Usability of Rich Internet Applications

September 6, 2009

Master Thesis

Noureddine Ou-aisa

Faculty Electrical Engineering, Mathematics and Computer Science
Master programme Multimedia and knowledge Engineering
Section Mens-Machine Interaction [Human-computer Interaction]

Supervised by Charles van der Mast
Abstract

This thesis reports the outcome of a study that is performed for Info Support. Info Support is a software house that is interested in new technologies. Unit finance and insurance is interested in the usability of Rich Internet Applications. Rich Internet Applications are new kinds of web applications that offer more possibilities to design user friendly and attractive graphical user interface (GUI) that is comparable to the GUI of desktop applications. To investigate if the usability of the Rich Internet Application is indeed better than the traditional web application, two car insurance web applications are built as a proof of concept. These web applications are built with the same usability principles and have the same content and functionalities. The only difference is the technology that is used for building the web applications. The RIA is built with JavaFX and the traditional web application with PHP/HTML. These two applications are tested in an experiment where 26 participants with different ages and sex have tested two web applications in a lab. The results of this experiment are statistically analyzed to find dependency.

From the analysis we have found that the satisfaction is significant for the tested web applications and therefore we can conclude that the usability of the Rich Internet Applications is indeed more usable than the traditional web application.

This finding can be useful for finance and insurance companies to convert their web applications into RIA to attract more clients to buy their products.

Title: Usability of Rich Internet Applications
Author: Noureddine Ou-aissa
Student number: 1100203
Date: 06 September 2009
Master programme: Media and Knowledge Engineering
Section: Man-Machine Interaction
University: Delft University of Technology
Supervisor: Dr. Ir. C.A.P. G. van der Mast
Graduation committee: Prof. Dr. M. A. Neerincx
Dr. Ir. C.A.P.G van der Mast
Dr. Ir. M. Pil
Ir. P. Bakker
Defense date: 09 September 2009
Preface

I would like to take this opportunity to express my gratitude to all people that have helped me during this project. First of all, I would like to thank my supervisor Dr ir Charles van der Mast for his help, guidance and advice during my thesis trajectory. I also want to thank Ir. Paul Bakker and Dr ir Marco Pil who have helped me with technical issues. I also want to thank Dr ir Willem-Paul Brinkman for his assistance with the statistical analysis with SPSS. Furthermore, I want to thank Marieke Keurntjes and Alex Hoeksma for their support and management that was necessary for the project.

Many thank to all the participants from the Bloodbank of Amsterdam who have helped me by their valuable input and time.

Finally, I would like to thank my friends and family, especially my mother for the help and support in difficult times.

September 2009,
5.2.5 Time for calculating the insurance fee ................................. 65
5.2.6 Time for reporting the damage ........................................ 66
5.2.7 Clicks for calculating the insurance fee .............................. 67
5.2.8 Clicks for reporting a damage ......................................... 68
5.3 Statistical analysis ............................................................ 69
  5.3.1 Subjective data .......................................................... 69
  5.3.2 Objective data ........................................................... 71
5.4 Participants comments ..................................................... 74

6 Discussion .................................................................. 76
  6.1 Limitations ................................................................. 77

7 Conclusion and recommendation ........................................... 78

8 Abbreviations ............................................................... 84

9 Appendix ..................................................................... 85
1 Introduction

The Internet is now one of the basic things in our life. We use Internet in education, work, home and even during our travels. Because of the availability of Internet in the Netherlands, the consumers have a broad choice of websites where they can search for information or do their transactions.

The computer applications are continuously evolving. First, we had computer applications without a GUI (Graphical User Interface). The interaction with the users took place through use of command prompt. The GUI is a human-computer interaction mechanism that is better mapped to human hand [Galitz, 2007]. The first GUI is invented in 1973 by research group of Xerox PARC [Reimer, 2005]. After this Apple has developed the Macintosh GUI in 1984 which uses a desktop metaphor. This GUI style was very successful and user friendly. That is why Microsoft has taken up this GUI style for Windows.

The usage of the Internet also had a similar development. The first commercial Internet was only used by experts, engineers and scientist [Galitz, 2007], because it was difficult to use and poorly accessible for non-experts. After introducing HTML (Hyper Text Markup Language) and web browsers the Internet became popular. This popularity is partly caused by the increase of usability of web applications.

One of the advantages of the Internet is accessibility. The only thing needed is a computer and an Internet connection and you can access information from every place in the world. That is why computer applications started moving from our desktops to the web platform.

The problem with these kind of web applications is that they were very basic. The GUI was simple and did not have the features of a desktop application with many interactive components that could be helpful for the users. To solve this problem a new concept was developed. This concept is called RIA (Rich Internet Application). According to many authors RIA delivers a better performance of Internet applications and a higher usability [Staley, 2007, Nada and Helwig, 2005]. This improvement of the usability leads to better designed web applications that are easy to use and can persuade the user about the benefit of the business that is offered by this web applications [B.J.Fogg, 2002].

Unfortunately the statement that the usability of the RIA is better than the usability of the traditional web application is not proved yet.

Info Support is a software house that has many customers in different area’s. Info Support provides solutions and implements high quality software in order to make the business of the costumers more profitable. Because many costumers especially in banking and insurance branch are dependent on the existing technology, they have to be convinced of the benefits and profits of the new technology in order to switch to it. Therefore Info Support unit finance/insurance has started this research project to prove if the RIA indeed delivers a better usability then the traditional web technologies that are based on (D)HTML.
1.1 Research Goal

The question that has to be answered is:

*Are RIA’s more usable than traditional Internet applications?*

With traditional Internet applications we mean all web applications that are built with (D)HTML technology. In order to answer this question, the following sub-questions need to be answered:

- What are RIA’s?
- What is the difference between RIA’s and other web applications?
- What is usability?
- How can we measure usability?

1.2 Research approach

This research will exist of four steps.

First a literature study will be conducted. In this literature study we will give answer to our sub-questions. After we have collected enough information, a proof of concept should be built. This proof of concept will exist of two web applications. One is built with a RIA technology and the other is built using the traditional technology. The choice of technology used for implementing the RIA will be based on the information that is collected in the literature study.
The third part of the research will be an experiment. In this experiment the two implemented web applications will be tested by participants. The collected data of this experiment will be statistically processed. Based on the result of this data we can answer our main research goal.

1.3 Thesis outline

Chapter 2 will describe relevant literature about RIA and usability. This chapter will provide enough knowledge to be able to set up an experiment. In chapter 3 the web applications that are implemented will be discussed. In chapter 4 the setup of the experiment will be explained. In chapter 5 and 6 the results will be presented and analyzed, finally in chapter 7 the conclusion will be given that provides an answer to our main research question.
2 Background

In this chapter the literature review will be given. All important aspect of the RIA and the usability will be explained. Based on this chapter the experiment will be set up.

2.1 RIA

After the invention of Internet, HTML (HyperText Markup Language) was used as predominate markup language to represent web pages. A markup language describes the identity of elements of a document, like paragraphs, headings etc. At that time HTML was used to present and structure the data and little attention was paid to the UI (User Interface). The GUI of the web applications was considered not so important because Internet was only accessible for a minority that was using the Internet for specific goals, in other words, Internet was not for broad public, but rather for small groups like military and research institutions [Galitz, 2007].

Today, almost 1.5 billion people [Group, 2009] have access to the Internet and the number is growing every day. A lot of business owners want to present their information or services to a large group of consumers. To achieve this a good designed website that contains all the information and functionalities that are needed. To persuade the consumers the website must be also usable and enjoyable [B.J.Fogg, 2002].

HTML was never meant to provide web application with attractive and interactive User Interfaces. Because of this limitation of HTML a new powerful technology that could make the use of web applications interactive and enjoyable to users was needed.

In March 2002, Macromedia has introduced the term RIA [Alaire, 2002]. This new kind of application should provide a solution to the limitations of traditional web applications that are built with HTML technology. This is done by combining the strengths of the traditional web applications with the strengths of desktop applications. This results in the concept of richness of the data model and the user interface. Nada and Helwig [2005] have summed up some of the strengths of both desktop applications and web applications.

Advantages of web applications:

- Standardized tags/scripts are easy to develop (rapid development, low cost development because HTML is not difficult to learn)
- User interface (UI) is simple and standardized (low delivery and maintenance costs)
- Applications can run on different operating systems (availability, flexibility)
- Applications are accessible from networked computers (platform independence)
- No installation, updates or patches are necessary

Advantages of desktop applications:

- Richer user experiences (audio, video, widgets)
- Efficient interface reloading (no synchronize data transfer)
- Support both online and offline deployment
- Enable more complex applications because of the computing power of the desktop machine.
• More responsive and interactive GUI.

The main disadvantage of web applications is the lack of tools to build a sophisticated GUI and the dependency of the synchronize communication model. On the other hand the desktop applications have the disadvantage that they have to be installed and configured before they can be used.

Today the gap between desktop applications and web application is smaller than ever, because RIA technologies make it possible to build desktop applications that use the Internet as a resource. This offers the possibility to let the browser render web applications that have a User Interface comparable to the User Interface of a desktop application and beside that it allows the communication across the web to take place just like a web applications.

The popularity of this kind of web applications is rising very fast, because some of the companies have showed that the use of RIA’s has an increase in revenues [Duhl 2003].

The concept of RIA was known before 2002. Microsoft has named it “Remote scripting” and Forrester Research had named it the X Internet. Some authors [Fain et al. 2007] suggest that the Java Applets, which were based on the AWT library, were the first attempt to build this new kind of applications. The idea behind the Java Applet was good, but the problem was that this technology was used at the wrong time [Brocklehurst 2007]. Java applets had to download the JRE (Java Runtime Environment) that required much higher bandwidth than was available at that time. There was also a lack of rapid GUI builders for Java which results in mostly poor designed User Interfaces. These are the main reasons why Java applets never became popular. Java has reintroduced Java applets in update (6.10) along with speeding up the loading time and offering the possibility to build widgets with Java applets. These improvements will certainly help the Java applets to become more usable and improve the user experience. However, in my point of view, Java applets are still not offering the desired usability and cannot rise to the expectation of the user. That is why I am still not convinced to see Java applet as a RIA technology.

This section will give the definition of the RIA and explain the differences between traditional web applications and the Rich Internet Applications. The traditional web applications refer to applications that are built with (D) HTML technology. The second part of this section will explore some of the technologies that are used to build RIA’s in combination with the Java back-end.

2.1.1 RIA characteristics

There are many technologies rising that stimulate the architecture of RIA such as JavaFX [Sun 2008a] and Adobe Flex [Adobe 2009c]. In this part we will mention the common properties of the RIA concept. These properties will then be explained in depth.

The most common properties of RIA’s are [Nada and Helwig 2005; Staley 2007; Fain et al. 2007; Allaire 2002; Hompe 2006]:

• They communicate asynchronously.
• They offer the possibility to perform some operations on the client side.
• They are only functional if an Internet connection is present. Some of the RIA can work offline, but to update the data an Internet connection is needed.
• They have a sophisticated User Interface.

These four common properties stimulate creation of fast responding and user-oriented web applications that often have an attractive User Interface. For the sake of simplicity, we will use these four properties to classify web applications as RIA.
RIA’s have an intermediate layer that extends the functionalities of the web browser. This extension is downloaded at the start of the session. This intermediate layer is most often referred to as a thick client because it offers more functionality than the traditional web applications offer. It serves as an engine that can take over some of the tasks that normally are done in the server. This layer also provides the ability to communicate asynchronously.

RIA’s can be categorized according to their architecture structure. In figure 2 the different architecture structures of RIA’s is presented. The gap between the applications and the web applications is filled with different RIA architectures. In the RIA architecture the business logic can be placed in the client side or in the server side. The components in the client side are executed in the RIA engine.

![RIA Architecture Diagram](image)

Figure 2: The different architectures of RIA [Moritz, 2008].

The main disadvantage of the traditional web applications is the synchronous communication. That means that when the user triggers an action, the HTTP request goes to the server and the server sends a complete page with the information and images that are required back. This mechanism consumes time, because the speed of the refresh rate is dependent on the speed of Internet connection. This procedure can result in a waiting user who is willing to move to another site that performs the same job in a shorter time period [Staley, 2007]; [Hompe, 2006].

RIA’s are giving solution to this problem by introducing the asynchronous interaction between the user and the RIA engine [Loosley, 2006]. Most people think that the whole communication is
asynchronous, but that is not the case. The data transfer between the server and the client is synchronous but it is not completely dependent on the user’s action. The data can be fetched from the server while the user is working in a User Interface. The RIA replaces only the components of the User Interface that have to be updated. In figure 3 the difference in the communication between the traditional applications and the RIA’s is illustrated.

According to Loosley [2006] the following points are applicable to the communication model of RIA’s:

- Information can be fetched from a server in anticipation of the user’s input.
- In response to an input, the screen can be updated incrementally instead of all at once.
- Multiple user inputs can be validated and accumulated on the client before being sent to the server.
- Responses to some inputs can be generated without communicating with the server.
- Processes previously handled by servers can be offloaded to the client desktop.

![Figure 3: The left model represents the communication model of the traditional web application and the right model represents the communication model of the RIA.](image-url)

This kind of interaction provides fast feedback to the user which is necessary for some applications like financial application. In financial applications like in stock exchange applications the data changes all the time. The GUI of this kind of applications is data intensive. By applying the RIA concept the user can have many benefits like, fast notification of changed exchange rates. This gives the user the feeling that he is communicating real time with the server and can focus only on the important parts of the GUI.

Another advantage of this interaction model is that the network load will be reduced when the RIA is applied. It is true that the user has to download the engine at the beginning of the session but during the usage, there will be less data transfer from and to the server [Farrell, 2007].

The thick client does not only serve as a render engine, but it can also perform calculations and act as an application. This means that some operations can be done on the client side, resulting in reducing the load of the expensive server [Izaks, 2007].
The main problem with RIA’s according to Gwardak and Pahlstorp [2007] is that they operate with a technology that is not or less compatible with RIA technologies. The HTTP is state-less and uses some functions of the browser that are not used by the RIA’s. Examples of these functions are back, next, refresh buttons and the book marking of the URL’s.

Another drawback is the limited indexability by popular search engines like Google and Yahoo [Moritz 2008]. However Adobe and Google are trying to improve the indexing of Flash websites [Farrell 2007].

RIA’s are running inside sandbox. This is a security mechanism that provides the service for running programs safely by blocking access to some functionalities of the operating system. The sandbox protects the systems but it limits the functionalities for the programmers, for example the possibility to inspect the host system.

To give a solution to this problem, some RIA technologies offer the possibility to create RIA’s that can be deployed as desktop applications, for example Adobe AIR (Adobe Integrated Runtime) [Adobe 2009]. AIR applications are applications that can work offline just like a desktop application and when an Internet connection is available data can be transferred. AIR web applications have the structure of group 2 in figure 2.

2.1.2 Technologies

There is now a broad variety of RIA technologies. Each of these technologies has its own advantages and disadvantages. Based on the way they are deployed we can divide these technologies into three groups [Farrell 2007]:

- Plug-in-based RIA’s. This type of RIA is dedicated to the platform. It can be embedded as a solution or launched from a browser as a standalone application. Examples of this kind of RIA are Adobe Flex, JavaFX, Open Laszlo and Microsoft Silverlight.

- Script-based-RIA. This type of RIA uses a combination of technologies such as XHTML/HTML, CSS, DOM and JavaScript. HTML and CSS are used to style and present the interface and JavaScript is used to make the asynchronous request to the server. An example of this RIA is AJAX.

- Browser-based RIA. This incorporates a XML based User Interface language, where the elements and the interaction in a declarative format can be specified. An Example of this technology is XUL.

The script-based RIA are in my point of view not a new technology for building RIA’s. It is just a combination of existing technologies that were used in the traditional web applications. This kind of RIA has a better indexability and are popular because programmers are familiar with most of the used technologies. In the next part some of the RIA technologies will be discussed.

AJAX

AJAX (Asynchronous JavaScript And XML) seems to be a hype at the moment. There are now many AJAX toolkits available. Some of the toolkits are open source. Google has proved that it is possible to build very dynamic fast interacting web application with AJAX technology.

AJAX is a combination of several technologies that helps creating fast interactive web applications. It is based on the following web standards:
- JavaScript
- XML
- XHTML
- CSS

XHTML and CSS are used for the presentation, data interchange is done by XML and XSLT. The asynchronous server communication is achieved by using XMLHttpRequest and JavaScript is used to bind everything together.

In figure 4 a scheme of a communication model shows how these different standards interact with each other.

![Ajax based web application](image)

Figure 4: The communication model of AJAX based web application [Kumar, 2008].

As mentioned above, AJAX uses many technologies that have some drawbacks. The various browsers tend to interpret JavaScript a little differently. Web application that is developed with AJAX may have an unexpected behavior if it is launched from various web browsers. The developers should take this in account and try to provide different solutions when developing RIA’s.

The User Interface is based on HTML and XHTML, which means that the User Interface will not contain flashy animated elements that are now typically used in RIA. Therefore AJAX does not fit in the definition of RIA that we have mentioned earlier, because of the fast development the user now expects web applications that deliver a high level of user experience AJAX now can not deliver. This problem is now solved by integrating AJAX with other RIA technology like Adobe Flex and Microsoft Silverlight.

The User Interface is not the only problem with AJAX web applications. Some of known AJAX applications have a low performance. Gmail is an example of user’s frustration. Some known problems are searching e-mails and the accessibility of the e-mail from different browsers. A example of a AJAX User interface can be seen in figure 5.
Adobe Flex

Adobe offers developers a framework to build Rich Internet Applications. The core of this framework is the Flash player that allows RIA’s to be launched from the browser. This player is according to Adobe installed on more than a billion desktops [Adobe 2009b]. As you can see in figure 6 Adobe offers many possibilities to merge technologies for building a complete RIA. This gives the opportunities to companies that are using a Java back-end to easily switch to RIA.
To allow developers building RIA’s in an easy and rapid way, Macromedia, the predecessor of Adobe, has developed the Flex framework in March 2004. This is now a stable and mature cross-platform technology for developing RIA [Fain et al., 2007].

The core of Flex is the open source Flex SDK (Software Development Kit), which consists of Flex framework, Flex compiler, a debugger, the MXML and the ActionScript 3.0 programming languages.

MXML is an XML based declarative programming language that first was meant to build GUI. In the last version of the MXML language it is possible to define non visual aspects like access to server-side data sources and data binding between the User Interface and components on the server side. Most of the MXML tags correspond to ActionScript 3.0 classes. Here below in figure 7 a simple "hello world" example in MXML is shown.

```xml
<?xml version="1.0" encoding="utf-8"?>
    <mx:Text x="10" y="30" text="Hello world" width="335" color="#000000" fontSize="35"/>
    <mx:Button x="63" y="115" label="Click me"/>

    <mx:Panel x="235" y="10" width="223" height="166" layout="absolute" title="Find Panel" />
    <mx:Panel x="235" y="174" width="223" height="166" layout="absolute" title="Result Panel" />
</mx:Application>
```

Figure 7: Code example of Flex 2.0
ActionScript 3.0 is an object-oriented ECMA Script language that allow developers to implement custom behavior within the Flex applications. It is also possible to implement the whole web application in ActionScript.

One of the disadvantages of Flex is that the developers are limited to the available defined components. It is difficult or sometimes impossible to create customized components.

Another disadvantage is the limited library of ActionScript. It is true that ActionScript is an object-oriented language, but it still does not have a powerful library just like Java. Here below in figure 8 an example is given of an operational web application that is built with Adobe Flex.

![Figure 8: Example of Adobe Flex web application](Volkswagen, 2008)

**JavaFX**

JavaFX is a technology for building RIA’s. It has been introduced in May 2007. JavaFX has evolved in last two years to a stable RIA technology with many possibilities.

It is based on the JavaFX Script language which offers an easy way to declarative construct GUI’s. JavaFX differ completely from the Java language. The JavaFX is a declarative and statically - typed programming language. The idea behind JavaFX is to optimize and make the development of the GUI creative by using existing components like Java Swing, Java 2D and Java 3D. It is also a cross platform technology that runs on systems that have a Java Virtual Machine (JVM). That JVM is installed on almost 91% of the desktop that are connected to the Internet over the world according to Sun Microsystems [Canoo, 2008]
One of the benefits of using JavaFX is that it is easy to create RIA’s that can run on different platforms and devices, since the Java platforms is present on a lot of devices like mobile phones and robots. Java FX mobile is introduced in February 2009 and it has shown that it is able to compete Flash on mobile phones.

Security is also an advantage of JavaFX. JavaFX can call existent security API (Application Program Interface) that is available for Java, which ensure a high degree of security.

JavaFX supports all video compressions and animations can be defined very easy. A powerful concept that is used in JavaFX are the bindings. With bindings attributes or objects can be binded to each other and when an attribute changes, the binded attribute change then automatically. This concept makes it easy to link GUI elements to some data. Here below in figure 10 a simple “hello world” example in the JavaFX script is given.
The problem with JavaFX is that there is still no GUI builder available and there are also limited GUI components available, therefore the developer has to define everything. The developers will spend a lot of time to create the GUI, but on the other hand, the developers have many possibilities to define the look and feel of the customized components because they are not limited to the out-of-the-box components.

An another known problem with JavaFX is the absence of a layout manager. The position of every component must be declared which may take a lot of time.

Summarized we can say that JavaFX is a RIA technology with many possibilities for especially programmers that do not have too much experience in the design but the drawback of this technology is that it takes a lot of time to implement everything that is needed. In figure 11 an example is given of web application that is built with JavaFX.
2.1.3 GUI of RIA

Graphical User Interface is very important for the interaction between the user and the system. RIA’s are characterized by the additional functionalities that existed for desktop applications. This functions help the web applications to become more user experience oriented and more usable. In this section some of these functionalities will be highlighted.

**Direct manipulation**

Direct manipulation is an interaction style that was described by Shneiderman and Plaisant [2007] as:

“visibility of the objects and actions of interest; rapid, reversible, incremental actions; and replacement of typed commands by a pointing action on the object of interest”

This interaction style metaphor was initially meant for desktop application. The essence of this methodology is to let the user manipulate objects by pointing with the mouse and dragging it over the screen. An example of this is the “drag and drop” functionality that many RIA technologies offer.
This style has many advantages:

- Novices can learn very fast using the application by a demonstration of an experienced user.
- Experts can perform task very fast.
- Error messages are rarely needed.
- Users can get immediate response to their action.

The context of use is every important for applying this style. Some applications have faster performance if the user uses command typed actions by keyboard.

The developers should also take the response time in account. Too long response time can give the user the impression that his action was not performed and can cause further repeating of the same action, both frustration the user and putting more burden on the application, potentially slowing it even further.

**Resolution**

Resolution independency is very important for showing a GUI in many browsers. Most browsers now have the possibility to show zoomed images and text. This offers some advantages for users with disabilities but because most of the traditional web applications use pictures in pixel-based format, the zoomed images can result in some artefacts and aliasing problems.

Therefore some of the RIA technologies like Flash and JavaFX [Moritz 2008] offer the possibility to build a vector based GUI. This gives the advantage that the same GUI can be seen in different screen resolutions.

**Animation**

Animation makes things ‘cool’ and can contribute to a better explanation of some functionality. That is why a lot of designers use animations. There are two kinds of animations [van Nieuwenhuizen 2005]:

- Low-level animations
- High-level animations.

In low-level animations, a lot of work has to be done by the animator. This results in a better control of the animation. An example of this animation category is the key frame animation.

High-level animations are also called procedural animation. The animator has only to specify some parameters for a model. And the model creates an animation. An example of this is a particle system for creating fireworks.

With RIA’s one can achieve many more effects with animations. Flash and JavaFX [Sun 2008b] offer the possibilities to use low level techniques like key frame animations and time line animation. The use of animation in the GUI can be useful by moving, rotating or morphing objects on the screen.

This enriches the User Interface and can also deliver a better user experienced, but developers must be aware that using a lot of animations in a GUI can slow down the user to accomplish tasks that have to be performed. Loading and playing animation is dependent of the specification of the video card and usually consumes time. This time may slow the efficiency of the tasks that have to be performed.
2.1.4 Conclusion

In this chapter we have defined RIA as a web application that can be launched from the Internet browser or as a desktop application and can asynchronously communicate between the user and the RIA engine which often is referred as a thick client.

RIA’s are presented as web applications that have the benefits of desktop applications combined with the benefits of the web applications. This results in web applications that are able to keep the attention of the user with their user oriented interface and their quick response capability.

Adobe Flex and JavaFX are examples of technologies that can be used to build RIA’s. These technologies can be used to build RIA’s with a java back-end. This is especially useful to (partly) convert web applications to RIA’s.
2.2 Usability

Usability is an important issue in the development of application. Usability does not depend only on the product, but it has also to deal with attribute of interaction with the product in the context of use [Bevan and Macleod, 1994][Karat et al., 1992]. The context of use is dependent on the next components:

- User
- Task
- Equipment
- Environment

Usable software leads to successful software. In Figure 12 we can see the attributes that are important in the acceptance of a system. Usability is a subtree that has to be considered when a system or a web application has to be accepted.

![Acceptability tree](image-url)  
**Figure 12: Acceptability tree [Nielsen, 1993]**

ISO 9241-11 [ISO, 1998] defines usability as:

*"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".*

We can define usability as a quality of use of an interactive system that has to deal with many human characteristics. This quality of use can be measured through usability attributes. [Nielsen, 1993] has defined five usability attributes:

- Learnability
- Efficiency
- Memorability
- Error
- Satisfaction

Usability has a link to cognitive science. Cognitive science is the interdisciplinary study of mind and intelligence [Thagard, 1996].

In this chapter we will explain the usability attributes and how usability can be applied in the design. At the end some usability evaluation methods will be explained.
2.2.1 Usability attributes

The usability attributes are important measures of the quality of use [Bevan and Macleod 1994]. These attributes are user dependent and are measurable. The importance of these attributes is dependent on the context of use. For example, when an application that is going to be used in the aircraft is evaluated, the error-attribute is very important. If a pilot can not restore an error in a short time, the consequence could lead to a disaster of many victims.

Learnability

Learnability is one of the important issues when software has to be used. A system has to be easy to learn independent of the experience of the user.

Learnability of a system depends on the context of use. For example, web applications that are not used very frequently must be designed in a way that they will be learned in a short time. But for an application used in a nuclear facility the designer has to pay a lot more attention to the efficiency of use especially when an error has occurred. The user must react in an efficient way to minimize the damage.

Learnability is an attribute that can be measured easily. This can be done by taking a novice user and measure the time that he needs to learn working with the system.

Efficiency

According to [Nielsen 1993], the efficiency refers to the expert user’s steady-state level of performance at the time when the learning curve flattens out. This means that the user has reached a level that he can’t learn much more about the system. The time to reach this state is dependent of the user and the complexity of the system.

Efficiency can be measured by finding experienced users and let them perform tasks. During these tasks, the time has to be measured. This process is done until the user reached a steady-state level. One of the problems with this measurement is defining what an experienced user is.

Most of the test organizer let the user work with the system for a couple of hours. After that they become an experienced user in their point of view.

Memorability

Memorability has to deal with the use of the system after the user has stopped using the system for a while. For example if a user become sick or goes on vacation.

People who don’t use the system frequently are called casual users. Measurement of this third attribute has to be performed with casual users.

There are two way’s of performing a measurement for memorability. The first method is by letting the casual user perform a typical task after he has not used a system for a while. The time that this casual user has spent to do the tasks will be compared to the time that he used to spend when he worked frequently with the system. If this data is not available, the average time that a moderate user spends to perform the same task can be used for comparison.

The other method is by letting the user answer some questions about how some tasks could be performed and explaining commands and icons.
Error

Errors are actions that do not accomplish the desired goal. A usable interface should not contain errors and if an error occurs, the system should guide or help the user to return to stable state.

There are several kinds of errors. We can distinguish errors that happen without side effects and just slow down the user in his task and errors that can have a huge impact on the performance and task accomplishment.

This attribute can be tested by counting the second category errors.

Satisfaction

Subjective satisfaction refers to pleasantness the user experiences during the use of the system. This attribute is very hard to measure because the user feeling about the system can differ. A lot of researchers have tried to measure satisfaction with help of tools like EEG (electroencephalography). Unfortunately because the feeling is affected by the environmental setting of the test, the test results were not valid.

The test can be done by giving a questionnaire to the user after the test is conducted. The questions should be as short and simple as possible. The tester has also to define an appropriate scale for the score.

2.2.2 Usability engineering

Usability engineering is a multi-disciplinary field. Usability engineers define guidelines and methods that help software developers to improve the usability of the software. Some of the guidelines will be discussed in the next section. The usability engineers must understand the user by analyzing the limitations and the cognitive state and tries to take this in account when the design is performed [T.A.O’Connell and E.D.Murphy 2007].

In this report we define user as the end product user. This user will work with the application or the website.

There are many variables that can influence the experience of the user. Some of them are intrinsic, for example; age, gender, aesthetic preferences and cognitive disabilities [Karat and R.G.Bias 2005]. Other variables that can influence the experience are the extrinsic variables such as, working environment and the user’s goals. These variables have to be analyzed carefully in order to make an usable design. The engineer has to think about which category users they include and exclude in the design. For example if a web application has to be accessible for the elderly people, the engineer has to pay attention to the cognitive disability such as limitation of memory or visual disability.

In the design the mental model of the user has to be mentioned [D.A.Norman 1988]. Every user that works with a particular software, creates a mental model about how to use a software or a web site. The usability engineer has to know this mental model [Karat and R.G.Bias 2005].

To achieve a good usability for the software, it is recommended to let the user to take part of the design. However one of the disadvantages of this way of engineering is that the design could be influences by the preference of the participant [Nielsen 1993].

Usability engineering can be integrated in the lifecycle of the software engineering process [Karat and R.G.Bias 2005]. In each of the phases the usability engineer has his own responsibility.
In the planning phase the usability engineer has to help with addressing milestones and the accessibility issues that has to be considered when building or improving software. For example if a web application is going to be used by a broad audience. The website must generate tags, than can be used by special web browsers to read the text for users with visually disabilities.

In the requirement definition phase, the usability engineer must address the users need and convert them into non functional requirements. The job of the usability engineer exists of first analyzing the users in order to form user groups. After that the goals and the tasks that could be done with the website or software must be analyzed. This could be done by a task analysis (see section 2.2.4). From this analysis a persona will be created. A persona is a detailed description of the user. This persona will be used in scenarios. The output of the analysis will form the usability requirements. These usability requirements define the usability issues that have to be evaluated at the end of the development cycle.

In the design the usability engineer applies UCD (User-Centered Design). The UCD is based on iterative design that is characterized by evaluating the design by the users. This can be done by paper prototype or a more complex prototype method. In the design, the look and feel of the website or the software has to be defined.

In the testing and evaluation, the requirements that have been defined are tested and evaluated according to evaluation methods.

### 2.2.3 Usability guidelines

Different researches have tried to give some guideline for the creation of highly usable interfaces. These guidelines are used to design usable interactive systems. Shneiderman and Plaisant [2007] and Nielsen [1993] have developed guidelines for usability. Shneiderman called his guidelines the eight golden rules. These guidelines are:

- Consistency
- Plasticity
- Feedback
- Closure of dialogs
- Error prevention and handling
- Reversal of action
- Internal locus of control
- Memory load reduction

The guidelines that are developed by Nielsen are called the “ten usability heuristics” [Nielsen 2005b] and they contain the next guidelines:

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
• Error prevention
• Recognition rather than recall
• Flexibility and efficiency of use
• Aesthetic and minimalist design
• Help users recognize, diagnose, and recover from errors
• Help and documentation

In this section we will explain the guidelines that are mentioned above and how they could be integrated in RIA's based on the heuristics of McMullin and Skinner [1993].

Consistency

Consistency is one of the most important issues in design [Shneiderman and Plaisant, 2007]. All of the interface elements should be consistent in terms of layout, color, use, fonts, menu order etc. Because of consistency the user does not have to learn new or different procedures, therefore the chance of error occurrence will be reduced.

Consistency could be achieved with RIA technologies by using standard components and using the same layout for different parts of the web application.

Plasticity and efficiency of use

The need of the different users must be recognized. For example novice users will need more explanation and expert users will need shortcut to perform action quickly. By taking in account of the needs of different user groups, the quality of the systems improves.

Auto-completion should help the advanced users to save time filling in fields. It is also possible to have a kind of intelligence in the web applications to collect time spent on particular operation. This data could be used to divide the GUI components according to the relevance of the operations.

Feedback and visibility

The User Interface should provide the user feedback. The feedback is needed to inform the user about the actions that are taken. Frequent used actions should contain less feedback than the actions that are taken rarely.

According to McMullin and Skinner [1993] RIA's have a rich display capability that could provide real-time status indicators. This is especially useful when processes take too much time and the user has to be notified. Using graphical bars for example could provide the user a better feedback than showing numbers.

Closure of dialogs

This is an extension of the previous guideline. The sequences of actions have to be grouped in a way that a user is aware in which part of the process he is. This guideline is mostly seen in web-shop applications.
Match between system and the real world

The developer has to use the language that the user can understand. This means that the developer has to use the same language the end user understand and tries to avoid technical word or abbreviations that are not understood by the user.

Error prevention and user control

The User Interface should be designed in a way that prevents the user to make fatal errors and if an error occurs the interface must help the user to return back to a stable state. The occurrence of an error must not have a high influence on the performance of the system. The user also must have the possibility to stop an operation if an error has happened. Preventing errors can be done by checking the types of fields and not allowing users to do operations that are not allowed in a particular state.

RIA technologies offer many possibilities to prevent the user from making errors. Some of common used techniques are blurring or highlighting fields/buttons that can cause an error.

Because RIA supports integration of multimedia elements, interactive video wizard could give the user a fast feedback about how to recover from an error. The asynchronous data transfer offers more time for the client to check fields automatically when the user is filling in the forms.

Reversal of action

The user interface must offer the possibility to reverse an action or a sequence of action if an error has occurred. This should be designed in a way that the user does not have to start the whole sequence of actions.

Because the undo and redo functionalities are still not functional for most RIA technologies, the developer should take this into account. He could for example store temporary data of a session. In case an error happens, the user can retrieve the filled data. It could be helpful if in a RIA an undo and a redo functions is implemented.

Internal locus of control

The user interface must be designed in a way that allows the user to be the initiator of actions rather than a responder. The user must have the ability to do different processes in one screen. RIA’s could have more components on the screen which supports working with different processes at the same time. It will not be necessary for the user to wait until one process has finished.

Memory load reduction and Recognition

The human has a limited capability of processing the information in the short term memory. Therefore he should not have to remember information from one part of a dialog to another. That is why the User Interface should be kept simple; multiple page displays should be consolidated.

RIA’s could use rollover options and tool tips to give clues about the formal dialogs. Animations that are used for wizards could also be very helpful to remember data that is entered in former dialogs.
Aesthetic and minimalist design

Dialogs must not contain information that is not necessary to complete the dialog. Advanced features have to be applied carefully and only in components where they can provide significant advantage. A balance must be found between aesthetic and functionalities.

In RIA’s the use of animations and sound effects must be limited. Too much use of effects can distract the attention of the user and consume more time to load. This can result in a less efficient use of the web applications.

Help and documentation

Help and documentation are necessary for using an application. The documentation should be short and easy to use.

RIA’s could provide this help through the use of video tutorials and animations.

2.2.4 Usability evaluation

Usability evaluation is essential in design. It is also important to evaluate the system in all different stages of the system life cycle. The earlier the evaluation is done the better, because it can reduce a lot of development costs. According to Nielsen [1993] validity and reliability are very important in usability. Validity is the question if the evaluator has measured what he really wants to measure. Reliability refers to if the results of the measures will be the same if the test is repeated.

The testing can be done in different ways [Scholtz 2004]. Evaluation can be done with users to identify the problems and checking the quality of the User Interface. There are two ways of testing with users:

- Formative evaluation
- Summative evaluation

Formative evaluation is done during the development [Scholtz 2004, Nielsen 1993]. The goal of this evaluation is to let users find positive and negative aspects in the design. This will help improve the development of the User Interface.

A summative evaluation is a formal evaluation that has to be done at the end of the development lifecycle. Nielsen [1993] has suggested testing with five - seven users of the end user group. The goal of this evaluation is to check the usability of the end product and to compare it with the desired level. This evaluation contains measurements that measure the usability attributes of the User Interface.

Another testing strategy is to evaluate the User Interface by experts. Experts have a lot of experience and can find problems that a developer over sees. A third evaluation strategy is by evaluating the User Interface according to existent model.

The evaluation strategy and methods that will be used for the evaluation are dependent of the budget, costs and the facility of the project. There is not a best evaluation strategy. In this section I will explain some methods that can be used in evaluation and how a test can be conducted.
User testing

User testing involves users that represent the end user group. These testers have to perform some tasks that represent the tasks that are normally done by the end user.

Finding the users that represents the end users can be a problem. Before searching for participants a user profile has to be developed. In this user profile the user characteristics has to be mentioned. Based on this characteristics user groups have to be formed. Every subgroup must contain at least three - five users [Nielsen, 1993]. It is recommended to not use friends or people who are familiar with the development of software, because they can influence the results of the test.

The tasks that should be done by the participants should cover most of the functionalities of the User Interface. The instruction of the task should be clear and written in the language that is understandable by the participant. The task should be given as goals that must be achieved, without giving the instructions how to do it [Hanson, 1991].

The location where the test is done is important, because it could influence the test. Therefore many researches are against doing the evaluation in test labs.

Hanson [1991] described testing with users in ten steps:

- Find the background information about the usability evaluation
- Write a test plan
- Design the test
- Arrange a test environment and equipment
- Conduct a pilot test
- Recruit participants
- Set up the test room
- Conduct the test
- Compile and analyze the results
- Recommend changes

Before the development has started, scenarios should be created. Scenarios are according to Nielsen [1993] prototype version that describes a single interaction session without any flexibility for the user. They can be constructed with help of a target group. The engineer tries to understand how the applications should work.

The important parameters that are encapsulated in the scenario are:

- The description of the user
- The use of specific set of computer facilities
- The achievement of specific outcome
- The understanding of specified circumstances
- The time interval.
Scenarios are cheap and flexible in use and let the evaluator thinks about the use of the product in a particular context [M.B.Rosson and J.M.Caroll, 2002] ; [Nielsen, 1993]. Scenarios deliver information about the use of the system and can be used for making tasks for the test sessions.

In the test plan the goals and the test objectives should be specified carefully [Nielsen, 1993] ; [Hanson, 1991]. The test plan should include the following issues [Nielsen, 1993] ; [Neerincx, 2007] ; [Hanson, 1991] :

- Purpose of the test
- Objectives
- Participants description and the amount of the testers
- Evaluation methods that will be used
- The tasks that the participator has to do.
- The measures description and data collection
- Time that is needed
- Settings of the test environment
- The needed resources and the system requirements
- Data analysis
- Documentation and the content presentation

The test plan is needed to give an estimation of the costs and gives an overview to the managers and people who are involved in the test.

There are many methods that can be used to provide feedback from the test participants. Examples of feedback methods are interviews and questionnaires. These two feedback methods could be very useful to get especially objective information from the participants of the test. Interviews are more flexibility. They can be used to ask the participants in depth about issues of the User Interface.

Questionnaires are more used to get a large amount of information that can be presented in quantitatively data. Some researchers have made questionnaires templates to make it easy for testers to gather information. Examples of such tools are SUMI [J.Kirakowski, 1994] and WAMMI [Claridge and Kirakowski, 2008]. Unfortunately these two templates are commercial and a license is needed to be purchased.

**Task analysis**

Task analysis is used to determine goals, tasks and actions. This is done by observing and hearing the comments of the users while they are performing the tasks.

Another way to perform a task analysis is by interviewing the end user group. This analysis must not only contain what the users are doing, but also the goals of the tasks.

These tasks can be broken in subtasks. The subtasks can be presented in HTA (Hierarchical task analysis). HTA is a graphically hierarchical structured chart. In figure 13 an example of a VCR is shown.
This method has to be done at the beginning of the development. It is desirable to do this method with at least three users that represent the user group. This method is unfortunately expensive. Additionally, presence of an engineer in the vicinity of the user can influence the results of the task analysis.

**Thinking aloud**

In the thinking aloud method the users are asked to speak their thoughts as they perform the tasks. During this method the test organizer can take notes or just record the test session.

Applying this method can help the engineer to get feedback about the elements and the interactions that are not understood by the users.

**Cognitive walkthrough**

Cognitive walkthrough is a usability inspection method that can be performed in different stages of the design. In this method one or a group of evaluators inspects the User Interface. It is recommended that the evaluator is an expert in User Interfaces. The evaluators can inspect a paper prototype, working prototype or a full working interface by following a sequence of actions that have to be performed. By inspecting each action, the evaluator can identify the problems that the end user can expect when he is using the User Interface.

The evaluator must first identified the functionality of the User Interface, then the goals and subtasks to execute the goals must be known. After that the evaluator should inspect the (prototype) User Interface by answering some questions about each step that is taken.
This method can estimate the learnability of the User Interface and can highlight some problems with the User Interface in an early stage of the development. A disadvantage of this method is that this process is time consuming.

Heuristic evaluation

Heuristic evaluation is also an inspection method and can be used to find problems in the User Interface according to the heuristics of Nielsen [2005a]. Nielsen recommends that three to five evaluators conduct this evaluation because according to his research, involving a lot of testers does not contribute to a better result.

The output of this method exists of a report that contains findings of the evaluator or a video recording. The evaluators are not supposed to communicate with each other during the evaluation because they can influence the results of each other.

This evaluation should be set up in the following steps:

- The evaluators should have a short training session
- The evaluators evaluate individually the User Interface
- A debriefing session
- Combining the problems into a list
- Estimation of the problems severity

In the first step, the evaluators should get an introduction to the heuristics of Nielsen if they are not familiar with it. They are allowed to get information about the User Interface if they are not familiar with the functionalities. If the User Interface is very complex, the test sessions should be divided in more test sessions. The test session should not take more than two hours.

For the severity of the problems Nielsen recommends the use of a five point rating scale. This scaling is also known as a Likert scale. An example of this scaling could be:

- 0 = this is not a usability problem at all
- 1 = Cosmetic problem only- does not need not to be fixed unless extra time is available on project
- 2 = minor usability problem- fixing this should be given low priority
- 3 = major usability problem- important to fix, so should be given high priority
- 4 = usability catastrophe- imperative to fix this before product can be released

GOMS

GOMS is a model that consists of Goal, Operators, Methods and Selection rules. It was introduced by S.K. Card et al. [1986]. This method is used to evaluate the efficiency of the User Interface.

The goals indicate what the user wants to achieve. Operators are a set of actions that the User Interface allows to take. Methods are well learned sequence of subgoals and operators that leads to accomplishment of a goal. If more methods are defines for the same goal, rules should be applied to know which method has to be performed in a particular case. The user has to choose one of the methods for the presented context.

With the GOMS model we can estimate the exact time of the expert user, it is also possible to estimate the learning time of a novice user. This method is very cheap because no users are involved.
2.2.5 Conclusion

Usability is important for acceptance of the product. It is dependent on several attributes that can be measured with use of different methods. Usability has to be measured in the context of use.

The attributes that can be measured to estimate the quality of the product are according to Nielsen:

- Learnability
- Efficiency
- Memorability
- Error
- Satisfaction

ISO has defined usability as:

“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”.

Usability engineering helps to add usability issues in the development of the software. Usability engineering focuses on user’s needs and satisfaction and are responsible for adding non functional requirements.

The usability attributes can be evaluated with different evaluation methods. The choice of the evaluation method is dependent on the budget and the complexity of the User Interface.

Prior to the evaluation a test plan has to be written. This test plan contains different aspect of the testing that has to be performed.

2.3 Discussion and conclusion

The motivation for this background chapter is to explore the RIA concept and the usability in order to use this information to prove whether the application of RIA in the finance area delivers benefits compared with the traditional way.

First we have introduced the RIA concepts, which was introduced as a solution to the limitations of the traditional web applications. RIA’s are characterized by their GUI, asynchronous communication, deployment on desktop as well as in web browser and the use of the client to perform operations that used to be done by the server side.

This delivers fast reacting web applications that have both the advantages of web applications and desktop application.

Some technologies that can be used to build RIA’s are adobe Flex, JavaFX and AJAX.

The second part of this background chapter was to explain what is meant by usability and how it can be measured. Usability is a quality of use of a product in a particular context with efficiency, effectiveness and satisfaction.

Usability can be integrated in software design by usability engineers. By applying guidelines and iterative evaluations a better usability can be achieved. The goal of the usability engineers is to understand the user and his needs.
The usability evaluations are methods that can measure the usability attributes. These measurements quantify the quality of the product. The choice of the evaluation methods is dependent on the budget and the complexity of the User Interface.

In order to prove the benefits of the RIA’s compared to the traditional web applications, the usability of the RIA’s has to be measured. Usability is one of the issues that can stimulate acceptance of a product.

Many authors claim that the usability and the user experience of the RIA are better than the traditional applications. But unfortunately there is still no research done that can prove this claim.

In further research I will try to prove or disapprove this claim by measuring the usability of two different web applications.

We have chosen to build the RIA with JavaFX, because this new technology offers more flexibility for me as a software developer. JavaFX is also a new technology that best can be used with Java back-end which is interesting for finance and insurance companies. The (traditional) web application will be built with HTML/PHP. By comparing and analyzing the test results we can find answer to the main research question.
3 Applications

To be able to answer our research question two web applications are built with different technologies. These two web applications are not completely functional but they are only used as a proof of concept. The RIA is built with JavaFX technology and the not RIA or what is referred as the traditional web application is built with HTML/PHP.

The subject of the applications that are used for the experiment is a car insurance web application. The reason why this topic has been chosen is because this research has been started by Info Support unit finance/insurance. Most finance web applications that Info Support develop are business to business web applications and therefore it is difficult to inspect such web applications because an authorization is needed. The second reason why we have chosen for this topic, is because we believe that insurance companies especially car insurance companies will have more benefit if the usability of their web application is improved because if the consumers are satisfied about their website, they will trust the company and will accept the insurance fee. According to [Berkositz and Cahil 2006], that have done a research about the satisfaction of car insurance websites of 10 companies. Their findings are:

- 72 % of the participants have reported that they will use the website for insurance coverage
- 74 % have indicated that they would like to make a fee calculation on the website.
- 64 % have indicated that the usability is very important for the choice of which insurance website they use.

For this study two web applications are built. These two web applications have the same content and are built with same usability principles. The content of the two web applications will contain two functionalities:

- **Calculating a car-insurance fee**: here the user can fill in information that is related to the person and the car that will be insured. Based on this information the web application calculates the annually fee. If the user accepts the fee, he can sign in for the car-insurance.
- **Reporting damage to the car**: here the user can report a damage to a car that may be caused by an accident.

The difference between the two web applications are the technologies and the interaction. In this chapter the two implemented web applications will be discusses. First the technical issues and the User Interface of the RIA will be explained, after that the traditional web application will be explained.

3.1 The user

Before building the system we have to think about who are the end users of the web applications. The users are very important in usability engineering and therefore they have to be mentioned.

The users of the website could be all users that have a driving license and have or are managing to buy a car. The age of this large group has a range between 18 and 80. This group can be separated in different groups according to the characteristics of the users.

We will not design the system for users with disability because that is out of the scope of this study.

One major distinction of the end users can be done by classifying the group according to their age. We can distinct older adults and young adults. Elderly users have some limitations that have to be taken in account when designing the web application. Some of the limitations of the elderly are:
- Slow motion of control
- Low learning ability
- Reduced kinaesthetic sensitivity
- Visual limitation
- Limited knowledge of using computer

Slow motion of control means that fast actions and animation in the website have to be avoided. Because the elderly have a limitation in the learning curve, the structure of the data should be placed in a logical way. This will speed up the learning and the use of the web application.

The visual limitations will not weigh too much, because it is supposed that drivers of the vehicles should have a good visual ability to be able to drive.

Another distinction could be done according to their experience with Internet and the use of the computer. People who don’t use or rarely use the Internet will need much guiding.

### 3.2 Functional requirements

The functional requirements define tasks and functions of the web application that have to be implemented. These tasks are related to functions of the car insurance web application. The functional requirements are for the web applications are:

- Calculate a car insurance fee.
- Accept a car insurance fee.
- Send a damage form.

### 3.3 Use cases

In this part use cases will be given to all the functional requirements. To each requirement two use cases will be given one for the RIA and the other for the traditional web application.
### Use case name
Calculate a car insurance fee (RIA)

### Actor
Edward: web application user

### Assumptions
Started the web application

### Steps
1. Edward presses the "Calculation" Button.
2. Edward chooses the brand of his car by selecting "Renault".
3. Edward chooses the manufacture date by selecting "2001".
4. Edward chooses the fuel of the car by selecting "gasoline".
5. Edward chooses the type of the engine by selecting "1.2".
6. Edward chooses the type of the transmission by choosing "hand transmission".
7. The applications presents some pictures of the found models according the information that is filled.
8. Edward clicks on the picture of the model "Clio".
9. Edward clicks on the "Next" button.
10. The system shows an animation of a changing form.
11. Edward fills in the birthday by choosing the "2" in day drop down menu, "12" in the month drop down menu and "1983" in the year drop down menu.
12. Edward enters his zip code by typing "1095KL".
13. Edward has chosen the amount of years he did not have an accident.
14. Edward clicks on the "Next" button.
15. The system shows an animation of a changing form.
16. Edward chooses the insurance type by selecting "WA".
17. Edward chooses the amount of accessories that he want to insure. He has chosen for below "1000 Euro".
18. Edward wants to insure other passengers; he has done this by selecting "Yes".
19. Edward wants also to have a law legal assistance, so he selected "Yes".
20. Edward clicks on the "Next" button.
21. The system shows an animation of changing form. The fee is given with the annually cost of the insurance.

Table 1: Use case for the RIA: calculate a car insurance fee
<table>
<thead>
<tr>
<th>Use case name</th>
<th>Calculating a car insurance fee (traditional web application)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Edward: web application user</td>
</tr>
<tr>
<td>Assumptions</td>
<td>Started the web application</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Edward presses the &quot;Calculation&quot; button.</td>
</tr>
<tr>
<td></td>
<td>2. Edward chooses the brand of his car by selecting &quot;Renault&quot;.</td>
</tr>
<tr>
<td></td>
<td>3. Edward chooses the manufacture date by selecting &quot;2001&quot;.</td>
</tr>
<tr>
<td></td>
<td>4. Edward chooses the fuel of the car by selecting &quot;gasoline&quot;.</td>
</tr>
<tr>
<td></td>
<td>5. The system shows a new field of the engine type.</td>
</tr>
<tr>
<td></td>
<td>6. Edward chooses the type of the engine by selecting &quot;1.2&quot;.</td>
</tr>
<tr>
<td></td>
<td>7. The system shows two new fields of the transmission type and the model of the car.</td>
</tr>
<tr>
<td></td>
<td>8. Edward chooses the type of the transmission by choosing &quot;hand transmission&quot;.</td>
</tr>
<tr>
<td></td>
<td>9. Edward chooses the model of the car by selecting &quot;Clio&quot;.</td>
</tr>
<tr>
<td></td>
<td>10. Edward clicks on the &quot;Next&quot; button.</td>
</tr>
<tr>
<td></td>
<td>11. The system shows the next form.</td>
</tr>
<tr>
<td></td>
<td>12. Edward fills in the birthday by choosing the &quot;2&quot; in day drop down menu, &quot;12&quot; in the month drop down menu and &quot;1983&quot; in the year drop down menu.</td>
</tr>
<tr>
<td></td>
<td>13. Edward enters his zip code by typing &quot;1095KL&quot;.</td>
</tr>
<tr>
<td></td>
<td>14. Edward has chosen the amount of years he did not have an accident.</td>
</tr>
<tr>
<td></td>
<td>15. Edward clicks on the &quot;Next&quot; button.</td>
</tr>
<tr>
<td></td>
<td>16. The system shows the next form.</td>
</tr>
<tr>
<td></td>
<td>17. Edward chooses the insurance type by selecting &quot;WA&quot;.</td>
</tr>
<tr>
<td></td>
<td>18. Edward chooses the amount of accessories that he wants to insure. He has chosen for below &quot;1000 euro&quot;.</td>
</tr>
<tr>
<td></td>
<td>19. Edward wants also to insure other passengers; he has done this by selecting &quot;Yes&quot;.</td>
</tr>
<tr>
<td></td>
<td>20. Edward wants also to have a law legal assistance, so he selected a &quot;Yes&quot;.</td>
</tr>
<tr>
<td></td>
<td>21. Edward clicks on the &quot;Next&quot; button.</td>
</tr>
<tr>
<td></td>
<td>22. The system shows a form of the annually fee.</td>
</tr>
</tbody>
</table>

Table 2: Use case for the traditional web application: calculating a car insurance fee
<table>
<thead>
<tr>
<th>Use case name</th>
<th>Accept a car insurance fee (RIA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Edward: web application user</td>
</tr>
<tr>
<td>Assumptions</td>
<td>Calculating the car insurance has been done</td>
</tr>
</tbody>
</table>
| Steps         | 1. Edward presses the "accept" button.  
2. The system shows an animation of a changing form.  
3. Edward chooses his gender by clicking the radio button of "male".  
4. Edward types his name.  
5. Edward types his last name.  
6. Edward fills in the birthday by choosing the "2" in day drop down menu, "12" in the month drop down menu and "1983" in the year drop down menu.  
7. Edward types his address.  
8. Edward types his zip code.  
9. Edward types his house number.  
10. Edward types his telephone number.  
11. Edward types his e-mail address.  
12. Edward fills in the start date of the insurance by choosing 3 in the day drop down menu, 3 in the month drop down menu and 2009 in the year drop down menu.  
13. Edward types the registration id of his car.  
14. Edward types the car code.  
15. Edward chooses the period he has a driving license. He has selected "5=<"  
16. Edward chooses the payment method by selecting "automatically payment"  
17. Edward chooses the payment period by selecting "annually payment"  
18. Edward types his bank account number.  
19. Edward presses on the send button.  
20. The system confirms the sent car insurance enrollment. |

Table 3: Use case for the RIA: accept a car insurance fee.
<table>
<thead>
<tr>
<th>Use case name</th>
<th>Accept a car insurance fee (traditional web application)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Edward: web application user</td>
</tr>
<tr>
<td>Assumptions</td>
<td>Calculating the car insurance has been done</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Edward presses the &quot;accept&quot; button.</td>
</tr>
<tr>
<td></td>
<td>2. The system shows the next form.</td>
</tr>
<tr>
<td></td>
<td>3. Edward chooses his gender by clicking the radio button of &quot;male&quot;.</td>
</tr>
<tr>
<td></td>
<td>4. Edward types his name.</td>
</tr>
<tr>
<td></td>
<td>5. Edward types his last name.</td>
</tr>
<tr>
<td></td>
<td>6. Edward fills in the birthday by choosing the &quot;2&quot; in day drop down menu, &quot;12&quot; in the month drop down menu and &quot;1983&quot; in the year drop down menu.</td>
</tr>
<tr>
<td></td>
<td>7. Edward types his address.</td>
</tr>
<tr>
<td></td>
<td>8. Edward types his zip code.</td>
</tr>
<tr>
<td></td>
<td>9. Edward types his house number.</td>
</tr>
<tr>
<td></td>
<td>10. Edward types his telephone number.</td>
</tr>
<tr>
<td></td>
<td>11. Edward types his e-mail address.</td>
</tr>
<tr>
<td></td>
<td>12. Edward presses on the &quot;Next&quot; button.</td>
</tr>
<tr>
<td></td>
<td>13. The system shows the next form.</td>
</tr>
<tr>
<td></td>
<td>14. Edward fills in the start date of the insurance by choosing 3 in the day drop down menu, 3 in the month drop down menu and 2009 in the year drop down menu.</td>
</tr>
<tr>
<td></td>
<td>15. Edward types in the registration of id of his car.</td>
</tr>
<tr>
<td></td>
<td>16. Edward types the car code.</td>
</tr>
<tr>
<td></td>
<td>17. Edward chooses the period he has a driving license. He has selected &quot;5=&lt;&quot;</td>
</tr>
<tr>
<td></td>
<td>18. Edward chooses the payment method by selecting &quot;automatically payment&quot;</td>
</tr>
<tr>
<td></td>
<td>19. Edward chooses the payment period by selecting &quot;annually payment&quot;</td>
</tr>
<tr>
<td></td>
<td>20. Edward types his bank account number.</td>
</tr>
<tr>
<td></td>
<td>21. The system confirms the sent car insurance enrollment.</td>
</tr>
</tbody>
</table>

Table 4: Use case for the traditional web application: accept a car insurance fee
<table>
<thead>
<tr>
<th>Use case name</th>
<th>Send a damage form (RIA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Edward: web application user</td>
</tr>
<tr>
<td>Assumptions</td>
<td>Started the web application</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Edward presses the &quot;Report damage&quot; button.</td>
</tr>
<tr>
<td></td>
<td>2. The system shows an animation of a changing form</td>
</tr>
<tr>
<td></td>
<td>3. Edward chooses the reason of the damage, he has clicked on the radio button of &quot;Accident&quot;</td>
</tr>
<tr>
<td></td>
<td>4. Edward fills in the date of the accident by choosing the &quot;15&quot; in day drop down menu, &quot;3&quot; in the month drop down menu and &quot;2009&quot; in the year drop down menu.</td>
</tr>
<tr>
<td></td>
<td>5. Edward types the car registration id of the involved car.</td>
</tr>
<tr>
<td></td>
<td>6. Edward chooses the radio button that indicates that it was his fault.</td>
</tr>
<tr>
<td></td>
<td>7. Edward types the name of the involved party.</td>
</tr>
<tr>
<td></td>
<td>8. Edward types the name of the insurance company of the involved party.</td>
</tr>
<tr>
<td></td>
<td>9. Edward wants his car to be repaired in the garage of his insurance company, he has clicked the &quot;Yes&quot; radio button</td>
</tr>
<tr>
<td></td>
<td>10. Edward sketches the accident by drag and drop the car that refers to his car and drag and drop the car that refers to the car of the involved party.</td>
</tr>
<tr>
<td></td>
<td>11. Edward presses on the &quot;Send&quot; button.</td>
</tr>
<tr>
<td></td>
<td>12. The system confirms the sent damage report.</td>
</tr>
</tbody>
</table>

Table 5: Use case for RIA: Send a damage form

<table>
<thead>
<tr>
<th>Use case name</th>
<th>Send a damage form (traditional web application)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Edward: web application user</td>
</tr>
<tr>
<td>Assumptions</td>
<td>Started the web application</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Edward presses the &quot;Report damage&quot; button.</td>
</tr>
<tr>
<td></td>
<td>2. The system shows an animation of a changing form</td>
</tr>
<tr>
<td></td>
<td>3. Edward chooses the reason of the damage, he has clicked on the radio button of &quot;Accident&quot;</td>
</tr>
<tr>
<td></td>
<td>4. Edward fills in the date of the accident by choosing the 15 in day drop down menu, 3 in the month drop down menu and 2009 in the year drop down menu.</td>
</tr>
<tr>
<td></td>
<td>5. Edward types the car registration id of the involved car.</td>
</tr>
<tr>
<td></td>
<td>6. Edward chooses the radio button that indicates that it was his fault.</td>
</tr>
<tr>
<td></td>
<td>7. Edward types the name of the involved party.</td>
</tr>
<tr>
<td></td>
<td>8. Edward types the name of the insurance company of the involved party.</td>
</tr>
<tr>
<td></td>
<td>9. Edward types the story how the accident is happened.</td>
</tr>
<tr>
<td></td>
<td>10. Edward chooses where he has damage to his car, he has chosen the front side of the car.</td>
</tr>
<tr>
<td></td>
<td>11. Edward wants that his car will be fixed at the garage of the car insurance company, he has clicked the &quot;Yes&quot; radio button.</td>
</tr>
<tr>
<td></td>
<td>12. Edward has pressed on the &quot;Send&quot; button.</td>
</tr>
<tr>
<td></td>
<td>13. The system confirms the sent damage report.</td>
</tr>
</tbody>
</table>

Table 6: Use case for traditional web application: Send a damage form

### 3.4 RIA

The RIA is built with JavaFX 1.0. The reason why this technology is chosen is because this technology provides many possibilities to build customized components. As mentioned before it is not needed to
be a designer to work with this technology. Other technologies like Adobe Flex require more designer’s skills. With this technology it is also possible to use Java classes to define the business logic.

3.4.1 Application structure of the RIA

This web application consists of different parts. As has been shown in figure 14, the RIA exists of three parts.

![Architecture of the RIA](image)

**Figure 14: Architecture of the RIA**

The RESTful (REpresentational State Transfer) [S.Tyagi 2006] web service is responsible for accessing the data of the back-end. The web services can simply be accessed by their URL’s. This way of accessing is chosen because the resources can be accessed in a simple way and all the HTTP methods like POST, GET can be performed.

The application exists of three packages.

![Packages of the RIA](image)

**Figure 15: Packages of the RIA**

To have a better overview of the different classes and their functionality we have chosen to separate the classes into different packages. The application contains three packages (figure 15).

- The carinsurance package contains classes that define the screens.
• The Data package contains classes that define the business logic and the communication with the back-end.

• The Components package contains classes that define customized components that are used in the GUI.

The Carinsurance package contains classes that are important for the GUI. In figure 16 the structure of the different classes in the GUI is presented. The classes of GUI are connected to each other according to a tree structure. This structure is chosen to simplify the overview of the used screens and they also could be extended very easy.

![Class diagram of the GUI]

Some of the classes in figure 16 rely on classes from the Data package and the Component package.

3.4.2 User Interface

The User Interface of the RIA differs from the User Interface of the traditional web application. In the RIA a Microsoft Vista/Apple metaphor is used. The reason why this metaphor is chosen is because the use of metaphor makes it easy for users to find their why in the application. Windows Vista and Apple are now very popular in use and are characterized by animations.

The buttons are chosen to be bigger than they used to be in traditional web applications because the elderly could have some limitations of movements. The buttons are decorated by icons. These Icons have to clarify the purpose of the button (figure 17).
In every screen the user can navigate back to the main menu. RIA’s still have problems with the "next" and "back" buttons on the web browsers; therefore we have implement these functionalities in places where the user might need them. (figure 18).

To provide help and instruction we have chosen to present this help as a video. These video recordings give feedback and notify the user if the text fields are not filled. According to [Galitz 2007] communication by speech is less complex then textual communication therefore we hope by applying video assistance to achieve an improvement of the interaction. In the next figures some of the screenshots of the traditional web applications are presented.
Figure 19: Screenshot of the main screen (RIA)

Figure 20: Screenshot of one of the calculate screens (RIA)
3.4.3 Development platform of the RIA

The development platform that is used to implement the web application is NetBeans 6.5. The reason why this IDE is chosen is because this was the only available IDE with a JavaFX 1.0 plug-in. This development platform has many advantages such as a large Java plug-in collection. Unfortunately because this was the first release of the JavaFX 1.0 it contained some errors, for example the real time verification of the methods and unexpected behavior of some variables.

One of the disadvantages of this IDE is the absence of a graphical tool to design the GUI. As said before JavaFX 1.0 does not have a layout manager. To layout all the element in NetBeans consumes a lot of time.

3.5 Traditional web application

The traditional web application is built in HTML and PHP on the server side. CSS is used to give the application a attractive look and feel. To build the web application we have first analyzed existing web applications of insurance companies like Central Beheer Achmea and FBTO.

In the next figure 22 the structure of the web applications is presented.

As can be seen the application structure is almost is the same as the structure of the RIA. The web page contains the HTML code for the forms and CSS. In the PHP server, the PHP code is executed in the PHP server. The business logic and the data transfer is done in the PHP server.
3.5.1 The user interface

For the traditional web application we have also added some features to have a better usability. Tooltips are used to provide information about the fields that have to be filled in. We have also added an indicator of the steps, to inform the user about which part of the process he is. In figure 23 you can see the indicator that refers to the first part of the four steps.

Because this web application represents existing insurance web application, we have decided to use only the attributes that are used for the existing web applications. For example the buttons, Text fields and the menu bars are standard. In the next figures some of the screenshots of the traditional web applications are presented.
Figure 24: Screenshot of the main screen (traditional web application)

Figure 25: Screenshot of one of the calculate screens (traditional web application)
Figure 26: Screenshot of the reporting damage screen (traditional web application)
4 Experiment

In this research an experiment has been done. As has been explained the Chapter 3 two different web applications have been implemented. These two web applications have to be evaluated by participants.

During this experiment, 26 participants of different ages have to perform two tasks and after these tasks (within subjects design) a questionnaire has to be filled in. During this experiment usability test software “Morae” is used to collect data such as the amount of clicks and the time spent on tasks. After this experiment has been done the collected data has to be analyzed with statistical methods.

In this part all the different subjects that are relevant to the experiment will be explained.

4.1 Users

The participants for the experiment are chosen in a way that they represent the actual users of a car insurance website. As mentioned before, the end users of the website could be all users that have a driving license and have or are considering to buy a car. The age of this large group has a range between 18 and 80.

The participants must not have disabilities that are essential for controlling the mouse and keyboard.

It is required for the participants to be familiar with using the computer and especially using the Internet.

For the purpose of this experiment 26 participants have been recruited to do the experiment. Most of the participants are recruited from the Bloodbank of Amsterdam (Sanquin). The recruited people did not receive a reward for their participation. This sample exists of twenty males and six females. The age of the participants deviate from 18 to 61 (Mean = 37.8 SD = 13.56)

4.2 Environment

The experiment was conducted in a lab environment. In figure 27 the constructed lab is shown.
The lab exists of two rooms. One room is used to conduct the test by the participant (figure 28) and the other room (figure 29) is used by the observer to see what the test participant is doing. The test participant must use the application on his own during the test and is not allowed to be disturbed by anybody.
The test organizer is not allowed to enter the room except if a fatal failure has occurred during the test although the chance was very low that a fatal error could happen. This is done to avoid bias during the test session.

The testing room must not contain objects that can distract the attention of the test participant. The testing room is therefore furnished like an office room with a desk, chair and a lamp for additional illumination if needed. During the test the door of the testing room is closed to not disturb the test participant.

4.3 Equipment

In this experiment some equipment are used to perform the test. For testing a laptop is used with T2400 1.83GHz Intel Centrino duo processor and 2 GB internal memory. The screen resolution that is used during the experiment is a 1024 x 768 pixel. This laptop has an optical mouse, but the touch pad can also be used if needed. To create the same condition for all participants, the use of the optical mouse is obligatory.

The observer uses a computer with 2.66 GHz Intel Pentium 4 and 768 MB internal memory. To connect the two machines a switch is used to make a small network.

In both machines a windows XP Professional operating system is used. The browser that is used for the test is FireFox 3.0. To record the testing session Morae 3.0 of Techsmith usability testing software is used. Morae is used to record and register the data like the time spent in a task, mouse clicks and errors.

4.4 Tasks

The participants have to perform two task for each web application. The web applications are chosen randomly to have a consistent data.
4.4.1 Questionnaire

The questionnaires exist of three parts (Appendix). The first part exists of personal questions. The next information is asked during this questionnaire:

- Gender
- Age
- Education level
- Usage of Internet

The second part of the questionnaire exists of questions about the web applications. The questions are meant to evaluate the opinion of the participant about the usability of the web applications. For these 29 questions a seven point Likert scaling is used. The last four questions are open-ended questions. Most of these questions are constructed with help of existing example questionnaires [Perlman, 2009].

The third part is meant to let the participant choose between the two tested web applications. The choice is based on:

- Pleasantness
- Ease of use
- Quickness
- Clarity

4.5 Procedure

For this experiment the organizer has tried to take each time three participants a day. The participants had to make an appointment to indicate when they were available. Most of the participants are transported by the organizer. The average duration of the test session including filling in the questionnaire ranges from 30 to 45 minutes. The experiment consists of several steps. First the participants get an introduction about the purpose and the goal of this experiment. After this introduction the participants get a manual to read (Appendix). In this manual the tasks that have to be performed are described. After reading the manual the participants are allowed to ask questions about the tasks that have to be performed.

To avoid biases the order of the web applications is changed randomly. First the participant has to fill in the personal questionnaire. This questionnaire exists of five questions to know some background of the participant. After filling in this questionnaire the participant is guided to the test room. The illumination is adjusted up according to the preferences of the participant. Morae Recorder is started to record the screen. This is an anonymous experiment; therefore the webcam recording is turned off.

The participant is asked to call the organizer when he is finished with the tasks that belong to one web application. This is needed to stop the recording of Morae. The participant has to fill in the questionnaire about the web application that is used to perform the tasks. This questionnaire exists of 33 questions. After filling in this questionnaire the Morae recorder is started to record the screen for the second web application and the second application is launched.

Finally the participant has to fill in the questionnaire about the second web application which is exactly the same as the questionnaire of the first web application. After this a general questionnaire about the two web application has to be filled in.
4.6 Pilot

Prior to the experiment a pilot was conducted. This pilot was needed to trace if the web applications contain errors. The web application contained some spelling errors and icons that were not understood. In the RIA some errors where found in the interaction between different screens. In the traditional web application some the validation of text fields did not work as they should have to work. All these errors are corrected in order to have a stable web application that does not influence the test results.

4.7 Statistics

In our experiment each participant is exposed to two different conditions. That is why our experiment is a within subject experiment. We can distinguish different variables that describe the characteristics of the participants and the situation. These variables can be divided into two groups; independent variables and dependent variables.

Independent variables describe the condition that is given to a group of participants [Morgan et al., 2004]. The dependent variables are meant to measure the effect of the independent variable.

The independent variable is the web application that is given to the participant. The condition that is given to the participants is RIA or not RIA (traditional web application). Each participants gets both conditions. The order or the condition has been randomly chosen to avoid bias.

The independent variables are:

- Effectivity
- Efficiency
- Learnability
- Error
- Satisfaction.
- Time
- Mouse clicks

To analyze the results we are going to use two different statistical analysis methods:

- GLM ANOVA repeated measure
- Chi square
5 Results

After conducting the experiment the data is collected in order to perform the statistical analysis. The analysis is done with help of SPSS In this part all the aspects that deal with data preparation and the analysis will be discussed.

5.1 Data preparation

From the experiment we have collected questionnaires data and screen recordings of Morae. Before performing the analysis this data has to be structured and converted in a numerical data.

5.1.1 Subjective data

The questionnaire exits of statements that have to be give a score by the participants and they also have the possibility to write their comments (Appendix).

These outputs have to be converted in suitable numerical values that can be used for statistical analysis. For the demographic information we have chosen to assign numbers to the different categorical choices. The Age is a continuous parameter therefore it was not needed to convert this parameter.

The 7 point Likert scale is converted into a score between 1 and 7. The statements that were filled as “not applicable” are left blank. This is done because assigning a much higher score could manipulate our results. Missing values have been replaces by a mean substitution of the score of all the participant for a particular web application.

After we have entered the data in SPSS, we have noticed that question 29, 30, 31, 32, 33, 34 contain too many blank values. Substituting these values with the mean may give a wrong interpretation of the data. Therefore we have excluded these statements for the analysis.

The statements are divided as follow to create internal groups:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effectivity</td>
<td>9,10,11</td>
</tr>
<tr>
<td>Efficiency</td>
<td>16,17,18,20</td>
</tr>
<tr>
<td>Learnability</td>
<td>13,14,15,19,21,35,36,37</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>12,22,23,24,25,26,27,28</td>
</tr>
</tbody>
</table>

Table 7: Usability variables and the corresponding questionnaire statements

The error variable is unfortunately not calculated because the corresponding statements are missing.

To check the reliability of the internal variables we have used the Cronbach’s alpha:

\[
\alpha = \frac{N \times r}{1 + (N-1) \times r}
\]

This alpha should be positive and near 0.7 to have a good grouping measurement.

Here below the tables of the Cronbach’s alpha are presented:
Effectivity

<table>
<thead>
<tr>
<th>Cronbach's Alpha</th>
<th>N of Items</th>
<th>Cronbach's Alpha if Item Deleted</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.693</td>
<td>3</td>
<td>Q9_RIA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.782</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q10_RIA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.425</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q11_RIA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.522</td>
</tr>
</tbody>
</table>

Table 8: Cronbach's alpha for effectivity for the RIA

Efficiency

<table>
<thead>
<tr>
<th>Cronbach's Alpha</th>
<th>N of Items</th>
<th>Cronbach's Alpha if Item Deleted</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.395</td>
<td>3</td>
<td>Q9_Tra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.718</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q10_Tra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.341</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q11_Tra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.119</td>
</tr>
</tbody>
</table>

Table 9: Cronbach's alpha for effectivity for the traditional web applications

In the table 8 and 9 we can see that the Cronbach’s alpha has not reached 0.7 yet. After deleting question 9 we can achieve a Cronbach’s alpha of 0.7. So for effectivity we are going to take the mean of the next questions:

- Question 10
- Question 11

Efficiency

<table>
<thead>
<tr>
<th>Cronbach's Alpha</th>
<th>N of Items</th>
<th>Cronbach's Alpha if Item Deleted</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.518</td>
<td>4</td>
<td>Q16_RIA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.552</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q17_RIA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.546</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q18_RIA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.461</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q20_RIA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.146</td>
</tr>
</tbody>
</table>

Table 10: Cronbach’s alpha for efficiency for the RIA

<table>
<thead>
<tr>
<th>Cronbach's Alpha</th>
<th>N of Items</th>
<th>Cronbach's Alpha if Item Deleted</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.662</td>
<td>4</td>
<td>Q16_Tra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.523</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q17_Tra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.420</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q18_Tra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.737</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q20_Tra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.573</td>
</tr>
</tbody>
</table>

Table 11: Cronbach’s alpha for efficiency for the traditional web application

The Cronbach’s alpha for efficiency is very low. For the questions of the RIA we see that there is a low improvement. For the traditional web application we can see that if we exclude question 18 we can reach a Cronbach’s alpha of 0.737 but this will this is not the case for the RIA. Therefore we will take all this values to calculate the efficiency.

57
Learnability

<table>
<thead>
<tr>
<th>Cronbach's Alpha</th>
<th>N of Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.845</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 12: Cronbach's alpha for learnability for the RIA

<table>
<thead>
<tr>
<th>Cronbach's Alpha if Item Deleted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q13_RIA</td>
</tr>
<tr>
<td>Q14_RIA</td>
</tr>
<tr>
<td>Q15_RIA</td>
</tr>
<tr>
<td>Q19_RIA</td>
</tr>
<tr>
<td>Q21_RIA</td>
</tr>
<tr>
<td>Q35_RIA</td>
</tr>
<tr>
<td>Q37_RIA</td>
</tr>
<tr>
<td>Q36_RIA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cronbach's Alpha</th>
<th>N of Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.679</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 13: Cronbach’s alpha for learnability for the traditional web application

For the learnability the Cronbach’s alpha is sufficient to take all the variables to calculate the mean of the learnability.
Satisfaction

<table>
<thead>
<tr>
<th>Cronbach’s Alpha</th>
<th>N of Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.792</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 14: Cronbach’s alpha for satisfaction for the RIA

<table>
<thead>
<tr>
<th>Cronbach’s Alpha</th>
<th>N of Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.807</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 15: Cronbach’s alpha for satisfaction for the traditional web application

For the satisfaction the Cronbach’s alpha is also sufficient. So we will use all the variables to calculate the satisfaction.

5.1.2 Objective data

During the experiment the screen was recorded by usability software Morae. This tool is very powerful for collecting information such as duration of the tasks and the amount of clicks that is needed to perform a task. Before the collection of this data can be performed the screen recordings has to be analyzed. In figure 30 an example analysis of the screen is shown.
As mentioned before, 26 participants have participated in the experiment. For every participant, two recordings have been saved. The recordings contain the tasks that belong to one website.

From Morae, we are going to use 4 tables:

- Time for task 1 in RIA (seconds)
- Time for task 2 in RIA (seconds)
- Time for task 1 in traditional web application (seconds)
- Time for task 2 in traditional web application (seconds)
- Mouse clicks for task 1 in RIA
- Mouse clicks for task 2 in RIA
- Mouse clicks for task 1 in traditional web application
- Mouse clicks for task 2 in traditional web application
5.2 Data exploration

In this section we will explore a part of the data that we have collected. The results of this data exploration consists of plots and short description of the data.

5.2.1 Age

The age of the participants differs from 18 to 61 years with a Mean = 37.4 and SD = 13.7. In figure 31 the distribution of the participants according to their age can be seen.

Figure 31: Histogram of age
5.2.2 Education

The distribution of the finished Dutch education level can be seen in figure 32. Five of the participants have finished a WO level (19.2%), thirteen have finished a HBO level (50%), seven have finished a MBO level (26.9) and one has finished a LBO level (3.8%).

Figure 32: Histogram Education
5.2.3 Purpose

From the histogram in figure 30 we can see most of the participants use the Internet for entertainment. Six of the participants use the Internet most for their work (23%), twelve of the participants use the Internet most for entertainment (46%), five use the Internet most for their work (19%) and 3 for something else (12%)
5.2.4 Internet usage

Most of the participants that use the Internet less than 3 hours a day. 8 use the Internet less than an hour a day (31%), 10 use the Internet between 1 and 3 hours a day (38%), 5 participants use the Internet between 3 and 6 hours a day (19%) and only 3 participants use the Internet more than 6 hours a day (12%).

Figure 34: Histogram of Internet usage
5.2.5 Time for calculating the insurance fee

In figure 32 a histogram of the time spent on calculating the insurance fee can be seen. 10 of the 26 (38.5\%) participants have spent more time with using the RIA and 16 of the 26 (60.5\%) have spent more time with the traditional web application.

For the RIA the fastest participant has calculated the insurance fee in 97 seconds and the slowest in 363 seconds (Mean = 182 and SD = 67)

The fastest participant has calculated the insurance fee in the traditional web application in 54 seconds and the slowest 409 seconds (Mean = 163 and SD = 84)
5.2.6 Time for reporting the damage

As we can see in figure 33 we have 10 out of 26 (38.5\%) participants that spent more time in the RIA and 16 out of 26 (60.5\%) spent more time in the traditional web application.

The fastest participant had done the task in the RIA in 67 seconds and the slowest in 324 (Mean = 154 and SD = 71)

For the traditional web application the fastest participant required 80 seconds to perform the task and the slowest 311 (Mean 166 and SD = 64)

Figure 36: Histogram time task 2
5.2.7 Clicks for calculating the insurance fee

17 of the 26 (65.4%, Mean = 43 and SD=8) participants have clicked more to calculate the insurance fee in the RIA and 9 out of 26 (34.6%, Mean = 40 and SD= 6) and had clicked more for the traditional web application.

Figure 37: Histogram clicks task 1
5.2.8 Clicks for reporting a damage

13 out of 26 (50\%, Mean = 23 and SD = 7) participants have used more mouse clicks to report a
damage in the RIA and 13 out of 26 (50\%, Mean 23 and SD = 3) used more mouse clicks for the
traditional web application.

Figure 38: Histogram clicks task 2
5.3 Statistical analysis

The data that we have collected can be divided into two groups:

- Subjective data
- Objective data

The data is analyzed using repeated measurements ANOVA (Analysis of Variance) for the within subject design. The confidence interval is chosen to be 0.95 and the significant level has to be p=0.05.

5.3.1 Subjective data

Effectivity

From the table we can see that there was not a significant main effect for effectivity F(1,25) = 1.18 with a p= 0.288

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>Mean square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web application</td>
<td>1</td>
<td>1.923</td>
<td>1.178</td>
<td>0.288</td>
</tr>
</tbody>
</table>

Table 16: Effect of the web application on effectivity (ANOVA)

Efficiency

From the table we can see that there was not a significant main effect for the efficiency F(1,25) = 0.698 with a p= 0.411.

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>Mean square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web application</td>
<td>1</td>
<td>0.347</td>
<td>0.698</td>
<td>0.411</td>
</tr>
</tbody>
</table>

Table 17: Effect of the web application on efficiency (ANOVA)

Learnability

For the learnability we can see that there was not a significant main effect F(1,25) = 0.351 with a p= 0.559.

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>Mean square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web application</td>
<td>1</td>
<td>0.019</td>
<td>0.351</td>
<td>0.559</td>
</tr>
</tbody>
</table>

Table 18: Effect of the web application on learnability (ANOVA)
Satisfaction

There is significant main effect for the satisfaction $F(1,25)= 4.91$ $p = 0.036$ and that is lower then 0.05. In figure 36 the estimated means of satisfaction for the two web applications is show.

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>Mean square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web application</td>
<td>1</td>
<td>1.349</td>
<td>4.914</td>
<td>0.036</td>
</tr>
</tbody>
</table>

Table 19: Effect of the web application on satisfaction (ANOVA)

![Estimated Marginal Means of Satisfaction](image)

Figure 39: Estimated marginal means of satisfaction
Pleasan tness

As we can in table 20, there is a significant effect for the pleasantness. \( \chi^2(1, N=26) = 0.00 \ p<0.05 \)

<table>
<thead>
<tr>
<th>CHI-Square</th>
<th>df</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pleasant</td>
<td>12,462</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 20: Effect of the web application on pleasantness (Chi-square)

Ease of use

There is a significant main effect for the ease of use, \( \chi^2(1, N=26) = 0.01 \ p<0.05 \)

<table>
<thead>
<tr>
<th>CHI-Square</th>
<th>df</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy of use</td>
<td>7,538</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 21: Effect of the web application on ease of use (Chi-square)

Quickness

There is not a significant effect for the quickness, \( \chi^2(1, N=26) = 0.43 \ p>0.05 \)

<table>
<thead>
<tr>
<th>CHI-Square</th>
<th>df</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy of use</td>
<td>0.615</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 22: Effect of the web application on quickness (Chi-square)

Clarity

For clarity the there is not a significant main effect \( \chi^2(1, N=26) = 0.12 \ p<0.05 \)

<table>
<thead>
<tr>
<th>CHI-Square</th>
<th>df</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy of use</td>
<td>2,462</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 23: Effect of the web application on clarity (Chi-square)

5.3.2 Objective data

Time for calculating the fee

In table 24 we can see that the \( F(1,25)= 0.249 \) and that means that the time for the first task is not significant.

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>Mean square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web application</td>
<td>1</td>
<td>4807.692</td>
<td>1.393</td>
<td>0.249</td>
</tr>
</tbody>
</table>

Table 24: Effect of the web application on time for the first task (ANOVA)
Time for reporting the damage

The time is also not significant for the second task. \( F(1,25) = 0.246 \).

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>Mean square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web application</td>
<td>1</td>
<td>2081.558</td>
<td>1.412</td>
<td>0.246</td>
</tr>
</tbody>
</table>

Table 25: Effect of the web application on time for the second task (ANOVA)

Mouse clicks for calculating the fee

Mouse clicks for calculating the fee \( F(1,25) = 0.114 \) and that means that this variable is also not significant.

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>Mean square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web application</td>
<td>1</td>
<td>111.077</td>
<td>2.681</td>
<td>0.114</td>
</tr>
</tbody>
</table>

Table 26: Effect of the web application on mouse clicks for the first task (ANOVA)

Mouse clicks for reporting the damage

The mouse clicks for the second task are also not significant. \( F(1,25) = 0.791 \).

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>Mean square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web application</td>
<td>1</td>
<td>1.231</td>
<td>0.071</td>
<td>0.791</td>
</tr>
</tbody>
</table>

Table 27: Effect of the web application on mouse clicks for the second task (ANOVA)

Time and tasks

The \( P \) value the time if we distinguish web applications and the tasks is 0.009. This means that time is significant for the web applications and the task that is performed.

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>Mean square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web application * task</td>
<td>1</td>
<td>6608.087</td>
<td>7.971</td>
<td>0.009</td>
</tr>
</tbody>
</table>

Table 28: Effect of the web application and task on time (ANOVA)
Figure 40: Estimated marginal means of time and tasks

**Mouse clicks and tasks**

The P value the time if we distinguish web applications and the tasks is 0.248. This P value is higher then 0.05. This means that this variable is not significant.

<table>
<thead>
<tr>
<th>Source</th>
<th>df</th>
<th>Mean square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web application * task</td>
<td>1</td>
<td>44,462</td>
<td>1,402</td>
<td>0.248</td>
</tr>
</tbody>
</table>

Table 29: Effect of the web application and tasks on mouse clicks (ANOVA)
5.4 Participants comments

Most participants have given some comments about the web applications. There are 4 kind of comments that the users have to give; positive aspects of the RIA; negative aspects of the RIA, positive aspects of the traditional web application, negative aspects of the traditional web application. Overall we can say that most users comments were positive.

<table>
<thead>
<tr>
<th>Participants</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Easy to use</td>
</tr>
<tr>
<td>2</td>
<td>Nice animations</td>
</tr>
<tr>
<td>5</td>
<td>Nice design</td>
</tr>
<tr>
<td>5</td>
<td>Good overview</td>
</tr>
<tr>
<td>2</td>
<td>Fun to use</td>
</tr>
<tr>
<td>1</td>
<td>Functional</td>
</tr>
</tbody>
</table>

Table 30: Participants comments: positive aspects of the RIA

<table>
<thead>
<tr>
<th>Participants</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Entry of data took too much time</td>
</tr>
<tr>
<td>1</td>
<td>The information that is been asked is not logical</td>
</tr>
<tr>
<td>1</td>
<td>The animation consumes time</td>
</tr>
<tr>
<td>1</td>
<td>Can’t find the button</td>
</tr>
<tr>
<td>1</td>
<td>Difficult structure</td>
</tr>
<tr>
<td>2</td>
<td>Boring colors used in the interface</td>
</tr>
<tr>
<td>2</td>
<td>Slow to use</td>
</tr>
</tbody>
</table>

Table 31: Participants comments: negative aspects of the RIA

<table>
<thead>
<tr>
<th>Participants</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Clear overview</td>
</tr>
<tr>
<td>1</td>
<td>Easy to navigate</td>
</tr>
<tr>
<td>2</td>
<td>Simple</td>
</tr>
<tr>
<td>1</td>
<td>Attractive colors</td>
</tr>
<tr>
<td>5</td>
<td>Easy to use</td>
</tr>
<tr>
<td>1</td>
<td>Functional</td>
</tr>
<tr>
<td>2</td>
<td>Response fast</td>
</tr>
</tbody>
</table>

Table 32: Participants comments: positive aspects of the traditional web application

<table>
<thead>
<tr>
<th>Participants</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Entry of the situation of the accident is difficult and time consuming</td>
</tr>
<tr>
<td>1</td>
<td>The information that is been asked is not logical</td>
</tr>
<tr>
<td>1</td>
<td>Displaying of new combo boxes in the calculating section is annoying</td>
</tr>
<tr>
<td>4</td>
<td>Boring, simple</td>
</tr>
<tr>
<td>1</td>
<td>More explanation is needed</td>
</tr>
</tbody>
</table>

Table 33: Participants comments: negative aspects of the traditional web application

74
<table>
<thead>
<tr>
<th>Participants</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The animation must be played without pressing on the &quot;next&quot; button</td>
</tr>
<tr>
<td>1</td>
<td>The &quot;send&quot; button has to be improved because it is not clear.</td>
</tr>
<tr>
<td>1</td>
<td>The speed the interaction has to be improved</td>
</tr>
<tr>
<td>1</td>
<td>More explanation is needed to the drag and drop area.</td>
</tr>
<tr>
<td>1</td>
<td>Video feedback about the errors must be deleted.</td>
</tr>
<tr>
<td>1</td>
<td>Use more attractive colors</td>
</tr>
</tbody>
</table>

Table 34: Participants comments: recommendation for the RIA

<table>
<thead>
<tr>
<th>Participants</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Change the entry of the accident</td>
</tr>
<tr>
<td>3</td>
<td>The User Interface has to be more attractive</td>
</tr>
<tr>
<td>1</td>
<td>Make a connection with the RDW webservice to find the car information</td>
</tr>
<tr>
<td>1</td>
<td>It has to be more fun to use</td>
</tr>
</tbody>
</table>

Table 35: Participants comments: recommendation for the traditional web application
6 Discussion

In Chapter 5 the results are presented. In this Chapter these results will be discussed. We will start with the influence of the different web applications on the user satisfaction. From the results a statistically significant preference for the RIA is shown. In the comments the users indicated that the RIA has a nicely designed User Interface. The users appreciate the attractive look and feel of the RIA. This finding can be related to the significance of the pleasant variable for the RIA. There may be a relation between pleasure that the participants have while using the web application and the user satisfaction.

From the results we can also see that the tasks combined with web applications have a significant influence on the time that is spent to complete the tasks. For the first task where the participants have to calculate the insurance fee, we can see that the mean of the RIA is higher than the traditional web application; for the second task the opposite is the case. Therefore, we can conclude that the RIA does not perform better in all cases. The performance of the RIA is context dependent. The animations took the users more time to complete the first task and slowed down the users. In the participants comments we can see that some participants were not happy with the duration of the animations. We can therefore conclude that the application of animations is time consuming and must only be applied in parts where the user will not be interrupted. Another factor which can have impact on the time that is spent on the tasks, is the need to get familiar with the used interaction style. Video and voice can attract the attention of the user.

In the second task where the participants have to report a damage, a drag and drop interaction style was applied. This application has saved the users more time to achieve the same goal as the traditional web application. During the observation, it was clear that the testers spent more time in the traditional web application because most users had some difficulties with typing a story about how the accident was happened. By applying the drag and drop components to draw the situation, we could achieve a faster and enjoyable accomplishment of the task. In the participants comments we can also see that three participants were not satisfied with the entry of the story by typing in the traditional web application.

The web applications have also an influence on the ease of use. There is a statistically significant influence on the easy of use. Nine of the participants have indicated that RIA was easy to use in the comments. The expectation was that the traditional web application should be easy to use for the users because most users are familiar with the use of Internet and they all have experience with the traditional web applications therefore they should have created a mental model about the traditional web applications. We think that the application of the video feedback and the drag and drop interaction style has contributed in the easy of use of RIA. So we can say that the efficiency is significant if the RIA functionalities are applied in places where the user can have benefit of it.

The effectivity was not significant for the web applications, because all the participants have managed to complete the tasks without difficulties. We think that the tasks were easy to perform and the instructions were clear enough. Another reason could be that all the web applications were clear and have a good structure. In the participants comments we can see that s. five participants have indicated that the RIA has a good overview and nine participants have indicated that the traditional web application has clear overview.

The learnability was also not significant for the web applications. The objective learnability could not be measured because we did not repeat the same measures for the same tasks. In the subjective learnability there was no difference between the two web applications. The two web applications were easy to work with and did not require too much time to get used with them. It is true that most of the participants have indicated that the RIA was easier in use, but that does not mean that the traditional web application was not easy to use.
The users were positive about the two web applications. Most of the participants have given a high rating to the two web applications, this has caused a small difference between the two web applications.

6.1 Limitations

In this study only 26 participants are involved. This amount is very limited to have a convincing conclusion. For a further research we recommend much more participants. The sample set of the participant must contain equal amount of female and men and the age must also be normal distributed.

Further more the design of the web applications that have been evaluated during the experiment are designed and implemented by the author of this thesis. The developer did not have a lot of experience in developing or designing professional web applications. Therefore the used application can be influenced by the lack of skills of the developer.

The tasks that are used for the experiment are simple and do not cover all the functionalities of a car insurance web application.
7 Conclusion and recommendation

The goal of this study is to evaluate the usability of the Rich Internet Application in order to convince finance and insurance companies of the benefits and profits of the new technology in order to switch to it. The main question that has to be answered is:

*Are RIA’s more usable than traditional Internet applications?*

We can say that we have managed to answer this question. First we have defined usability in five attributes; effectiveness, efficiency, learnability, satisfaction and error rate.

In order to answer this question we have built two not fully functional web applications as a proof of concept. The RIA is built with JavaFX and the traditional web application is built with PHP and HTML. This web applications are tested in an experiment by 26 volunteers. Most of the 26 participants that have participated in our experiment were quite positive about the RIA and could all manage to complete the tasks for both web applications.

The data that is collected during this experiment is used to see if there is a significant different between the usability variables for the RIA and the traditional web application. For the effectiveness, learnability and we could not see a significant different between the RIA and the not RIA.

The efficiency in terms of time spent for the tasks is significant. This means that engineer has to think carefully were and how to apply the RIA functionality in order to achieve a better efficiency. Using animations can slow down the user, but animations can deliver more information and help to the user. The drag and drop can be useful and can save more time if it is applied in the correct way.

The satisfaction is significant for the web applications. The participants were satisfied about the RIA. They enjoyed working with the RIA and it was easier to use than the traditional web application. The sophisticated User Interface and the animations have contributed in the satisfaction of the participants.

Now we have summed the key findings of the experiment we can conclude that the RIA is usable then that traditional Internet application. The users are more satisfied and the efficiency can be improved it the RIA functionalities are used correctly. Insurance and financial companies are advised to switch to this new technology in order to persuade the consumers about their products.

We think that there is more research is needed in this domain. For a further research the usability can be investigated between elderly and young people or between women and man. RIA’s can be built in a way that the User Interface can adopt to personal preferences. For Insurance and finance company’s it is very useful to know the preference of the target costumers to be able to build a web application that satisfy the need of the costumers.


P. van Nieuwenhuizen. 3d computer animation techniques, 2005. Lecture slides, Technical University Delft, MKE.

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Research construction</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>The different architectures of RIA [Moritz, 2008]</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>The left model represents the communication model of the traditional web application and the right model represents the communication model of the RIA.</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>The communication model of AJAX based web application [Kumar, 2008]</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>Example of an AJAX web application [Isomorphic Software, 2008]</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>Adobe technologies for building RIA [Adobe, 2008]</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>Code example of Flex 2.0</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>Example of Adobe Flex web application [Volkswagen, 2008]</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>Overview of the JavaFX framework</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>Code example of JavaFX</td>
<td>19</td>
</tr>
<tr>
<td>11</td>
<td>Example of JavaFX web application [Campbell, 2008]</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>Acceptability tree [Nielsen, 1993]</td>
<td>23</td>
</tr>
<tr>
<td>13</td>
<td>Example of HTA [Shneiderman and Plaisant, 2007]</td>
<td>32</td>
</tr>
<tr>
<td>14</td>
<td>Architecture of the RIA</td>
<td>43</td>
</tr>
<tr>
<td>15</td>
<td>Packages of the RIA</td>
<td>43</td>
</tr>
<tr>
<td>16</td>
<td>Class diagram of the GUI</td>
<td>44</td>
</tr>
<tr>
<td>17</td>
<td>Icons of the RIA</td>
<td>45</td>
</tr>
<tr>
<td>18</td>
<td>Navigation icons of the RIA</td>
<td>45</td>
</tr>
<tr>
<td>19</td>
<td>Screenshot of the main screen (RIA)</td>
<td>46</td>
</tr>
<tr>
<td>20</td>
<td>Screenshot of one of the calculate screens (RIA)</td>
<td>46</td>
</tr>
<tr>
<td>21</td>
<td>Screenshot of the reporting damage screen (RIA)</td>
<td>47</td>
</tr>
<tr>
<td>22</td>
<td>Architecture of the traditional web application</td>
<td>48</td>
</tr>
<tr>
<td>23</td>
<td>Indicator of the steps</td>
<td>48</td>
</tr>
<tr>
<td>24</td>
<td>Screenshot of the main screen (traditional web application)</td>
<td>49</td>
</tr>
<tr>
<td>25</td>
<td>Screenshot of one of the calculate screens (traditional web application)</td>
<td>49</td>
</tr>
<tr>
<td>26</td>
<td>Screenshot of the reporting damage screen (traditional web application)</td>
<td>50</td>
</tr>
<tr>
<td>27</td>
<td>Experiment lab</td>
<td>52</td>
</tr>
<tr>
<td>28</td>
<td>Test room</td>
<td>52</td>
</tr>
<tr>
<td>29</td>
<td>Observer room</td>
<td>33</td>
</tr>
<tr>
<td>30</td>
<td>Morae analysis example</td>
<td>60</td>
</tr>
<tr>
<td>31</td>
<td>Histogram of age</td>
<td>61</td>
</tr>
<tr>
<td>32</td>
<td>Histogram Education</td>
<td>62</td>
</tr>
<tr>
<td>33</td>
<td>Histogram purpose</td>
<td>63</td>
</tr>
<tr>
<td>34</td>
<td>Histogram of Internet usage</td>
<td>64</td>
</tr>
<tr>
<td>35</td>
<td>Histogram time task 1</td>
<td>65</td>
</tr>
<tr>
<td>36</td>
<td>Histogram time task 2</td>
<td>66</td>
</tr>
<tr>
<td>37</td>
<td>Histogram clicks task 1</td>
<td>67</td>
</tr>
<tr>
<td>38</td>
<td>Histogram clicks task 2</td>
<td>68</td>
</tr>
<tr>
<td>39</td>
<td>Estimated marginal means of satisfaction</td>
<td>70</td>
</tr>
<tr>
<td>40</td>
<td>Estimated marginal means of time and tasks</td>
<td>73</td>
</tr>
</tbody>
</table>
List of Tables

1. Use case for the RIA: calculate a car insurance fee .......................................................... 38
2. Use case for the traditional web application: calculating a car insurance fee ......................... 39
3. Use case for the RIA: accept a car insurance fee ................................................................. 40
4. Use case for the traditional web application: accept a car insurance fee ................................. 41
5. Use case for RIA: Send a damage form ............................................................................... 42
6. Use case for traditional web application: Send a damage form .............................................. 42
7. Usability variables and the corresponding questionnaire statements ....................................... 56
8. Cronbach's alpha for effectiveness for the RIA ....................................................................... 57
9. Cronbach's alpha for effectiveness for the traditional web applications ................................. 57
10. Cronbach's alpha for efficiency for the RIA ........................................................................... 57
11. Cronbach's alpha for efficiency for the traditional web application ....................................... 57
12. Cronbach's alpha for learnability for the RIA ....................................................................... 58
13. Cronbach's alpha for learnability for the traditional web application .................................... 58
14. Cronbach's alpha for satisfaction for the RIA ....................................................................... 59
15. Cronbach's alpha for satisfaction for the traditional web application .................................... 59
16. Effect of the web application on effectiveness (ANOVA) ....................................................... 69
17. Effect of the web application on efficiency (ANOVA) ............................................................. 69
18. Effect of the web application on learnability (ANOVA) ........................................................... 69
19. Effect of the web application on satisfaction (ANOVA) ........................................................... 70
20. Effect of the web application on pleasantness (Chi-square) .................................................... 71
21. Effect of the web application on ease of use (Chi-square) ..................................................... 71
22. Effect of the web application on quickness (Chi-square) ......................................................... 71
23. Effect of the web application on clarity (Chi-square) .............................................................. 71
24. Effect of the web application on time for the first task (ANOVA) ........................................... 71
25. Effect of the web application on time for the second task (ANOVA) ....................................... 72
26. Effect of the web application on mouse clicks for the first task (ANOVA) ............................... 72
27. Effect of the web application on mouse clicks for the second task (ANOVA) ............................. 72
28. Effect of the web application and task on time (ANOVA) ....................................................... 72
29. Effect of the web application and tasks on mouse clicks (ANOVA) ......................................... 73
30. Participants comments: positive aspects of the RIA ............................................................... 74
31. Participants comments: negative aspects of the RIA .............................................................. 74
32. Participants comments: positive aspects of the traditional web application ............................ 74
33. Participants comments: negative aspects of the traditional web application ........................... 74
34. Participants comments: recommendation for the RIA ............................................................ 75
35. Participants comments: recommendation for the traditional web application ......................... 75
8 Abbreviations

AIR: Adobe Integrated Runtime
API: Application Program Interface
ANOVA: ANalysis Of Variance
CSS: Cascading Style Sheet
DOM: Document Object Model
GUI: Graphical User Interface
HTML: Hyper Text Markup Language
JRE: Java Runtime environment
MXML: Magic eXtensible Markup Language
NLJUG: Nederlandse Java User Group
PHP: Hypertext Preprocessor
RIA: Rich Internet application
SUMI: Software Usability Measurement Inventory
UCD: User Centered Design
UE: User Engineer
WAMMI: Website Analysis and Measurement Inventory
SD: Standard deviation
9 Appendix

Experiment manual

Bruikbaarheid van autoverzekering websites
Enquête

Bedankt dat u deel wilt nemen aan dit onderzoek. Dit onderzoek wordt uitgevoerd in het kader van mijn Master opleiding aan de TU-Delft.

Het doel van dit onderzoek is om de gebruikersvriendelijkheid van twee verschillende autoverzekering websites te peilen. Deze websites zijn uitsluitend voor dit onderzoek ontwikkeld.

Deze vragenlijst is compleet anoniem en zal alleen gebruikt worden voor dit onderzoek.

Op pagina twee vindt u de taken die uitgevoerd dienen te worden. Het is de bedoeling dat u in totaal vier taken uitvoert, twee taken met website A en twee taken met website B.

De vragenlijst bestaat dan ook uit twee onderdelen. Het eerste onderdeel dient ingevuld te worden na het uitvoeren van de taken horende bij website A. Het andere onderdeel dient ingevuld te worden na het uitvoeren van de taken horende bij website B.
Taken

Hieronder zal een globale beschrijving gegeven worden van de taken die uitgevoerd dienen te worden. Na het uitvoeren van elk van beide onderstaande taken dient er een vragenlijst ingevuld te worden.

Lees de onderstaande twee taken zorgvuldig en als er vragen zijn over de taken dan kunt u deze gerust stellen.

**Taak 1: Berekenen van de auto verzekering**

U heeft vandaag een auto gekocht en bent benieuwd naar de kosten van de autoverzekering. De auto die u heeft gekocht is een **Renault** van model **Clio** met het bouwjaar **2001**. Het is een **handgeschakelde** auto met een **1.2 Benzine** motor.

U wilt een **WA verzekering** aansluiten met een **inzittende verzekering** en **geen rechtsbijstand verzekering**. U heeft minder dan **1000 euro** aan accessoires. U heeft vorige jaar een auto gehad dus heeft u **1 schadevrij jaar**.

**Taak 2: Melden van een schade**

Gisteravond bent u betrokken geweest bij een auto ongeluk waarbij uw auto flink is beschadigd. Om de procedure te versnellen voor de verzekering maatschappij, heeft u besloten om de schade online te melden.

Het ongeluk is veroorzaakt door **u zelf**. U bent **frontaal gebotst** tegen de andere partij. U heeft dus schade aan de voorzijde van uw auto en de tegenpartij heeft schade aan de achterzijde.

U heeft **gisteravond** een schadeformulier ingevuld met de tegenpartij. Daarop staan de gegevens van de tegenpartij. De naam van de tegenpartij is **dhr. R. Jansen** en het kenteken van de auto van de tegenpartij is **29-ZWV-6**. Hij is verzekerd bij **OHRA**. U wilt de schade aan uw auto bij uw eigen dealer laten maken, dus **niet bij de garage van de verzekering maatschappij**.

Voer de twee bovenstaande taken uit en vul daarna het enquête onderdeel dat bij deze website hoort.
Onderstaande vragen zijn bedoeld voor achtergrondinformatie.

Zet een kruis tussen de accolades van het juiste antwoord.

1 Wat is uw geslacht?
[ ] Man
[ ] Vrouw

2 Wat is uw leeftijd? (vul uw leeftijd op de stippelijn)
...... Jaar

3 Wat is uw hoogst afgeronde opleiding?
[ ] Wetenschappelijke onderwijs
[ ] Hoger beroeps onderwijs
[ ] Middelbare beroeps onderwijs
[ ] Lager beroeps onderwijs
[ ] Lagere school

4 Voor welke doeleinden gebruikt u het internet het meest?
[ ] Werk gerelateerde activiteiten
[ ] Ontspanning en hobby's
[ ] Studie
[ ] Overig

5 Hoeveel tijd zit u gemiddeld aan het internet per dag?
[ ] Minder dan 1 uur
[ ] Tussen de 1 uur en 3 uur
[ ] Tussen de 3 en 6 uur
[ ] Meer dan 6 uur
Vragen over website B (HTML based application)

Deze vragen zijn bedoeld om uw mening over de website te krijgen.

Hieronder treft u verschillende stellingen. Geef bij elke stelling aan door het invullen van een cirkel, in welke mate u het met de stelling eens bent. De cirkels zijn gerangschikt in de volgende volgorde:

1 = Helemaal mee eens
2 = Mee eens
3 = Een beetje mee eens
4 = Neutraal
5 = Een beetje mee eens
6 = Mee eens
7 = Helemaal mee eens

Als de stelling niet toepasbaar is voor de geteste website of als u het antwoord niet weet, dan kunt u de cirkel invullen in de laatste klokm.

<table>
<thead>
<tr>
<th></th>
<th>Helemaal mee eens</th>
<th>Neutraal</th>
<th>Helemaal mee eens</th>
<th>Niet van toepassing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ik had moeite met het uitvoeren van deze taken.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Het gebruik van deze website maakt het makkelijk voor mij om een verzekering af te sluiten.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Het gebruik van deze website maakt het makkelijk voor mij om een schade te melden.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Deze site voldoet aan de wensen die ik zou willen zien in een autoverzekering website.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Ik vind de website makkelijk in gebruik.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Ik weet wat ik moet invullen in de website.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Ik begrijp de informatie die de website toont.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Het kost mij veel tijd om een verzekering af te sluiten.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Het kost mij veel tijd om een schade te melden.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>De website reageert snel op mijn handelingen.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>De website is duidelijk.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Zin</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Het invoeren van de gegevens kost veel tijd.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>De structuur van de website is duidelijk</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ik ben tevreden over de website</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ik zal de website aanraden aan een vriend</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deze website is mooi</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deze website is leuk om te gebruiken</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deze website werkt op de manier hoe ik het zou willen</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deze website is frusterend</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ik heb het gevoel dat ik weet wat de website doet</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ik vind het gebruik van animaties leuk</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>De foutmeldingen zijn duidelijk</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foutmeldingen door middel van filmpjes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ik kan makkelijk terugkeren na een gemaakte fout</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>De website biedt genoeg hulpinstructies voor het invoeren van de benodigde data.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Over het algemeen ben ik tevreden over de aangeboden hulp in de website</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>De structuur van de website is logisch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Het leren gebruiken van de website is makkelijk</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ik denk dat ik over één jaar nog steeds zal kunnen werken met deze website</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Wat is/zijn de positieve aspecten van deze website?

---

Wat is/zijn de negatieve aspecten van deze website?

---

90
<table>
<thead>
<tr>
<th>Heeft u nog aanbevelingen om de website te verbeteren?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heeft u nog opmerkingen?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
**Vragen over website A (Rich Internet application)**

Deze vragen zijn bedoeld om uw mening over de website te krijgen.

Hieronder treft u verschillende stellingen. Geef bij elke stelling aan door het invullen van een cirkel, in welke mate u het met de stelling eens bent. De cirkels zijn gerangschikt in de volgende volgorde:

1 = Helemaal mee eens
2 = Mee eens
3 = Een beetje mee eens
4 = Neutraal
5 = Een beetje mee eens
6 = Mee eens
7 = Helemaal mee eens

Als de stelling niet toepasbaar is voor de geteste website of als u het antwoord niet weet. Dan kunt u de cirkel invullen in de laatste krom.

<table>
<thead>
<tr>
<th></th>
<th>Helemaal mee eens</th>
<th>Neutraal</th>
<th>Helemaal mee eens</th>
<th>Niet van toepassing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ik had moeite met het uitvoeren van deze taken.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Het gebruik van deze website maakt het makkelijk voor mij om een verzekering af te sluiten.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Het gebruik van deze website maakt het makkelijk voor mij om een schade te melden.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Deze site voldoet aan de wensen die ik zou willen zien in een autoverzekering website.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Ik vind de website makkelijk in gebruik.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Ik weet wat ik moet invullen in de website.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Ik begrijp de informatie die de website toont.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Het kost mij veel tijd om een verzekering af te sluiten.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Het kost mij veel tijd om een schade te melden.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>De website reageert snel op mijn handelingen.</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>
Algemene vragen over beide websites

Welke website vond u het prettigst om mee te werken?
[ ] Website A (RIA)
[ ] Website B (traditionele website)

Waarom heeft u voor deze website gekozen?

Welke website vond u het makkelijkst om mee te werken?
[ ] Website A (RIA)
[ ] Website B (traditionele website)

Waarom heeft u voor deze website gekozen?

Welke website vond u het snelst om mee te werken?
[ ] Website A (RIA)
[ ] Website B (traditionele website)

Waarom heeft u voor deze website gekozen?

Welke website vond u het duidelijkst om mee te werken?
[ ] Website A (RIA)
[ ] Website B (traditionele website)

Waarom heeft u voor deze website gekozen?
<table>
<thead>
<tr>
<th>Heeft u nog aanbevelingen om de website te verbeteren?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heeft u nog opmerkingen?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
Algemene vragen over beide websites

<table>
<thead>
<tr>
<th>Welke website vond u het prettigst om mee te werken?</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ] Website A (RIA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ ] Website B (traditionele website)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Waarom heeft u voor deze website gekozen?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Welke website vond u het makkelijkst om mee te werken?</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ] Website A (RIA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ ] Website B (traditionele website)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Waarom heeft u voor deze website gekozen?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Welke website vond u het snelst om mee te werken?</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ] Website A (RIA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ ] Website B (traditionele website)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Waarom heeft u voor deze website gekozen?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Welke website vond u het duidelijkst om mee te werken?</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ] Website A (RIA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ ] Website B (traditionele website)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Waarom heeft u voor deze website gekozen?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>