays in the train would be a good situation for using a crowd sourcing application. This is because the information on the displays comes from a complex chain of systems, failure can be everywhere and interaction between these systems is critical. In other words, a critical human look at the screens would be very welcome. The product manager adds that there are in fact two types of information that can be derived from such an application; objective factual information and perceptual information about one's experience. An example of the first type is feedback on the passenger information or other systems in the train, an example of the second is to measure the experience passengers have in the train. An interesting additional thought, especially applicable implemented in the form of a mobile application, is to add photos to your feedback posts. According to the business consultant this is input that would lead to a better problem analysis than only scripted feedback.

Summarized, we can derive the following information from the expert interviews at NS:

- NS is interested in crowd sourcing in the train, although the specific application of it is unclear.
- There are two different channels that could be used for implementation: smart phones and the portal website. There are three strategies for implementation of these channels: use only smart phones, use only the portal website, or use both.
- There should be a fit between the quality of the back-end and the size of the crowd. A bigger crowd means that the back-end needs to be better organized.
- Implement a crowd sourcing initiative on the train's portal website, limits the applicability of crowd sourcing to in-train. Using a smart phone application extends the applicability to before and after the travel, on stations, et cetera.



- When using a smart phone application, passengers could be given the option to post pictures with their feedback. According to the experts, this makes problem analysis better and faster.
- Publicity of the initiative is a key factor in the success of it.

## 6.2 Validation questions from the regulative cycle

In the final section of this chapter, we want to recap the validation questions as proposed by Wieringa in the regulative cycle model. The following validation questions were already stated in section 1.4:

- **Internal validity:** Would this solution design, implemented in this context, satisfy the criteria identified in the problem investigation phase?
  - **Causal question:** In this specific problem domain, would this solution have the intended effects?
  - **Value question:** Do the effects satisfy stakeholders?
- **Trade-offs:** How would slightly different designs, implemented in the same specific context, satisfy the criteria?
- **External validity:** Would the design, implemented in a slightly different context, also satisfy the criteria? This kind of validation is also known as sensitivity analysis.

During this research project we gained results from different validation processes; two expert interviews and a questionnaire. The results from the questionnaire are used to provide an answer to the causal question. The expert interview with NS is used to provide an answer to the value question and the trade-offs. All results together are used to answer the final validity question, the external validity.

### 6.2.1 Internal validity

When reflecting on the internal validity of this research project, two questions are proposed by our research model by Wieringa. The two questions are stated in the previous section, being the causal question and the value question. In the introduction we stated that this research was meant to explore the potential of crowd sourcing in the context of NS. The results from the literature study show that there are two preconditions for crowd sourcing to be successful; a technical platform and a willing crowd. Within the introduction of NS we already found that a possibility for the technical platform for a crowd sourcing application is partly present, namely the OBIS, which has enough capacity and possibilities for extension. However, this system can only be used for deploying the front-end of the intended application, which means that the back-end needs to be developed specially for this purpose. Moreover, the OBIS cannot be used if a crowd sourcing initiative is deployed for use on smart phones only. In that case we could look at the possibilities of using the NS Xtra smart phone application. Summarizing, we can indicate that OBIS can play a role in the technical platform, and so does the Xtra application, but the technical platform is not complete without the back-end systems.

The results from the questionnaire indicate that there is enough support from passengers to participate in the crowd. Moreover, results also indicate that people are willing to use the intended application and keep using it. This means that both preconditions are met in this specific context. If we summarize the intended effect we are looking for in this research as gaining knowledge about what passengers in the train think of their experience, then we can answer the causal question positively.



The second question in analyzing the internal validity, is the value question. Although it is quite difficult to analyze the value from a prototyping experiment, the expert interview at NS indicates that both the business consultant and the product manager can see the value in our crowd sourcing idea. Of course the real value can only be assessed once the intended application is running in the real environment. With this information we cannot state with certainty that the value question will be answered positively, but we have a strong indication that this will be the case. Combining the answer on both questions, we can assess the internal validity as strong.

### 6.2.2 Trade-offs

Trade-offs are something that will always occurs during development projects, especially when development happens in an iterative process. Since the development of a crowd sourcing initiative would be project where iterative development is most applicable, trade-offs will always happen in the development process of such a crowd sourcing initiative. In this research project we recognized two different kinds of trade-offs, the first being the trade-offs that are not or minimally critical for the success of our crowd sourcing initiative. This kind of trade-offs are for example the look of the user interface, the fields that are included in the application and the names of certain fields. The second category contains the trade-offs which are critical for the success of a crowd sourcing initiative. In other words, these trade-offs affect the main idea behind the crowd sourcing application.

Since the main idea of our crowd sourcing initiative is to receive feedback on the passenger information, critical trade-offs are the ones that affect this idea. Since we made a lot of smaller and less important trade-offs, which do not affect our main idea, we will describe only the more important trade-offs. The first trade-off we made, is to choose for the portal website as a medium for implementation. We made this choice because we had the indication that with this channel we would the most passengers. Results from our questionnaire indicate that we would have needed a smart phone application, or at least an application for the mobile portal, as well. The second important trade-off we made, was to require users of the application to fill in their name and email address. We realized that this requirement could create a barrier, but in our perception we needed the credentials to evaluate users, let them participate in compensation programs and to mitigate noisy contributions. The final and biggest trade-off we made, was to choose to save the contributions in a simple database. The explorative character of our research justifies this choice, but in a real organizational setting, a better back-end system needs to be used. This something we elaborate more on in section 7.3.

To conclude, trade-offs are a phenomenon that will always show up during development projects. Moreover, the effect will be stronger if an iterative development cycle is used. However, the results from our questionnaire, expert interview and our experience during this project, indicate that the minor trade-offs will not affect the success of a crowd sourcing initiative (much). This means that trade-offs that do not affect the main idea behind the initiative, are relatively save to make. On the other hand, the trade-offs that are critical for the success of a crowd sourcing initiative need to be analyzed in detail before making a decision. For example, we choose to implement the application only on the portal website. The results from the questionnaire show that smart phone users are unsatisfied with the performance and possibilities of the application on the smart phone. In this specific context this choice has no consequence, but in real organizational settings this would have been the case.

### 6.2.3 External validity

The final kind of validity, external, can be assessed by analyzing the possibilities of the intended solution in other fields. Since this research project is the application of general idea to a specific context, we can analyze two different kinds of external validity. The first contains the possibilities of our general idea, crowd sourcing, in other areas than public transportation. Considering the crowd sourcing examples we stated in chapter two, it may not

be a surprise that the idea of crowd sourcing is working in all different kinds of fields. Moreover, the success of crowd sourcing initiatives in the real world, is one of the requirements for starting this research project. It is therefore not that interesting to discuss the external validity in this context, since the idea is already validated in dozens of different contexts.

The other kind of external validation, our specific crowd sourcing initiative applied to other fields, is something which is more interesting to discuss. The main idea of our crowd sourcing initiative is to receive feedback from passenger, customers in general, on the passenger information in the train. The novelty in this project is that we apply this technology to the domain of public transportation, but we can also change the domain. Asking for feedback is something which could be applied to almost every organization. Moreover, it something which can also be applied to governmental institutions like provinces, municipalities and cities. Logica is already testing an application which asks inhabitants of a certain municipality, Westerveld, to comment on the experiences they have with the municipality. Inhabitants can use this application, which is smart phone based, to report a bad sidewalk in their municipality. Other Logica initiatives, like an application for the city of Amsterdam and the 'Verbeter je buurt' project, have similar characteristics like our research project. In other words, the concept of using the crowd for feedback or ideas for process improvements can be applied in many different fields.

Throughout this thesis we already presented the idea of taking our crowd sourcing initiative to the next level, by introducing other sources for data; Twitter, Facebook, log-files from other systems et cetera. An example of a project where similar characteristics play a role is Compronet: the Community Protection Network. This project is a joint venture of Logica and the Dutch Police. The goal of this project is to create a real-time information system about the disturbance of the public order. This should lead to a faster recovery of the public order in case of incidents, with help of optimal information exchange by witnesses, civilian participants, organizations and professional first responders. The choice has been made to prototype this idea within a small number of police forces, connecting Facebook, Twitter and other social media to a so-called Complex Event Processing (CEP) engine. Such an engine is meant to combine several data sources and extract information from these different sources. Initial results show that this idea is working and the prototype will be further developed in the forthcoming months. This is just one of the examples that our idea could also work when taken to a next level.

## 7. CONCLUSION

*After we completed the validation phase with a questionnaire and expert interviews, we use this chapter to present the conclusions of our research. After the conclusions we reflect the outcomes of this project in the context of each of the stakeholders. We will discuss what these outcomes mean in that specific context and what could be future steps for each of the stakeholders.*

### 7.1 Results

In this section we present the results of our research project. In other words, we present what we discovered during this research. To recall what we already stated in the introduction; to make crowd sourcing applications a success, both a technical platform and a crowd should be available. Since both of these preconditions are met in the context of NS, we chose to conduct a case study in this domain. The results are presented in this section.

#### **A validated set of requirements for future crowd sourcing initiatives at NS has been developed**

To develop a prototype of the crowd sourcing application, we first elicited requirements and developed a functional and technical design. The functional and technical design present a set of requirements to a crowd sourcing application that could be implemented in the context of NS. One of the results of this research project is therefore the set of requirements to a future crowd sourcing application in the Dutch intercity trains. The set of requirements is validated by information engineers from Logica and stakeholders from NS and the University of Twente. The results from the questionnaire do not show any indication that the set of demands should be changed.

#### **There is enough support from passengers to make in-train crowd sourcing a success**

With less than 10% of the respondents answering they would not participate in a future crowd sourcing application in the train, we can conclude that there is enough support from passengers. The technical platform, which is necessary, is partly provided by OBIS. The questionnaire shows that 94% of the respondents own a smart phone and respectively 83% and 48% own a laptop or tablet. The majority of respondents make use of these devices in the train on a regular basis, which means that the second precondition for crowd sourcing, the crowd, is also sufficiently present. Together with the fact that 73% of the respondents indicates that the prototype would be useful or very useful to them. We can conclude from this that there obviously is enough support from passengers to make the crowd sourcing application a success.

#### **The technical platform is incomplete**

Throughout this research we stated several times that both a technical platform and a crowd are necessary for crowd sourcing. We also stated that the OBIS in the train could function as technical platform, but this is only a start. When deploying a crowd sourcing initiative on smart phones, OBIS is out of scope. In this case, the NS Xtra application could form a starting point for the technical platform. However, the technical platform is not complete without a back-end system that processes the information that comes in. The real challenge lies in implementing the back-end system, which completes the technical platform.

#### **The prototype application is not suitable for use on smart phones**

On the trains that are equipped with OBIS, a special portal website is available for smart phone and other mobile platforms. Whenever a passenger connects a mobile device with the portal website, the portal

website recognizes that the passenger is on a mobile device and redirects the passenger to the mobile portal website. The prototype presented in this thesis is developed only for the normal portal website, which is in hindsight a design flaw. When implementation of the crowd sourcing initiative goes via the portal website, a second application needs to be built specifically for the mobile device users.

#### **Smart phone users do not use the NS internet often**

Due to the fact that smart phones are often purchased in combination with a data plan from the service provider, passengers seem to use their own 3G connection more often than the internet in the train. Several reasons could be presented for this, the most important being the speed of the internet in the train. An effect of people using their own connection is that they are not presented the portal website and can therefore not participate.

#### **An application for smart phone users is needed**

During this research we also found a slight mismatch, presented in the two previous conclusions. The fact that the prototype application was not that suitable for smart phones, along with the fact that most smart phone users do not use the NS internet, pleads for a special application for smart phones. A simple solution would be to integrate the application into the Reisplanner Xtra. This application is already provided by NS for multiple services and on multiple smart phone platforms.

#### **Frustration, boredom and compensation are important incentives to participate**

In chapter two we presented a model with crowd sourcing incentives, by Malone (2009). Results from our questionnaire indicate that providing passengers with glory, as Malone calls it, is little incentive in this specific context. Compensation for contributions seems to be the best incentive, with almost half of the respondents indicating this as a main motivator. Additionally, contributing out of frustration and boredom also seem to be incentives not to be ignored – two factors not described by Malone. Future research can point out in more detail how these incentives work, how they can be amplified and what other incentives fit this context.

## **7.2 Reflection in scientific context**

From a scientific point of view, this research project utilizes four scientific models. Three of these models are considered methodological models; the regulative cycle and the testing & evaluation platforms (TEP) model. Furthermore we utilize the four challenges model.

#### **The incentives model as proposed by Malone is incomplete**

Our interpretation of the results gained from the questionnaire, indicates that the incentives model as presented by Malone et al. (2009) is incomplete. Initial results on the questions related to this model indicate that 38% of the respondents answer with an option that is not included in the model. Analyzing these answers in more detail shows that 45% of these answers can be considered money-related in terms of Malone. Moreover, 21% of the answers can be labelled 'love', which means that 34% of the answers is really outside the model of Malone. We validated this result with a second question, letting passengers choose between six statements, which show that 36% of the answers cannot be categorized according to the Malone model. Incentives outside the model that came up in the questionnaire are frustration and boredom. In this context, frustration often occurs when people have to wait during disruptions or when they miss a connection. We indicate that this incentive is specific in this context. Boredom probably occurs since people are 'locked in' in the train for a certain amount of time. We

indicate that boredom is an incentive that is not specific to this context, but could occur in other contexts as well.

In chapter two we presented the publication by van der Wees et al. (2011) on the potential of in-train crowd sourcing. Since this research is in the same context, we want to compare our results with the results as obtained by van der Wees et al. in their research paper. Their most important conclusions were as follows:

1. The most valuable data for NS is a better understanding of who is travelling their trains and at what trajectories. Within the train [...], feedback on the sentiment of the journey, the attractiveness of the train environment, and the feeling of safety are most sought after.
2. Most people clarified that if they felt that it was useful, they would likely keep using the application.
3. Customers are most likely to adopt applications that give them the feeling they contribute to something worthwhile. [...] Applications closer to the core business are more likely to get adopted.

The first conclusion that van der Wees et al. draw is based on interviews with employees from NS. In the same context, crowd sourcing in the train, we found similar results, namely that customer satisfaction lags behind in certain areas. The information NS needs according to van der Wees' interviews, also seems to be information that we found necessary in this research. The second conclusion, also derived by van der Wees' interviews, is also a much given answer in our questionnaire. People want to participate if they have the idea that NS is taking their feedback messages seriously and really trying to solve the problems. The third conclusion is less relevant in our context, because in this conclusion van der Wees et al. conclude what kind of application the passengers would like to have. Since the feasibility of our application was never a point of discussion, we cannot reflect on this conclusion.

In all, there are two things that this research project adds to the conclusions of the publication by van der Wees et al. The first is that their second conclusion, passengers keep using the application if they feel it is useful, is not complete enough. The conclusion is certainly true, but there is more to it. Our research shows that people are not only motivated by the fact that they want to do something useful, but that there are other incentives like frustration, boredom and compensation for example. Moreover, where van der Wees et al. explored the first concepts of crowd sourcing in the train, we took the idea one step further and user-tested a prototype. The results from our research project support the conclusions of the research by van der Wees et al.. These conclusions present a strong indication that passengers in the train are willing to participate in crowd sourcing initiatives.

## 7.3 Reflection in the context of Dutch Railways

In the first part of this section we present our findings, additional to the findings in section 7.1, in the context of NS. The final part of this section is used to reflect on these findings and what impact these findings have on NS.

### **The back-end of the application is crucial**

The results of our questionnaire show that one-third of the respondents want to contribute to a crowd sourcing application in the train because they want to improve the experience of their travel. In the open-ended question on how NS could motivate passengers to participate, an often-stated incentive is NS doing something with the information that is provided. The questionnaire also shows that the willingness to participate depends on whether the contributions are actually used. Thus, for this crowd sourcing application to be successful, the back-end needs to be developed first and integrated with the current

business processes. Only if the contributions that passengers make really reach the organization and the organization takes them seriously, NS could benefit from large amounts of relatively inexpensive data.

### **Publicity is another key factor in the success of the crowd sourcing application**

The support from passengers could improve if NS would give more publicity to the existence of the crowd sourcing application and the right incentive in passengers to participate. Results from the questionnaire show that only 29% of the respondents indicate that they know the current procedure for providing feedback to NS. The same results also show that the majority of respondents indicate that they would be mostly triggered by the money incentive. This means that participation would probably increase if NS offered contributors a money-related compensation for their contributions.

### **Implementation happens as stand-alone application or part of a social media environment**

The research shows that there are two different projects that could follow from this research. The first project would be to implement a crowd sourcing application as a stand-alone application. This means that the application as proposed in this thesis, could be implemented and integrated with the business processes of NS. A second project could be to implement a crowd sourcing application as part of a bigger whole, the social media environment as described by Divol et al. (2012). The latter project is also related to Logica's Compronet project as briefly presented in section 6.2.3, which presents the idea of a Complex Event Processing (CEP) engine. Such an engine could be part of the back office system, which makes integration with Twitter, Facebook and other social media platforms possible. Furthermore, the CEP-engine could also be used for the existing Twitter-desk at NS, to shift the messages. In this way the NS Twitter-desk employees do not need to read every tweet, something that is the case in the current situation.

After we presented the results in the context of NS, we want to reflect on what these results mean for NS. First of all, we want to state that the content of this research project is just a starting point. We indicate that there are four aspects that are important to consider in the implementation of a crowd sourcing initiative. These four aspects, which we will reflect on in the following four sub sections, are as follows:

- (1) Users in a crowd sourcing initiative (section 7.3.1)
- (2) User interaction (section 7.3.2)
- (3) Back office system (section 7.3.3)
- (4) Incentives to participate (section 7.3.4)

#### **7.3.1 Users in a crowd sourcing initiative**

As we presented throughout this thesis, there are two preconditions to make crowd sourcing a success. The first of these preconditions is the crowd. Without the crowd, crowd sourcing is not possible, so this is an aspect to consider. We indicate that in this specific context, in-train crowd sourcing, the crowd could be defined in different ways. On one side of the spectrum we see a closed group of people acting as the crowd, on the other side we see all passengers as the crowd. As stated before, the choice on this spectrum is connected with the quality of the back-end system; the bigger the crowd, the better the back-end system needs to be.

When we think of small groups of people as the crowd, we can think of special interest groups (SIG), focus groups of passengers, NS employees or the so called signal passengers. The advantage of having a small group of people as the crowd, is that the back-end system does not need to be perfect. Small groups of people could already make use of a crowd sourcing initiative in an early stage. On the other hand, a small group of people,



crowd, can still make valuable contributions. Since a crowd sourcing initiative is relatively easy and cheap to start up, a small crowd could already turn out to be profitable. On the other hand of the spectrum that defines the crowd, we indicate the option to include all passengers in the crowd. The advantage of this option is that more contributions are potentially made. But more contributions means more data, which on its turn means that the back-end system needs to be organized in a much better way. We reflect more on the quantity of data in section 7.3.2.2 and more on the back-end system in section 7.3.3.

## 7.3.2 User interaction

This section describes how the interaction between the user and the application or organization can be implemented. As with every other means of communication, two important things should be considered here: the medium and the message.

### 7.3.2.1 Medium for user interaction

In this specific context, in-train crowd sourcing, we see two different mediums or channels for implementation of crowd sourcing initiative. Within these two channels, we indicate three strategies for use of the channels. The first channel is the train's portal website. The advantage of using the portal website is that it runs locally on each train. This means that each train can run a different or specified version of the crowd sourcing initiative. Moreover, since each train runs its local portal website, messages that come in via the portal website can be submitted included with the train number, trajectory and even compartment number. The big disadvantage of this channel is that it is not available in every train. The portal website is only available in trains that are equipped with OBIS, which currently about 150 trains are. A second disadvantage is that not every member of the crowd, user, is using the portal website. For example, passengers that are using their own mobile device and 3G connection, never see the portal website.

The second medium is via smart phones. This could either mean a dedicated smart phone application or implementation within the NS Xtra passenger information application that is currently available. The advantage of this channel is that many people today have a smart phone with an internet connection. The same goes for passengers in the train, something that is supported by results from our research and questionnaire. This means that with this channel, more people can be reached than with the previous channel. An disadvantage of using this channel is that localization of the passengers that contribute a message cannot be done in an automated way.

We propose three strategies for using specific channels during the deployment of a crowd sourcing initiative. The first strategy is to only make use of the channel portal website. The second strategy is to exclusively make use of a mobile application. The third strategy is hybrid, which means that we can include both channels in this strategy. According to the situation in which the crowd sourcing initiative is going to be used, each strategy has its advantages and disadvantages. Choosing the strategy of using both channels is most likely to have the highest number of messages received. If the crowd sourcing initiative is meant for the signal passengers, the strategy of using the portal website is most likely to be the best fit. In other words, each strategy has its own applicability.

### 7.3.2.2 Messages for user interaction

In our prototype, we asked passengers for feedback about the passenger information in the train, which means that the message is a feedback message on passenger information. In the solution design phase we already stated that the same application, with a different front-end, could also be used to report on the cleanliness in the train. But additional to our ideas, there are plenty of other examples where a crowd sourcing application could be useful: design of new stations, design of new trains, information during disruptions, estimates on how crowded

the train is, real time passenger information, and so on. Not only new ideas could benefit from crowd sourcing, but also existing NS projects could benefit from it. For example the project of counting passengers in the train, checking which kind of ticket people have, and benefit to surveyors in the train. In the current situation, these projects are done by external bureaus or agencies. If NS could perform these projects in-house and automated with use of crowd sourcing, this would save money.

Looking at the data that will be gathered with such an application, it is quite difficult to give estimates on the expected number of messages. During this research we learned that 660.000 internet sessions are consumed by passengers each month (Sloos, 2012; Tweakers, 2012). This means that 660.000 passengers browse through the portal website to accept the terms of agreement. If only 1% of these people would use a crowd sourcing application to contribute, this means that 6600 new data entries would be created each month. This means that the back-end of such a crowd sourcing application should be capable of doing more than just saving the contributions and alerting employees. We described a back-end process capable of aggregating and analyzing the contributions. In this way the contributions on the same subject or same compartment could be clustered. This could also indicate the importance or urge of a certain contribution or cluster of contributions. Such a back-end has another significant advantage; other data sources could also use the same back-end system. Lots of systems produce log-files, which could be processed in an automated way to add extra information to the back-end. The back-end could be extended by feeding on Tweets from Twitter or other messages from social networking sites. We presented the idea of a complex event processing engine as back-end in section 6.2.

### 7.3.3 Back office system

As we stated before, this research is just a starting point for in-train crowd sourcing. Receiving messages from the crowd, as we did with our prototype, is just one thing. The real challenge lies in designing and implementing a back-end system which is capable of handling these messages. If more message are submitted, something that is likely to be positive for NS, the back-end needs to be of better quality. We already stated that 660.000 passengers visit the portal website each month and we indicated that a 1% usage of the system would lead to 6600 messages a month. A good system that can interpret and process these messages is needed, because there is no possibility that every message is going to be read by an employee.

With the addition of features like machine input and feeds from social media, the implementation of a crowd sourcing system as a stand-alone system seems to fade away. That is one of the reasons why we would suggest to look at the options of implementing a crowd sourcing system as part of a bigger whole. Since NS is already trying to use social media sites like Facebook and Twitter, this seems a logical next step. The bigger whole we suggest here could be in the form of a social media environment as presented by McKinsey (Divol et al., 2012) in chapter 2. Such a social media environment could be a nice integration of the current social media efforts and crowd sourcing systems, in order to improve the brand and services of NS. The image of NS is a hot topic in the media for the past years. The stages as proposed by Divol et al. are a perfect match to the efforts of NS and the problems they currently encounter. Monitoring the social environment, in other words knowing what is said about NS, is something that is being done right now. Reacting to these messages is something that is being done, but could probably be improved. Crowd sourcing application could be a suitable tool for improving this process. Finally amplifying the positive brand messages and leading passengers toward long-term behavioural changes could improve the image of NS and therefore the customer satisfaction.

Something that is also related to the back-end system and the additional features, is the Complex Event Processing engine as presented in Logica's Compronet project. If additional features like analysing tweets and



log-files included in the crowd sourcing initiative, a Complex Event Processing engine is something not to be ignored. Combining messages from a crowd sourcing application, Twitter, Facebook and other social media, processing them in a complex event processing engine and thereby gaining valuable new information. Moreover, this new valuable information can then be used to improve the internal business processes, interaction with the passengers and the overall experience that customers encounter when travelling with an NS train.

#### 7.3.4 Incentives to participate

During this research we found that participation in a crowd sourcing initiative does not just happen, results indicate that there are incentives why people participate. Before our validating questionnaire we proposed three incentives as found by Malone: money, love and glory. Results indicate that glory does not play a role in this specific context. A large part of the respondents indicate that their incentive to participate is love. This means that they want to participate in a crowd sourcing initiative to help NS, because they somehow feel connected with NS. Another large part of the respondents indicate that money is their incentive to participate. This means that they want to participate in a crowd sourcing initiative if NS offered them some kind of compensation for their contribution. The importance of incentives depends on the context and specific use of the crowd sourcing initiative. If, for example, a crowd sourcing initiative is used as a tool for signal passengers, incentives are most likely to not play any role. If a specific crowd sourcing initiative needs more messages submitted, then the role of incentives becomes more primary.

## 7.4 Reflection in the context of Logica

In this section we want to present the results, in addition to the ones in section 7.1, of this research project in the context of Logica. The latter part of this section is used to reflect on the results of our research project and how these results could be useful for Logica.

### **The potential of crowd sourcing is proven and results indicate suitability in other fields of practice**

With the results we gained from this case study, we have proven that crowd sourcing applications in organizational context can give organizations great benefits. This research and current running projects at Logica, apps for the city of Amsterdam and the municipality of Westerveld, show that crowd sourcing applications really work and are not to be ignored as a tool for organizations. With the indication that these kinds of applications are not only working at NS and in governmental environments, the idea can be applied to much more contexts.

### **Logica can play a role as 'business integrator' in the social media environment implementation**

As mentioned before, there is a thin line between implementing the NS crowd sourcing application stand-alone or as part of a bigger whole. Should NS decide to go for the latter option, Logica's nature as business integrator is a perfect fit for leading such a project. Moreover, since Logica and NS have a solid relationship in doing projects together, connections have already been made and sincere interest has been shown. We would therefore suggest to pitch the ideas presented in this research project to NS and see if they are interested to build a social media environment jointly.

For Logica this research project could be useful in both a general way and a specific way. In general, this research and preliminary results from current projects, indicate that crowd sourcing is relatively easy and cheap way of getting consumers or customers involved in the business process. In this case study example we involve passengers to improve the business process of NS, but there are dozens of other examples where a crowd sourcing application could be useful. Logica already started to explore these possibilities with apps for the city of

Amsterdam and the municipality of Westerveld, but the possibilities are much bigger than that. This research shows that crowd sourcing is a concept not to be ignored. Moreover, the innovativeness of crowd sourcing fits the innovative image of Logica.

The more specific reason why this research project is of interest for Logica, is purely a business reason: the implementation of a crowd sourcing application at NS. Although NS could implement the application in two different ways, as stand-alone or as part of the social media environment, in both situations Logica can and should play a role in this project. Although we have to admit that Logica's role in the stand-alone implementation would not be as big as it would be in the social media environment implementation. Should NS decide to make the idea of a stand-alone crowd sourcing application more concrete, then Logica could be an important systems integration partner in this trajectory; this thesis might be a starting point as such. Although we presented just one example of getting information through the crowd, NS should take more examples into consideration. For Logica this is a great opportunity to develop a complete structure of what information could be extracted from the crowd, information which is of value to its customers. Once this vision is laid down, Logica has several experts on developing user-friendly interfaces and mobile applications. These people could work with NS on realizing a suitable interface or mobile application. The back-end of the application, which we concluded is most crucial for the success, is something that likely needs to be built from scratch. Since this back-end preferably needs to be integrated into the existing business processes and systems, Logica's role as business integrator is a perfect fit for this job.

Should NS decide to implement the crowd sourcing application as part of the social media environment described in chapters 2 and 7, the magnitude of such an implementation would be even larger. In that case the service and knowledge that Logica could offer are of even greater value and importance. Additionally, the implementation of such a social media environment would again fit the innovative image of Logica.

Last but not least a personal side note. During the last 8 months, I gained a lot of experience with Logica as an IT-services organization. I can conclude that these 8 months have been challenging, to say the least. A large reorganization, 1200 job losses, mentors that found new assignments and the upcoming merger with the Canadian IT-services organization CGI did not prevent me from getting a clear image of what Logica can do for organizations. Although the current situation in the IT-services field is challenging, Logica manages to get new projects, to detach professionals to their clients and to take care of outsourced it-projects for their clients. However, I am under the impression that Logica is mainly busy with detaching professionals to clients, the so-called hourly billing (in Dutch: uurtje factuurtje). To be truly innovative, which Logica wants to be, I would suggest that more in-house research and development could be done. Commercially of the shelf (COTS) products, new innovative applications and applying new technologies to business problems creates opportunities. In times when the market is in a dip, new opportunities are necessary for continuation of the business. Moreover, these new opportunities seem to have a higher innovative character than the well-known hourly billing. In other words, innovative projects pay off!

## **7.5 Challenges and future research**

In this section we want to share some comments on future research and challenges. Throughout this thesis we used the scientific model of Doan's (2011) four challenges for crowd sourcing applications. To explicate the future research that could be done involving this model, we first restate the four challenges:

- (1) How to recruit and retain users?

- (2) What contributions can users make?
- (3) How to combine user contributions?
- (4) How to evaluate users and their contributions?

From these challenges we used the first and the second in this research project. We found that recruiting users is mainly influenced by the publicity the organization gives to the crowd sourcing applications. Retaining users is something that can be done with compensation and incentives. During our research we proposed the incentives as proposed by Malone (2009), but we concluded that this model is incomplete. Future research could show what a better incentives model is for this specific context and, thereby, how users can be retained. The second challenge is used to think about what contributions passengers can make in this context. Since we wanted to test if there was enough support from passengers, we kept the contributions quite simple by making them scripted. Future research should point out what other contributions can be useful in the context of NS, which is a job that requires involvement from NS. An interesting sub domain in this challenge is the use of machine contributions. With machine contributions we mean contributions that have not been submitted by human users. An example of such an application would be the log files from different systems that are analysed automatically and added to the crowd sourcing application. Additionally, NS could perform data mining technologies on several social networking sites to find mentions about their services. These mentions could be added to the crowd sourcing system as well. Adding machine contributions to the intended crowd sourcing application is also a leap into the introduction of the social media environment as described in this chapter and chapter two.

The third and fourth challenge are not specifically addressed in this research project, but do seem useful. Before implementing the intended crowd sourcing application, both challenges should be analyzed in more detail. We already stated in the reflections that aggregating feedback messages could turn out to give more accurate and more valuable information. To do this, we proposed to use a Complex Event Processing engine, which is presented in section 6.2. Future research should prove the applicability of such an engine. The fourth challenge, how to evaluate users and their contributions, is something that was out of the scope of this project. However, if NS really gets 6600 messages a month, or more, as indicated in the reflections, there is a certainty that noisy contributions are among them. Since evaluating contributions is a job that needs human involvement, noisy contributions take time and thus money. Eliminating or mitigating noisy contributions saves money, which is the main reason to spend future research on this challenge.

## REFERENCES

- Abowd, G. D. (1999) Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal*, 38(4), 508-530.
- Abran, A., Moore, J. W., Dupuis, R., Dupuis, R.I., & Tripp, L. L. (2001). Guide to the software engineering body of knowledge (swebok). IEEE Press, 204. IEEE. Retrieved from <http://www.swebok.org/>
- Agile Alliance, (2001a). Manifesto for Agile Software Development. In: AgileAlliance.org. Retrieved, July 20, 2012, from: [www.agilealliance.org/](http://www.agilealliance.org/)
- Agile Alliance, (2001b). Principles: The Agile Alliance. In: AgileAlliance.org. Retrieved, July 20, 2012, from: [www.agilealliance.org/principles.html](http://www.agilealliance.org/principles.html)
- Alexander, B., Levine, A., (2008). Web 2.0 Storytelling: emergence of a new genre, in: *EDUCAUSE review*, vol. 65, November/December, 2008, pp. 40-56.
- Ballon, P., Pierson, J., Delaere, S., (2005). Test and experimentation platforms for broadband innovation: examining European practice, in: Conference Proceedings of 16th European Regional Conference by the International Telecommunications Society (ITS), Porto, Portugal, 4-6 September, 2005.
- Bergvall-Kareborn, B., Hoist, M., Stahlbrost, A., (2009). Concept design with a living lab approach, in: Conference Proceedings of Hawaii International Conference on System Science (HICSS), Big Island, Hawaii, 5-8 January, 2009.
- Borst, W.A.M., (2010). Understanding Crowdsourcing – Effect of motivation and rewards on participation and performance in voluntary online activities. *ERIM PhD Series in Research Management*, 221.
- Brabham, D.C., (2010). Moving the crowd at threadless, in: *Information, Communication & Society*, 13:8, pp. 1122-1145.
- Brabham, D.C., (2009). Crowdsourcing the public participation process for planning projects, in: *Planning Theory* August 2009, vol. 8, no. 3, pp. 242-262.
- Brabham, D.C., Sanchez, T.W., and Bartholomew, K., (2009). Crowdsourcing public transportation in transit planning: preliminary results from the next stop design case. *Transportation Research Board, Annual meeting 2010*.
- Budde, R., Kautz, K., Kuhlenskamp, K., and Züllighoven, H., (1992). What is prototyping? In: *Information Technology & People*, Vol. 6 Iss: 2/3, pp.89 – 95.
- CBS, (2012). ICT, Kennis en Economie 2012. In *CBS.nl*. Retrieved July, 2012, from <http://www.cbs.nl/NR/rdonlyres/130F8419-05C1-43AE-B5ED-4C373F34EC82/0/2012i78pub.pdf>
- Chui, M., Miller, A., and Robert, R.P., (2009). How companies are benefiting from Web 2.0. In: *McKinsey Quarterly*, February 2009. Retrieved, July, 2012, from: [http://www.mckinseyquarterly.com/How\\_companies\\_are\\_benefiting\\_from\\_Web\\_2\\_0\\_McKinsey\\_Global\\_Survey\\_Results\\_2432](http://www.mckinseyquarterly.com/How_companies_are_benefiting_from_Web_2_0_McKinsey_Global_Survey_Results_2432)
- Chung, L., Prado Leite, J. do, Borgida, A., Chaudry, V., Giorgini, P., and Yu, E., (2009). On non-functional requirements in software engineering, in: *Conceptual Modelling: Foundations and Applications*. Springer Berlin, Heidelberg, pp. 363-379
- Clegg, D., and Barker, R., (1994). *Case Method Fast-Track: A Rad Approach*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1994
- Cook, S., (2008). The contribution revolution; letting volunteers build your business, in: *Harvard Business Review*, October 2008, pp. 60-69.
- CoreLabs., (2007). Living labs roadmap 2007-2010: Recommendations on networked systems for open user-drive research, development and innovation, in: *Open Document*. Lulea University of Technology, Centrum for Distance spanning technology, Lulea. pp. 1-61.

- Divol, R., Edelman, D., and Sarrazin, H., (2012). Demystifying social media, in: McKinsey Quarterly, April 2012.
- Doan, A., Ramakrishnan, R., and Halevy, A.Y., (2011). Crowd sourcing Systems on the World-Wide Web. *Communications of the ACM*, April 2011, Vol. 54, No.4. doi:10.1145/1924421.1924442
- Folstad, A., (2008). Towards a living lab for the development of online community services, in: *The Electronic Journal for Virtual Organizations and Networks (eJOV)*, Volume 10, August, 2008.
- Fowler, M., and Scott, K., (2000). *UML Distilled (2nd Ed.): A Brief Guide to the Standard Object Modelling Language*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Howe, J. (2006a). "The Rise of Crowd sourcing". *Wired Magazine*, issue 14.06. Retrieved from: [http://www.wired.com/wired/archive/14.06/crowds\\_pr.html](http://www.wired.com/wired/archive/14.06/crowds_pr.html)
- Howe, J. (2006b). "Crowd sourcing: A definition". *Crowd sourcing*. Retrieved from: [http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing\\_a.html](http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html)
- Kaner, C., (2006). Exploratory testing. Florida Institute of Technology, Quality Assurance Institute Worldwide Annual Software Testing Conference, Orlando, FL, November 2006
- Kittur, A., Chi, E.H., Suh, B., (2008). Crowd sourcing user studies with Mechanical Turk, in: *Proceedings of the 26th annual SIGCHI conference on Human factors in computing systems*. Florence, Italy, April 5-10, 2008.
- Koch, A., (2009). De ICR-rijtuigen, veranderingen in het park sinds 2001. *Op de rails*, issue 9-2009, ISSN 0030-3321.
- Lakhani, K., Panetta, J., (2007). The principles of distributed innovation, in: *innovation*, summer 2007, pp. 97-112.
- Lampel, J., Bhalla, A. 2007. The role of status seeking in online communities: Giving the gift of experience. *Journal of Computer-Mediated Communication*. 122, <http://jcmc.indiana.edu/vol12/issue2/lampel.html>.
- Logica, (2012a). Logica innovatie evenement: how to create value in a connected world (n.d.). In *Logica.nl*. Retrieved January 2012, from <http://www.logica.nl/we-are-logica/media-centre/events/2011/transform11/>
- Malone, T. W., Laubacher, R., & Dellarocas, C. (2009). *Harnessing Crowds: Mapping the Gnome of Collective Intelligence*. MIT Sloan Research Paper No. 4732-09.
- Markopoulos, P., Rauterberg, G.W.M., (2000). LivingLab: a white paper, in: *IPO Annual Progress Report*, 35, 2000.
- McAfee, A.P., (2007). Enterprise 2.0: The dawn of an emergent collaboration. In: *MIT Sloan Management Review*, spring 2006, vol. 47, no. 3.
- Matuszak, G., (2007). Enterprise 2.0: fad or future? Retrieved July 09, 2012, from [http://195.145.215.8/media/20070501\\_ICE-Enterprise20\\_fertig.pdf](http://195.145.215.8/media/20070501_ICE-Enterprise20_fertig.pdf)
- Mirjamdotter, A., Ståhlbröst, A., Sällström, A., Niitamo, V.-P., and Kulkki, S., (2006). The European Network of Living Labs for CWE - user-centric co-creation and innovation. *E-Challenges 2006*. Barcelona, Spain, October 25-27, 2006.
- Nielsen, J., (2003). Usability 101: Introduction to Usability (n.d.). In *Useit.com*. Retrieved June 4, 2012 from: <http://www.useit.com/alertbox/20030825.html>
- NS, (2010a). Tevredenheid treinreizigers gestegen. In *NS.nl*. Retrieved March 05, 2012, from <http://www.ns.nl/over-nieuwscentrum/persberichten/2010/10/tevredeheid-treinreizigers-gestegen-q3-2010.html>
- NS, (2012a). Onze reizigers. In *NS.nl*. Retrieved June 07, 2012, from <http://www.ns.nl/jaarverslag2011/ns-in-2011#onze-reizigers>
- NS, (2012b). NS in een oogopslag. In *NS.nl*. Retrieved January, 2012, from <http://www.ns.nl/over-ns/wat-doen-wij/ontdek-ns/ns-in-een-oogopslag.html>
- O'Mahony, S., West, J., (2007). The emergence of governance in an open source community. *Academy of Management Journal* 50, no. 5 (October 2007): 1079-1106

- PCextreme, (2012). Webhosting pakketten. In *PCextreme.nl*. Retrieved December, 2011, from <https://www.pcextreme.nl/webhosting/hosting-pakketten/>
- Ramaekers, P., Wit, T. De, Pouwels, M., (2009). Hoe druk is het nu werkelijk op het Nederlandse spoor? In *InfraSite.nl*. Retrieved January 2012, from [http://www.infrasite.nl/articles/articles.php?ID\\_articles=197](http://www.infrasite.nl/articles/articles.php?ID_articles=197)
- Sloos, F., (2012). Maandrapportage OBIS, NS Reizigers IT Operations, February 2012.
- Steinfeld, A., Zimmerman, J., Tomasic, A., Yoo, D., & Aziz, R. (2011). Mobile transit rider information via universal design and crowdsourcing. Proceedings from Transportation Research Board 2011 Annual Meeting.
- Strien, P.J. van (1997). Towards a methodology of psychological practice: The regulative cycle. *Theory & Psychology*, 7(5):683-700, 1997.
- T-Mobile, (2012). Onderweg met de trein, altijd gratis online. In *T-Mobile.nl*. Retrieved February, 2012, from <http://www.t-mobile.nl/promo/ns/zakelijk-index.html>
- Tweakers, (2012). Treinreizigers verbruiken bijna 10TB via NS-hotspots. In *Tweakers.net*. Retrieved July, 2012, from <http://tweakers.net/nieuws/82828/treinreizigers-verbruiken-bijna-10tb-via-ns-hotspots.html>
- Vijaykumar, N., (2010). Crowd Sourcing: Leveraging Social Networks to Drive Innovation. SETLabs Briefings, Vol. 8, No. 4, pp. 9-18.
- Wees, B.J. van der, Moonen, J.M., (2011). The Potential of In-Train Crowdsourcing. 24th BLED eConference on eFuture, 12-15 June 2011.
- Wieringa, R.J., (2009). Design science as nested problem solving. In Proc. of the 4th Int. Conf. on Design Science Research in Information Systems and Technology, pages 1-12. ACM, 2009.
- Wieringa, R.J., and Heerkens, J.M.G, (2006). The methodological soundness of requirements engineering papers: A conceptual framework and two case studies. *Requirements Engineering Journal*, 11(4):295-307, 2006.
- Wieringa, R.J., Maiden, N., Mead, N., and Rolland, C., (2006). Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requirements Engineering*, 11(1):102-107, March 2006.
- Wu, C. Gerlach, J., Young, C., (2007). An empirical analysis of open source software developers' motivations and continuance intentions. *Information and Management*. 44, 253 – 262.
- Xampp, (2012). Apache Friends – Xampp. In *ApacheFriends.org*. Retrieved December, 2011, from <http://www.apachefriends.org/en/xampp.html>
- YouTube, (2012a). Statistics (n.d.). In *YouTube.com*. Retrieved February 16, 2012, from [http://www.youtube.com/static?hl=en&template=press\\_statistics](http://www.youtube.com/static?hl=en&template=press_statistics)
- Zhong, X., Chan, H., Rogers, T. J., Rosenberg, C. P., Coyle, E. (2006) The development and eStadium testbeds for research and development of wireless services for large-scale sports venues. The 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, TRIDENTCOM 2006, IEEE.

## APPENDIX A: COLLECTIVE INTELLIGENCE GENES

Table A.1 shows an overview of the organizational genes model according to Malone et al. (2009) and when these genes are best applicable.

<b>Question</b>	<b>Gene</b>	<b>When useful</b>
<b>Who</b>	<b>Crowd</b>	<ul style="list-style-type: none"> <li>- Resources useful in doing activities distributed widely or in places not known in advance</li> <li>- Activities can be divided into pieces satisfactorily (necessary information can be shared, gaming and sabotage can be managed)</li> </ul>
	<b>Hierarchy</b>	<ul style="list-style-type: none"> <li>- Conditions for crowd gene are not met</li> </ul>
<b>Why</b>	<b>Money Love Glory</b>	<ul style="list-style-type: none"> <li>- Many factors, too complex to list here, are relevant, with two rules of thumb:               <ul style="list-style-type: none"> <li>- Appealing to Love and Glory, rather than Money, can often reduce costs</li> <li>- Providing Glory and Money can often influence group's direction and speed</li> </ul> </li> </ul>
	<b>Collection</b>	Conditions for crowds gene, plus... <ul style="list-style-type: none"> <li>- Activity can be divided into small pieces that can be done (mostly) independent of each other</li> </ul>
<b>How – Create</b>	<b>Contest</b>	Conditions for collection gene, plus... <ul style="list-style-type: none"> <li>- Only one (or a few) good solutions are needed</li> </ul>
	<b>Collaboration</b>	<ul style="list-style-type: none"> <li>- Activity cannot be divided into small independent pieces (otherwise collection)</li> <li>- There are satisfactory ways of managing the dependencies among the pieces</li> </ul>
	<b>Group decision</b>	Conditions for crowd, plus... <ul style="list-style-type: none"> <li>- Everyone in the group needs to abide by the same decisions, plus...</li> </ul>
<b>How - Decide</b>	<b>Voting</b>	<ul style="list-style-type: none"> <li>- It is important for the crowd to be committed to the decision</li> </ul>
	<b>Averaging</b>	Conditions for voting, plus... <ul style="list-style-type: none"> <li>- Decisions consist of estimating a number</li> <li>- Crowd has no systematic bias about estimating the number</li> </ul>
	<b>Consensus</b>	Conditions for voting, plus... <ul style="list-style-type: none"> <li>- Achieving consensus in reasonable time is feasible</li> </ul>
	<b>Prediction market</b>	<ul style="list-style-type: none"> <li>- Decision consists of estimating a number</li> <li>- Crowd has some information about estimating the number</li> <li>- Some people may have (or obtain) much better information than others</li> <li>- Continuously updated estimates are useful</li> </ul>
	<b>Individual decisions</b>	Conditions for crowd gene, plus... <ul style="list-style-type: none"> <li>- Different people can make their own decision, plus...</li> </ul>
	<b>Market</b>	<ul style="list-style-type: none"> <li>- Money is needed to motivate people to provide the necessary effort or other resources</li> </ul>
	<b>Social network</b>	<ul style="list-style-type: none"> <li>- Non-monetary motivations are sufficient for people to contribute</li> <li>- Individuals find information about other's opinions useful in making their own choice</li> </ul>

**Table A.1: Overview of the collective intelligence genes model**



## APPENDIX B: FUNCTIONAL DESIGN

*In this appendix we elaborate more on the functional design for the reporting system for train compartments. This chapter extends the work that is started in chapter three of this thesis. The appendix is constructed in a way that this appendix on itself could serve as functional design. This means that some information is already stated in the summary of this functional design in chapter three.*

### B.1 Requirements elicitation process

The requirements elicitation process is meant to state the functional and non-functional requirements. (Sommerville et al., 1998; Chung et al., 2009). For this project we used the brainstorming method for eliciting requirements. During the brainstorming process different iterations have been performed, under supervision of the product owner and two experts from Logica. We started with a set of initial requirements from a brainstorm session and discussed these requirements with an information analyst from Logica. This initial set of requirements was then refined and extended, by means of brainstorming sessions with a second Logica expert. The second expert, a requirements engineer with fifteen years of experience, was used nine sessions of an hour each. During the process, we validated the requirements two times with the product owner. The final set of non-functional requirements for both systems is presented in table 3.4. The functional requirements for both systems are presented in tables 3.5 and 3.6

### B.2 Use case modelling

The second step in the functional design is defining the actions between actors and the system. Therefore, the requirements are transformed into use case specifications. Since the requirements for both systems are alike, the use case models apply to both systems. The main part of this section consists of Unified Modelling Language (UML) Use Cases. A use case is a list of steps, typically defining interactions between an actor and the system. The steps are typically taken to pursue a goal. Actors in this case can be humans, but also other systems. Use cases are expressed in the UML. UML is a standardized general-purpose modelling language, which is mostly used in the field of object-oriented software engineering (Fowler et al., 2000). In order to come to the use cases, first an overview of the use cases and their coherence is presented. The overview is called a use case diagram. After the use case diagram, a use case list is given, which is a table that contains the use cases in more details. The use case list shows for example which requirements form the basis for a certain use case. Generally, the use case diagram is used to create an overview of the use cases, while the use case list is used to provide more insight. The use case diagram is presented in Figure 3.2, the use case list is shown in table 3.7.

### B.3 Use case matrix

Since the use cases form the basis for the implementation of the application, it is necessary to check if every functional requirement is used within at least one of the use cases. This is done by making a use case matrix. In the top row the numbers of the use cases are stated. In the first column, the functional requirements are stated. For each intersection a check is performed if the use case under attention entails the requirement. If so, the intersection is marked yellow, if not the cell remains empty. If all requirements are checked at least once and all use cases are also checked at least once, each requirement is entailed in a use case. The use case matrix is presented in table B.1.



	UC1	UC2	UC3	UC4	UC5	UC6	UC7
FRA01	X						
FRA02		X					
FRA03		X					
FRA04		X					
FRA05			X				
FRA06				X			
FRA07					X		
FRA08							X
FRA09						X	

Table B.1: Use case matrix

## B.4 Prioritization

Since there is a time constraint on this graduation project, not all requirements can be implemented during this project. Therefore, prioritization needs to be performed. A widely known and accepted method for prioritization in business analysis and software development is the so called MoSCoW-method. This method was first developed by Dai Clegg of Oracle UK Consulting (Clegg and Barker, 1994), but afterwards the intellectual property rights were donated to the Dynamic Systems Development Method (DSDM) consortium. The MoSCoW-method is often used in combination with time boxing, in projects with a fixed deadline. In this way, the focus can stay on the most important requirements. This makes the model a core aspect of the more rapid software development processes like Agile and DSDM.

According to this prioritization model, four different categories can be stated (Clegg and Barker, 1994);

**M** – Must: Describes a requirement that must be satisfied in the next release to make the release a success. This means that if a Must-requirement is not implemented, the project delivery should be considered a failure. Must-requirements can be downgraded, if all stakeholders agree upon it, for example when new requirements come into play that can be considered more important.

**S** – Should: This category describes requirements that should be included in the next release if possible. Often these requirements are crucial, but they can be satisfied in other ways or next releases.

**C** – Could: Requirements labelled as could, are less critical requirement and often seen as features that are nice to have.

**W** – Would: Would requirements are considered least critical or not appropriate in this release or at this time. This results in the fact that Would-requirements are often not implemented in the current release.

The following two tables contain the non-functional and functional requirements for the reporting system for train compartments and their prioritization using the MoSCoW-method. The categorization in tables B.2 and B.3 is created as part of the brainstorming session during the functional design process.

	Must	Should	Could	Would	Note
NFA01				X	(1)
NFA02	X				
NFA03	X				
NFA04	X				
NFA05	X				
NFA06		X			(2)
NFA07				X	(3)
NFA08		X			
NFA09			X		(4)
NFA10		X			(5)
NFA11				X	
NFA12		X			(6)

**Table B.2: Prioritization matrix for the non-functional requirements using the MoSCoW-method**

Whenever a number in parentheses is presented in the table above, the corresponding number below explains more details on the respective prioritization.

- (1) The availability of the system has a strong relation with the availability of the OBIS in the train. Since this is only a pilot of a crowd sourcing system, this non-functional requirement is not considered in the implementation of the prototype.
- (2) The feature of failure management is nice to have, but in this pilot situation not a necessary feature. It will therefore not be implemented in the prototype.
- (3) The feature to access the system from the 'shore-side' would be a nice extension to the prototype, but considering the time constraints and the fact that it will be a pilot, this feature is not implemented.
- (4) The possibility to extend the system in the future is also not considered in this prototype.
- (5) This requirement is very important for a real application, but for the prototype it would be enough if the information is stored in a safe database.

- (6) For this requirement, the same thing as with note 5 goes. For a simple prototype that will be developed during this project, it is enough that the data is not accessible, meaning that SQL injections and script cannot be ran from within the application.

	Must	Should	Could	Would	Note
FRA01	X				
FRA02			X		
FRA03		X			
FRA04		X			
FRA05			X		
FRA06		X			
FRA07		X			
FRA08				X	(1)
FRA09		X			

**Table B.3: Prioritization matrix for the functional requirements using the MoSCoW-model**

Whenever a number in parentheses is presented in the table above, the corresponding number in the section below explains some details on the respective prioritization.

- (1) Since this functional requirement has a relation with FRA05, this feature can only be implemented if the feature in FRA05 is also implemented. For this prototype it is not necessary to implement the feature.

Since this graduation project is on a time constraint, not all requirements can be implemented in the prototype. Therefore, only the requirements that are prioritized as Must will be implemented.

## B.5 Functional use case specifications

In the following sub sections, B.5.1 through B.5.7, each of the seven use cases for the reporting system for train compartments is specified. This is done by stating the number and name of the use case, along with the actors, preconditions and post conditions and finally the happy flow and the alternative flow. The happy flow is used to describe the flow through the system if everything goes according to plan. If anything does not go according to plan, an alternative flow should describe how the system is going to behave in this situation.

### B.5.1 Functional use case A1: Submit contribution

<b>Use case:</b>	UA1 (functional)	<b>Version:</b>	v0.4 (04062012)
<b>Title</b>	Submit contribution		
<b>Description</b>	The user can fill in his credentials (name and email address) and provide feedback on the		

---

general status of the train. The system stores this feedback in a database.

---

**Corresponding requirement** FRA01: As a user, I want to contribute on the general status of the train to improve my experience in the train.

---

**Actors** User

---

**Precondition** The user has opened the portal website of the train.

---

**Post condition** The contribution that the user made will be stored in the system for processing. The user receives a feedback message on the portal site.

---

---

**Happy flow:**

1. User fills in his name and email address and a category of feedback {complaint, compliment, other}
2. User specifies his feedback
3. User submits his feedback
4. System stores the feedback in a database

**Alternative flow: User does not fill in his name**

1. User does not fill in his name
2. System gives an error message on the screen, telling the user to fill in his name
3. Continue at step one of the happy flow

**Alternative flow: User does not fill in the email address or a valid email address**

1. User does fill in his name, but does not fill in his email address
2. System gives an error message, telling the user to fill in a valid email address
3. Continue at step one of the happy flow

**Alternative flow: User does not fill in the category of feedback**

1. User fills in his name and email address, but no category of feedback
2. System gives an error message, telling the user to fill in a category of feedback
3. Continue at step one of the happy flow

**Alternative flow: User does not specify his feedback**

1. User fills in his name, email address and a category of feedback {complaint, compliment, other}
2. User does not specify his feedback
3. System given an error message, telling the user to specify his feedback
4. Continue at step two of the happy flow

**Alternative flow: The store action failed**

1. User fills in his name, email address and a category of feedback {complaint, compliment, other}
2. User specifies his feedback
3. User submits his feedback
4. System cannot store the feedback in the database
5. System gives a feedback message, telling the user that his feedback could not be submitted
6. Return to step three of the happy flow

---

**Table B.4: RSTC functional use case 1**

B.5.2 Functional use case A2: Retrieve overview of the received contributions

<b>Use case:</b>	UA2 (functional)	<b>Version:</b>	v0.3 (04062012)
<b>Title</b>	Retrieve overview of the received contributions		
<b>Description</b>	The user, conductor or worker calls the function to create and show an overview of the contributions that have been submitted to the system.		
<b>Corresponding requirements</b>	FRA02, FRA03, and FRA04: As a user/NS Worker/Conductor, I want an overview of the contributions that have been made.		
<b>Actors</b>	User, conductor and worker		
<b>Precondition</b>	Actors do not all have the same permissions. Users for example can only see an overview of the contributions in their current train, whilst workers can see an overview of all contributions.		
<b>Post condition</b>	The user, conductor or worker is presented an overview of the contributions that fit with his selection options.		

**Happy flow:**

1. User/worker/conductor calls the function to create an overview
2. User/worker/conductor makes a choice out of the selection options
3. User/worker/conductor submits his preferences
4. System shows the data that is asked for

**Alternative flow: User/worker/conductor does not make a choice out of the selection options**

1. User/worker/conductor calls the function to create an overview
2. User/worker/conductor does not make a choice out of the selection options
3. System gives a feedback message, telling the user/worker/conductor to select an option
4. Continue at step two of the happy flow

**Table B.5: RSTC functional use case 2**

B.5.3 Functional use case A3: Close contribution

<b>Use case:</b>	UA3 (functional)	<b>Version:</b>	v0.3 (04062012)
<b>Title</b>	Close contribution		
<b>Description</b>	The conductor in the train selected a contribution in the overview and will now close the status of a contribution.		

<b>Corresponding requirement</b>	FRA05: As a conductor, I want the possibility to change the status of contribution to show that certain problems have been solved.
<b>Actors</b>	Conductor
<b>Precondition</b>	The conductor has checked or solved the problem that is stated in the contribution.
<b>Post condition</b>	The status of the contribution is now closed instead of open.
<b>Happy flow:</b>	
<ol style="list-style-type: none"> <li>1. Conductor creates an overview of the contributions (UA2)</li> <li>2. Conductor selects the specific row with the contribution</li> <li>3. Conductor checks the 'closed' field in the row</li> <li>4. System changes the status to closed</li> </ol>	

**Table B.6: RSTC functional use case 3**

B.5.4 Functional use case A4: Alert conductor on a new contribution

<b>Use case:</b>	UA4 (functional)	<b>Version:</b>	v0.3 (04062012)
<b>Title</b>	Alert conductor on a new contribution		
<b>Description</b>	Once a new contribution has been made, which violates certain pre-defined rules, the system alerts the conductor on the new contribution.		
<b>Corresponding requirement</b>	FRA06: As a system, I want to alert a conductor on certain contributions so that he can be in the right place quickly.		
<b>Actors</b>	Conductor		
<b>Precondition</b>	A new contribution has just been stored in the database by the system		
<b>Post condition</b>	The conductor receives an alert on his mobile device.		
<b>Happy flow:</b>			
<ol style="list-style-type: none"> <li>1. System detects that the new entry is urgent</li> <li>2. System sends an alert to the conductor</li> </ol>			

**Table B.7: RSTC functional use case 4**

B.5.5 Functional use case A5: Filter malicious contributions

<b>Use case:</b>	UA5 (functional)	<b>Version:</b>	v0.3 (04062012)
<b>Title</b>	Filter malicious contributions		
<b>Description</b>	The contributions that have been submitted are filtered based on their content, to see if multiple contributions are on the same problem or if users are playing with the system. Contributions that tend to have the same subject are aggregated.		
<b>Corresponding requirement</b>	FRA07: As a system, I want to detect and filter malicious contributions because they do not contribute.		
<b>Actors</b>	System		
<b>Precondition</b>	A new contribution has just been stored in the database by the system		
<b>Post condition</b>	The contribution in the train filtered before they are sent to the back office.		
<b>Happy flow:</b>	<ol style="list-style-type: none"> <li>1. System judges the contribution on the category of feedback {complaint, compliment, other}</li> <li>2. System will search for other contributions in the same category</li> <li>3. System analyzes and checks the credentials and content of the contribution with other contributions</li> <li>4. System alerts a worker if a certain threshold is exceeded</li> </ol>		

**Table B.8: RSTC functional use case 5**

B.5.6 Functional use case A6: Capture location of the user

<b>Use case:</b>	UA6 (functional)	<b>Version:</b>	v0.3 (04062012)
<b>Title</b>	Capture location of the user		
<b>Description</b>	The system captures the location of the user within the train.		
<b>Corresponding requirement</b>	FRA09: As a system, I want to detect from which train compartment a contribution has been made		
<b>Actors</b>	System, user		
<b>Precondition</b>			
<b>Post condition</b>	The contribution of the user contains data on his location within the train.		



---

**Happy flow:**

1. System is triggered by new contribution
2. System discovers compartment number from which the feedback is submitted
3. System adds compartment number to the feedback

---

**Table B.9: RSTC functional use case 6**

B.5.7 Functional use case A7: Capture ID of the conductor

---

<b>Use case:</b>	UA7 (functional)	<b>Version:</b>	v0.3 (04062012)
<b>Title</b>	Capture ID of the conductor		
<b>Description</b>	The system captures the ID of the conductor.		
<b>Corresponding requirement</b>	FRA08: As a system, I want to register which conductor changed the status of certain contributions to ask for further details if necessary.		
<b>Actors</b>	System, conductor		
<b>Precondition</b>			
<b>Post condition</b>	The contribution of the user contains data on his location within the train.		

**Happy flow:**

1. System is triggered by new contribution
2. System discovers the ID of the conductor the changed the status
3. System adds the new data to the existing contribution

---

**Table B.10: RSTC functional use case 7**

## B.6 Use case specification for the feedback system

Since the requirements of the feedback system are almost similar to the requirements of the reporting system, the use case list, use case diagram and prioritization are also the same. Therefore, the use case list, use case diagram and prioritization will not be stated again. Since only the first use case is fundamentally different from the use cases for the reporting system for train compartments, this is the only use case that will be stated.

B.6.1 Functional use case B1: Submit contribution

---

<b>Use case:</b>	UB1 (functional)	<b>Version:</b>	v0.1 (04062012)
<b>Title</b>	Submit contribution		
<b>Description</b>	The user can fill in his credentials (name and email address) and provide feedback on the		

---

passenger information. The system stores this feedback in a database.

---

**Corresponding requirement** FRB01: As a user, I want to contribute on the passenger information in the train to improve my experience in the train.

---

**Actors** User

---

**Precondition** The user has opened the portal website of the train.

---

**Post condition** The contribution that the user made will be stored in the system for processing. The user receives a feedback message on the portal site.

---

**Happy flow:**

1. User fills in his name and email address and a category of feedback {complaint, compliment, other}
  2. User specifies his feedback
  3. User submits his feedback
  4. System stores the feedback in a database
- 

**Alternative flow: User does not fill in his name**

1. User does not fill in his name
  2. System gives an error message on the screen, telling the user to fill in his name
  3. Continue at step one of the happy flow
- 

**Alternative flow: User does not fill in the email address or a valid email address**

1. User does fill in his name, but does not fill in his email address
  2. System gives an error message, telling the user to fill in a valid email address
  3. Continue at step one of the happy flow
- 

**Alternative flow: User does not fill in the category of feedback**

4. User fills in his name and email address, but no category of feedback
  5. System gives an error message, telling the user to fill in a category of feedback  
The error message is represented in a special CSS class; the message itself is set in a variable.
  6. Continue at step one of the happy flow
- 

**Alternative flow: User does not specify his feedback**

1. User fills in his name, email address and a category of feedback {complaint, compliment, other}
  2. User does not specify his feedback
  3. System given an error message, telling the user to specify his feedback
  4. Continue at step 2 of the happy flow
-

---

**Alternative flow: The store action failed**

1. User fills in his name, email address and a category of feedback {complaint, compliment, other}
2. User specifies his feedback
3. User submits his feedback
4. System cannot store the feedback in the database
5. System gives a feedback message, telling the user that his feedback could not be submitted
6. Return to step three of the happy flow

---

**Table B.11: FSPI functional use case 1**

# APPENDIX C: TECHNICAL DESIGN

## C.1 Technical use case A1: Submit contribution

---

<b>Use case:</b>	UA1 (technical)	<b>Version:</b>	v0.2 (04062012)
<b>Title</b>	Submit contribution		

---

### Happy flow:

1. User fills in his name and email address and a category of feedback {complaint, compliment, other}  
The first page of this application gives the user two HTML text fields (fieldnames: name and email) to enter his name and email address. Additionally, there are three radio buttons (fieldname: feedback\_type) with the values complaint, compliment and other. At the bottom of the page there is a button to continue to the next step, which stores the data in a session (PHP).
2. User specifies his feedback  
If the category of feedback is either compliment or other, the user gets an HTML text area (fieldname: explanation) to specify the feedback. If the category of feedback is complaint, the system presents four radio buttons (fieldname: complaint\_type). These buttons represent the specific topic of complaint, being the outside of the train, the interior in the train, the toilets in the train and other passengers. Each specific topic contains specific options, which is depicted in table 5.2 below this specification.
3. User submits his feedback  
If the specification is completed, the user can submit his feedback by clicking the HTML submit button.
4. System stores the feedback in a database  
By clicking this button, the system will store the session of the user in a MySQL database. The database scheme that is used is presented in table 5.3 below this specification.

---

### Alternative flow: User does not fill in the name

1. User does not fill in his name  
The code that is necessary for checking if a name is entered, will be provided with PHP
2. System gives an error message on the screen, telling the user to fill in his name  
The error message is represented in a special CSS class, the message itself is set in a variable
3. Continue at step one of the happy flow

---

### Alternative flow: User does not fill in the email address or a valid email address

1. User does fill in his name, but does not fill in his email address  
The code that is necessary for checking if an email address is entered or is valid, is provided by PHP. Valid in this context means that it exist out of the format name@domein.extension
  2. System gives an error message, telling the user to fill in a valid email address
-

---

The error message is represented in a special CSS class, the message itself is set in a variable

3. Continue at step one of the happy flow
- 

**Alternative flow: User does not fill in the category of feedback**

5. User fills in his name and email address, but no category of feedback
  6. System gives an error message, telling the user to fill in a category of feedback  
The error message is represented in a special CSS class; the message itself is set in a variable.
  7. Continue at step one of the happy flow
- 

**Alternative flow: User does not specify his feedback**

1. User fills in his name, email address and a category of feedback {complaint, compliment, other}
  2. User does not specify his feedback
  3. System given an error message, telling the user to specify his feedback  
The error message is represented in a special CSS class, the message itself is set in a variable
  4. Continue at step two of the happy flow
- 

**Alternative flow: The store action failed**

1. User fills in his name, email address and a category of feedback {complaint, compliment, other}
  2. User specifies his feedback
  3. User submits his feedback
  4. System cannot store the feedback in the database
  5. System gives a feedback message, telling the user that his feedback could not be submitted  
The error message is represented in a special CSS class, the message itself is set in a variable
  6. Return to step three of the happy flow
- 

**Table C.1: RSTC technical use case 1**

The following table represents the specification of complaints in the application. If the user chooses complaint as his type of feedback, then the user is directed to step two, where he can choose a topic for his complaint. The complete specification is given in the table below, together with the type of fields that are used.

---

<b>Step 1</b>	<b>Step 2</b>	<b>Step 3</b>
Complaint	Outside of the train (radio button)	Windows too dirty (radio button)
		Graffiti on the train (radio button)
		Something broken on the outside of the train (radio button)
		Other (radio button with text field)

---

Interior of the train (radio button)	Broken seats (radio button)
	Dirty seats (radio button)
	Trashcans full (radio button)
	Floor not clean (radio button)
	Other (radio button with text field)
Toilets (radio button)	Toilet needs cleaning (radio button)
	Lock is broken (radio button)
	Other (radio button with text field)
Other passengers (radio button)	Passengers too loud (radio button)
	Passengers noisy in silence coupe (radio button)
	Aggressive passengers in compartment (radio button)
	Smoking passenger in compartment (radio button)
	Other (radio button with text field)

**Table C.2: Structure of the feedback specification for the reporting system**

The table below presents an overview of the database scheme that is needed for the system to work. The data that is entered in the front-end of the system can then be saved in the database. The elements in the table are based on a MySQL database, if other databases would be used (for example Oracle) then some data types need to be translated.

Column	Type	Nullable	Standard value	Extra
Id	Int(8)	No	None	Auto_increment
Timestamp	Timestamp	No	Current_timestamp	
Name	Varchar(60)	No	None	
Email	Varchar(60)	No	None	
Feedback_type	Varchar(60)	No	None	
Complaint_type	Varchar(120)	Yes	Null	
Complaint	Varchar(120)	Yes	Null	
Explanation	Varchar(120)	Yes	Null	

**Table C.3: Database scheme for the reporting system**

## C.2 Technical use case B1: Submit contribution

---

<b>Use case:</b>	UB1 (technical)	<b>Version:</b>	v0.2 (04062012)
------------------	-----------------	-----------------	-----------------

---

**Title** Submit contribution

---

### Happy flow:

1. User fills in his name and email address and a category of feedback {complaint, compliment, other}  
The first page of this application gives the user two HTML text fields (fieldnames: name and email) to enter his name and email address. Next to that, there are three radio buttons (fieldname: feedback\_type) with the values complaint, compliment and other. At the bottom of the page there is a button to continue to the next step, which stores the data in a session (PHP).
  2. User specifies his feedback  
If the category of feedback is either compliment or other, the user gets an HTML text area (fieldname: explanation) to specify the feedback. If the category of feedback is complaint, the system presents four radio buttons (fieldname: complaint\_type). These buttons represent the specific topic of complaint, being outside information display (BBA), display in the train, announcement system or the portal website. Each specific topic contains specific options, which is depicted in table 5.5 below this specification.
  3. User submits his feedback  
If the specification is completed, the user can submit his feedback by clicking the HTML submit button.
  4. System stores the feedback in a database  
By clicking this button, the system will store the session of the user in a MySQL database. The database scheme that is used is presented in table C.5 below this specification.
- 

### Alternative flow: User does not fill in the name

1. User does not fill in his name  
The code that is necessary for checking if a name is entered, will be provided with PHP
  2. System gives an error message on the screen, telling the user to fill in his name  
The error message is represented in a special CSS class, the message itself is set in a variable
  3. Continue at step one of the happy flow
- 

### Alternative flow: User does not fill in the email address or a valid email address

1. User does fill in his name, but does not fill in his email address  
The code that is necessary for checking if an email address is entered or is valid, is provided with PHP. Valid in this context means that it exist out of the format name@domein.extension
  2. System gives an error message, telling the user to fill in a valid email address  
The error message is represented in a special CSS class, the message itself is set in a variable
-



- 
3. Continue at step one of the happy flow
- 

**Alternative flow: User does not fill in the category of feedback**

8. User fills in his name and email address, but no category of feedback
  9. System gives an error message, telling the user to fill in a category of feedback  
The error message is represented in a special CSS class; the message itself is set in a variable.
  10. Continue at step one of the happy flow
- 

**Alternative flow: User does not specify his feedback**

1. User fills in his name, email address and a category of feedback {complaint, compliment, other}
  2. User does not specify his feedback
  3. System given an error message, telling the user to specify his feedback  
The error message is represented in a special CSS class; the message itself is set in a variable.
  4. Continue at step two of the happy flow
- 

**Alternative flow: The store action failed**

1. User fills in his name, email address and a category of feedback {complaint, compliment, other}
  2. User specifies his feedback
  3. User submits his feedback
  4. System cannot store the feedback in the database
  5. System gives a feedback message, telling the user that his feedback could not be submitted  
The error message is represented in a special CSS class, the message itself is set in a variable
  6. Return to step three of the happy flow
- 

**Table C.4: FSPI technical use case 1**

The following table represents the specification of complaints in the application. If the user chooses complaint as his type of feedback, then the user is directed to step two, where he can chose a topic for his complaint. The complete specification is given in the table below, together with the type of fields that are used.

---

Step 1	Step 2	Step 3
Complaint	Outside information display (BBA)	Wrong end destination (radio button + text field)
		No end destination (radio button)
		Other (radio button + text field)
	Announcement system	Volume too high (radio button)
		Volume too low (radio button)

---

	No announcements (radio button)
	Wrong information (radio button)
	Unfriendly announcer (radio button)
	Bad quality of the announcement signal
	Other (text field)
Portal website	Wrong information (radio button + text field)
	No information (radio button)
	No internet (radio button)
	Other (radio button + text field)
Display in the compartment	No information (radio button)
	Wrong end destination (radio button)
	Wrong intermediate station (radio button)
	Times are wrong (text field)
	Information too late (radio button)
	Part of the trip no information (radio button + text field)
	State of the display (radio button + text field)
	Other (radio button + text field)

**Table C.5: Structure of the feedback specification for the feedback system**

The database scheme that is needed for this application is exactly the same as the database scheme for the reporting system for train compartments. The table below shows this database scheme.

Column	Type	Nullable	Standard value	Extra
Id	Int(8)	No	None	Auto_increment
Timestamp	Timestamp	No	Current_timestamp	
Name	Varchar(60)	No	None	
Email	Varchar(60)	No	None	
Feedback_type	Varchar(60)	No	None	
Complaint_type	Varchar(120)	Yes	Null	

Complaint	Varchar(120)	Yes	Null
Explanation	Varchar(120)	Yes	Null

**Table C.6: Database scheme for the feedback system**

### C.3 Implementation of the non-functional requirements

In the previous chapter twelve non-functional requirements for both solutions were stated and prioritized using the MoSCoW-method. Four out of the twelve non-functional requirements were prioritized as being Must-haves; these requirements are stated in the table below. In the following sub sections an explanation is given on how these four requirements can be preserved during the implementation of a prototype.

Number	Requirement	Description
NFA02	Usability	The system needs a simple (frame-like) and user-friendly user interface for the users
NFA03	Decomposition	There should be a clear decomposition between the front-end of the system and the back-end of the system
NFA04	Connectivity	The system needs an internet connection to communicate between user interface and back-end
NFA05	Platform independence	The front-end of the system should function the same on every user platform

**Table C.7: Overview of the Must-have non-functional requirements**

#### C.3.1 Implementation of usability

Since there is an already existing portal website on the train, the application to be developed needs to fit in the original portal website. This means that room on this page has to be preserved for the application and that the application should have a frame like design. This feature is implemented with use of html, CSS and PHP. The first two techniques make sure that the application has the right place on the page and the right look, PHP makes sure that the functionality of the application is working properly.

Next to that, the application needs a high usability. Although the term is quite hard to define, ISO defines it as follows: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" (Nielsen, 2003). Effective in this case can be looked at as the user being able to contribute feedback to the NS. Efficient in this case means that the contribution can be submitted in as little clicks as possible. The satisfaction is quite hard to measure out front, so this should be done if the application is ready and testable.

#### C.3.2 Implementation of decomposition

Decomposition in this requirement means that the front-end and the back-end of the application should be separated. This can be useful for several different reasons. The first reason is that with this separation the high

level architecture of the application can change. It could for example be possible to run the front-end on the train and the back-end in one of the offices. On the other hand is also possible that both parts are implemented on the train. Since the back-end can be implemented in a different way or different place, NS could also choose to implement it in the cloud. This makes the scalability (non-functional requirement 11) much better. The decomposition in this context is realized by building a front-end that communicates with a back-end. In the prototype that will be built, the back-end could be as simple as a database.

### **C.3.3 Implementation of connectivity**

Since the front-end and the back-end are two separate entities, there needs to be a means of communication between both. Since this a common problem, there are quite simple solutions for this. In this specific context, the requirement will be preserved by using an internet connection between both ends. In the prototype this can be as simple as writing the data via an internet connection to the database.

### **C.3.4 Implementation of platform independency**

Since the portal website of the train is already platform independent, the application to be developed will also be platform independent if it is implemented into the portal website. The only aspect to keep in mind is that no techniques or methods that are platform specific can be used during the implementation. The techniques and methods that are already stated in this technical design do not compromise this requirement.

## APPENDIX D: PROTOYPING THE FEEDBACK SYSTEM

### D.1 Deployment of the prototype

After the implementation of all the necessary features has been realized, the prototype should be deployed before it can be tested. The deployment of this prototype, the feedback system for passenger information, can be done in different ways. For the application to work, we can roughly divide it in three parts. The first part is the web server; the second part is the back end. Both are connected with an interface, which forms the third part. In the following sections each of these components will be described.

#### D.1.1 Web server

Since the application front-end is web based, a web server is a prerequisite for this kind of deployment. This prerequisite only applies for testing and validation purposes of this application. The final application would be implemented in the train and therefore be running on the platform in the train.

When choosing a web server, two different approaches can be followed. The first is the use of a local web server; the second is the use of an online web server. For testing and validation purpose, both solutions for the web server have been used in this project. The online web server that is used is a shared web-hosting package from PCExtreme (2012), which provides disk space and hosting for the application. During this project the tool Xampp (2012) has been used as a local web server. Xampp is an open source, and thus free to install and use, web server with some extras. These extras will be discussed in a later sub section.

#### D.1.2 Back-end

Since the front-end of the application collects information and data from the users, the back end should be able to save and process this data. In this simple prototype it would be sufficient to use a database to store the information. Another solution, which is too complicated in this context, is the use of Business Process Modelling as a back-end. In the final application, this could be of more value, since the front-end can then be connection to already existing business processes.

The information will therefore not be processed in this project, something that is already stated in the prioritization of the requirements. As with the web server, two different options have been chosen as a back-end solution. The first solution is a MySQL database, which is already included in the shared web-hosting package at PCExtreme. The other solution is an OracleXE 11g Database. This database has to be installed in a local environment and is chosen for its standard functionality and the in-house knowledge of Logica on this database package. There is also a solution that offers a combination of both solutions above, namely use MySQL in the local environment. This option can be realized quite easily, since MySQL is already part of the open source Xampp package that is installed. As stated before, the back-end will in this prototype not be used to process the data. However there could be some kind of processing with the OracleXE database. This database provides standard functionality that can set triggers on new additions and can take further action based on this trigger.

### D.1.3 Interface between front-end and back-end

Once the front-end and back-end solutions have been chosen, the only thing that is needed to test and validate the application is an interface between both. Since the back-end is a simple database, the interface is not difficult to realize. Many other solutions have done the same before, the only thing that needs to be done is connect to the database and write specific pieces of code for the different databases.

In the development of this specific application, two implementations have been developed. The first is to connect the online front-end with the MySQL database. This solution can also be used in the local environment if the connection details are changed. The second solution is to connect the local front-end with a local OracleXE database. The OracleXE database cannot be used in combination with the online front-end. In this application we implemented a connect script which can handle both the MySQL and the OracleXE database. Changing one variable in the connection script will lead the application to work with either one of them. Based on the selection of one of the methods, the code to insert new data in the database is also changed automatically. This means that only one variable needs to be changed and the connection details need to be provided in order to let the application work.

## D.2 Testing the application

Software testing is most of the times seen as an investigation conducted to provide stakeholders with information about the quality of the product or service under test (Kaner, 2006). Since it is a field of its own and the application is quite simple, we will only perform some small tests to see if the application really does what it promises, before we use the application in the train. According to the Software Engineering Body of Knowledge (SWEBOK), there are three levels on which software can be tested (Abran et al., 2001):

- Unit testing: these tests are normally performed to verify the functionality of a specific piece of code, usually at the function level. A function can have multiple tests, to catch corner cases other branches in the code.
- Integration testing: these tests are performed on the interfaces that have to integrate multiple different systems.
- System testing: this kind of testing is used to test the system as a whole, if it meets the requirements as specified out front.

Normally these kinds of testing are done by writing special test classes or special test applications. Since the application proposed in this thesis is quite easy to build, there is no need to write special test application. We have chosen to take another approach, but to stay within the levels as proposed by the Software Engineering Body of Knowledge. This means that we first tested all the functions in the application, a bit analogous to unit testing. This testing was done in a web browser, by selecting the different options. After each option, we checked if the contribution that was made during the test could be found in the database. If this was the fact, the integration of the front-end and back-end can be considered successful. This test looks a bit like the integration testing proposed in the SWEBOK. If all the functions perform well and all contributions can be found back in the database, the system as a whole is working. This last step can be seen as the system testing part as proposed in SWEBOK. The application as a whole was tested and considered working before the validation in the train was started.

LOGICA NEDERLAND BV  
PROF. W.H. KEESOMLAAN 14  
1183 DJ AMSTELVEEN  
TEL: +31 (0) 20 503 3000  
[www.logica.nl](http://www.logica.nl)