Automated Usage Tracing and Analysis: a comparison with web survey

Master of Science Thesis in Software Engineering

MIKAEL BOLLE
EMIL BACKLUND

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Gothenburg, Sweden, September 2013

Automated Usage Tracing and Analysis: a comparison with web survey
MIKAEL BOLLE
EMIL BACKLUND

Examiner: MIROSLAW STARON

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Gothenburg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Gothenburg, Sweden September 2013

Abstract

A challenge in taking decisions on how to improve a software product is to gain knowledge on how end-users interact with it. One way of getting this knowledge is by asking them through a web survey. Another approach is based on tracing what the users do and then run an analysis on that gathered data. This paper presents an approach called Automatic Usage Tracing and Analysis (AUTA) to automatically gather usage data for a software system with Aspect-Oriented Programming (AOP) and analyzes the gathered data through data mining. A brainstorming workshop with the developers of the software system was used to define a set of questions that the data mining should answer. The questions were implemented in AUTA and a web survey was conducted to compare the two methods. The comparison showed that there is a resemblance between AUTA and a conducted web survey. However, the resemblance is not strong enough to conclude with certainty that AUTA can replace the use of web surveys. It was also discovered that some questions identified in the workshop were not possible to be answered with a web survey but could be answered with AUTA. The recommendation is therefore that AUTA and web surveys are used as complement to each other.

Keywords: AOP, Aspect-Oriented Programming, Data mining, Web survey, Correlation, Usage tracing

Acknowledgements

First of all we would like to thank Matthias Tichy our academic supervisors and Carl Stein our industrial supervisor, they have both been a great support and guidance through out the process.

We would also like to thank ATEA Global Services for providing the possibility to conduct this study and to the employees' whom gave valuable input during the brainstorming workshop and took their time to answer our questions when needed. A big thanks is also sent out to all participants which were test subjects in the task workshop

Finally, we would like to thank our friends and family for their support, this study would not have been possible without them.

Mikael Bolle & Emil Backlund, Göteborg 12/08/13

Glossary

Advice Advices are the actions that will be executed when a Join-point is reached. 11–14, 39, 41, 43–45

AOP Aspect-Oriented Programming. 1–4, 8, 11–13, 37, 39, 79, 80, 88

Aspect A stand-alone module or class for implementing cross-cutting concerns. iv, 11, 12, 14, 39–45

Aspect language The programming language for implementing aspects. 11

Aspect weaver Aspect weaver is used to combine the aspect- and component-language. 12

AUTA Automated Usage Tracing and Analysis. 2, 3, 8, 9, 22, 37, 46, 79, 80, 88, 89

Bounce rate A bounce is considered as when a user navigates to a service and then navigates to another service in just a matter of second. The bounce rate is a value representing how often this kind of navigation happens for a service, it represent the action of bouncing in and out of a pages. 27, 29, 54, 64, 67, 69, 73, 77, 84

CA Cluster Analysis. 15

Component language The programming language for implementing the main concern of the system. Can be both a procedural- or object-oriented-language, in this study it is an object-oriented language. 11, 12

Join-point Is a point in the execution where aspects are coordinated with the Component language. 11, 12

Pointcut Pointcuts define which Join-Points an aspects should be applied to. 11, 14, 37, 41, 42, 44, 45, 61, 79, 80

PostSharp Is an Aspect-Oriented Framework for .NET. 8, 14, 39

Proxy-pattern A design pattern where the interaction with a class is passed through a proxy which implements the same interface. 13, 39, 79

SUMMON Chalmers library search system. 6

UI User interface. 1, 26, 27, 38, 40

UserControl Class responsible for the presentation of a web page. 37, 39

UX User experience design. 22, 26

Contents

List of Figures

Listings

1 Introduction

For the delivery of an appreciated software product user satisfaction is key. One crucial part for achieving high user satisfaction is to know where user interaction issues arise. Therefore, software development organizations must, in order to continuously reach high satisfaction, strive to continuously learn about the users' experience of the product. However, these organizations may not always have close contact with their end-users and it might require costly and complex procedures to gather and extract in-depth knowledge about the use of their product.

An approach that is relatively cheap for such situation is the use of surveys. Surveys, and especially web surveys, can gather a vast amount of data in a short amount of time and do not require the development organization to meet with the end-users [1]. Nevertheless, conducting a survey might be a time consuming task that is difficult to execute and get accurate data from. Also the required knowledge to conduct a successful survey [2] might not be present within the software development organization.

By automatically gathering data about users' interaction within the system, it might be possible for organizations to replace the use of surveys. For data collection in general Aspect-Oriented Programming (AOP), which is an approach to keep separation of concerns when implementing cross-cutting concerns, has been suggested as a good approach [3]. Some research has been conducted on automated data collection in the context of user interaction. Some papers in this research have presented implementations of automated usability tracing and evaluation using AOP [4, 5, 6]. However, these papers focus on data collection and apply the theory on small scale systems with less than tens of thousands lines of code. And, their implementations have only evaluated the feasibility of using AOP.

Extracted data, from the AOP implementation, needs to be analyzed in order for development organizations to make decisions on how to evolve their product. This type of analysis, data mining, is an area that is very well researched and used in many different applications. Data mining has for example been used for targeting in marketing management, fraud detection and for stock market forecasting [7].

Van der Shuur and Jansen has presented a software quality improvement solution by combining data gathering and data analysis. However, they are concerned with the overall usage of the system and not how parts of the User interface (UI) are being utilized. Their solution collects data from the service layer, which means that it does not have access to as data available in the presentation layer.

This paper presents an automated analysis approach, using AOP and data mining, that provide insights into how users are interacting with a software product. The purpose of the automated analysis is, in this case, to acquire information that can be used as a basis for strategically decisions when developing a software product further. The approach is applied to a large-scale enterprise web product, containing thousands of lines of code. In order to verify this approach an evaluation between data from the automated analysis and a web survey is conducted. The results of the evaluation suggest that the solution can not completely replace a survey but that there is a resemblance. Furthermore, it is discovered that the solution can gather more knowledge than what is possible with a survey.

1.1 Problem Statement

Due to challenges that arise when gathering data from end-users through surveys it might be easier to execute an automated approach. If it is proven that this kind of approach can be used with same or better result than a survey but with fewer resources, then it might yield substantial benefits for software development organizations. The creation of such solution for automated analysis is not without challenges and specifically contains two major challenges. Firstly, data on users' interaction with the system must be gathered in an efficient way. Secondly, the gathered data must be analyzed in order to gain knowledge about users' interaction.

To automate the gathering of usage data there is a need to implement usage tracing into the product, specifically at each execution point in the system where user interaction events are handled. With an object-oriented language usage tracing would have to be implemented at each such point of interaction. However, in a large business system this would be a very tedious task and since tracing is not part of the purpose of that interaction, concerns would be mixed. Separation of concerns, which is an important aspect in object-oriented software development, would thereby be lost and the code would be cluttered with usage tracing [8]. Previously AOP has been used to counter this issue and is therefore a strong candidate for successfully implementing usage tracing.

The usage data gathered from usage tracing is likely to be difficult to interpret by humans. Transforming the data, into what would be human perceivable, is essential for gaining any knowledge of the usage data. It is also important that the right kind of knowledge is gained. For example, the number of different IP addresses being used might prove little value compared to finding out where users have problem navigating the system. Since the employees at a software development organization have many opinions, ideas and great insight of the product that they are developing it might be beneficial to query these employees for wisdom on what should be analyzed.

In this study an approach using data mining with automated gathering of usage data using AOP, referred to as Automated Usage Tracing and Analysis (AUTA), is proposed. This approach ought to be evaluated for how it compares to a web survey, which is an existing method that could be used for the same purpose. By performing this evaluation, AUTA might be advocated as a possible replacement to the use of web surveys that are used to gather usage knowledge about end-users.

Due to the problems described above, this paper intends to answer the following research questions:

- Is the use of AOP suitable for usage tracing in a large business system?
- Is it beneficial to conduct a workshop to gain deeper insight of a software product, and use that information to form questions that provide guidance for the implementation of data analysis?
- Does AUTA provide equivalent results as a web survey and can it therefore be used as a substitution for the process of gathering information about users interaction with a software product?

1.2 Scope

This study is conducted on a system developed with C# and conclusions drawn are therefore not necessarily applicable to other programming languages. Furthermore, the scope of this study is limited to web systems with a workflow like user interface and is only concerned with analysis of server-side interactions, i.e. not interactions which are handled by client-side scripts. As the system in this case, see section 1.4, allows for gathering the majority of interactions this limitation will not affect the data mining, which could produce results that reflects the users' actual interaction. It should be noted though that this limitation in scope could affect the result for other cases, as some web systems have a lot of interaction driven by client-side scripts.

Visualization of data can play an important role in the knowledge discovery process and ease of interpretation. However, visualization of data is by itself large enough to be a separate research and has therefore been excluded from the scope of this study. Still some visualization of the analyzed data will be made but there will be no research into which visualizations approaches should be used.

1.3 Related Work

For automated usage tracing and analysis there is, from the best of the authors knowledge, a shortage of research. However, usage tracing and data analysis as two different subjects have a much greater amount of research, especially data analysis or as it is more often refereed to; data mining.

Van der Shuur and Jansen has presented a solution for improving software quality by automatically gather and report how a software service is being used. The data gathering was implemented in the service layer using AOP. Furthermore, the reporting was built using a set of metrics that were concerned with quality attributes like availability, accuracy, reliability and usability. They concluded that their solution was expected to contribute to an increase in software quality and that future work was needed on how to use data mining techniques for reporting on software utilization. [9]

For usage tracing one technique, AOP, stands out as having been tested for its suitability when when implementing automated data collection for usability evaluation and usage tracing. Tart and Moldovan showed that AOP could be used for automated usability evaluation [6] and equal results where gained by Tao who used the same AOP framework [4]. Tarby et al. compared AOP with Agent-Based Software Architecture concluding that they could be used as complement. The recommendation was to use AOP for defining traces while the agents would be used to ”produce traces whose visualization will be made in real time” [5]. A trace is referred to as a record of an action performed by a user. What all papers found on the subject lack is the comparison against other data collection techniques. For example, it might be of interest to know whether to utilize usage tracing or a web survey to gain appropriate knowledge to improve a software product.

Data mining is a wide research area with a substantial amount papers and books on the subject. The Data Mining and Knowledge Discovery Handbook is an example of an extensive handbook on data mining. This wide area covers several kinds of techniques for gaining knowledge from large sets of data. Examples of techniques are Cluster Analysis, Frequent Set mining and Outlier Detection. Kerr and Chung have for example made data mining research related to user interaction. They showed that it was possible to use Cluster Analysis to determine where instructions failed in a computer game [10], which can be seen as an advanced use of data mining to understand users interaction.

1.4 Case Description

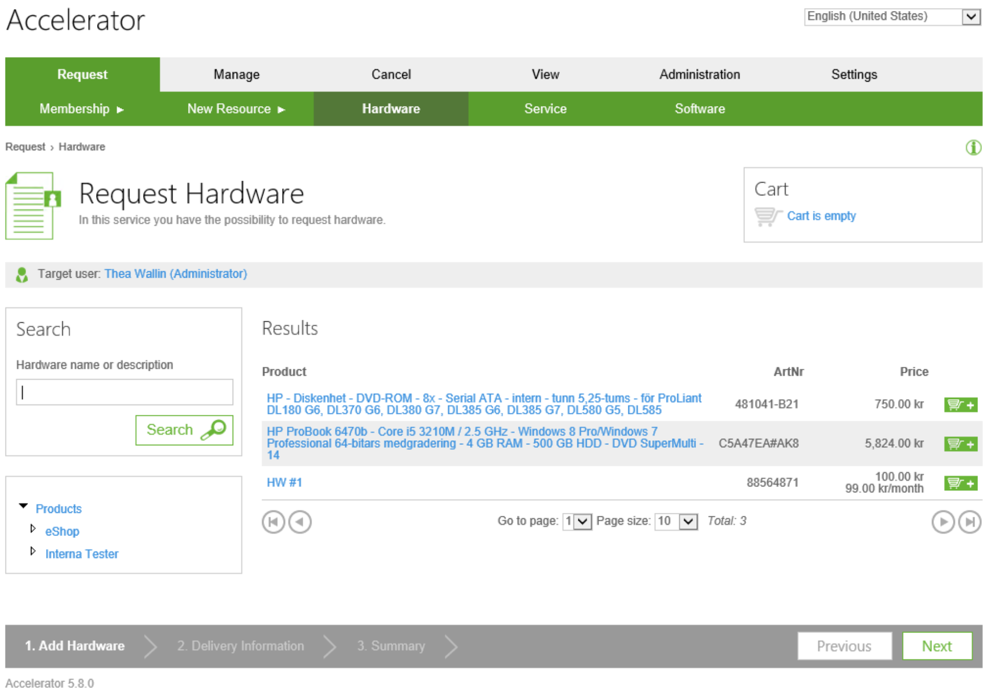
This section introduces the context in which this study took place by describing the organization and their product. It also explain and show why this is a topic of their interest by looking at their challenges.

Handheld devices have become a natural part of people’s life and are an essential tool for enhancing communication within organizations. For a software organization delivering a web platform it becomes important to explore the possibility and capability by targeting these devices. It is a question of meeting the market demands and could be the difference between success and failure.

ATEA Global Services is an organization that develops a software solution that automates the tasks of a Service Desk, there are several IT related tasks that employees within an organization need to handle. Ordering software, hardware and managing user’s passwords are examples of such tasks. ATEA Global Services goal has been to automate as many of these task as possible and by that become an essential part of the IT infrastructure within organizations.

To be a central part of the IT infrastructure, accessibility and understanding the needs of the end-users is essential. Therefore, ATEA Global Services wants to understand how these users interact with their software. They also want to explore the possibility of extracting the most important parts from their complex product for the creation of a lighter version with simplified interface, targeting handheld devices.

The product that ATEA Global Services provide consists of thousands of lines of



Request Hardware

In this service you have the possibility to request hardware.

Cart is empty

Target user: Thea Wallin (Administrator)

Search

Hardware name or description

Results

Product	ArtNr	Price
HP - Diskenhet - DVD-ROM - 8x - Serial ATA - intern - tunn 5,25-tums - för ProLiant DL180 G6, DL370 G6, DL380 G7, DL385 G6, DL385 G7, DL580 G5, DL585	481041-B21	750.00 kr
HP ProBook 6470b - Core i5 3210M / 2.5 GHz - Windows 8 Pro/Windows 7 Professional 64-bitars medgradering - 4 GB RAM - 500 GB HDD - DVD SuperMulti - 14	C5A47EA#AK8	5,824.00 kr
HW #1	88564871	100.00 kr 99.00 kr/month

Go to page: 1 Page size: 10 *Total: 3*

Accelerator 5.8.0

Figure 1.1: The interface of the service Order Hardware for the given case product

code, which makes it difficult and time consuming for the developers to implement a monitoring solution. On the other hand, a survey can also be difficult as they do not have direct contact with their end-users. This is one of the challenges that ATEA Global Service have in front of them today, while exploring the possibility of transitioning from a web application for desktop platforms to one for handheld devices.

An image of the provided case product can be seen in figure 1.1. The interface that is shown in this figure is the one for the service Order Hardware, which give the user the ability to order hardware that is predefined by the system administrator.

Accelerator is the name of the product provided by ATEA Global Services and used as case study in this paper. A set of terms are used to describe different functionalities within the Accelerator and there relation to each other is depicted in figure 1.2. The term *Task* is used to describe an action that the user want to achieve, e.g. ordering a computer or accepting request of access to a folder. A *Service*, which essentially is a type web page, is used to realize a specific *Task* and each service only provides the ability to perform one *Task*. Each *Service* contains a *Workflow* where the workflow provide guidance for the different *Steps* a users needs to take to perform a *Task* in a *Service*. Every *Step* in a *Workflow* has a link to the previous and the next steps. Furthermore, a step can have more than one possible previous step. Inside a *Step* there exists a set of *Features* where a *Feature* is a control, like a button, that a users can use to perform

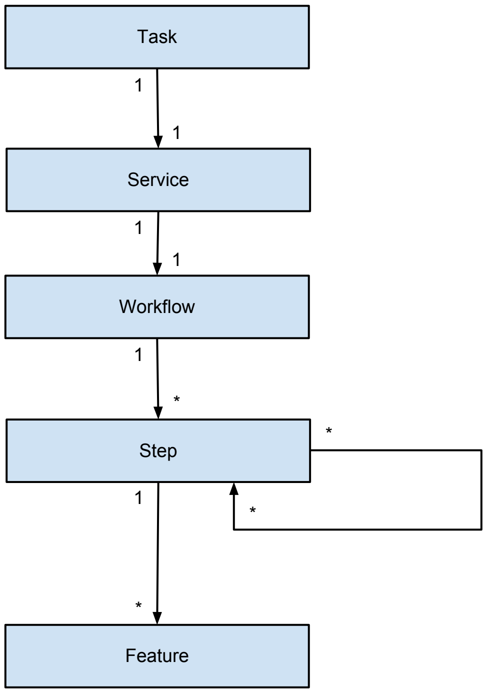


Figure 1.2: An overview of the relations between different terms in the product

actions like submitting a request.

1.5 Method

This study is, as discussed in the previous sections, concerned with evaluating the possibility to understand user interaction without conducting surveys targeting end-users. The study is conducted as a proof of concept and an overview of the process can be seen in figure 1.3.

In order to answer the proposed research questions this study was divided into eight main steps and two different tracks, one track for the literature study and one for the execution and validation of automated usage analysis. The reason of having a separate track for the literature study is to visualize that it was an ongoing process running in parallel with the other steps.

The purpose of the literature study was to gather knowledge about different areas that were encountered throughout the study. However, before the start, a related work study was conducted in order to have a general knowledge of the subject. Google Scholar and SUMMON were the main search engines used to find relevant papers. Focus was

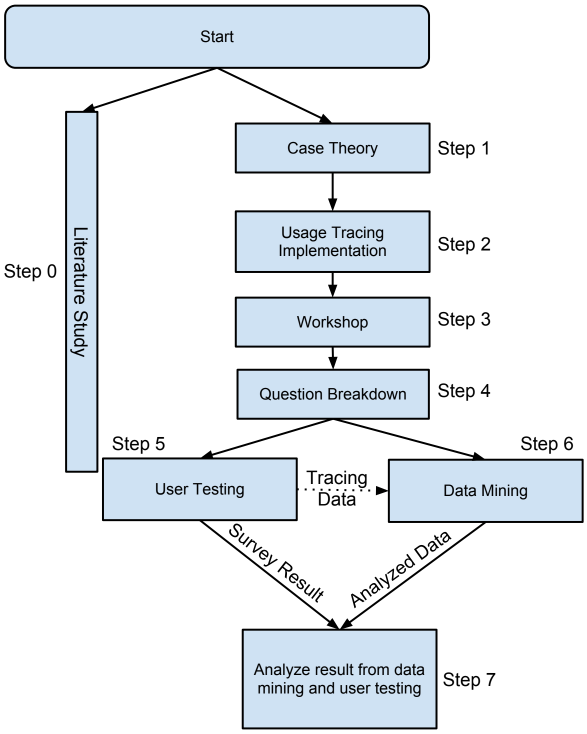


Figure 1.3: Workflow of the metod

put on the databases IEEExplore, ACM Digital Library and Springer Link with the following search terms:

- Usability Aspect-Oriented Programming, Usability AOP
- Usability Evaluation
- Aspect-Oriented Programming, AOP
- Aspect-Oriented Programming Systematic Review, AOP Systematic Review
- Data Mining
- Cluster Analysis, Clustering
- Workshop Brainstorming
- Survey
- Web Survey

- Outlier detection
- Correlation

A focus was put at papers published after year 2000, which contained the relevant keywords and had been used as references before.

The other track visualized in figure 1.3 consists of seven main steps: The understanding of the case and its challenges, implementing usage tracing in the current case product, conducting a workshop session to understand what should be analyzed, breaking down the outcome from the workshop into a set of questions, performing user tests to generate data, implementing data mining to analyze generated data from the user tests and finally analyzing the results from step five and six to evaluate the process. The fifth and sixth step ran in parallel because of their mutual dependency. Each step in this track is discussed further beneath.

To understand the case a brief introduction of the system was held by ATEA Global Services. Also, access to the system and the source code was provided for in depth analysis. Beside this there were both formal and informal meetings conducted with different employees within the organization. Based on this information insight and general understanding of the case was gained.

A usage tracing solution was implemented as the second step, see chapter 5. The reason that implementation of usage tracing occurred before the workshop, which can be argued as a better approach, was a hard deadline. A new version of the software was in the final stages and by prioritizing this step it could be made sure that the usage tracing was shipped with this release. The organization has around two or three releases each year and if the implementation would not have been in the release, the whole study might have been jeopardized or heavily delayed. The actual implementation was preceded with research on how to implement the tracing. The outcome of this research was the decision to use the AOP framework PostSharp. The development of usage tracing that followed was experimental in its process and during implementation it was also examined which data would be possible to gather. Completion of the implementation was successful before the deadline of the new version and was included with the new release.

A workshop was conducted with employees of different positions at ATEA Global Services, see chapter 3. The main goal of the workshop was to gain more insight of their perception of the system and their ideas on this study as well as extracting data that could be used as a foundation while defining question that should be answered by AUTA. A list of questions was extracted as a base to understand how to execute the data mining and survey.

Since the list of question where not detailed enough for data mining they needed to be broken down into more detail and made unambiguous, see chapter 4. This was achieved by creating a tree structures for each question and letting a lower level in tree represented a more detailed question. This is represented as step 4, Question Breakdown, in figure 1.3. At this stage it was also determined which questions could be answered by data mining respectively by conducting a web survey.

To support findings in AUTA the result needed to be verified with the result of a web survey with intention to gather the same findings. The plan was to design a survey and distribute it to end-users, which used a version of the case product that had usage tracing enabled. Unfortunately, due to lack of customers for the new version and low upgrade rate for current customers, the version with usage tracing enabled was only deployed for a few customers and was not being used to the extent that would be satisfactory for the data analysis. The solution to this issue was to let test subjects perform a set of tasks and then let the them answer questions in a web survey, this is annotated in figure 1.3 as User Testing. The tasks performed were related to a set of questions extracted from the workshop, see chapter 4.

The survey, see section 6.2, was designed by examining the extracted questions from the workshop and their break down in step four, figure 1.3. However, some of the questions from the workshop would not be possible to answer with the new approach. The reason for this was that data like ”how often a service was being used” require gathering of data from a deployed customer system that has been actively used for some time. With the questions for the survey selected tasks that would be sufficient to answer those questions by analyzing the data, were designed. When the tasks had been designed the testing was executed by inviting users with similar profile as possible end-users. They were given an introduction of the case product, before performing the tasks, and afterward they were given access to a web survey. By having this approach it was possible to simulate a real world scenario. It should be noted that along the design of both the survey questions and the tasks it was ensured that a correlation of the two would later be possible.

As previously mentioned, data mining of usage data was implemented and tested in parallel to the User Testing, see chapter 5. Steps six in figure 1.3, which is named Data Mining, includes two partial steps. Firstly the appropriate technique for each question was selected and then implemented. Secondly the implementation was analyzed using data generated from the user testing.

To asses the validity of the AUTA a correlation between results from data mining of data from tasks and result from the web survey was done, see section 6.3. By doing this it was possible to evaluate and understand if the AUTA could replace a web survey, that is conducted to understand user interaction. This is visualized in figure 1.3 as step 7, and is the final step in the method.

1.6 Outline

This paper consists of eight chapters. The current chapter, Chapter 1, give an introduction and overview of the study and describe the method. Chapter 2 presents the fundamental cornerstones. Chapter 3 describe why and how the workshop was conducted with employees at Atea Global Services. Chapter 4 present the questions broken down from the data of chapter 3. Chapter 5 present the solution, including implementations of automated usage tracing and data mining. Chapter 6 presents the evaluation of the solution using a correlation between usage data from user testing and responses from a survey. It also present how the survey and user testing was designed, conducted and analyzed. Chapter 7 contain a discussion of the results and threats to validity. The final chapter, Chapter 8, presents conclusions and potential future work.

2 Fundamentals

This chapter provides the foundations of the paper. First, the concept of AOP is presented along with an example of how it can be used for logging. Thereafter, the basics of data mining and different techniques used in this paper are presented. Lastly, fundamentals for conducting surveys and especially web surveys are described.

2.1 Aspect-Oriented Programming

Separation of concern is a vital aspect when developing software and its significance was discussed already in the beginning of the 1970th [11]. What separation of concern does is to only allow an object or method to be responsible for one aspect of the system. With this design, separation of concern, systems become simpler to create, understand, reuse and modify [8]. In object-oriented languages a system's functionality is decomposed of objects which have relationship to each other and with separation of concern each object models a single aspect of the modeled environment. Issues arise when some concerns of the functionality are not intuitively modeled in a single object, but rather overlapping several objects. For example, the ship entity in a ”port planning software” is not concerned with logging but logging code needs to be present in that entity. These concerns, or aspects, are so called cross-cutting concerns since they affect more than a single object. As cross-cutting concerns are implemented in an object-oriented language separation of concerns is lost along with its benefits.

AOP, which was first introduced in 1997 by employees at Xerox, can be used to maintain separation of concerns when cross-cutting concerns are present [3]. With AOP, cross-cutting concerns are implemented in so-called Aspects which can be executed at certain execution points in the system, e.g. on entering a method.

An AOP implementation contains a Component language and an Aspect language. The Component language, refers to an object-oriented or procedural language. While the Aspect language refers to the language used for implementing the cross-cutting concerns. The difference in an AOP implementation compared to a regular language is the Aspect language [3]. The Aspect language has three main concepts; Join-point, Pointcuts and Advices. A Join-point is a point in the execution where Aspects are coordinated with the component language, for example a join-point can be a call to a method. Pointcuts define which Join-points an Aspects should be applied to. Advices are the actions that will occur when a Join-point is reached. In order for the Aspect language to be used with

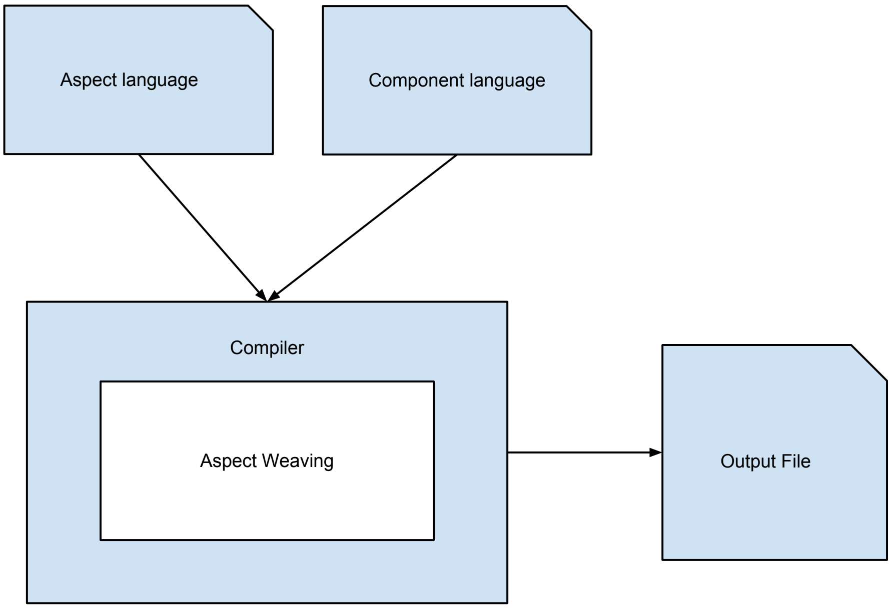


Figure 2.1: AOP weaving for a compile-time solution.

the Component language the two must be combined. This is done by an Aspect weaver which injects Advices into Join-points in the Component language during compile-time, shown in figure 2.1, or during run-time. The output file of the compiler will then have the instructions for the Advices weaved into the instructions of the Component language. [12]

Recent studies evaluating AOP are indecisive on whether significant benefits can be obtained. However, for most characteristics it has been concluded that AOP could have a positive effect. Endrikat and Hanenberg conducted an experiment on students using object-oriented programming and AOP [13]. The result showed no significant difference in development time. A study by Hanenberg et al. suggest that more than 36 code targets, points that the Aspect affects, are required before AOP will decrease the development time [14]. It should be noted that both studies used students as subjects. A systematic review of maintainability studies on AOP was conducted by Burrows et al. [15]. Reports included in the review have been leaning towards the fact that AOP would not contribute to significant improvements in maintainability. However, the authors of the review point out that the metrics used might be overlooking essential contributors to maintainability. Another systematic review looking at different empirical studies evaluating AOP was conducted by Ali et al. in 2010 [16]. This study concluded that AOP had positive effects on performance, the code size, evolution of system and modularity.

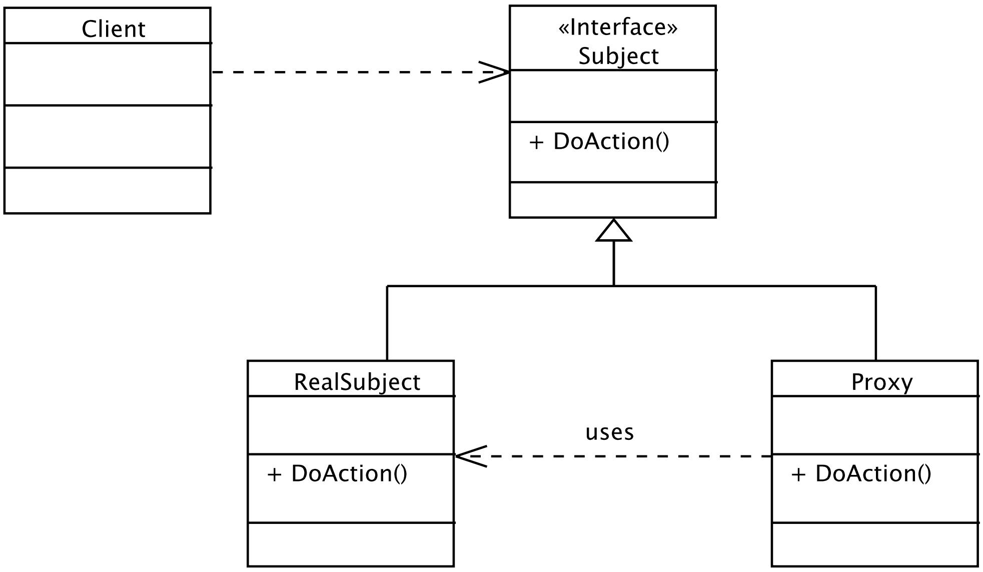


Figure 2.2: UML diagram depicting Proxy-pattern

Negative effects were only seen in studies evaluating AOP for cognition and language mechanisms. It can therefore be concluded that AOP provide many benefits but that it should be applied with care and since it is a different programming paradigm it is essential that developers have a good understanding of it.

To describe AOP in detail, the rest of this section will look at an implementation example, using PostSharp [17]. PostSharp is an AOP framework for .NET and like Aspect-J [18] its weaving is performed during compile-time, requiring changes to the compilation process. Other frameworks, like the AOP implementation in Spring.NET [19] is based on Proxy-pattern [20], figure 2.2. Which require a well defined interface to create proxies for the objects where Advices must be executed. Furthermore, this approach requires that any client object using one of the real subjects must only have references to the subject interface and not directly to the real subject. If this is not the case the changes to use such a solution would increase the upfront cost manifold.

The scenario for this example is logging, a typical problem that could be solved with AOP. Imaging a system that contains many service classes for which one is a service for sending email. A simple version of this service, EmailService, is show in listning 2.1. This class contains a method, SendMessage, which sends an email to the specified recipient. The goal is to log each time any of the service classes' public methods gets called. It is possible to write the logging code in the beginning of each public method but that would become tedious as the number of services grows.

Listing 2.1: An example class for sending emails.

```
namespace PostSharpExample.Services
{
    public class EmailService : IEmailService
    {
        private string from = "myemail@email.com";

        public void SendMessage(string to, string title, string message)
        {
            var client = new SmtpClient("myHost", 587)
                        {
                            Credentials = new NetworkCredential(from,
                                "myPassword"),
                            EnableSsl = true

                        };

            client.Send(from, to, title, message);
        }
    }
}
```

In order to avoid writing logging code in every service method an Aspect, LoggingAspect, is introduced. Its full implementation is presented in listing 2.2. The method ”OnExit”, line 18 to 22, is the Advice for this Aspect and will at execution log information about the method. In this specific example the name of the class and method along with its input and output is logged. The method ”RuntimeInitialize”, line 11 to 15, is called after PostSharp has deserialized the Aspect. This allows the programmer to perform initialization to objects that can not be serialized, in this example the serviceLogger instance. All services are put into the same namespace, ”PostSharpExample.Service”. In order to apply this Aspect to public methods in all services a Pointcut needs to be defined, in PostSharp referred to as Multicasting when applying to multiple methods at the same time. LoggingAspect define this Pointcut as being any public method for any class in the namespace ”PostSharpExample.Service”, see listing 2.2 line 1 and 2. If a base interface for all services existed the same could be achieved by applying the Aspect to all classes which implemented that interface.

During compile time PostSharp will inject the Advice at intended point of execution, weaving the component and aspect language together.

Listing 2.2: The aspect for logging the execution of all public methods in the ”PostSharpExample.Services” namespace.

```
1  [assembly: LoggingAspect(AttributeTargetTypes="PostSharpExample.Services.*"
2      , AttributeTargetMemberAttributes=MulticastAttributes.Public)]
3  namespace PostSharpExample
4  {
5      [Serializable]
6      public class LoggingAspect : OnMethodBoundaryAspect
7      {
8          [NonSerialized]
9          private ServiceLogger serviceLogger;
10
```

```
        public override void RuntimeInitialize(System.Reflection.MethodBase method)
        {
            base.RuntimeInitialize(method);
            serviceLogger = new ServiceLogger();
        }


        public override void OnExit(MethodExecutionArgs args)
        {
            serviceLogger.Log(args.Instance, args.Method, args.Arguments,
                args.ReturnValue);
        }
    }
}
```

2.2 Data Mining

Data mining concerns the finding of algorithms that investigate data and discover prior unknown patterns. Several different methods of data mining exist and can be divided into two main groups, verification and discovery. Verification is concern with the evaluation of a hypothesis proposed by for example an expert and includes the common methods of traditional statistics. Discovery deals with the task of automatically discover patterns in data. For this there are two subgroups, descriptive which try to interpret data and predictive which tries to create a behavioral model and use that to make predictions. For this study the subgroup of descriptive techniques is of most interest as it aligns with the purpose of this study. This group for example contains summarization and clustering. [21]

This section presents data mining techniques that are used in this study, these techniques were selected after studying multiple sources about data mining. Firstly, Cluster Analysis which groups elements with close proximity is described. This technique is in this study used to find when users tend to use the case product. Secondly, Outlier Detection, which has high value when trying to understand problematic parts in the system, is presented. This technique is used to find which services that are outliers in regard to completion time.

2.2.1 Cluster Analysis

Cluster Analysis (CA) is a data mining technique that tries to form groups having homogeneity within the groups and heterogeneity between groups, i.e. items within groups will have as close proximity as possible and items between groups will have as far proximity as possible. By defining the proximity measure in different ways different relations between entities in a data set can be discovered. [22]

Research related to CA has been conducted across a wide range of fields, from research in archeology and disease co-occurrence to media usage. Also, CA has many commercial usages, for example it has been used by startup companies to organize results from online searches [23]. Closer in relation to this study is the research that have been conducted by Huang et al., they have presented an approach to cluster data from a web server log to identify navigational patterns. Their paper mostly shows that cluster analysis is feasible to use with web server logs. They observed that the amount of sessions must be sufficiently large and that cluster with strong patterns usually contain few sessions. [24]

Two of the most popular cluster analysis techniques are K-Means clustering and Hierarchical clustering. K-Means clustering creates clusters by dividing objects in the set of data into k number of clusters, meaning that the number of clusters has to be predetermined. Hierarchical clustering creates clusters by making a hierarchy of clusters where a lower level represents a smaller cluster. In this paper Hierarchical clustering has been used and therefore this technique is described in detail. [22]

Hierarchical clustering, as mentioned above, results in a hierarchy of clusters that has the property of a binary tree. This technique contains a bottom-up approach, called agglomerative clustering, and a top-down approach, called divisive clustering. Agglomerative clustering creates a hierarchy by working its way up from the bottom with single object clusters by merging the clusters being closest to each other. Divisive clustering considers all objects to be in one cluster and then breaks the cluster into smaller clusters until each cluster only contains one object. [22]

An agglomerative clustering approach begins with every object being considered as a cluster and in each iteration the two closest clusters are merge into a parent cluster. The clustering continues the iteration until a root cluster has been found. Each cluster is assigned a height that represents the sum of summary distance between all of its elements and its centroid. With height calculated, two clusters which are at the same level in the binary tree of clusters can be positioned at different positions on the vertical axis to shows the relation between these two clusters. A cluster with high height will in general have many elements with close proximity since the height increase with "exponential" speed. It is required that the parent height is greater than its children's heights, satisfying a tree structure. [22]

One algorithm for performing agglomerative clustering is the Ward agglomerative algorithm, which uses the ward distance to calculate distance between clusters. The Ward distance is defined for clusters a and b as:

$$dw(a,b) = \frac{N_a N_b}{N_a + N_b} d(cen(a), cen(b)) \tag{2.1}$$

Where N_a and N_b are the cardinalities for clusters a and b, $cen(a)$ and $cen(b)$ are their centroids, and $d(cen(a), cen(b))$ is the euclidean distance between the two centroids. The algorithm involves the following steps (a more detailed description of the algorithm can be found in Mirkin's between pages 138 and 141): [22]

1. Let the set of maximal clusters be that of each element being its own cluster. Each cluster is initialized with cardinalities being one, heights being zero and the centroids being the clusters themselves.

2. In this step the two clusters, c_1 and c_2, that have the closets Ward distance, equation 2.1, are merged to make a parent cluster. The parent cluster gets its height as $h(c_1 \bigcup c_2) = h(c_1) + h(c_2) + dw(c_1, c_2)$, centroid as $cen(c_1 \bigcup c_2) = (N_{c1} * cen(c_1) + N_{c2} * cen(c_2))/N_{c1 \bigcup c2}$ and its cardinality as $N(c_1 \bigcup c_2) = N(c_1) + N(c_2)$.

3. The parent cluster is added to the set of maximal clusters and the two child clusters are removed. Then the Ward distance for the parent cluster to all other elements in the set of maximal clusters is calculated.

4. If the number of maximal clusters is larger than one return to step 2, otherwise the algorithm finishes.

A divisive clustering algorithm similar to the Ward agglomerate algorithm is the Ward-Like divisive clustering. This algorithm is essentially a reverse Ward agglomerate algorithm that. Instead of merging, this algorithm starts with one large cluster, containing all elements, that is split into two smaller clusters by maximizing the distance between them. This is repeated until each cluster only contains one element. [22]

2.2.2 Outlier Detection

Outlier detection finds elements that deviate heavily from the sample, which it is a part of and is an integral part of data mining. It can be used to detect elements which would yield incorrect result or for detecting elements which should be examined due to their deviation. For the second case its been suggested that outlier detection as a method could be applied for credit card fraud detection, network intrusion and voting irregularity analysis. [25]

There are two types of outlier detection methods, parametric and non-parametric. Parametric methods depend on observation having an underlying distribution, like the normal distribution. Non-parametric methods, does not assume prior knowledge to the underlying distribution and is therefore often more suitable for data mining. Further these non-parametric methods can be divided into three different methods; distance-based methods, clustering techniques and spatial outliers. Outlier detection methods also differ on whether the data is univariate or multivariate. Univariate methods are used when outliers in observation with one variable ought to be found. While multivariate methods are used to find outliers when the number of variables are manifold and the relationship between variables must be considered. Since parametric test are inappropriate for most cases of data mining this subsection will focus on non-parametric outlier detection methods for both multivariate data. [25]

Clustering techniques considers small clusters as outliers. Clustering, which is presented in section 2.2.1, does however not have outlier detection as its main objective. This means that clustering methods might not be optimized to the detection of outliers. [25]

Spatial outlier methods can be used to find local differences in respect to observations' neighboring values without these observations being significantly dissimilar than the rest of the population. Generally there exist two kinds of classification for spatial

X	5.0	6.0	3.0	7.0	5.0
Y	2.0	1.0	1.5	20	3.0

Table 2.1: Two group observations, X and Y, with almost equal mean due to the fact that on observation in Y is extremely large.

outliers: graphic approaches and quantitative tests. Graphic approaches try to highlight spatial outliers using visualization of spatial data. Quantitative tests uses, for example, Scatterplots to differentiate outliers from the rest of the data. [25]

Distance-based methods are based on a measure of the local distance and there are several definitions for what constitutes as an outlier. The original definition, presented by Knorr et al., is that an observation is an outlier if its distance to at least b of the other observations is greater than r [26]. A challenge with using this definition is how to define b and r. [25].

Challenges arise when outlier detection ought to be used for finding outliers among groups of observations. A simple way to compare the different groups is to use the average value for each group of observation. Where the averages are used to find which groups or group is an outlier. The problem with this approach is that outliers in each group of observations can heavily affect the mean. Two groups of observation can have very different ranges of data but still have the same mean. For example, let two groups of observation, X and Y, be those in table 2.1 which have mean of $\bar{X} = 5.2$ respectively $\bar{Y} = 5.5$. The observation have almost equal mean but most of the values, except one, in Y are smaller than the value in X. The reason for this is that the fourth observation in Y is substantially larger than it other observations. In fact, if this observation were to be removed the mean on Y would become $\bar{Y'} = 1.88$. The problem seen here is that the mean of different groups of observation can be similar even though the range of observation is very different.

When mining for outliers between groups of observation it may, as shown in previous section, be inappropriate to use the mean of each group to find outliers. Instead a multiple statistical test for significance could be used. In statistics, test for significance is used to check if one group of observations are significantly larger or smaller than other groups of observations. By using the ANNOVA F-test it is possible to obtain information on whether there is a statistical significant difference between the different groups of observations. Usually if such test shows a statistical difference the proceeding step is to do pairwise comparisons to discover which group of observations has a significance difference. [27]

In data mining, when trying to discover outliers between groups of observations the proceeding pairwise comparison is of more interest than just knowing that there is a difference. The reason for this is that the outcome should lead to gained knowledge that actions can be based upon, just knowing that there is a difference will due to this reason not be of good use. It is therefore more interesting to do multiple pairwise comparisons. By performing such a comparison it is possible to know which group or groups of observations that are statistically different to the other groups of observations.

Wilcoxon signed rank test, is a non-parametric significance test for comparing two samples. In this test the observations in each group are ranked for their position in the combination of the two groups of observations [28]:

$$E(U) = n_u(n+1)/2 \tag{2.2}$$

$$z = \frac{U - E(U)}{\sqrt{n_1 n_2 (N+1)/12}} \tag{2.3}$$

The probability that at least one of the comparisons have a type I error is called the family-wise error rate (FWER). When the number of comparisons increases there is also an increase in the FWER. Due to this it is important, when performing multiple comparisons, to adjust the FWER. Many methods have been presented for adjusting the FWER by trying to make FWER the same as the accepted type I error probability for a single comparison. [27]

Garcia and Herrera recommended the use of Shaffer's static procedure when making multiple pairwise comparisons. Their recommendation was based on the fact that Shaffer's static procedure does not perform as well as the Bergmann-Hommel's procedure but is much simpler to use and still performs better than most other procedures. [29] Shaffer's static procedure follows a step down model for adjusting the type I error probability α, the probability of rejecting a true hypothesis. For an ordered, smallest to largest, set of p-values p_i,...,p_n and hypotheses H_i,...,H_n. H_1 to H_{i-1} is then rejected if $p_i \leq \alpha/t_i$, where t_i is the maximal number of hypotheses which could be true if any $(i-1)$ hypotheses are false. The formula for the values of t_i is defined as: [30]

$$S(k) = \bigcup_{j=1}^{k} \{ \binom{j}{2} + x : x \in S(k-j) \} \tag{2.4}$$

As an example of Shaffer's static procedure, let there be 3 groups of observations where the accepted probability of type I error is $\alpha = 0.05$. Comparing these groups would then require 3, $\binom{3}{2}$, pairwise comparisons. For H_3 let the resulting p-value from a test be $p_3 = 0.02$. Since $\alpha/t_3 = 0.017$ is smaller than p_3 so H_3 is rejected. The adjusted p-value then becomes $APV_3 = min(v; 1), v = max(p_j t_j : 1 \leq j \leq 3) = 0.04$ since $t_3 = 3$, i.e. the last value in the set from $S(3) = \{0,1,3\}$.

2.3 Surveys

An efficient method for collecting data on a large population is the conduction of surveys. It is also a relatively simple research method but tends to not be beneficial in dealing with complex issues. This method is based on the notion that by having samples of subjects respond to questions it is possible to draw conclusions for the whole population. Such conclusion can however only be made if the sample is representative of the whole population. [31]

Surveys can be conducted in several different ways, so called modes. Example of modes are phone-, postal-, Internet- and face-to-face-survey. In a phone survey, test subjects are called and asked to answer questions on the phone. This kind of survey is both faster and less costly than face-to-face interviews and there is a tendency for subjects to answer truthfully. Also, it gives the researcher a chance to explain the goal of the questionnaire and handle misunderstandings. Face-to-face surveys, has many of the same benefits as phone surveys but additionally due to the direct contact the response rate is usually better than other approaches and researchers could tell if the subjects is providing false data [1]. In a postal survey the questionnaire is sent through post which is a relatively inexpensive mode compared to face-to-face and phone surveys. And, even though it is a faster approach than face-to-face surveys it is slower than phone or Internet surveys [31]. An Internet survey, can be done either by providing questionnaire in emails or web sites. It is both faster and cheaper than any of the other alternatives but come with challenges for getting a high response rate. [1]

2.3.1 Web Surveys

This sub section focuses on Internet surveys, specifically the mode of providing a web site, and is referred to as web survey. A web survey is the approach most likely used if an organization, similar to the organization for this case study, need to collect data from their end-users. As it has the desirable characteristics of being both fast and cost efficient. Another reason for selecting this mode is that it is the only practical and reasonable way of contacting end-users.

Web surveys are, as mentioned, beneficial when cost and time are of primary concern and are therefore an appropriate mode when in need of obtaining a great amount of data in a limited amount of time. The main issue is however, as pointed out by Fan and Yan [32], that they are prone to low response rate, which severely reduces the statistical power of the survey and therefore its credibility. It is thereby important to focus on conducting a web survey in a way that will yield a relatively high response rate.

The first time the subjects come into contact with the web survey is in the invitation and this is the first source of possible loss in response rate. For the invitation several precautions can be taken in order to limit this loss. For web surveys Kaplowitz et al. have concluded that the invitation text should not be reduced in length over the cost of persuasion and completeness. Their study also suggested that the URL to the survey should be placed in the end of the web survey instead of in the beginning. Moreover, the use of mixed modes, like e-mail and postcard, and providing accurate estimate of time to take the survey could also increase response rate [33]. And, Sauermann and Roach show that personalization of the invitation increase the chance of response by 48% [34]. It is also recommended to state scarcity, like deadline coming up and that the subject is part of a small selected group to further enhance response rates [32].

Once a subject decides to participate in the survey, by clicking the link, it must be made sure that the subject also completes the survey. Therefore the design of the questions should be done with care. First of all, questions should be kept simple, avoiding biased and vague questions: Biased questions can induce subjects to answer what the researcher wants the answer to be. Vague questions might lead to misunderstanding and potentially making subjects cancel the survey before completion. Another important aspect is whether to use screen-by-screen or scrolling surveys as they will yield different benefits. Screen-by-screen, where each question is displayed on a separate page, allows subjects to skip questions and makes it possible to remind subjects to give consistent answers in the right format and range. While the latter, where all questions are one one page and user scroll through them, is suggested to require less time and resources. [32] Furthermore, aesthetically displeasing visualization have been suggested to have insignificant effect on response rate but it is likely to have high impact on response quality. For example, responses to scalar questions can be expected to become negatively skewed. [35]

In the delivery of web surveys it has been shown that reminders increase response rate and that they should provide the possibility to "opt out" of the web survey. To further increase response rate the reminder should change wording from initial invitation without presenting new information. Also, changing the day for sending reminders and changing the time of day can also have positive effect, in increased response rate. [34]

3 Workshop

This chapter provides the purpose of the workshop by giving a general background and idea behind the workshop. It also describes how the workshop was conducted, the questions that were brought up, the outcome and the analyzed result that was extracted.

The persons with deepest knowledge of the software that ATEA Global Services provide are those that work with it every day, developing, testing and maintaining the solution. They are experts of the front end, back end and everything that surrounds it.

To understand ATEA Global Services employees' point of view on the potentials, limitations and challenges of transitioning the Accelerator to handheld devices a workshop was conducted. Invited to the workshop was parts of the support team, test team, development team, User experience design (UX) team and the product owner. This set up of participants was selected to get a wide range of ideas and to enhance discussion, as people from different teams are very likely to have different standpoints.

The goal of the workshop was to gain more insight of participants perception of the system and their ideas on this study as well as extracting data that could be used as a foundation while defining question that should be answered by AUTA. Therefore, the workshop was conducted in such a way that all ideas should be tackled and analyzed. The goal regarding evolution to handheld devices came as a sidetrack and was of interest by ATEA Global Services.

Workshops of this type are typically called brainstorming. A brainstorming session is defined as group activity of problem solving. This methods goal is to produce as many ideas as possible, the wider the ideas the better, and to compare, combine or improve proposed ideas. To succeed with this it is important to not be critical and to let participant think freely [36]. According to Osborn the average participant contribute with twice as many ideas compared to when thinking individually [36]. This finding has been questioned because of participants blocking each other's ideas [37]. However, it is suggested by Diehl and Stroebe that letting participants prepare and reflect on the problem individually, while writing down ideas, is a way to avoid the decreased generation of ideas caused by group blocking [38]. They also suggest that a time limit will increase the number of ideas that populate from a brainstorming session, this to the fact that a fixed time is most likely to make participants feel obligated to continue the activity for most of the time [38].

3.1 Set Up

The workshop had seven participants excluding the two authors of this study, whom lead the workshop. It started with a brief introduction to the idea of this study and the goal and structure of the workshop, this lasted for about ten minutes. The participant were then presented with three questions, one at a time, which all were first described for about two minutes so that everyone understood the question. After a question was introduced all participant wrote down his or her ideas to the question on sticky-notes, which were collected and put on a whiteboard. The time limit, which participants had while writing down their ideas, was strictly five minutes. The reason of not letting them work in groups, at this point, was to avoid them from influencing each other and decreasing productivity, as mentioned in above. The notes that were similar to each other were then grouped, and simultaneously there was an open discussion about each group of notes. The reason for this structure during the workshop was to first let people think alone and then to have an open discussion to reason about their answers and see if more could be extracted, each open discussion had a time limit of ten minutes. By grouping the questions it was easier to have a structured discussion and see which points seemed to be common perceptions between participants. In total, for all questions, 74 sticky-notes were written. The complete workflow of the workshop session can be seen in image 3.1.

The questions for the workshop were designed in such a way that they together would give a deeper knowledge of what was known, what was not known, what was interesting to know and what was important for handheld devices. This knowledge could then be used to define a number of questions that should be answered by the survey and the data mining. Each question and the outcome from the workshop is discussed further in the subsections that follows.

3.2 Which parts of Accelerator need improved user experience?

The purpose of this question was to understand what the employees at ATEA Global Service saw as points of improvement in regards to the user experience. Their deep knowledge, diversity and long term use of the software over different releases give them unique insight. By questioning them it would be possible to predict the outcome of the gathered data and understand if they are able to predict the need of their end user. This question was the foundation for defining which parts of the software were important to monitor and analyze.

The sticky-notes were gathered and grouped into five different categories, the following list describe the categories:

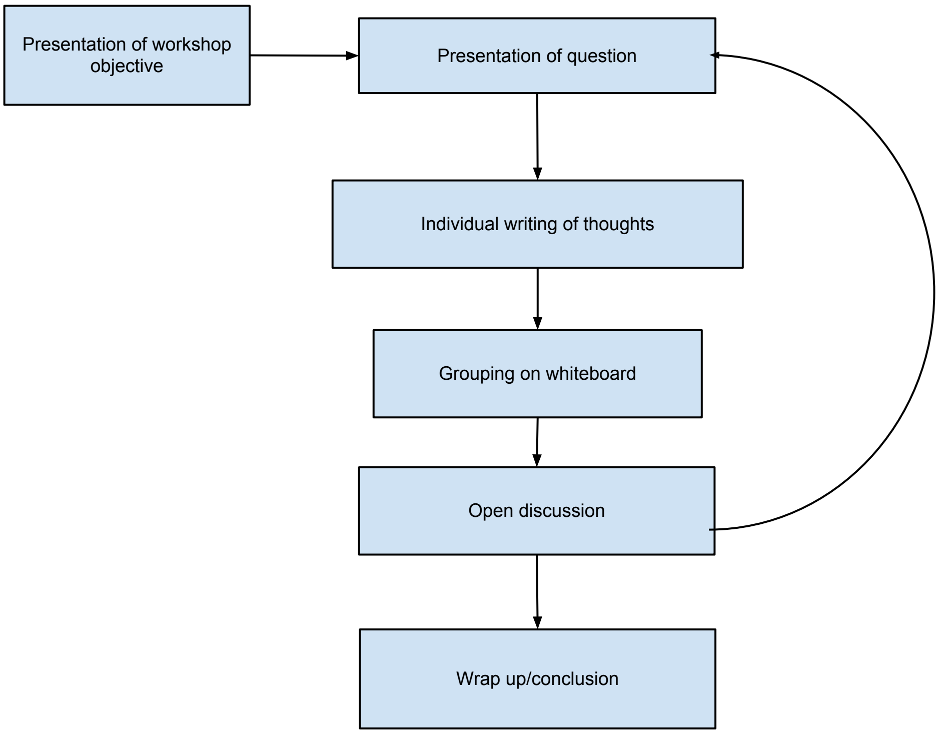


Figure 3.1: Workflow of the workshop process

1. "What happens" The transparency of the software towards the end-user. Example of sticky-notes: "Orders, what happen with the order when you complete your order."
2. "Grouping" The possibility to group and simplify some procedures. Example of sticky-notes: "The need to go to different services to order different types of products."
3. "Admin" The parts that are related to the Admin interface and advanced configuration.
4. "Menu" The ideas discussing the navigation. Example of sticky-notes: "Menu system, there is a limited space"
5. "Customization" The reasoning about a more personalized interface. Example of sticky-notes: "An Accelerator for different users' roles."

The diversity of participants was very important for the discussion of each category and it resulted in a clear view of possible parts to monitor and analyze.

3.3 What would you like to know about the users' interaction with Accelerator?

By asking this question it was possible to find points where the employees of ATEA Global Services felt uncertain, in regards to how the software is used. The idea was also to understand if parts of the software had been developed and maintained without clear and motivated reasons. The question was defined as guiding purpose to which data that should be gathered and extracted for analysis.

There were a lot of sticky-notes written during this question and the following categories were formed:

1. ”Misunderstandings” See where end user have issues to use the product. Example of sticky-notes: ”What makes a user confused, regards to how the service works”
2. ”Statistics” Information based on pure statistics. Example of sticky-notes: ”Least used service”
3. ”Time” Information based on time, when and how end user use the software. Example of sticky-notes: ”How long times does it take a user to complete different services/tasks”
4. ”Frequency” How often and how is the software used. Example of sticky-notes: ”How big is the user group of daily users”

From the open discussion it was clear that participants where uncertain of how the product was being used. Which seem to come from a lack of collaboration with end users. It was also clear that the participant found this question interesting and wanted to know more about their end users, as there were so many different ideas. The result from this questions was later used to define what data to gather, see chapter 5.1.

3.4 What is important in a mobile version of Accelerator, from the users' standpoint?

This question was asked for by the development organization of the case product. It was therefore a part of the workshop and the outcome of it is of interest as it gave a deeper knowledge of what could be of interest while evolving the case product.

As all participants have experience of handheld devices and a technical background they could easily define important points of software for handheld devices. They could also relate specifically to the case product in question, which is very valuable. The following groups were extracted from the notes:

1. ”Simplicity” The ease of use, ideas in this category are about the the importance of UX. Example of sticky-notes: ”Easy to find what I’m looking for”
2. ”Performance” Highlighting performance. Example of sticky-notes: ”No delay because of use on handheld device, otherwise I rather use the computer.”
3. ”Functionality” Different functionality that should be included in the software. This category describe functionality and reasoning regards functionality. Example of sticky-notes: ”Only display the most important for each specific user.”

These categories are clearly focused on handheld devices and the discussion highlighted the importance of reduced complexity by minimizing functionality. The reasoning was to exclude most functionality and keep the essential parts, while still delivering a good service.

3.5 Result

There were some obvious outcomes from the workshop, first of all there is a need for increased communication within the organization. It was also clear that the many felt uncertain about their end users usage of the software. This emphasizes the need of more statistical data of the software use or increased communication with end users.

By gathering and analyzing the information from all Workshop Questions the following eleven questions were extracted and define, to possibly be answer by the survey and automatic data analysis:

1) What is the frequency of use for service/functionality: By answering this question usage and possibly UI design decisions can be made better. It will define which services should be prioritized when evolving the product. This question can also be grouped by user roles to see how their behaviors differ.

2) Frequency of use for different roles: The reason for understanding frequency of use for different role is so that the UI could be tailored for some roles which use the system a lot.

3) Are there any functions which are not used: The question aim to reduce complexity by removing not used functionality. The goal is to provide a software for handheld devices which is simple to use and have as few distraction points as possible.

4) How is internal search of the website utilized: Since a lot of the services are based on search it is interesting to know if the search functionality is used once within a service session or several times. If search mostly occur ones it might be possible to suggest a simplified design.

5) How long time does it take a user to complete a task: If a part of the software is complex it most probably will take longer time to complete task. By understanding complex parts it would be possible to pinpoint where to focus on improvement or parts that should be revised.

6) How does software use differ between regular and non-regular users: Based on this question strategic decision can be made from the data. For example, which type of user should be prioritized.

7) Which service is the most difficult to complete: By understanding what scenario and part of navigation that are difficult for the end-user, it is possible to see where its most vital to simplify the UI.

8) Are there any trends, between different versions: Changes over time are important feedback for continues development and prof of successful strategical decisions.

9) Are there services with high Bounce rate: By finding services with high Bounce rate it is possible to revise these services. A bounce is considered as when a user navigates to a service and then navigates to another service in just a matter of second. This indicates that the service was not what the user was looking for. A high bounce rate can be a sign of design or content issues.

10) How often and where do drop offs occur: By finding misunderstandings it could be possible to pinpoint parts that have a complexity that is too high or where there are too many steps and options for the end-user to complete. A drop off occur when a user is a while on a page and start doing things but then abort.

11) What time of the day are tasks carried out: By understanding when the service is accessed and what is carried out, it might be possible to motivate what parts should be extracted to a handheld device as these are used in the mornings, evenings or during lunch.

These questions are further analyzed, broken down and redefined in chapter 4.

4 Question Analysis

This chapter describe how the questions defined in section 3.5 are redefined in a more detailed manner and broken down into a tree structure. Moreover, the questions are analyzed for which could be answered by data mining and respectively a web survey.

Each question has been broken down further and into leafs of ”Database” and ”Survey”. The ”Database” leaf represent a solution that analyze the data automatically and the ”Survey” leaf represent an analyzis of the gathered survey data. If the leaf is inside a box which is red and have horizontal lines it means that it is not feasible to answer. If it is green and have diagonal lines it means it is feasible to answer. The reasoning about feasible or not feasible to answer was done by arguing back and forth on the complexity for each leaf.

The following sections will go trough each question, from one to eight in the tree structure, and point out which questions in section 3.5 that it derives from.

4.1 Breakdown Question 1

Breakdown Question 1 (Time for a user to complete a task) derive from Workshop Question 5 in section 3.5 and is broken down into two levels, see figure 4.1. Sub level 1.1, in figure 4.1, look at each service and try to understand the completion time of them. The corresponding ”Survey” leaf in this branch is not feasible to answer, as the survey is unable to measure this kind of data. However, the ”Database” leaf is feasible to answer as the automatic gathering of data will retrieve data with more parameters. Sub level 1.2, in figure 4.1, look at all services and find outliers with respect to completion time. The leafs for level 1.2 encounters the same issue as 1.1, ”Survey” is therefore marked as not feasible to solve and ”Database” as feasible.

4.2 Breakdown Question 2

Breakdown Question 2 (Time of day that task are carried out) derive form Workshop Question 11 in section 3.5. It has two branches, see figure 4.2, and each of these branches have the two leafs, ”Database” and ”Survey”. Branch 2.1 in figure 4.2 question when different services are being used during the day and 2.2 in figure 4.2 question when the product is being used. To ask survey participants when they use the product is feasible but to ask them at what time during the day that they use a specific service is not.

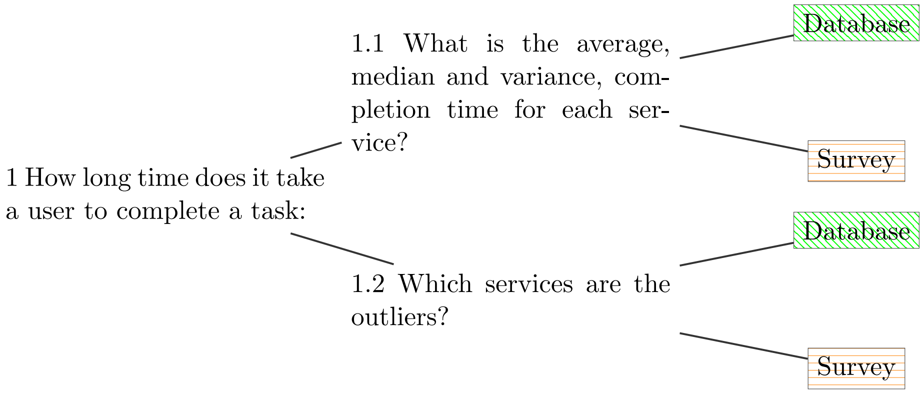

1 How long time does it take a user to complete a task:
1.1 What is the average, median and variance, completion time for each service?
Database
Survey
1.2 Which services are the outliers?
Database
Survey

Figure 4.1: Breakdown Question 1

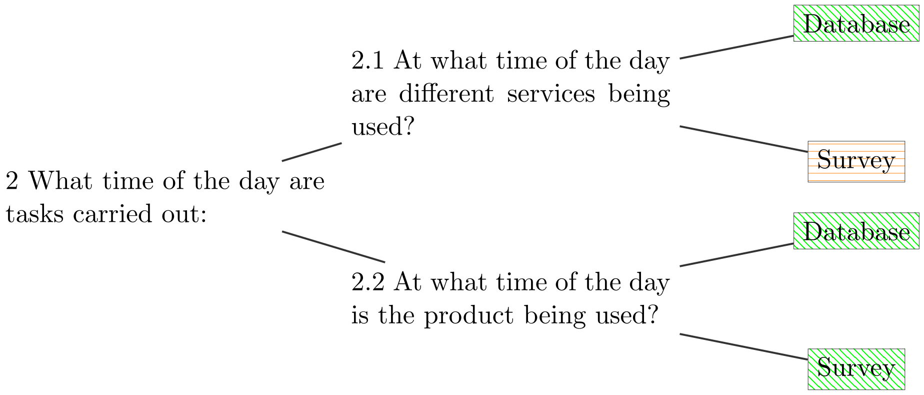

2 What time of the day are tasks carried out:
2.1 At what time of the day are different services being used?
Database
Survey
2.2 At what time of the day is the product being used?
Database
Survey

Figure 4.2: Breakdown Question 2

Therefore, it is feasible to answer the ”Survey” and ”Database” leaf for 2.2 but it is only feasible to answer the ”Database” leaf for 2.1.

4.3 Breakdown Question 3

Breakdown Question 3 (Product Bounce rate), see figure 4.3, derives from Workshop Question 9 in section 3.5. It focus at the product and the services and try to understand if there is an issue with Bounce rate. By dividing into two branches 3.1, ”In the different services?”, and 3.2, ”In general?”, from figure 4.3 it is feasible to understand if it is a common problem, that users have difficulties in finding what they are looking for. It is not feasible to answer 3.1 by a survey as the questions would need to be very specific and with unreasonable amount of options. However, the ”Survey” leaf in 3.2 is feasible to answer by a survey as it is simple to ask if participant have the perception that it is hard to find what they are looking, and that they need to ”click around” for a while

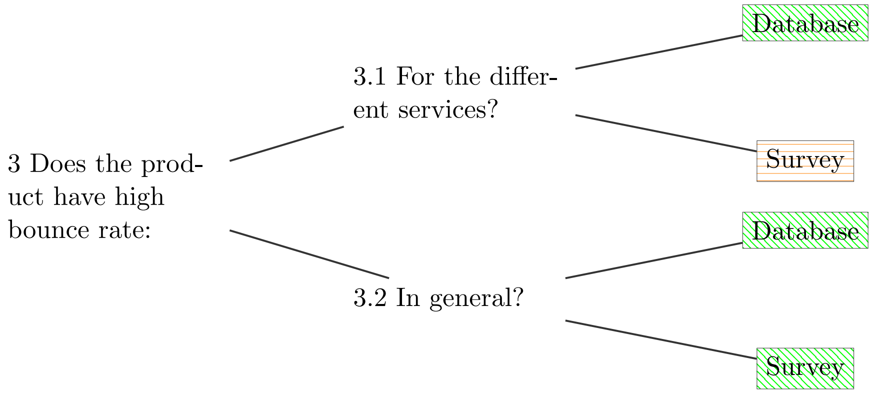

3 Does the product have high bounce rate:
3.1 For the different services?
3.2 In general?
Database
Survey
Database
Survey

Figure 4.3: Breakdown Question 3

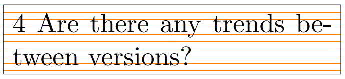


Figure 4.4: Breakdown Question 4

before they find it. The ”Database” leaf can be answered in both branches as an analysis of the gathered data could provide necessary information.

4.4 Breakdown Question 4

Breakdown Question 4 (Trends between product versions), see figure 4.4, derive from Workshop Question 8 in section 3.5. This question is marked as not possible to answer, this is to the fact that trends are not feasible to answer in the given case for this paper. In section 1.4 the case is described and as seen there is only going to be one release with tracing enabled during the given timespan. The problem arise from the fact that this study is performed within a time limit that has not provided the possibility to analyze the outcome over time, tweaking the product and analyzing again.

4.5 Breakdown Question 5

Breakdown Questions 5 (Utilization of internal search) derive from Workshop Question 4 in section 3.5. It questions how the internal search is used and is broken down into four branches, see figure 4.5.

Firstly, branch 5.1 tries to understand how the Product Tree/Catalogue is used. This is possible to find out both for the ”Database” and ”Survey” leaf. It is simply a question of analyzing the given data and asking the participants what they most often use to find what they are looking for.

Secondly, branch 5.2 analyzes how the search by text is used. Similar to 5.1 it is

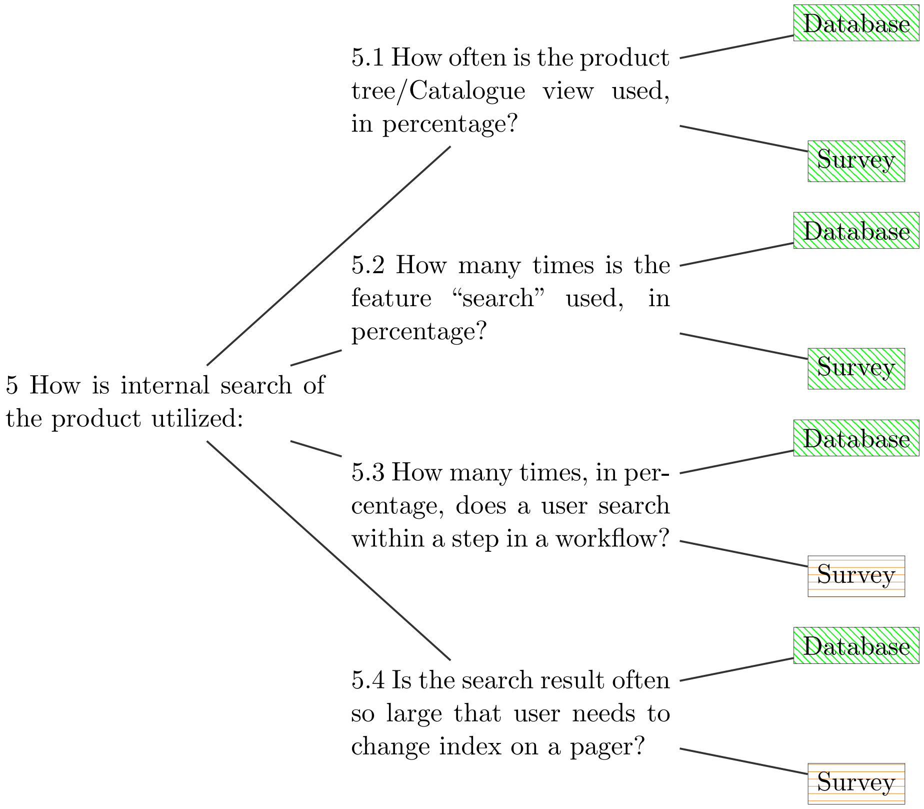

5 How is internal search of the product utilized:
5.1 How often is the product tree/Catalogue view used, in percentage?
Database
Survey
5.2 How many times is the feature “search” used, in percentage?
Database
Survey
5.3 How many times, in percentage, does a user search within a step in a workflow?
Database
Survey
5.4 Is the search result often so large that user needs to change index on a pager?
Database
Survey

Figure 4.5: Breakdown Question 5

feasible to find the answer by analyzing the data and asking the same question in the survey, to successfully answer the leaf ”Database” and ”Survey”.

Thirdly, branch 5.3 is looking for the use of search functionality within different steps of a workflow for a service. The leaf ”Database” can answer this by analyzing the data but the ”Survey” leaf will not be able to answer this as the question in a survey would be to complex.

Fourthly, branch 5.4 try to understand how the users act on the results given from the search that they perform. Is the result so large that they need to change page or will they find what they are looking for in the first couple of results, for the leaf ”Database” it is feasible to analyze and see how the user act but for the leaf ”Survey” it is not feasible to design a simple question within a survey to answer this.

4.6 Breakdown Question 6

Breakdown Question 6 (Occurrence of drop offs) derive from Workshop Question 10 in section 3.5. It has three main branches, see figure 4.6, that cover the question of when

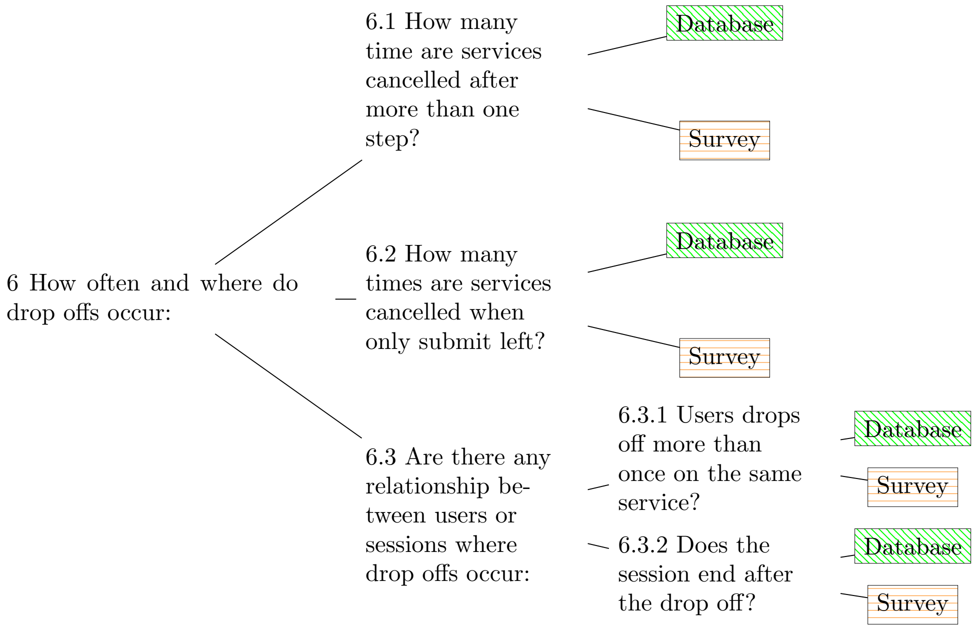

6 How often and where do drop offs occur:
6.1 How many time are services cancelled after more than one step?
Database
Survey
6.2 How many times are services cancelled when only submit left?
Database
Survey
6.3 Are there any relationship between users or sessions where drop offs occur:
6.3.1 Users drops off more than once on the same service?
Database
Survey
6.3.2 Does the session end after the drop off?
Database
Survey

Figure 4.6: Breakdown Question 6

users abort from something that they started to perform. None of the ”Survey” leafs can be answered in this structure due to the complexity of it. However, the ”Database” leafs can be answered with data analyzis. Looking at each branch it is feasible to see the different angels of drop offs that could be monitored.

Firstly, branch 6.1 count the number of times that a user stop going trough the procedure for a service, cancelled after being initiated. Secondly, branch 6.2 has the similar purpose as 6.1 but by only counting the drop offs at the last step in a workflow. Thirdly, branch 6.3 tries to understand and see connections between drop offs for a given user. It is therefore broken down even further with two branches. Branch 6.3.1 which try to understand if the user try to redo the operations for a service more than once, and drop of several times, and branch 6.3.2 which try to understand if the user log out or close the browser upon drop off.

4.7 Breakdown Question 7

Breakdown Question 7 (Most difficult service to complete) derive from Workshop Question 7 in section 3.5. It has one leaf, ”Survey”, and two branches with leafs ”Database” and ”Survey”, see figure 4.7. The purpose of this structure is that a survey could answer Breakdown Question 7 directly by asking the participants what is difficult to do with the service. However, to design a survey question for the other two branches, 7.1 and 7.2, would be to complex. The two branches, in figure 4.7, try to understand which services

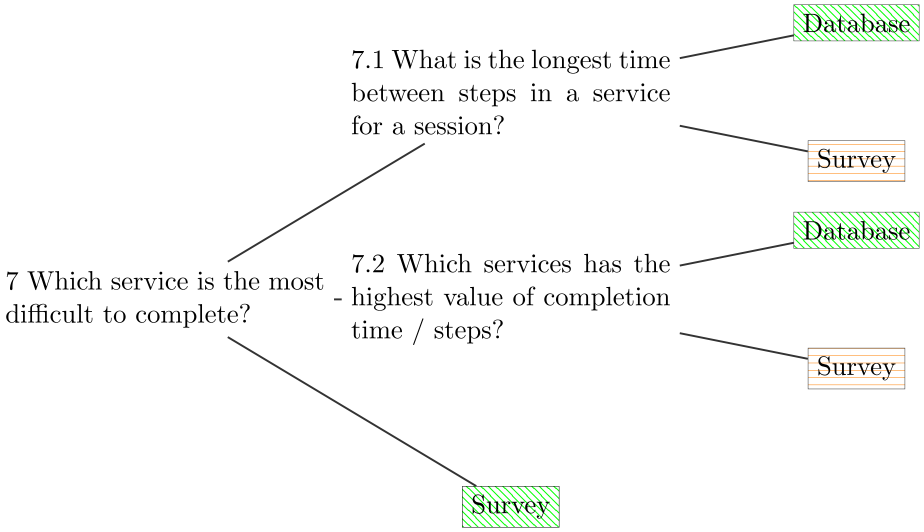

7 Which service is the most difficult to complete?
7.1 What is the longest time between steps in a service for a session?
Database
Survey
7.2 Which services has the highest value of completion time / steps?
Database
Survey
Survey

Figure 4.7: Breakdown Question 7

are difficult to complete by analyzing the longest time it takes a user to complete a step during the completion of a workflow, represented in branch 7.1, and by looking at the service which has the highest value of completion time per steps, represented in branch 7.2. Analysis of data could answer 7.1 and 7.2, which makes the leaf ”Database” feasible to answer.

4.8 Breakdown Question 8

Breakdown Questions 8 (Frequency of use for service and functions) derive from Workshop Question 1, 2, 3 and 6 in section 3.5. The reason for combining so many of the questions is that they are all related to summary statistic. Workshop Question 8 has three main branches, see figure 4.8, and each of these branches have a set of branches and leafs. Beneath is a walkthrough of each main branch; 8.1, 8.2 and 8.3.

Firstly, Breakdown Question 8.1 (Navigation to a service) has five branches, see figure 4.9, which all relates to the analysis of how many times a service is navigated to. The first branch, 8.1.1 is simply looking at the number of times that a service is visited. The ”Survey” leaf in this branch is not feasible to answer due to the fact that there are too many services to select between and the participants of a survey can not be expected to memorize all the services they have visited. However, for the ”Database” leaf it is feasible as all visits to a service is registered and an analysis of this would give the answer. The second branch, 8.1.2 does the same thing as 8.1.1 but by grouping the request related to the user’s roles. The conclusions regarding the leafs is therefore the same as for 8.1.1. The third branch, 8.1.3 focus on the finding of services that are not used very much. It has two branches that reflect on the general case and the case for

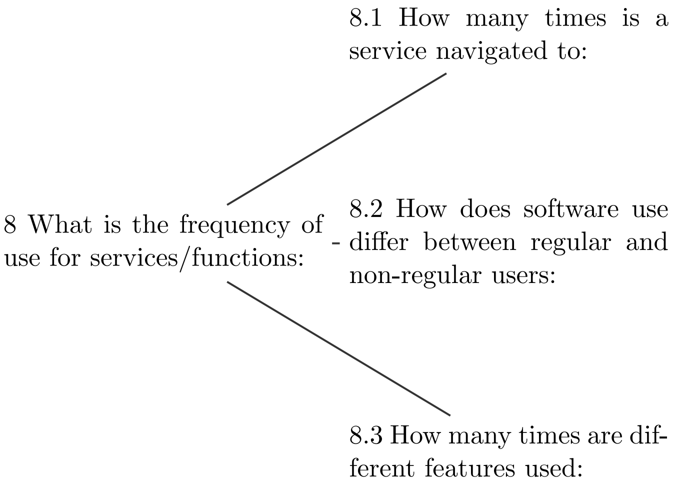

8.1 How many times is a service navigated to:
8 What is the frequency of use for services/functions:
8.2 How does software use differ between regular and non-regular users:
8.3 How many times are different features used:

Figure 4.8: Breakdown Question 8

different user groups/roles. The ”Survey” leaf in both branches are not feasible to answer as the complexity of this question is to deep. However, the ”Database” leafs for 8.1.3 are feasible to answer by combining different results for the data analysis. The fourth branch, 8.1.4 is analyzing the use of different services. It has two branches, 8.1.4.1 and 8.1.4.2, which are analyzing the overall usage of different services respectively within a session. By analyzing 8.1.4.2 it might be feasible to understand which services are most frequently used during the same session and thereby should be easier to reach or be combined. Both the leaf for 8.1.4.1 could be answered by a survey and an analysis of the automatically gathered data. The ”Database” leaf of 8.1.4.2 could be answered, in the same way as 8.1.4.1, but the ”Survey” leaf is difficult to answer due to the hardness of designing such a survey question.

Secondly, Breakdown Question 8.2 (Difference between regular and non-regular users) try to analyze the differences of service usage between regular and non-regular users, non-regular users are defined as users that uses the solution less than every 2 days. It has two branches, see figure 4.10, to analyze the general case and for specific a specific user group/role. The ”Survey” leaf for both branches is not feasible to answer due to the issue of asking such a question in a survey. The ”Database” leaf can be answered by using the automatically gathered data.

Thirdly, Breakdown Question 8.3 (Use of features) is focusing on features. By analyzing on feature level it is feasible to understand what features within a service that are being used. It has three branches, see figure 4.11, looking at the general case, the service specific case and for what case that features within a service are used less. The general case 8.3.1 can answer both the ”Database” and ”Survey” leaf by analyzing the data. The specific case within a service and the least used feature have the possibility to be answered for the ”Database” leaf but not for the ”Survey” leaf as they are to specific.

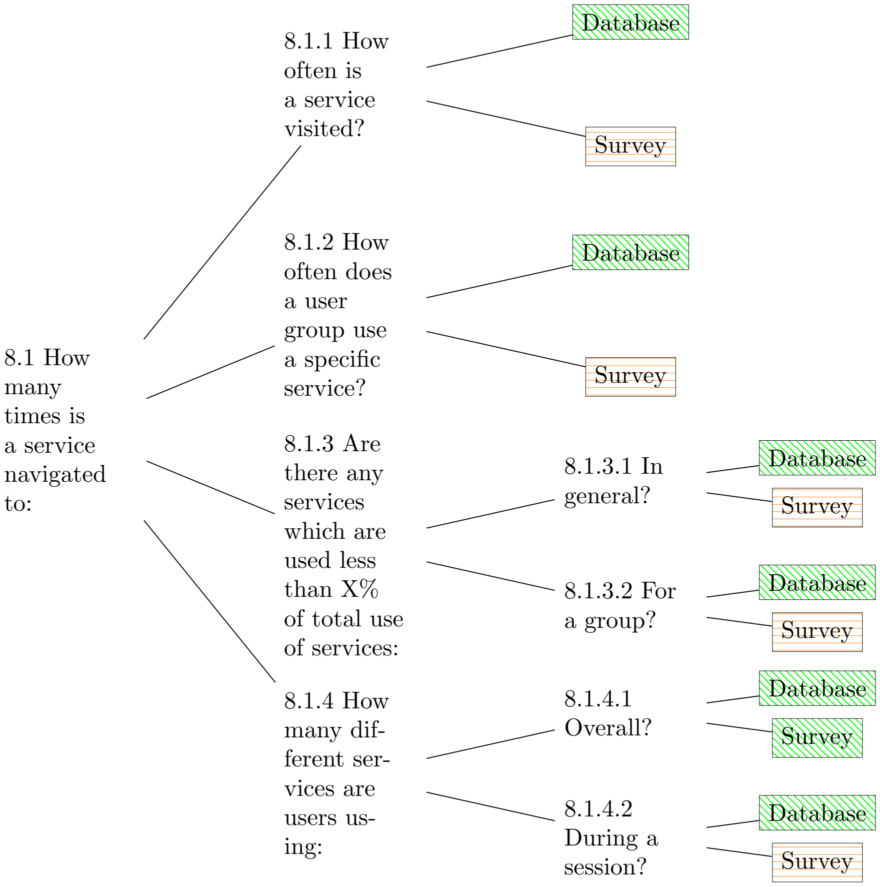

8.1 How many times is a service navigated to:
8.1.1 How often is a service visited?
Database
Survey
8.1.2 How often does a user group use a specific service?
Database
Survey
8.1.3 Are there any services which are used less than X% of total use of services:
8.1.3.1 In general?
Database
Survey
8.1.3.2 For a group?
Database
Survey
8.1.4 How many different services are users using:
8.1.4.1 Overall?
Database
Survey
8.1.4.2 During a session?
Database
Survey

Figure 4.9: Breakdown Question 8.1

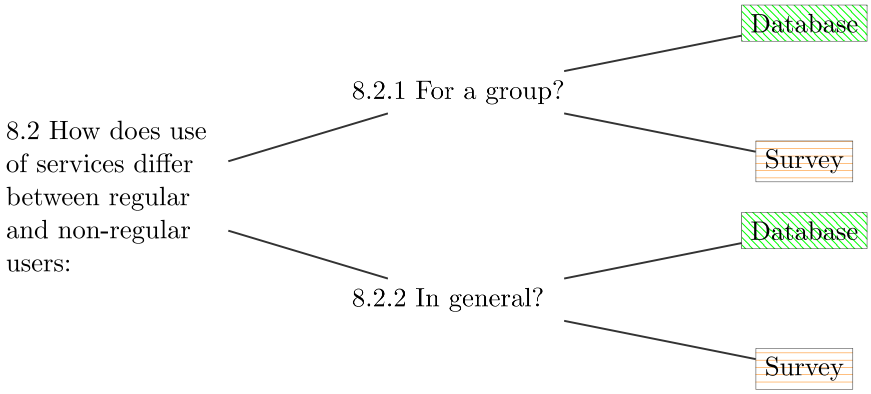

8.2 How does use of services differ between regular and non-regular users:
8.2.1 For a group?
Database
Survey
8.2.2 In general?
Database
Survey

Figure 4.10: Breakdown Question 8.2

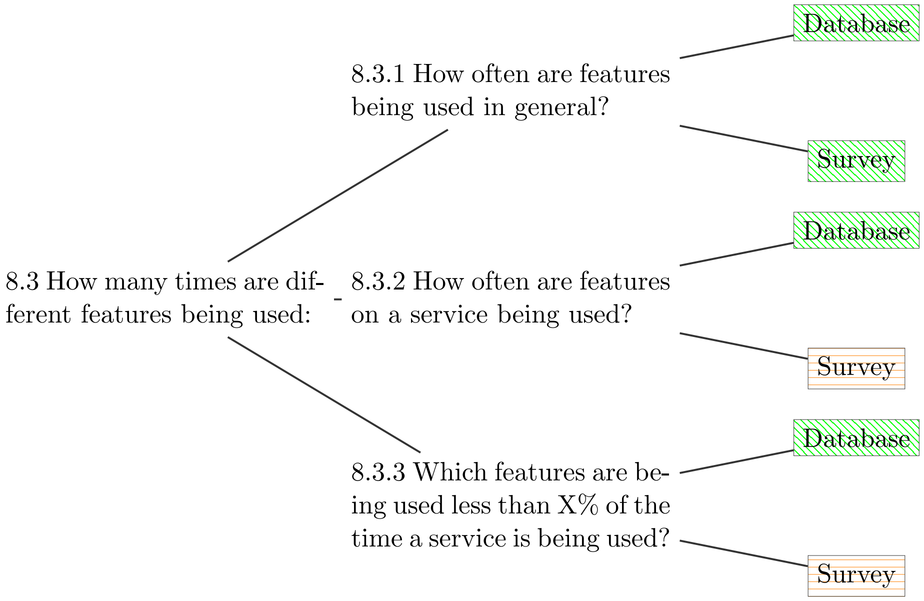

8.3 How many times are different features being used:
8.3.1 How often are features being used in general?
Database
Survey
8.3.2 How often are features on a service being used?
Database
Survey
8.3.3 Which features are being used less than X% of the time a service is being used?
Database
Survey

Figure 4.11: Breakdown Question 8.3

Due to the level of detail that an automatization of data gathering can provide it is feasible to answer most of the questions using data mining, which is seen above. The reversed situation applies for the survey, as it provides a lower level of detail it is harder to answer the questions.

5 Solution

This chapter presents, AUTA, the solution to the problem presented in section 1.1. It comprises of automatically gathering usage data and analyzing that data using data mining techniques. The design and implementation of automatic gathering of usage data, or usage tracing, will first be described. Thereafter, the data mining implementation will be presented.

5.1 Usage Tracing

The act of gathering data from some of the users interaction with a system is in this paper referred to as usage tracing. The gathered data is a collection of traces, where a trace is a record of a user's interaction with the system.

The process of developing usage tracing included three steps. Firstly, determining what data that needs to be gathered in order to perform data mining. The reason this was done as an initial step for this case was, as mentioned in section 1.5, to meet an upcoming deadline. Secondly, selecting appropriate technology for implementing usage tracing. Finally, executing the implementation of usage tracing with the selected technology.

The usage tracing part of the solution, in chapter 5, was implemented during a time period of two weeks, about 160 man hours. Since the implementation was experimental, authors learned the AOP framework while implementing tracing, the actual time to implement the tracing was even less. During these two weeks usage tracing was implemented on about 140 UserControls by applying eleven different Pointcuts. The preceding subsections will in describe the different steps taken during development of usage tracing.

5.1.1 Selection of Data Points

A consequence of implementing usage tracing as an initial step was the uncertainty of which questions the data mining ought to answer and by that not knowing which data points that would be required. Ensuring that the usage tracing would gather data points that would be sufficient for the data mining was for this reason a primary goal. This goal was achieved through in-depth walkthroughs of the system, studying what kind of data was possible to gather and looking at what kind of data would be vital for the data mining. The strategy was to add a column for a data point when in doubt of whether it would prove valuable. The reason for this strategy was that the addition of an extra column in the database table would not contribute to noticeable increase of response times. Furthermore, this approach decreased the risk of excluding data points which might have proved valuable during the data mining part of this study.

It was determined that the following data would provide sufficient information to capture user interaction:

”Service & type of service”	Name of a service available for navigation.
”User ID”	Identifier of a user.
”User action”	Name of buttons, lists and etc. used in the system.
”User role”	The roles of the user working with the case product.
”Session ID”	Identifier of a web session.
”Timestamp”	Time and data of executed action in case product.
”Execution time”	Time for the system to respond to a users action.

The data points above had to be examined for how they could contribute to the data analysis. Since the aim in this study is to look at usage of user controls and features in those controls the ”Service & type of service” and ”User action” data points were the most essential ones. The ”Service & type of service” was determined sufficient to cover all navigations in the system and user actions were covered by the data point ”User action”. Other data points that were selected were viewed as supporting data points, which when used together with either or both of ”Service & type of service” and ”User action” could help in gaining deeper insights.

As different users could behave very differently depending on their level of experience with computers and the given case product, ”User ID” had to be included. Users in different roles would most probably also perform different sets of task. By knowing which those sets of tasks were it would be possible use that information to customize the UI design for different roles and due to this the data point ”User Role” was included. The data point ”Session ID” give the ability to identifying a session and would for example give hints on whether users perform a set of tasks or a single task when they are using the product. The ”Timestamp” data point would possibly give insights to when users most often use the product in general but also specifically for different services. If users are often using a service after office hours it might be valuable for them to use that service when not being able to use a computer, for example during a commute. For that reason the service would be a candidate service to include in a version for handheld devices. The data point ”Execution time” could show which services involve requests that take long time to serve. If requests take long time to serve, then the service should be revised for possible improvements.

All other possible data points were excluded, most of them intentionally. Unintentionally, the result size of requests was excluded which could have given a notion of how much data users viewed in a service. Similar notion could be gained by looking at whether users were using the next button to see next page of search results after performing a search.

5.1.2 Selection of Technology

Since Google Analytics© is by far the most popular analytics technology [39] it had to be considered when selecting a technology for the gathering of usage data. It seemed like a promising alternative the but due to fact that it uses JavaScript to send usage data to its servers meant that it could not be used. This design had two major flaws for the type of application and type of tracing that this study is concerned with. First of all, much of the data needed will not be available to client-side scripts like JavaScript. Second of all, continuously sending data from an intranet website to servers of a third-party solution would most certainly not be accepted by customers. Due to these two flaws a custom built usage tracing implementation was determined to be a better solution.

The main contributing factor for how a custom usage tracing solution should be implemented was the size of the the product which is tens of thousands lines of code, as mentioned in section 1.4. The regular approach of implementing tracing at each execution point where tracing is needed would have proven a very tedious task. Such implementation would also have violated separation of concerns and heavily affected the ongoing development. Since AOP, which was presented in chapter 2, maintain separation of concerns and since the tracing could be implemented in Aspects, the effect on the ongoing development would be kept at a minimum. For these reasons AOP was selected as the appropriate approach for implementing usage tracing.

With AOP selected the next, and last, question was which AOP framework to use. The case product presents its pages using classes, refereed to as user controls, which inherit from the class UserControl in the ASP.NET Framework [40]. A user control handles all user actions for a specific part of a page and communicates with internal components to achieve different tasks. The selection of AOP framework was affected by the fact that user controls inherit from UserControl. One AOP framework, Spring .NET [19], had been developed using the Proxy-pattern [20] and since a UserControl is not defined with a clear interface which can be mimicked, that framework could not be used. The framework PostSharp however did not rely on this pattern and was therefore selected for the implementation. Also, it is a framework that continues to have a high development frequency.

5.1.3 Implementation with AOP

The implementation of usage tracing comprises of Aspects which trace user actions, a repository class that handle the database interaction and a database, see figure 5.1 which depicts the core structure these elements. The event of a user interaction that trigger a method in a user control will execute an Advice for that interaction. This Advice will gather the necessary data and send it to the repository, ”TraceRepository”, which stores the data in a database table. It should be noted that this structure is mainly used to maintain consistency with the rest of the code base.

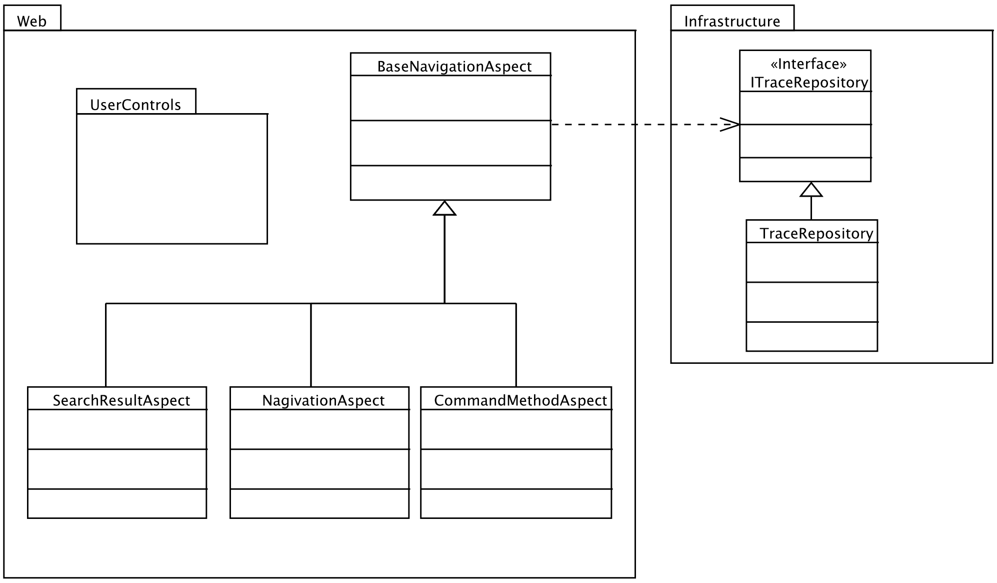


Figure 5.1: Structure of Aspects and repository type in tracing solution. In the ”Web” namespace the Aspects for tracing, which monitors classes in the ”UserControls” namespace, are implemented. The ”Infrastructure” namespace contains the repository class which handles all communication with the database.

Three Aspects are implemented into this solution to gather the necessary data, all of them are shown in figure 5.1 and with more detailed structure in figure 5.2. The Aspects are:

- ”NavigationAspect”, which is used for tracing navigations in the product and for user actions like clicking on a button, selecting items in a list and etc.
- ”SearchResultAspect”, which gather usage data when the search was made using a control called SearchResult.
- ”CommandMethodAspect”, which gather usage data when user used commands in different UI collection objects.

The three Aspects derive from the same base aspect, ”BaseNavigationAspect” which contain functionality shared between the three Aspects. The ”OnEntry” and ”OnExit” methods in the three Aspects all directly call the base class. The base Aspect then performs gathering of all common data points before calling the ”FillMethodAndWizardStep” method on the derived Aspects. This method gather data specific to each derived Aspects, including at least what method was executed and which step in the workflow a service is on.

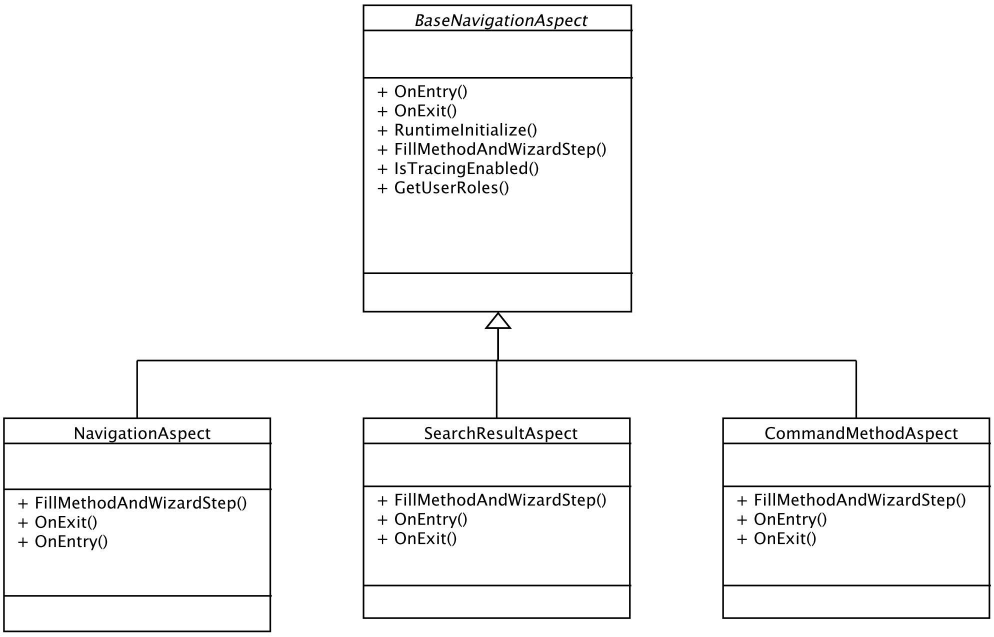


Figure 5.2: Structure of the Aspects in the solution. Shows the derived Aspects ”NavigationAspect”, ”SearchResultAspect”, ”CommandMethodAspect” and the base Aspect ”BaseNavigationAspect”.

Aspects used in this solution are all method boundary type Aspects, where the Advice is applied before and after execution of a method. Moreover, Advices are only applied to methods in a specific namespace and Pointcuts are based on naming conventions, i.e. they rely on the use of letter casing and keywords. What method an Advice is applied to is thereby determined by the use of wildcards, asterisk, and naming conventions for specific types of methods. For example, all buttons should trigger methods that contain ”Click” and therefore ”*Click*” is used. Listings 5.1, 5.2 and 5.3 shows examples of pointcuts. For these examples it should be noted that listing 5.1 and 5.2 have AttributeTargetTypes set to ”Accelerator.*.UserControls.*”.

Listing 5.1: Example of a pointcut for ”NavigationAspect” which is being applied to methods that have ”Click” in them. And, that apply to classes inside a namespace that begin with ”Accelerator” and ends with ”UserControls”.

```
[assembly: Accelerator.Web.Trace.NavigationAspect(AttributePriority = 5,
    AttributeTargetTypes = "Accelerator.*.UserControls.*",
        AttributeTargetMembers = "*Click*")]
```

Listing 5.2: Example of a Pointcut for ”CommandMethodAspect”. This Pointcut is applied to any method which contain the word command and to classes in the same way as listing 5.1

```
[assembly: Accelerator.Web.Trace.CommandMethodAspect(AttributePriority =
    10,
    AttributeTargetTypes = "Accelerator.*.UserControls.*",
        AttributeTargetMembers = "*Command*")]
```

Listing 5.3: Example of a Pointcut for "SearchResultAspect" being applied to a specific class and method. Note that the "AttributeTargetTypes" and "AttributeTragetMembers" does not end with a wildcard.

```
[assembly: Accelerator.Web.Trace.SearchResultAspect(AttributeTargetMembers
    = "OnSelectedIndexChanging",
 AttributeTargetTypes = "Accelerator.Web.ServerControls.SearchResult")]
```

By default the usage tracing is disabled so that it can be included in the standard solution while waiting for customers consent of activation. To allow instant activation once a customer have given their consent a tag in a configuration file is used to switch between active and inactive tracing, the tag is shown in listing 5.4. By setting the value to "True" instead of "False" the Aspects instantly begin to trace user interaction. This solution of using a tag in a configuration file mean that no new deployment is required after the customer has accepted to turn on usage tracing.

Listing 5.4: XML tag in configuration file used to enable and disable usage tracing

```
<add key="NavigationTracingEnabled" value="False" />
```

The "TraceRepository" is implemented with principles of the repository-pattern [41] to decouple the Aspects and the storing of data. Apart from simple storing of data in the database the repository also included functionality to determine whether a page load should be stored. When a user click a button or other action which does not trigger a navigation the "NavigationAspect" will still be triggered for loading a page. This is due to the fact that the “Page_Load“ method is called every time the page is updated and the same apply for user actions such as clicking the search button, the page is reloaded even though there is no navigation to a new page. By not storing “Page_Load“ when a page is just reloaded all “Page_Load“ rows in the database table represents navigation in the system.

A separate database is used for this solution since its objectives are very different from that of the main database. For the needs of usage tracing a single table is enough to cover all data points. An overview of the table can be viewed in table 5.1, below follows an explanation of each column:

- "StartTimeStamp", relates to the data point "Timestamp" and represents the date and time that a usage tracing event was triggered.
- "ExecutionTime", relates to the data point "Execution Time" and represents the time to handle the user request.
- "SessionID", relates to data point "Session ID" and represents a unique identifier for a web session.

Column name	Data type	Data point
ID	int	-
StartTimestamp	datetime	Timestamp
ExecutionTime	int	Execution Time
SessionID	varchar(250)	Session ID
UserID	varchar(250)	User ID
WizardStep	int	Service & type of service
Name	varchar(250)	Service & type of service
ControlName	varchar(250)	Service & type of service
Method	varchar(250)	User action

Table 5.1: This table shows the structure of the tracing table in the tracing database and which data point each column is related to.

- ”UserID”, relates to the data point ”User ID” and contains a unique identifier for a specific user.
- ”WizardStep”, is part of the data point Service & type of service and represents the index of a workflow on a service.
- ”Name”, is part of the data point Service & type of service and represents the type, subtype and name of a user control.
- ”ControlName”, is part of the data point Service & type of service and represents the user controls name in the code base.
- ”Method”, relates to the data point ”User action” and represents the method that is executed.

5.1.4 Example: User Action to Database Storage

This subsection will present an example describing the process from when a user performs an action to that a trace is inserted into the database table. Figure 5.3 shows the coordination between handling of user action and execution of tracing Aspects and figure 5.4 depicts the flow of usage tracing when an user action is performed. Note that the figures reflect the compiled language after aspect weaving has been performed. In this specific case the user navigates to a page where she can order software and in order to find the item that she intend to order she performs a search. After typing the name of the item she clicks on the ”Search” button that starts a flow of executions in the system.

When the user clicks on the search button that triggers the method in the user control called ”SearchButton_OnClick”. But, before executing that method the Advice

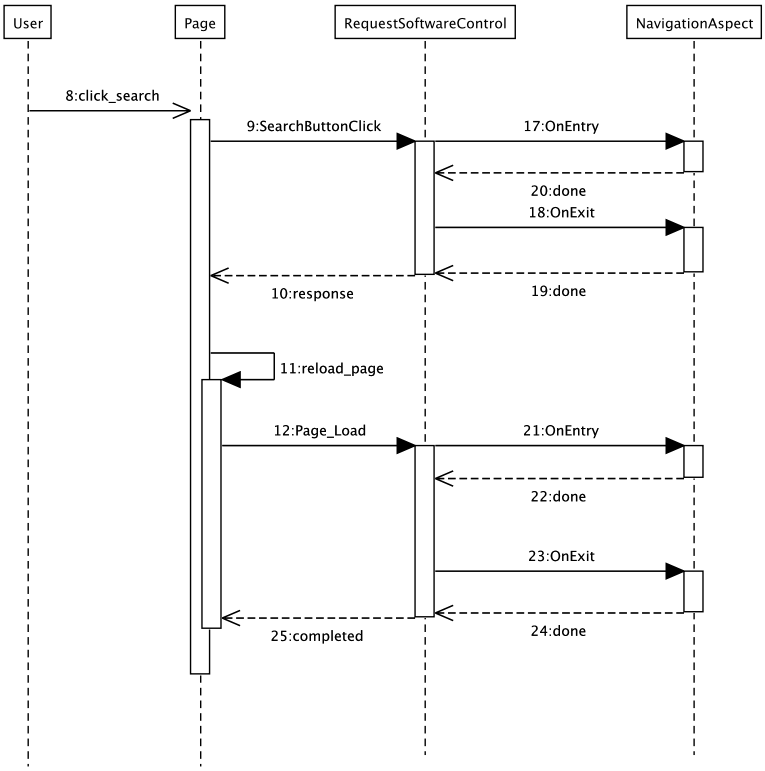


Figure 5.3: The flow and synchronization of execution between a user control, RequestSoftwareControl, and an Aspect, NavigationAspect. This sequence diagram reflects flow of the complied language.

”OnEntry” in ”NavigationAspect” is executed. The reason for this is that the ”NavigationAspect” has a Pointcut with ”AttributeTargetMembers” set to ”*Click*”. Furthermore, Advices for execution before and after the execution of any method containing ”Click” are present. Therefore the ”OnEntry” Advice is executed before the execution of the ”SearchButton_OnClick” method and the ”OnExit” Advice is executed afterwards.

When ”OnEntry” is executed it calls the ”OnEntry” method on base class which starts a timer to record the length of the request. After that the ”OnEntry” method finishes and the code in the Search method is executed, querying the database for items matching the search term. When query is completed the method updates the list in the view with the matching items and the method is exited.

With ”SearchButton_OnClick” finished the next Advice, ”OnExit”, is executed. This

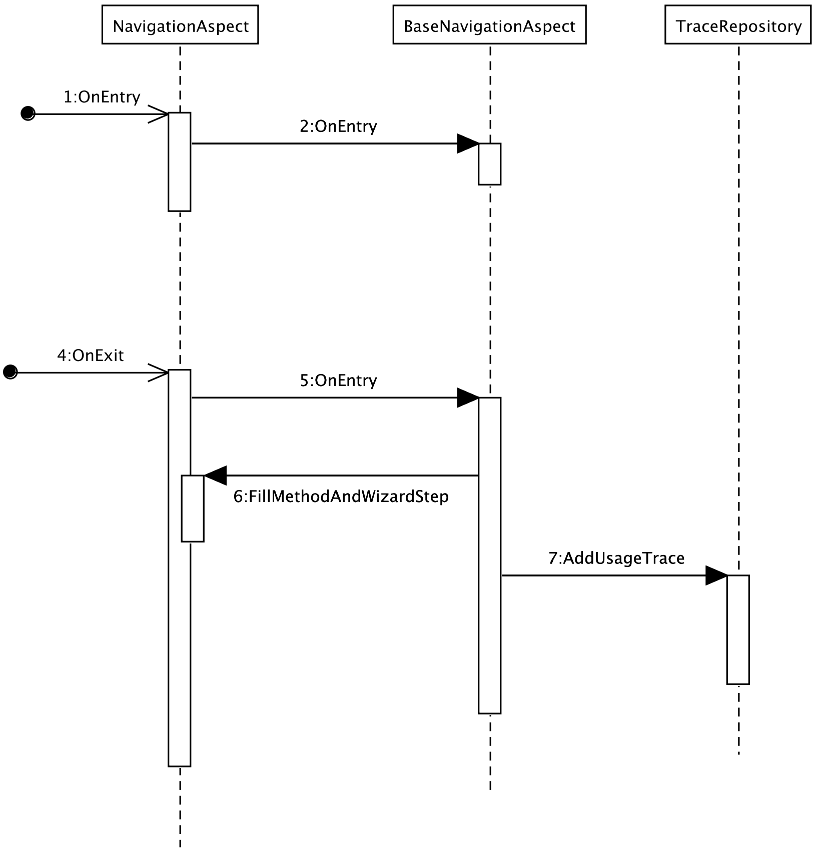


Figure 5.4: The flow of execution for when an event triggers the NavigationAspect.

Advice will call the same method on the base class which gather necessary usage data that is common for all Aspects, call the ”NavigationAspect” to fill in Aspect specific data, stop the running timer and call the ”TraceRepository” to save the usage data.

Once the usage trace has been stored the Advice finishes and an update of the page will be triggered, which in turn calls the “Page_Load“ method. Since ”NavigationAspect” also has a Pointcut for ”Page_Load” the ”OnEntry” and ”OnExit” Advices will be executed again. This time the ”TraceRepository” will notice that the latest instance of “Page_Load“ for the provided ”SessionID” has the same ”ControlName” and ”WizardStep” as the not yet persisted instance sent to ”TraceRepository”. And, because of this the trace instance will be discarded instead of inserted in to the database table. When all executions for this user event are completed the database table will consist of one more entry representing the search click that the user performed.

5.2 Data Mining

Data gathered from usage tracing must be analyzed and presented in such a way that the development organization can gain valuable insights. In this study it is done through different data mining techniques. The techniques that have been used can be divided into three areas; outlier detection, cluster analysis and aggregation. Outlier detection and cluster analysis has been described in the fundamentals chapter. Aggregation is used as a collection name for techniques related to summarization and grouping in different ways and therefore not in need of deeper analysis.

Two of the Breakdown Questions, in chapter 4, were due to lack of sufficient data not implemented. Breakdown Question 4 (Trends between product versions), see figure 4.4, has not been implemented since it is concerned with trends which would require usage tracing data from different versions of the software. Breakdown Question 5.4 (Size of search result), see figure 4.5, as there was no data point gathering data on the size of search results. Data for whether the pager was used existed but that would not be sufficient since the pager would not be used when the item the user is looking for is on the first search result page.

The main purpose of this study was to see if AUTA can replace the use of web survey. Therefor all those Breakdown Questions that could not be answered by a web survey and applicable to the correlation evaluation were not prioritized. Two Breakdown Question were due to this not implemented, Breakdown Question 8.3.3 (Features used less than X within a service) and Breakdown Questions 5.3 (Utilization of internal search within a workflow for a services).

The data mining solution was developed as a framework in .NET. The framework consists of two libraries; a main library written in the object-oriented programming language C# and a library in the functional programming language F#. The F# library is used for computations which requires a lot of math and is used by the C# library. This data mining framework only computes values for answering the breakdown questions presented in chapter 4. Visualization is not part of the framework but any client application, being a web site or a desktop application, could leverage this framework. For this study a console application was used to extract comma separate files for each question. Using Excel, visualization in form of charts where made using these files.

The remainder of this section will describe how each Breakdown Question was implemented, table 5.2 and 5.3 give a summary of which and how each Breakdown Questions have been implemented. For each Breakdown Question results from execution of the data mining framework on usage tracing data, is presented. The data used for the results was collected from a system used by testers at the software development organization. Hence, the result is merely an indicator that the implementation provide correct type of result but does not give any indication of end-users perception of the system.

5.2.1 Breakdown Question 1

Breakdown Question 1 (Time for a user to complete a task), see figure 4.1, was implemented specifically for services with workflows, where the last action for completing the

BQ	Question	Input	Approach	Output
1.1	Average, median and variance for completion times	Completion times	Aggregation	Average, Median and Variance for completion times
1.2	Which services are outliers for completion times	Completion times	Outlier detection	Significant outliers
2.1	Time of the day services are used	Time stamps grouped by services	Cluster Analysis	Hierarchical Clusters of time of use
2.2	Time of the day the product is used	Time stamps	Cluster Analysis	Hierarchical Clusters of time of use
3.1	Bounce rate for services	Loadings of a service	Aggregation	Bounce rate per service
3.2	Bounce rate in general	Loadings of a service	Aggregation	Bounce rate in general
4	Trends between product versions	—	—	—
5.1, 5.2	Does user prefer catalog or search	Catalog and search traces	Aggregation	Preference of the two options
5.3	In percent how often does a user search within a workflow	—	—	—
5.4	Search result too large for one page	—	—	—
6.1	Times a service is cancelled after more than one step	All traces	Aggregation	Services and number of cancellations
6.2	Times a service is cancelled when only submit left	All traces	Aggregation	Services and number of cancellations
6.3.1	Users drops off more than once on for the same service	All traces	Aggregation	Services and number of recurring drop offs
6.3.2	Session end after drop off	All traces	Aggregation	Services and number of recurring drop offs

Table 5.2: A summary of the implementation of child questions to Breakdown Question 1, 2, 3, 5 and 6. Display the Breakdown Question number (BQ), what the question aim to answer, the input to the implementation, the implementation approach and the output of the implementation. Rows with ”—” have not been implemented, for reasons discussed above.

BQ	Question	Input	Approach	Output
7.1	Longest time between steps in a service for a session	All steps for a service	Aggregation	Average longest step time
7.2	Services completion time per steps	Steps & completion time	Aggregation	Completion time per steps
8.1.1	How often is a service visited	All loadings of services	Aggregation	Visited count for each service
8.1.2	How often is a service used by a specific groups	All loadings of a service	Aggregation	A count for each service and user group
8.1.3.1	Which services are used less than X%	All loadings of a service	Aggregation	Services used less than X%
8.1.3.2	Which services are for a group used less than X%	All loadings of services	Aggregation	Services used less than X% for each group
8.1.4.1	How many different services a user use overall	All loadings of services	Aggregation	The number of services each user use
8.1.4.2	How many different services a user use per session	All loadings of services	Aggregation	Average number of services each user use per session
8.2.1	Regular and non-regular use behaviors	All loadings of services	Aggregation	Times the different groups use each service
8.2.2	Regular and non-regular use behaviors	All loadings of services	Aggregation	Times the different groups use each service
8.3.1	Features being used in general	Feature traces	Aggregation	How often features are being used
8.3.2	Features being used for a service	Feature traces for a service	Aggregation	Features usage for each service
8.3.3	Features used less than X%	—	—	—

Table 5.3: A summary of the implementation of child questions to Breakdown Question 7 and 8. Display the Breakdown Question number (BQ), what the question aim to answer, the input to the implementation, the implementation approach and the output of the implementation.

workflow is a submit. From the usage tracing data all loadings of services and submits are queried from the database containing usage tracing data and order by session and timestamp. The page load, which marks the beginning of a service, is paired with a submit to form a completion. The completion time is then computed by subtracting the timestamp of the submit with the timestamp of the page load.

With completion time calculated different statistic methods are used to answer Breakdown Question 1.1 (Time to complete service) and 1.2 (Outliers regards to completion time for services). For question 1.1 the implementation computes the average, median and the standard deviation of completion time for each service. For question 1.2 the procedure is not as straightforward as it tries to find service that completion times stand out among other services. Determining which services thare are outliers is done by running an outlier detection algorithm that uses multiple pairwise comparisons. Also, ideas from distance-based outliers, discussed in subsection 2.2.2, have been used by requiring that an outlier must be significantly larger than t of the other observations. The result of the algorithm is a set of services which are significant outliers. As the outlier detection implementation require a lot of mathematical computations the implementation was made in F#, part of the implementation can be found in listing 5.5. During test runs of the algorithm it was noticed that the type I error level had to be increased, since no outliers where found at $\alpha = 0.05$. Therefore, the level was increased until a satisfying result was achieved, which was at $\alpha = 0.2$. Furthermore, since the correct level for α had to be found by incrementally increasing α to an appropriate level there would be no value of decreasing it again by using methods to adjust for the family wise error. The outlier detection algorithm contain the following steps:

1. Set the threshold t indicating how many services a service must have significantly greater completion time than to be considered an outlier. This is sent as input to the algorithm on line 12 in listing 5.5.
2. For each list of completion times for a service compare with all other lists.
   (a) Sort the two lists and give both lists a ranking, taking ties into account when ranking, see line 22 in listing 5.5.
   (b) Make a significance test, method call on line 23, using Wilcoxon signed rank test, implemented on lines 1-9 in listing 5.5. The in parameter of 0.2 is the type I error value.
   (c) Increment a counter if signifigance test returns true, see line 24 in listing 5.5.
3. If counter larger than t then add service to a list of outliers, see line 25-26 in listing 5.5.

Listing 5.5: Question 1.2 implementation

```
1 let signifigant(t:(float * float), n1, n2, a) =
2     let EU = (float (n1 * (n1+n2+1)) / 2.0)
3     let z = ((fst t) - EU) / (sqrt (float (n1*n2*(n1+n2+1)) / 12.0))
```

```
    let p = (1.0 - getCriticalValue(z)) * 2.0

    if p < a then
        true
    else
        false


let SignifigantOutliers(threshold:int,
   tuples:System.Collections.Generic.List<(string *
   System.Collections.Generic.List<float>)>) =

    let outliers = new System.Collections.Generic.List<string>()
    let mutable t = (0.0, 0.0)
    for tuple1 in tuples do
        let mutable count = ref 0
        for tuple2 in tuples do
            if tuple1 <> tuple2 then
               let list1 = List.ofSeq (snd tuple1)
               let list2 = List.ofSeq (snd tuple2)
               t <- sortAndRank(list1, list2)
               if signifigant(t, list1.Length, list2.Length, 0.2) then
                incr count
        if count.Value > threshold then
            outliers.Add(fst tuple1)

    outliers
```

Result

For Breakdown Question 1.1 (Time to complete a service) the mean, median and standard deviation of the completion times for each service is shown in figure 5.5. On particular service, ”NOBT_RequestSoftware”, stands out when it comes to the median and standard deviation. It should be noted that the result is affected by how testers during the time of data collection were focusing the testing effort on some of the services.

Few outliers were found by the algorithm implemented for Breakdown Question 1.2 (Outliers regards to completion time for services). Of the services in figure 5.5, ”IM_RequestUser”, ”IM_RequestComputer”, ”IM_ManageUser” and ”IM_ReinstallComputer” were marked as outliers. It should be noted that the services determined to be outliers does not have bars that are consistently much larger or smaller than other services.

5.2.2 Breakdown Question 2

To answer what time of the day different tasks are carried out Breakdown Question 2 (Time of day that tasks are carried out), see figure 4.2, was implemented using cluster analysis. The reason for using cluster analysis was that the goal was not to see the frequency at any given time but to see which points during the day that had most activity. The idea is that product use, that is close in time, form clusters which show

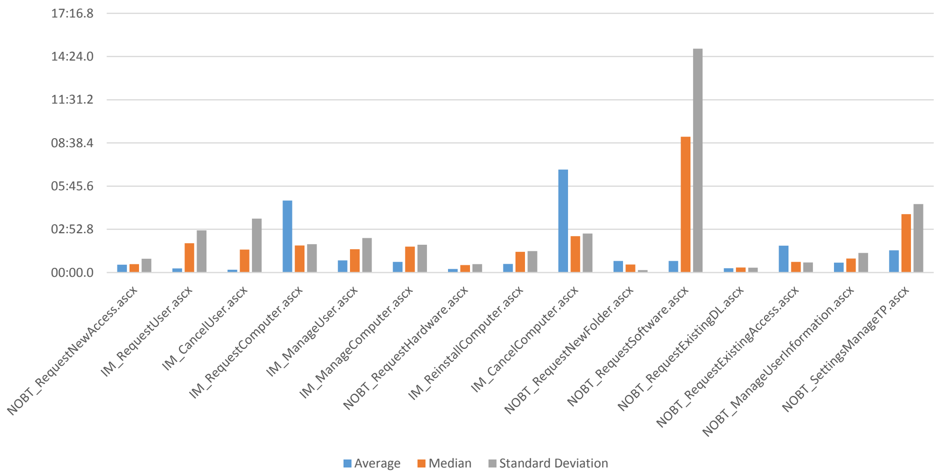


Figure 5.5: Chart showing the mean, median and standard deviation for services that have been used at least 6 times.

when users tend to use the product. The algorithm for the cluster analysis is based on the Ward-Agglomerate Hierarchical clustering method, see subsection 2.2.1. The deviation from this method is the calculation of the distance which is defined as $|c_1 - c_2|$, where c_1 and c_2 are the cluster's centroid, the centroids are here representing a point in time of a day. The core of the clustering algorithm can be found in listing 5.6. The implementation of the clustering algorithm contains the following steps.

1. Let the clusterSet be a set of time stamps and let distMatrix be a distance matrix containing the distance between all time stamps.

2. Find the two elements, A and B, that are closest, i.e. having lowest value in the distance matrix.

3. For the two elements create a new parent element and calculated its height and centroid.

4. Put the parent into the distance matrix and remove the two elements A and B.

5. If there are less than 2 elements left in the set of maximal clusters make a top cluster containing the two element that are left. Else repeat step 2-4.

In order to answer Breakdown Question 2.1 all completions are collected and sent to the clustering algorithm. For Breakdown Question 2.2 the only difference is that only completions for a specific services is sent to the clustering algorithm.

Listing 5.6: Question 2 implementation

```
let rec clusterUpdate(clusterSet:List<ClusterNode>,
    distMatrix:Matrix<float>) : List<ClusterNode> =
         let (a,b) = distMatrix.Dimensions;
         // find clusters closets to each other
         let (index1, index2) = FindShortestDistance(distMatrix)

         // merge into parent
         let item1 = set.Item(index1)
         let item2 = set.Item(index2)
         let newCentroid = CentroidMerge item1.Centroid
             ((int64)item1.Count) item2.Centroid ((int64)item2.Count)
         let height = WardDistance(item1, item2);
         let parent = new ClusterNode(item1, item2, new
             System.TimeSpan(0,0,0), null, height, newCentroid, item1.Count
             + item2.Count)


         // put parent into maximal cluster set and remove child clusters
         let mutable parentIndex = index2
         let matrix = Matrix.create (a-1) (a-1) 0.0
         let mutable newMatrix =
             match index1 with
             | 0 -> distMatrix.Columns(1, a-1).Rows(1,a-1)
             | i when i = a-1 -> distMatrix.Columns(0, i).Rows(0,i)
             | _ -> // Take everything except the column and row
                 representng index1
                      matrix.[0 .. index1-1, 0 .. index1-1] <- distMatrix.[0
                          .. index1-1, 0 .. index1-1]
                      matrix.[index1 .. a-2, 0 .. index1-1] <-
                          distMatrix.[index1+1 .. a-1, 0 .. index1-1]
                      matrix.[0 .. index1-1, index1 .. a-2] <- distMatrix.[0
                          .. index1-1, index1+1 .. a-1]
                      matrix.[index1 .. a-2, index1 .. a-2] <-
                          distMatrix.[index1+1 .. a-1, index1+1 .. a-1]
                      matrix




         let updatedMaximalClusters = HelperModule.remove index1
             (HelperModule.replace parent parentIndex set)
         parentIndex <- parentIndex - 1; // Update after index1 item was
             removed


        if(updatedMaximalClusters.Length <= 2) then          //Must not be
            smaller than 2!!
             let item1 = updatedMaximalClusters.Item(0)
             let item2 = updatedMaximalClusters.Item(1)
             let newCentroid = CentroidMerge item1.Centroid
                 ((int64)item1.Count) item2.Centroid ((int64)item2.Count)
             new ClusterNode(item1, item2, new System.TimeSpan(0,0,0),
                 null, WardDistance(item1, item2), newCentroid, item1.Count
```

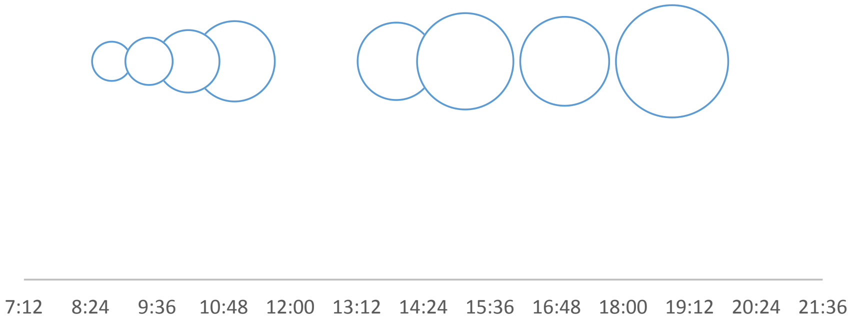


Figure 5.6: Bubble chart of timestamp clusters, where each bubble represents a cluster which origin is the clusters centroid and the radius represents the clusters height.

```
                        + item2.Count)
            else
                let updatedDistMatrix = updateDistanceMatrix(parentIndex,
                    updatedMaximalClusters, newMatrix)
                clusterUpdate(updatedMaximalClusters, updatedDistMatrix)
```

Result

In order to extract a result from Breakdown Question 2.1 (Time of day that tasks are carried out) the data mining framework was queried to find the eight largest clusters and export their centroid, maximal child centroid, minimum child centroid and the cluster's height. The clusters are visualized using a bubble chart where the center of each bubble is the centroid and the radius represents the height of the cluster. By using the height as radius it is possible to show the magnitude of each clusters.

Using the test data from testers the results shows eight bubbles of varying size, see figure 5.6. It seems that tester are not testing between 12 and 13, which is lunch hour. Furthermore, the testing activity is more frequent after lunch than before lunch. It is also shown that no testing is done before office hours but there is a lot of activity after office hours until around 20:30.

5.2.3 Breakdown Question 3

Breakdown Question 3 (Product Bounce rate) in figure 4.3, related to whether users are able to find what they are looking for in the system, is implemented using aggregation. Since the only difference between Breakdown Question 3.1 (Product Bounce rate in different services) and Breakdown Question 3.2 (Product Bounce rate in general) is that 3.2 is the combined bounce rate for all services, the implementation is done in one method. For answering these questions all possible bounces are first gathered, a possible bounce is defined as the load of a new service. Then the possible bounces where a user switched to a new service within ten seconds is marked as being a bounce. The detailed description of the steps in the implementation can be found below.

1. Get all services that have been loaded.
2. Sort all services timestamp and session.
3. For each service loading:
   (a) If the next loading is for the same session and the difference in time of loading is less than ten seconds, register the service loading as a bounce.

Result

The bounce rate result from data mining of testers usage tracing data is shown in figure 5.7. ”NOBT_RequestNewAccess” has by far the highest bounce rate, noticeable is that this service is also the one with most completions. For this case it is only possible to conclude that ”NOBT_RequestNewAccess” was used a lot during the testing which resulted in a high bounce rate. For a real case this result would have indicated that users often go to ”NOBT_RequestNewAccess” without starting to use the service. With such indication further investigation of the data could be performed to find out why this services have such a high bounce rate.

5.2.4 Breakdown Question 5

The product in this study provides two ways of finding items in services, search or catalog view. Breakdown question 5 (Utilization of internal search), see figure 4.5, is concerned with how these are being used.

For Breakdown Question 5.1 (Utilization of Product Tree/Catalogue for search) and 5.2 (Utilization of search by text for search) it was discovered that usage numbers are not as interesting as what the user want to do. If a user for some reason is forced to use one of the two approaches the data will reflect what the user is doing and not what she wants to do. In order to know what the user wants to do the implementation only register what a user did first when entering a service. The implementation, see listing 5.7, is the following; all traces of catalog and search uses are used as input, these are then grouped on session and service. For each group only the first trace is selected, i.e. what the user first tried to do. If the trace is a search than a counter for searches for

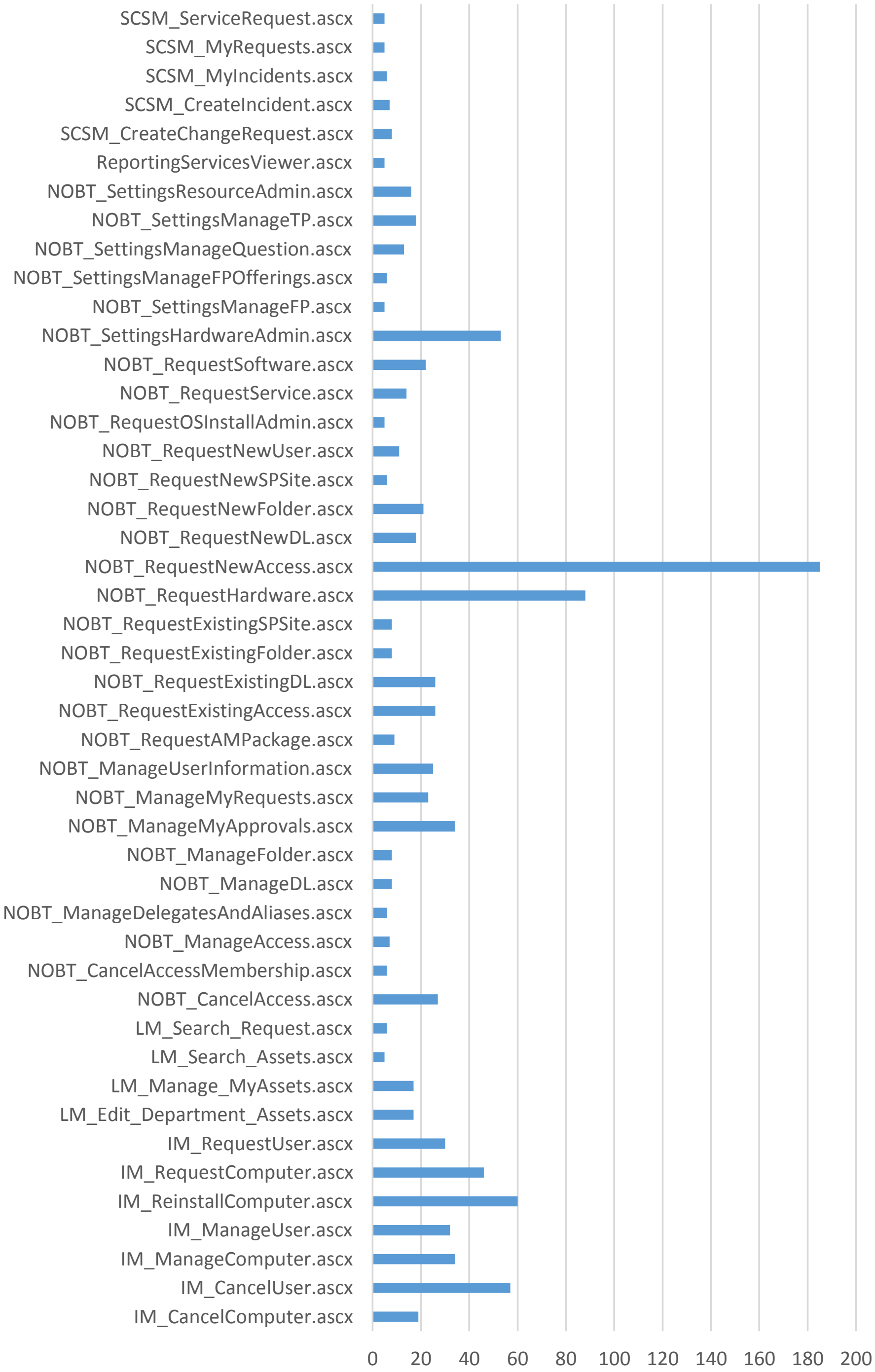


Figure 5.7: Chart showing the bounce rate for services with bounce rate of 5 or higher.

a user is incremented and vice versa. To determine if a user prefers one option to the other a threshold of 60% is used. When a user tend to use one option more than 60% of the time that is strong enough indication that they prefer that option. If both options are within the range 40-60% the user is marked as having no preference of how to find products.

For Breakdown Question 5.3 (Utilization of internal search within a workflow for a services) the same input is used as for Breakdown Question 5.1 and Breakdown Question 5.2. However, the set of inputs are grouped on Session, Service and WizardStep. Where WizardStep indicates which step in a service that a user searched on.

Listing 5.7: Question 5.1 and 5.2 implementation

```
public void ComputeSearchOrCatalogTendencies(ref Computation computation, IEnumerable<UserControlTrace> searchAndCatalogUses)
    {
        var list = new List<UserControlTrace>();

        // use of catalog for one use of a service should only be represented by as one use
        foreach (var item in searchAndCatalogUses)
            if (IsSearchAndServiceNotInListForSession(list, item) || IsCatalogAndServiceNotInListForSession(list, item))
                list.Add(item);

        var groupedList = list.OrderBy(u => u.StartTimestamp).GroupBy(u => new { u.SessionID, u.ControlName }, (key, group) => new { Key = key.ControlName, Group = group });

        var userPreferences = new Dictionary<string, int[]>(); // First integer in array for search and the other for catalog
        foreach (var group in groupedList)
        {
            var item = group.Group.First(); // Select what the user tried to do first

            if (!userPreferences.ContainsKey(item.UserID))
                userPreferences.Add(item.UserID, new int[2]);

            if (Regex.IsMatch(item.Method.ToLower(), searchRegex))
                userPreferences[item.UserID][0]++;
            else if (Regex.IsMatch(item.Method.ToLower(), catalogRegex))
                userPreferences[item.UserID][1]++;
        }

        computation.UserSearchOrCatalogPreferences = userPreferences;
    }
```

Result

From the results create from data mining of testers usage tracing it was found that all users, testers, preferred the search option. 12 difference system users had been used during the data collection and for all search was preferred. In reality the 12 system users were used by two testers, i.e. the result only shows that the two testers tended to prefer.

5.2.5 Breakdown Question 6

Drop offs can show where users have problem completing a service. Four questions were extracted from Breakdown Question 6 (Occurrence of drop offs). A drop off has been defined as when a user leaves a service after completing one or more steps. The method that is used to decide if a trace is a drop off is seen in listing 5.8. First it checks that the service has more than one step and then that the trace is not a submit trace. After this, that a step was taken and finally that the previous trace was not a submit.

Breakdown Question 6.2 (Service cancellation at last step) requires the knowledge of which step is the last before submit, for each service. For this there are two options. Either, the second last step for each service must be manually registered, or it must be assumed that a service was completed at least once, so that the second last step can be found in the usage tracing data. This option increases in validity as the use of the product increase and does not require manual work when new services are created. The second option was selected since there would be enough data for this approach and that it would be less time consuming.

The implementation for answering Breakdown Question 6.1 (Service cancellation after more than one step), 6.3.1 (User drop off more than once for the same service) and 6.3.2 (End of session after drop off) in 4.6 is grouped into one method. This method finds all traces that are drop offs and registers drop offs along with answers for the questions. A step-by-step description of this method is presented below.

1. Get all traces order first on session and then on timestamp.
2. Find the items that have a workflow with more than one step and group hem by service.
3. For all traces:
    (a) Take the current trace and the next trace.
    (b) If the trace is a drop off, see listing 5.8, crate a new drop off. For the new drop off mark the drop off as a session ending after drop off if the next trace is not for the same session.
    (c) Add new drop off to the set of drop offs.

Listing 5.8: Question 6 implementation

```
private bool IsDropOff(List<UserControlTrace> usedServices,
    IEnumerable<IGrouping<string, UserControlTrace>> wizardMethods, int
    index, UserControlTrace possibleDropOff, UserControlTrace
    nextPossibleDropOff)
{
    //If the current step contains a wizardstep grater then 0 it is a wizard and we shall check for a drop off
    // If there is a submit then the wizard was finished correctly and no drop off occurred
    //If next is not the same session or not the same method and previous was not a submit, then we have a drop off
    return HasWizardSteps(possibleDropOff.ControlName, wizardMethods) && !possibleDropOff.Method.Equals(”Submit”) &&
        ((!possibleDropOff.SessionID.Equals(nextPossibleDropOff.SessionID) || !possibleDropOff.Method.Equals(nextPossibleDropOff.Method)) && !usedServices.ElementAt(i − 1).Method.Equals(”Submit”));
}
```

Result

In the data from the test environment used by testers there existed quite a high number of drop offs. This is likely linked to how testers act when testing the system. Scenarios like what happens when a user do not complete a service are done by purpose which would not be the case for end-users. From the data mining a comma separated file with data per service on number of drop offs, number of users with multiple drop offs and number of session that end after a drop off was exported. From this it could be noticed that ”IM_RequestUser” and ”IM_ManageUser” had highest drop off rate, see figure 5.8. In an actual system this would be indications of service which could be improved for better user interaction.

It should also be noted that the bar representing ”End after drop off” is very small and most often none existent. This is result is probably due to the fact that the data used is from testers using the system, which do not end the sessions after that they have done what they would like to test. Instead they probably keep testing other parts of the system.

The results for the values ”Drop off count” and ”Multiple drop offs” are very alike and can be questionable. Therefore the raw data was analyzed and it was discovered that there are small variations, this can also be seen if looking closer to the graph. The reason that the result looks like this is that testers have used one user account heavily when they test the system. This mean that the result for the occurrence of drop off in most cases have multiple drop offs as testers, using the same account have tried a verity of scenarios.

For Breakdown Question 6.2 (Service cancellation at last step) it was discovered during implementation and testing that it would not be possible to get a correct result. For the case product the number of steps in some services depends on what is ordered. Hence, the last step in a service could differ between completions. So when the implementation tries to determine what the index for the second last step is it would only be able to determine it for one possibility.

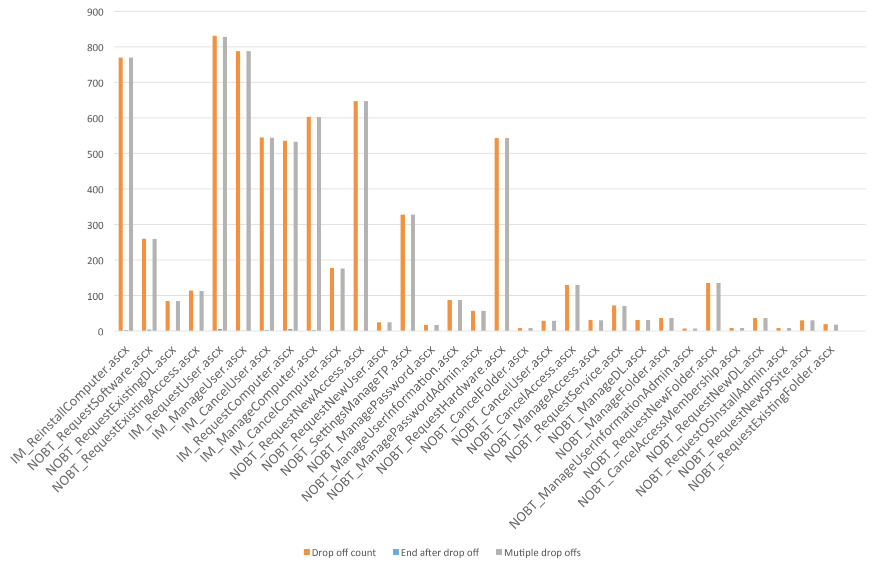


Figure 5.8: Chart showing for each services the number of drop offs, multiple drop offs and time session end after drop off.

5.2.6 Breakdown Question 7

Breakdown Question 7 (Most difficult service to complete), see figure 4.7, is concerned with finding where users have problems completing a service. The completion time only gives and indicator of how long in comparison it takes to complete a service but not whether one service is more difficult to complete than the other. By taking the number of steps into consideration it is possible to compare how easy different services are to complete.

The implementation for answering Breakdown Question 7.1 (Longest time between steps in a service) and 7.2 (Completion time per step for a service) is combined with the answering of Breakdown Question 1 (Time for a user to complete a task), seen in subsection 5.2.1, due to their shared similarities in computation. While iterating through completions the time between each step in a service is calculated along with the number of steps.

Result

Result of Breakdown Question 7, see figure 5.9, shows both child question 7.1 and 7.2. For Completion time per steps it can be noted that ”NOTB_RequestSoftware” is much higher than the other services but that the rest of the services have fairly similar val-

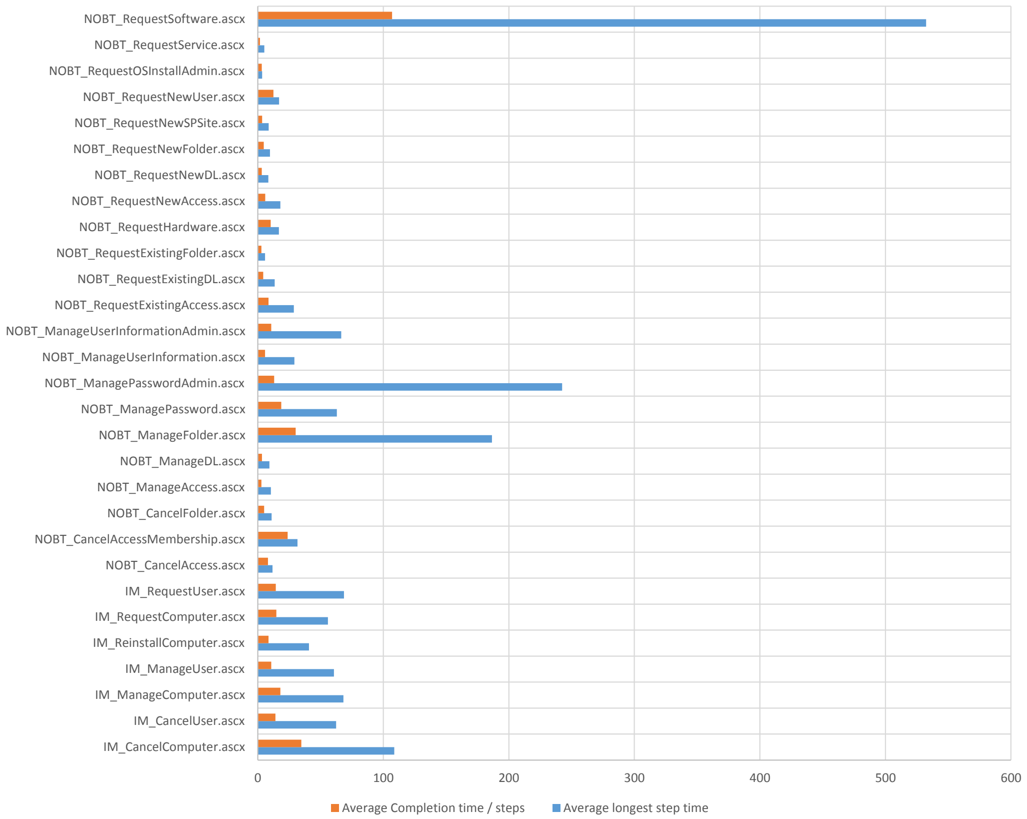


Figure 5.9: Show the average completion time per steps and average longest step time for services with with completions.

ues. The other bar, ”Average longest step time”, shows what the longest step for each completion of a service is on average. This bar shows if there is any particular step, in a service, that is time consuming. If both bars are at the same level for a service that mean that all steps have similar levels of time consumption. From the result it is depicted that ”NOTB_RequestSoftware”, ”NOBT_ManagePasswordAdmin” and ”NOBT_ManageFolder” have a high longest time per step compared to other services.

5.2.7 Breakdown Question 8

Breakdown Question 8 (Frequency of use for service and functions) was divided into three parts, see figure 4.8. The first part, 8.1 (Navigation to a service), is concerned with the extent of uses for service. The second part, 8.2 (Difference between regular and non-regular users), is concerned with differences between regular and non-regular users. The third part, 8.3 (Use of features), is concerned with how feature are used where features can be any actionable item on a page, for example buttons, links and etc.

Breakdown Question 8.1 (Navigation to a service), see figure 4.9, has four branches. The implementation for Breakdown Question 8.1.1 (Visit to service) and 8.1.2 (User groups/roles use of service) takes all traces for loading services and groups them per service. After that the implementation proceeds by looping through all the grouped set, for each counting the number of service loadings and number of service loadings per user role. Breakdown Question 8.1.3 (Services used less than X) also use all service loadings but the determine which services are used less than 5% of the time both in general, 8.1.3.1, and per role, 8.1.3.2. For Breakdown Question 8.1.4 (Services used per user) all services loadings are used to find how many services users use in general, 8.1.4.1, and per session, 8.1.4.2. This is done counting all service loadings per user and all service loadings per user and session.

For Breakdown Question 8.2 (Difference between regular and non-regular users), see figure 4.10, the implementation is based on regular users being users that that visit the system at least once every two days. Since Breakdown Question 8.2.1 (Difference between regular and non-regular users, regards to groups) and Breakdown Question 8.2.2 (General difference between regular and non-regular users) are quite similar they have been implemented at the same time. The steps in the implementation are described below.

1. Get all traces for loading services (X) and order them first by user and then by timestamp
2. From X extract all unique session.
3. For each unique session:
    (a) If the next unique session for the user is within two days add the user to the list of regular users.
4. Let all users not in the list of regular users be in the list of non-regular users.
5. For all traces count the number of services used for each regular and non-regular user.
6. For all traces count the number of services used for each regular and non-regular user per user role.

For Breakdown Question 8.3 (Use of features), see figure 4.11, the usage of features used by users to interact with the system is analyzed. For this implementation the name of features used is stored in a trace's element Method. In order to distinguish which features are used a set of regular expression strings are defined to find the different features that are available on services. They are defined so that the same feature will be found on different services even though the name of the feature might deviate. Essentially the expression are based on the Pointcuts in section 5.1. For example adding a product to the cart is found by "\\w*.Add\\w*". A complete description of the steps in the implementation of Breakdown Question 8.3.1 (General use of features) and Breakdown Question 8.3.2 (Use of features inside services) can be found below.

1. Get all traces which method matches one of the defined regular expressions
2. Group the traces on service.
3. For each set of service feature traces:
   (a) For each group of feature traces:
      i. Add trace to the corresponding feature group on the the service.
      ii. Add trace to the corresponding feature group for the system in general.

Result

From the first part of Breakdown Question 8 (Frequency of use for service and functions) many of the sub questions are concerned with the extent that the product and its services are used, both for all users but also for different user groups. Breakdown Question 8.1.1 provides a good overview of the general uses in the system and has therefore been selected as the one most interesting results to show. Charts where created for the top used services, from the usage data exported. From the top ten services, see figure 5.10, it is evident that "NOBT_RequestAccess" have been used the most and that it has almost been used the double amount of times than "IM_ManageComputer".

For Breakdown Question 8.3 (Use of features) it was discovered that all features are not comparable, more specifically a comparison would not give any valuable information. For example, the amount of searches compared with how many times the users clicks the next button in a workflow will not be of value. Comparing the use of next, previous and submit button however can yield some valuable insights on users flow through a service. Few previous clicks compared to next clicks can show that users easily go through a service without having to correct their input. Using the test environment data the data mining for Breakdown Question 8.3.1, see figure 5.11, shows that next clicks are orders of magnitude higher than previous clicks. This is mostly likely linked with the fact that the testers using the system are familiar with all service and seldom need to make corrections. Even though the number of previous clicks overall the diagram shows that there is some problem with "IM_ReinstallComputer".

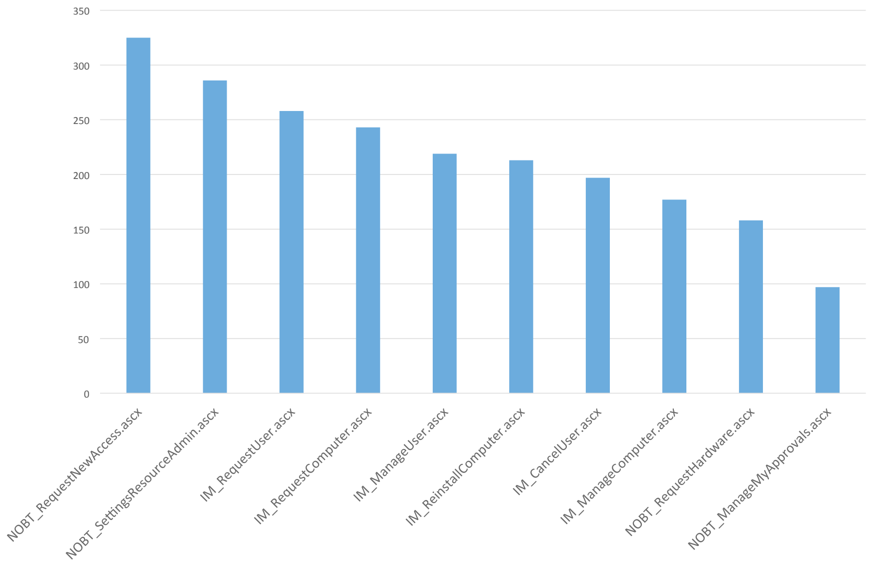


Figure 5.10: Chart showing the 10 most used services for the product.

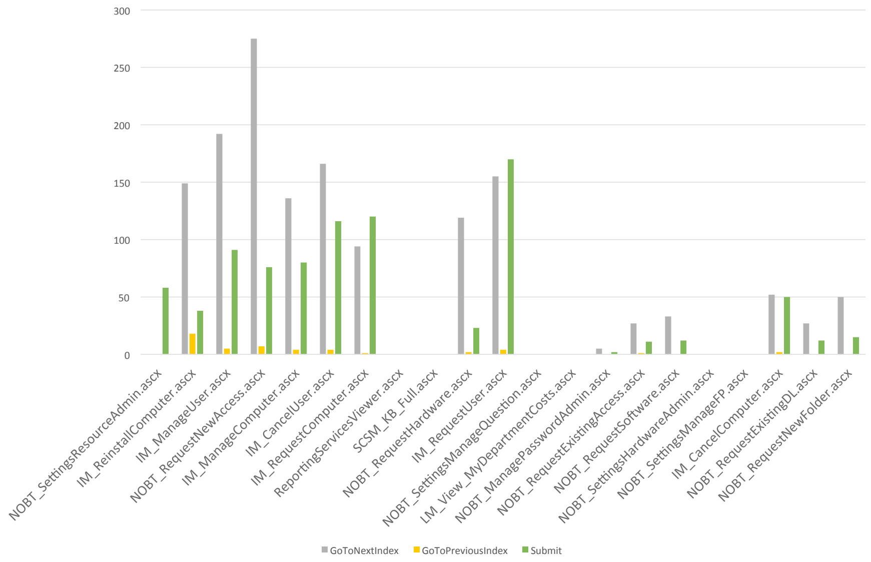


Figure 5.11: Chart showing uses of the buttons Next, Previous and Submit.

6 Evaluation

This chapter describes the verification that the solution in chapter 5 produces knowledge that conforms to the users perception expressed in a web survey.

The evaluation was performed by letting test subjects take part in user testing workshops. In the workshop each user was asked to perform a set of tasks, see Appendix C. After completion of the tasks they were sent an e-mail with a link to a web survey, see Appendix B. They then had to answer the four questions that the survey consisted of. After data mining had been run on the user testing data and the data from the web survey had been collected correlations were performed.

All Breakdown Questions were not possible to use for correlation when data was acquired from user testing. The selection criteria were that a question had to be possible to answer by both the survey and data mining with accurate data. Foremost, the user testing had only tasks for a predefined set of services meaning that any question looking at service frequency of use had to be ruled out. For this reason all questions in Breakdown Question 8 (Frequency of use for service and functions), Breakdown Question 6 (Occurrence of drop offs) and Breakdown Question 2 (Time of day that tasks are carried out) were excluded. Exclusion was also made to questions which could not be answered by a survey. Four Breakdown Questions were left and selected, number 1 (Time for a user to complete a task), 3 (Product Bounce rate), 5 (Utilization of internal search) and 7 (Most difficult service to complete).

6.1 User Testing

The reason for performing a simulated user testing, as mentioned in section 1.5, was the inability of conducting a survey on end users and activating the tracing for a deployed system running in production. Simulating a similar scenario was decided to be the best possible approach to handle this issue.

6.1.1 Subject selection

The criteria for the sample of selected test subject were to reflect expected end-users. Therefore they needed to be of different gender, at different age and have a basic computer experience that could be expected from someone using a computer at their workplace on a daily basis. It was also decided that none of the test subjects should be developers of the case product, see section 1.4, as they would possess more knowledge about the product than the average end-user. The number of test subjects was set to 15 as this was assumed to be a reasonable amount to approach and still get valuable input and diversity for possibility to analyze. The selected subjects fulfilled the above stated criteria hence the requirements were satisfied.

6.1.2 Task Design

The six different tasks where designed to provide knowledge of different service but also to use some similar services for comparison. The following tasks where therefore defined, the document explaining each task to the test subjects can be seen in Appendix C.

1. Request Hardware - user ought to complete a request for a computer.
2. Change Password - user ought to successfully change the password.
3. Request Software - user ought to complete a request for a software product.
4. Request Access - user should request membership to a user group.
5. Approve Order - user should approve a request, while using a manager account.
6. Cancel Order - user should cancel a request.

These six tasks covers the essentials of the case product and what user most often use the product for, according to ATEA Global Services. Since request of different kinds are the core idea of the case product three tasks were used to cover this. Requests need approval in some cases and due to this the Approve Order and Cancel Order were selected. To also cover some of the manage user part of the product one task was designed for changing password.

Five services were used for performing the six tasks. The task Request Hardware utilized the service Request Hardware, Change Password used the Manage Account service, Request Software utilized the Request Software service, Request Access used the Request Existing Access service. Approve Order and Cancel Order both was performed on the services Manage Approvals respectively Manage Requests.

As mentioned in section 1.5 the tasks were designed to reflect a real world scenario. With this in mind the task were formulated in such way that they represent a possible scenario that end-users could encounter. As a test subject performs a set of tasks there can arise an issue of learning- and boredom-effect [42]. To counter undesired variations related to learning- and boredom-effect the ordering of tasks for each test subject was made so that no subjects performed the tasks in the same order.

6.1.3 Execution

Each test subject was given a brief introduction to the study, the case product and the purpose of the user testing. After the introduction they were given a paper with the six tasks, ordered in the way they should be completed, and as mentioned in section 6.1.2 the order was unique for each subject. They were also given a computer on which they should perform the tasks, if they had any questions they were able to ask these before they started. After they had started they were not allowed any help and had to solve the tasks as good as they could on their own. The subjects were given as much time as they needed and were not supervised, so that they would not feel any pressure or stress while working.

6.1.4 Result

As mentioned tracing was enabled while test subjects performed the different tasks and all this usage tracing data has been analyzed with the methods mentioned in chapter 5. From the data acquired, out of the usage tracing with test subjects, it was possible to use data mining of this studies solution to get answers for the Breakdown Questions. Results for the Breakdown Questions that are part of the evaluation are presented below.

Breakdown Question 1

For Breakdown Question 1 (Time for a user to complete a task) it was only possible to get results for four of the six services used during the testing. The reason for this was that for the services Manage Approval, used for tasks 5, and Manage Requests, used for task 6, there were no clear distinction of beginning and completing. From the calculated completion times a chart with completion time per subject was created for each service. The time scale is in seconds and each chart also contains an orange line depicting the mean.

For the Order Hardware service, see figure 6.1, most users had a completion time between 50 and 150 seconds. One subject however had a completion time of 350 seconds for reasons which are unknown.

For the Order Software service, see figure 6.2, the range is slightly larger but there are no significant outliers. Compared with Ordering Hardware the mean time to complete the service is substantially lower, about 70 instead of about 140.

The result for Manage Account service, see figure 6.3, shows relatively high completion times compared to the two service above. Completion time ranges here from 50 to 370 seconds 12 of the 14 subjects have completion times within 160 and 280 seconds. The mean completion time, orange line in the diagram, shows a value right below 210 seconds making it the highest completion time mean among the four services.

The last service with result of completion time is the Request Access service. The result, depicted in figure 6.4, shows a mean just below 50 seconds which is the lowest for the four services. Moreover, the range of completion time is for most services between 17 and 47 seconds while 3 of the subjects have completion times above 85 seconds.

It should be noted that some of the diagrams have outliers but that no subjects is consistently outlier across all services. Furthermore, the manage account has a substantially higher mean than the other services. This can either be due to the fact that the

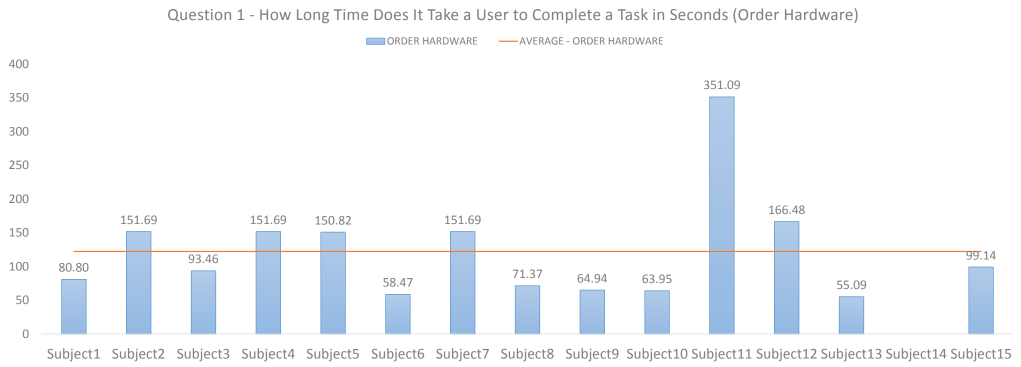


Figure 6.1: User testing answers for Breakdown Question 1 - Order Hardware, ”How long time does it take a user to complete a task”.

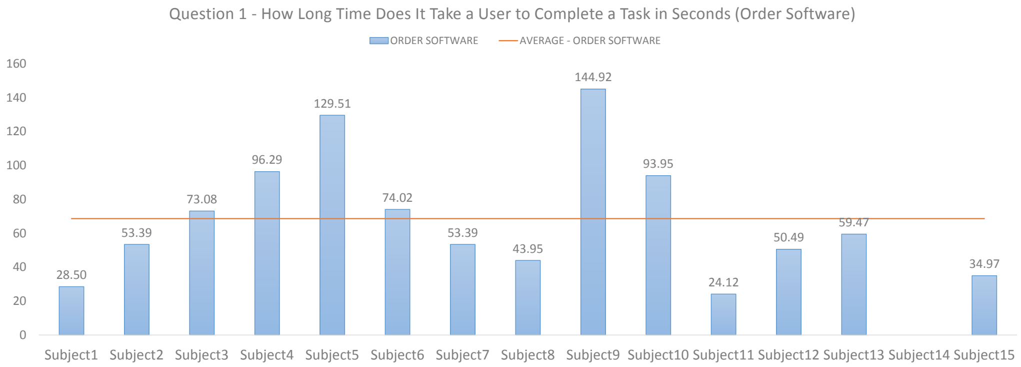


Figure 6.2: User testing answers for Breakdown Question 1 - Order Software, ”How long time does it take a user to complete a task”.

service is time consuming in the number of steps that users have to perform or that users are struggling with completion the service.

Breakdown Question 3

Bounce rates was calculated to correlate if it in general is difficult to find the right service in the product. Some manual preprocessing of the data was required, as some test subjects had looked around in the product after the fact that they had completed the tasks. This created bounces which were not part of the actual test and therefore created false results. The data from between the completion of the last task and the end of the session was for this reason removed for subjects which had used the application after completing the tests.

The result for Breakdown Question 3 (Product Bounce rate), see figure 6.5, shows that in general there was quite few drop offs. However some subject have a relatively

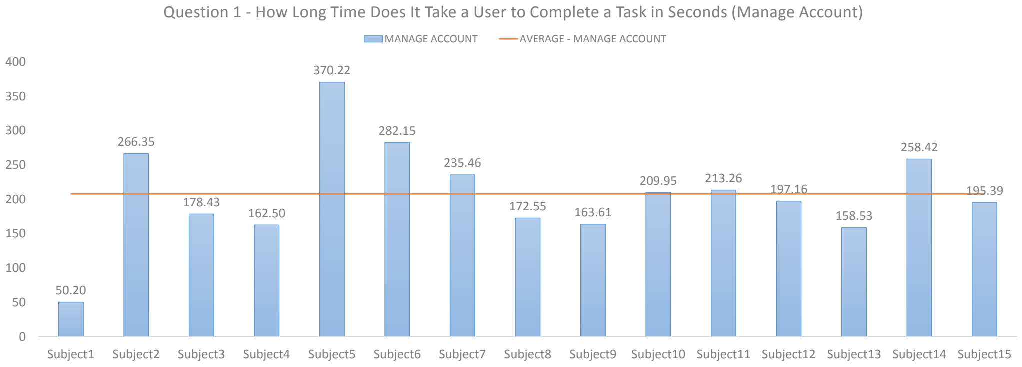


Figure 6.3: User testing answers for Breakdown Question 1 - Manage Account, ”How long time does it take a user to complete a task”.

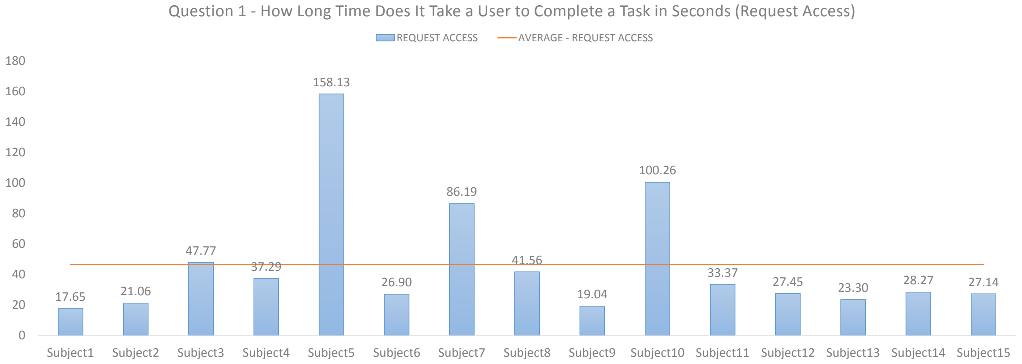


Figure 6.4: User testing answers for Breakdown Question 1 - Request Access, ”How long time does it take a user to complete a task”.

high bounce rate indicating that they might have struggled finding their way in the system.

Breakdown Question 5

The data mining for internal search showed that most user tend to prefer the search option, see figure 6.6. 33% of the user seem not to have a preferred way to use the product while 67% seem to prefer the search option. Note that no subject tended to prefer the Catalog option according to the data mining, see section 5.2.

Breakdown Question 7

As for Breakdown Question 1 (Time for a user to complete a task) result for Cancel Order and Approve Order could not be obtained due to the service lack of clear start

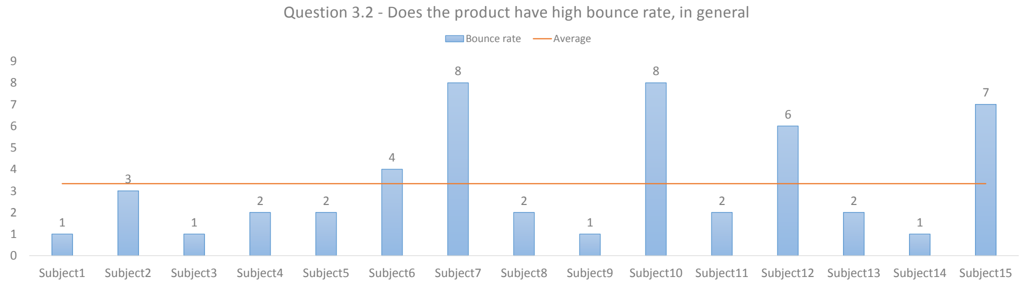


Figure 6.5: User testing answers for Breakdown Question 3, ”Does the product have high Bounce rate”.

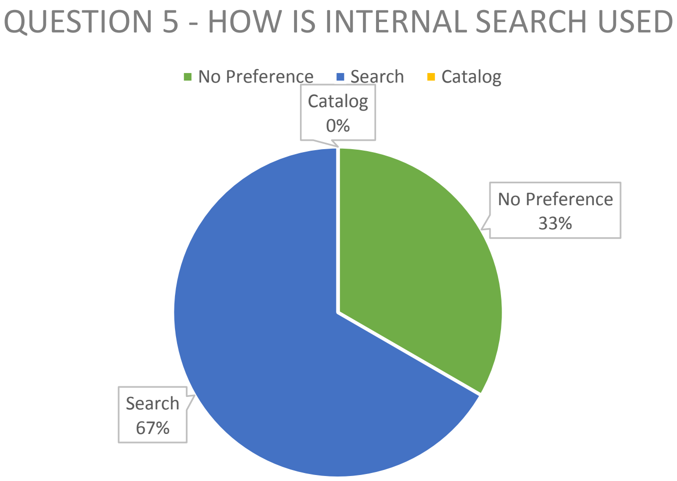


Figure 6.6: User testing answers for Breakdown Question 5, ”How is internal search used”.

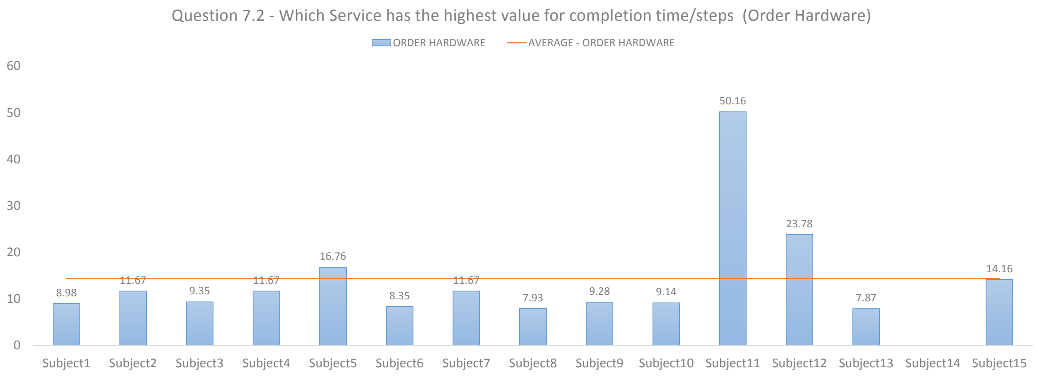


Figure 6.7: User testing answers for Breakdown Question 7.2, ”Which service has the highest value for completion time / steps”, where time is in seconds.

and endpoint. For the other tasks results where obtained and exported for correlation. The data mining result of each service is described beneath where time scale is in seconds.

The Order Hardware service had an average completion time per steps of approximately 14 seconds, see figure 6.7. The diagram shows a few outliers, subject 11 and subject 12, but the rest of the subjects have values close to each others. It should be noted that, as for Breakdown Question 1, there is no data for subject 14.

For Order Software the data mining result, see figure 6.8, shows completion time per step with a bit more variation but with a mean of about 9 seconds which is lower than the mean of Order Hardware. It should also be noted that, as for Breakdown Question 1, there is no data for subject 14.

The service with the highest mean completion time per steps is the Manage Account service, see figure 6.9. The mean for this service is about 47 seconds which indicates that there is a lot of time between each step that the user performs. This is a further indication that subjects struggled with completing this service. The result of Breakdown Question 1 indicated that there is a long completion time but with this result it can be concluded that the long completion time is not due to a complex service with many steps.

The Request Access service has a mean completion time per steps of about 11 seconds which is the lowest of all services, see figure 6.10. There is some variation amoung the subjects but most values are fairly close to the mean. For this service it can be concluded that it is the best service both for taking short time to complete, Breakdown Question 1, and for being easy to complete, Breakdown Question 7.

6.2 Web Survey

This section will describe the design and result of the survey that was conducted with test subjects whom completed the user testing workshop. The questions of the web survey can be found in Appendix B.

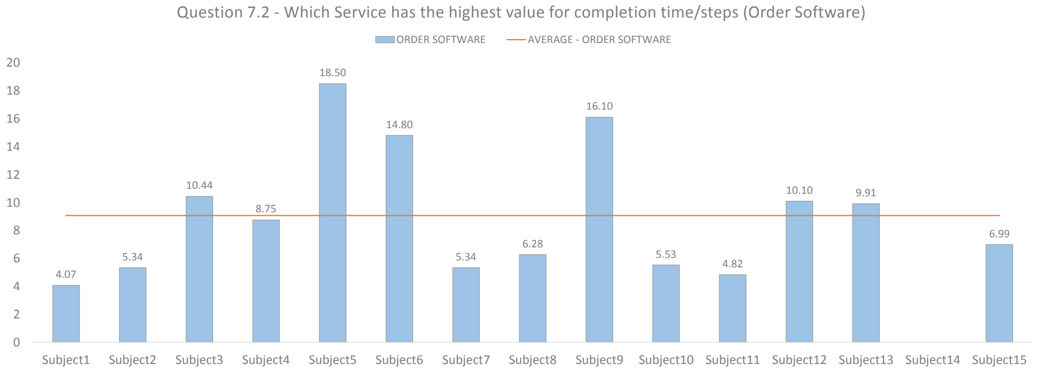


Figure 6.8: User testing answers for Breakdown Question 7.2, ”Which service has the highest value for completion time / steps”, where time is in seconds.

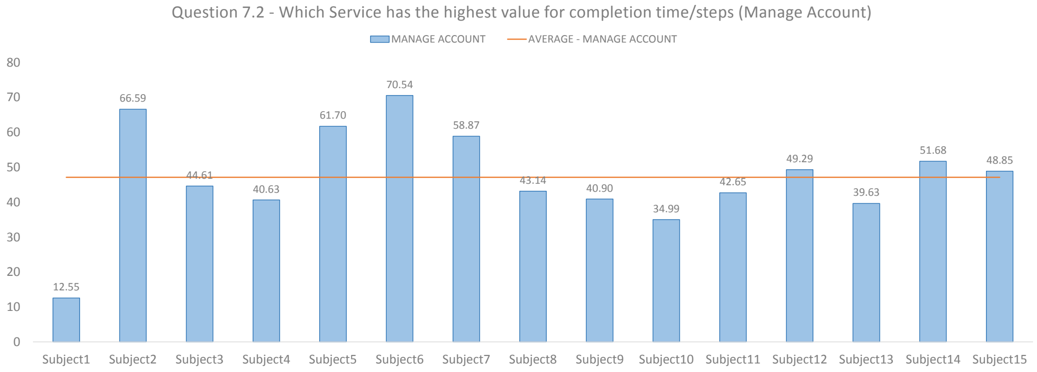


Figure 6.9: User testing answers for Breakdown Question 7.2, ”Which service has the highest value for completion time / steps”, where time is in seconds.

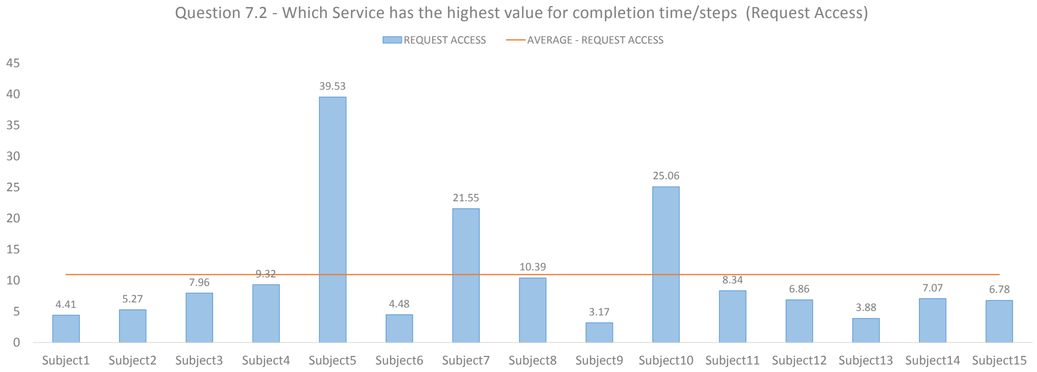


Figure 6.10: User testing answers for Breakdown Question 7.2, ”Which service has the highest value for completion time / steps”, where time is in seconds.

6.2.1 Design

When designing a web survey there are, as mentioned in section 2.3, some critical considerations to be made. The questionnaire must be designed to ensure that participants answers truthfully, independently and that they respond to the whole survey without dropping out. Invitation and reminders must be designed to induce the subject to participate in the survey. Moreover, the execution of the survey must be planned for when to send invitation, reminders and etc. Since the survey was given to test subjects during the user testing workshop focus was primarily put on the questionnaire design. The decisions described in this section are based on the web survey fundamentals in section 2.3.

The questionnaire was designed to ensure high response rate and a high quality answers by keeping the number of questions short. The expected time to complete the web survey was set to 3-5 minutes and this time limit was confirmed by performing test cases on employees at Atea Global Services. To achieve a short completion time the structure of the survey was determined to be a survey of type scrolling and the number of questions were kept low. Moreover, closed ended questions were utilized to further ease the completion of the survey. By keeping the survey short it was more likely that participants would not be affected by the boredom-effect, and drop out our giving random answer just to get done.

An effort was made in making sure not to push participants in any direction. This was done by avoiding unambiguous terms, using positive or negative phrasing and using an uneven number for grading questions, so that the participant need to pick side.

For each data mining question that were to be correlated with the web survey a question was designed. This resulted in 4 questions which also would be a good balance between participant effort and the receiving valuable data. The following survey questions were designed, based on the Breakdown Questions which was determined to need a survey questions:

1. Rate how difficult each service was to complete?
2. Which tasks did you find most time consuming?
3. When you order products do you prefer to use the search box or to use the catalog tree?
4. What is your general experience navigating and finding the correct service for each task?

The first question in the survey relates to Breakdown Question 7 (Most difficult service to complete). The hypothesis is that a high value for completion time per step will show which services are difficult to complete. The second question relates to Breakdown Question 1.1. If a user perceives that a task is time consuming that should also show in the completion time of each task. The third question relates to Breakdown Question 5.1 and 5.2. By asking what the user prefers the answers can be correlated for whether the

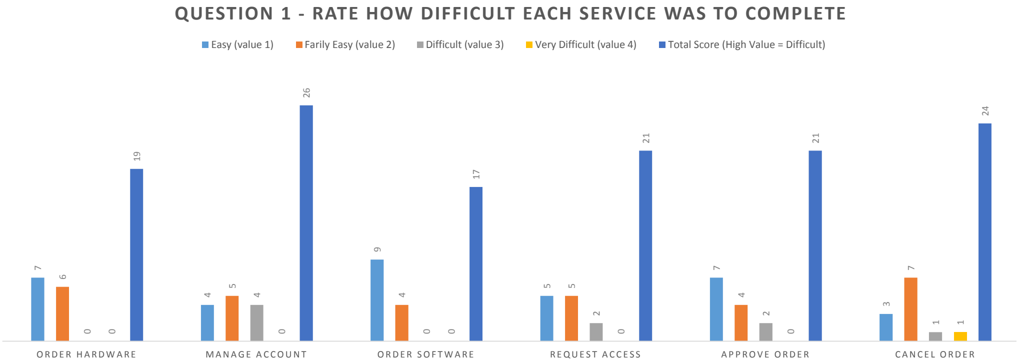


Figure 6.11: Survey answers for Survey Question 1, ”Rate how difficult each service was to complete”.

data mining is apply to compute what the subjects prefer. The forth question relates to Breakdown Question 3 (Product Bounce rate). When a user have problem navigating the systems it is likely that the user will have a lot of bounces.

Subjects in the user testing, section 6.1, received and completed the survey right after they had completed the user testing. Therefore, extensive efforts was not needed on invitations and reminders, which in general is of high importance for web surveys. Before sending out the survey a test run was made to make certain that questions were understandable, gave expected data and that the time limit could be met.

6.2.2 Result

The web survey was answered by 14 of the 15 test subjects, several reminders were sent out to one of the test subjects but without any reply or completion of the web survey. The result for each questions is shown beneath.

Survey Question 1

The result for Survey Question 1, where the test subjects had to rate how difficult each service was to complete, is seen in figure 6.11. Each answer was given a score from 1 - 5 and added to all the other answers so that it was possible to calculate a total for how difficult a service was to complete, a high value symbolize a increased difficulty. The worst ranked service, that user found most difficult, was Manage Account with a score of 26. After comes Cancel Order with a score of 24 followed by Request Access and Approve Order which both have a score of 21. The two answers with lowest score were Order Hardware with a score of 19 and Order Software with a score of 17 which is seen as the easiest task to complete, according to test subjects.

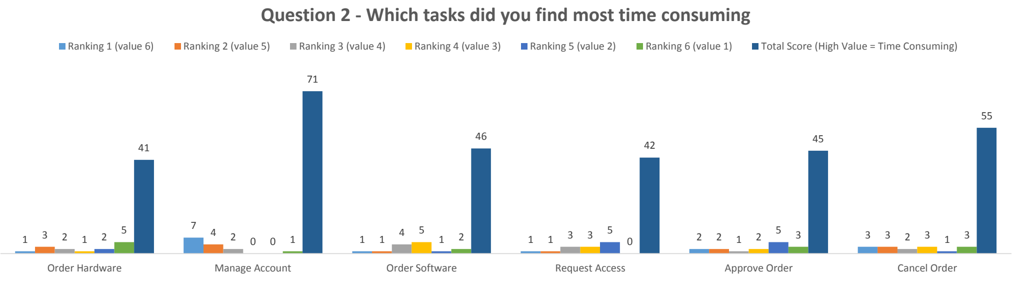


Figure 6.12: Survey answers for Survey Question 2, ”Which task did you find most time consuming”.

Survey Question 2

The result for Survey Question 2, where the test subjects had to rate which service was most time consuming to complete, is seen in figure 6.12. All services needed to be ranked by the test subjects and they ranked them from 1 - 6 with rank 1 being the most time consuming task and 6 being the least time consuming task. Each answered was then multiplied with a the corresponding value shown next to the color description in figure 6.12. This mean that a higher value for total score indicates a more time consuming task. The results show that Manage Account is by far the most time consuming task with a score of 71. Thereafter Cancel Order is found as quite time consuming with a score of 55 but then rest of the services are seen as equally time consuming with small variations; Order Software has a score of 46, Approver order has a score of 45, Request access has a score of 42 and Order Hardware has a score of 41.

Survey Question 3

The result for Survey Question 3, where the test subjects had to tell if they prefer to use the Search box or the Catalog tree while using the case product, is seen in figure 6.13. The obvious preference is to use the Search box, 50%. The Catalog tree is preferred by 29% and 21% answered that they do not have any preference between the two options. This shows that even though only 50% prefer the Search box it is a much more appreciated functionality as only 29% prefer the Catalog tree.

Survey Question 4

The result for Survey Question 4, where the test subjects had to rate their general experience while navigating and looking for the correct service in order to complete the different tasks, is seen in figure 6.14. They had three different options; ”I always find what I'm looking for”, ”I sometimes find what I'm looking for” or ”I often have difficulties finding what I'm looking for”. The results show clearly that a majority, 57%, answered that they always find what they are looking for. A minority, 7%, answered that they

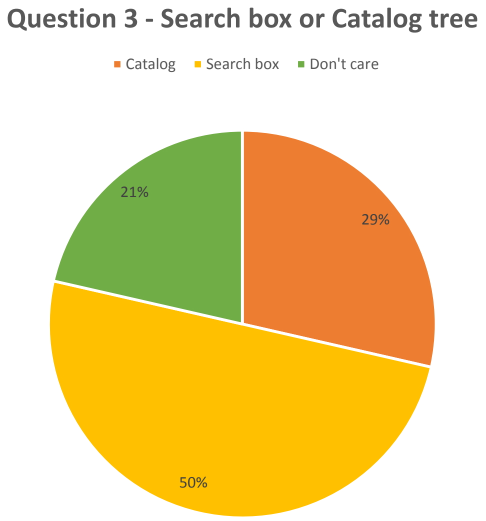


Figure 6.13: Survey answers for Survey Question 3, "Do you prefer to use the search box or the catalogue".

often had difficulties finding what they are looking for and the rest, 36%, answered that they sometimes find what they are looking for.

6.3 Correlation

Correlation of the results from the two methods, the user testing and the survey, was made to evaluate how the result of the two compares to each other. As discussed in the beginning of this chapter, four Breakdown Questions where selected for correlation.

Spearman's rho, Kendall's tau, the Pearson Product-moment correlation coefficient are some of the most well known correlation coefficients. Unlike the Pearson Product-moment correlation, Spearman's rho and Kendall's tau are non-parametric, i.e. do not assume normal distribution. [27] And, since the data acquired from user testing and the survey would not be sufficiently large in size to assume a normal distribution one of the non-parametric approaches needed to be selected for making the correlation. It has been argued that Kendall's tau provide a more reliable and interpretable confidence intervals than Spearman's rho. Moreover, the ease of calculating without computer that Spearman's rho provide was not of interest. Kendall's tau was therefore selected as the correlation coefficient to be used in this study. [43]

Using Kendall's correlation coefficient results in a value between -1 and +1. This value represents the relationship between the two sets of observations used in the correlation. Negative values mean that there is an inverse relationship, one set's values grow while the other's decrease, and -1 correspond to a perfect inverse relationship. Positive

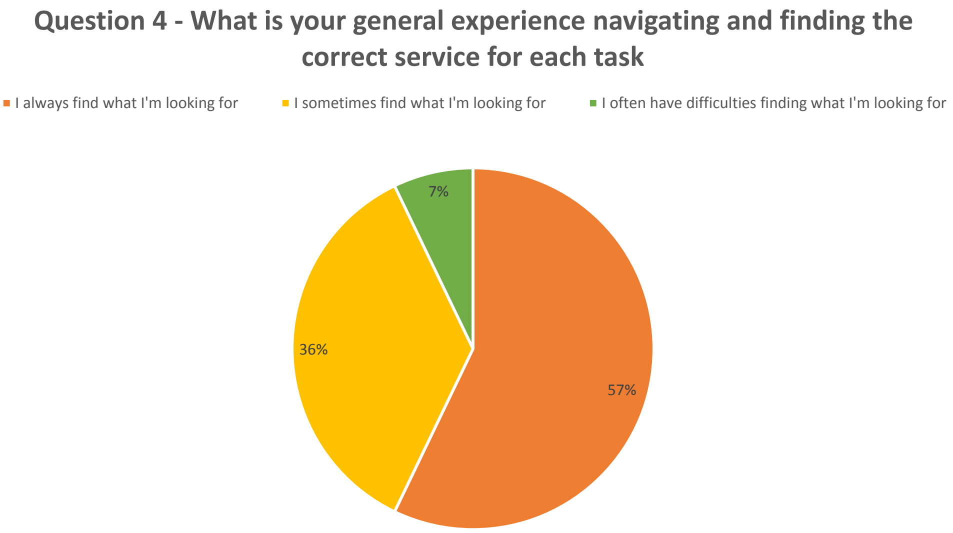


Figure 6.14: Survey answers for Survey Question 4, ”What is your general experience navigating and finding the correct service for each task”.

values correspond to a level of relationship, both sets of values grow or decrease, where +1 mean a perfect relationship. In this paper a negative value is referred to as an inverse relationship of some strength and a positive value is referred to as a relationship of some strength. If the absolute value is 0.1-0.3 the strength is weak, for 0.3-0.5 the moderate and for 0.5-0.7 strong. Values above 0.7 are referred to as very strong.

Kendall’s tau is based on concordant and discordant pairs. For a pair of observations x_j, y_j and x_k, y_k in a sample where $x_i \in X$ and $y_i \in Y$ for two variables X and Y. The pair is considered concordant if $x_j < x_k$ and $y_j < y_k$ or $x_j > x_k$ and $y_j > y_k$. If the pair instead has the property $x_j < x_k$ and $y_j > y_k$ or $x_j > x_k$ and $y_j < y_k$ it is considered discordant. With S being the score of subtracting discordant pairs from concordant pairs, the formula for Kendall’s tau is the following:

$$\frac{S}{\frac{1}{2}n(n-1)} \tag{6.1}$$

Where n is the number of observations.

An important factor when working with ranking coefficients is how to handle ties. Kendall’s tau, see equation 6.1, does not handle tied ranks but with the modified formula of Kendall’s tau, Tau-b, it is possible to handle tied ranks. In this formula, see equation 6.2, the denominator is changed by subtracting the number of tied ranks for observations in each variable X and Y. [44]

$$\tau = \frac{S}{\sqrt{(\frac{1}{2}n(n-1)-T)(\frac{1}{2}n(n-1)-U)}} \tag{6.2}$$

For $T = \frac{1}{2}\sum t(t-1)$ and $U = \frac{1}{2}\sum u(u-1)$, where t and u is the number of tied observations in X respectively Y.

The preceding subsections present the correlation for each of the selected Breakdown Questions. For all correlations let an observation be a value pair (x_i, y_i), where $x_i \in X$ and $y_i \in Y$. X represents the user testing values and Y represents the survey values. Note that as previously discussed completion time for Approve Order and Cancel Order was not possible to compute. For this reason Approve Order and Cancel Order are not included in correlation for Breakdown Question 1 (Time for a user to complete a task) and Breakdown Question 7 (Most difficult service to complete). Also as mentioned in the user testing result of Breakdown Question 3 (Product Bounce rate) data that was not part of the tests where removed. The correlation has been performed using data from 14 of the 15 test subjects since one subject never responded to the web survey, see subsection 6.2.2. Each correlation computation can be found in Appendix A.

6.3.1 Breakdown Question 1

For Breakdown Question 1 (Time for a user to complete a task), which concerns the time to complete a service, the scale of the variables X and Y are different but are both interval scale. Using the data mining framework presented in section 5.2 completion times was exported for each subject and service. From the survey the same format existed in the data and did not have to be transformed.

For this Breakdown Question the correlation was done per subject. For each subject a correlation coefficient was calculated from the completion time per service from the user testing and the ranking per service from the survey. Since one of the subject, subject number 13 of 14, had not completed Order Hardware and Order Software this subject was excluded from the correlation.

First the values in Y was transformed into an ordinal numerical scale and per subjects. This was done by looping all columns, representing each ranking, and all rows, representing each user. For each column and row the ranking for a task, the column index + 1, added to per user and per service matrix. Thereafter a correlation coefficient is calculated for each user. Summing all correlations give a total value of 3.667 which gives an average of correlations of $tau = 0.28$.

6.3.2 Breakdown Question 3

From X for Breakdown Question 3 (Product Bounce rate) services that are difficult to find will have a high bounce rate prior to the use of that service. The format for this data was exported as number of drop offs per user. For the survey users were provided with three options of which they could only select one. The options where as follow:

- I always find what I'm looking for
- I sometimes find what I'm looking for
- I often have difficulties finding what I'm looking for

For X a frequency list was created for three levels of bounce rate that correspond to the three options in the survey. The three levels was determined by dividing the difference between the highest and lowest bounce rate by three. Bounce rates lower than a third was determined as equal to option one above. Bounce rate between a third and two thirds was determined as option two above. And, bounce rate above two thirds was determined as equal to option three above. For Y another frequency list was created by using the three options above.

With frequency lists created for X and Y the correlation coefficient was calculated. This resulted in $\tau = 0.333$ which mean that there is a moderate relationship X and Y.

6.3.3 Breakdown Question 5

Breakdown Question 5 (Utilization of internal search) concerns users preference for finding products using either a catalog view or a search. For this question correlation was done on how the data mining and the survey relate when ranking the use of the three different options; Search, Catalog and No preference.

The correlation was performed by first creating frequency lists of X and Y for the three options stated above. With the frequency lists Kendall's tau is used to calculate the correlation coefficient. The result of the correlation is $tau = 0.333$ which suggest a moderate relationship between X and Y.

6.3.4 Breakdown Question 7

Like for correlation of Breakdown Question 1 (Time for a user to complete a task) the values for X are different but on the same type of scale as Y. The completion time per step for X was exported, from the data mining, in the format of per subject and per service. A format of per subject and service was already present in the raw survey data so no changes had to be done to the format of Y.

The correlation was performed per subject. For each subject the values in Y was transformed into an ordinal scale by ranking the answers for each task. Thereafter a Kendall correlation was performed on the completion time per steps for each service for that subject. As for correlation of Breakdown Question 1 subject number 14 was excluded due to missing completions of two of the four services used in this correlation. A summarization of the correlation above gives a total of 6.436 which means an average correlation of $tau = 0.495$.

7 Discussion

The following chapter discuss the result of the study by dividing it in sections that represent the targeted research questions, that are defined in section 1.1:

- Is the use of AOP suitable for usage tracing in a large business system? See section 7.1.
- Is it beneficial to conduct a workshop to gain deeper insight of a software product, and use that information to form questions that provide guidance for the implementation of data analysis? See section 7.2.
- Does AUTA provide equivalent results as a web survey and can it therefore be used as a substitution for the process of gathering information about users interaction with a software product? See section 7.3.

7.1 Aspect-Oriented Programming for Usage Tracing

For this specific case the use of AOP proved very valuable. It is not sure though that the same result would be found using a different AOP framework. But it is likely that similar AOP frameworks, using compile time weaving and not being based on the Proxy-pattern, would provide as valuable result.

Three main advantages were found when developing usage tracing with AOP:

- Reduced development time
- Separation of concern
- No disruption of the ongoing development

The use of AOP was an important factor for the fast implementation, which is mentioned in chapter 5. This is especially positive as the given case product consisted of such a large code base. The reduced development time is achieved by using Pointcuts, which makes it possible to apply tracing to several methods at the same time.

Separation of concern was confirmed as the main advantage and the base reason for using AOP. It gave the benefits that the code for the other developers was not cluttered with this new code, the avoidance of merge issues at code check-in even though the implementation affected all the services and that none of the developers were required to be aware of the tracing code. It also gave the benefit of being able to perform changes to the tracing in one place.

Having no disruption of the ongoing development was valuable as no changes needed to be made to the current code base. AOP provided the possibility to do a completely separated implementation where no internal code was interfered with. As no changes were being made to the current code base less synchronization with the developers was required. More synchronization would probably have meant an increase in development time both for the authors and for the development team. This was very positive as there were hard deadlines coming up and developers could not afford to risk any delays.

One potential issue, with the implementation of AOP, is the reliance on naming conventions. If some developer has misspelled a variable so that it does not conform to the naming conventions or just have not followed the naming conventions, then it can yield potential loss of valuable data. For example, if a developer by mistake names a method ”Clck” instead of ”Click” then that method would not be weaved, for a Pointcut applied to ”Click”, and all usage of ”Clck” would not be traces. The outcome of this could result in false conclusions, since in the data it would look like the method resembled is never being used, while the reality is something else. In large systems this problems multiplies as it becomes increasingly difficult to check that all Pointcuts follow these conventions. This could be a major issue for the results of the analysis as input data is incorrect.

7.2 Conducting a Workshop For Guidance of Implementation of Data Analysis

Conducting a workshop to gain deeper insight for the implementation of the proposed solution proved much valuable. It was noticed that participants of the workshop had many great ideas and could formulate and spread them as knowledge during the workshop. This then provided a good basis for the Breakdown Questions in chapter 4 and also point towards a greater knowledge if performed before the implementation.

In section 1.5 it is mentioned that the implementation of the usage tracing occurred before the conduction of the workshop, with employees at the development organization. This was not seen as an issue at the time but after the workshop it was noticed that some data points had been missed out. If the workshop would have been conducted before the implementation this would most certainly have been avoided.

Many of the questions extracted during the workshop were placed in Breakdown Question 8 (Frequency of use for service and functions) as they related to some kind of summary statistics. And, it was also later seen that many of the questions could be answered by AUTA but would not be feasible for a web survey. This is due to the fact that AUTA can gather a high quantity of data and process it without having to ask complex or many questions to the end-users.

Even though the implementation was feasible to perform before the conduction of the workshop it would have been difficult to construct the eleven questions without the input gained during this session, see section 3.5. The workshop therefore provided a valuable input as the questions were more accurate of what the organization wanted, they were based on their employees' thoughts. For example, the fact that many questions where discovered to be some kind of summary statistics guided the authors in what kind of data mining needed to implement. The reason that the extracted Workshop Questions were so accurate might have to do with the predefined questions that were asked during the workshop. These questions helped the workshop to stay on track and not drift away from the objective.

7.3 Possible Replacement Of Web Survey

The following section describe and discuss results from the Data Mining, section 5.2, and all parts of the evaluation, chapter 6, in order to provide summarized basis, which a conclusion can be made upon.

7.3.1 Data Mining

Based on the information collected from a test system used, by testers in the software development organization, it was seen that some behaviors in the data did not reflect that of actual end-users. In one case there was for example the issue that a tester used the same service multiple times to generate different scenarios. Some of these targeted scenarios did not require completion of the service. This creates an illusion that drop offs happen quite regularly, while the reality was that they actually were created on purpose. Another behavior that is noticeable in the result of Breakdown Question 6 (Occurrence of drop offs) is that the testers used a limited amount of user accounts. Drop off and multiple drop offs in figure 5.8 have, from what it look like, the same values. However, investigating the raw data shows that there are some small differences. Since the majority of usage is from one user, 13285 of 16605 traces are traces from one user account, it is likely that this user also made more than one drop off on most services.

As mentioned above it was found that the result of Breakdown Question 6 (Occurrence of drop offs) unfortunately did not reflecting actual drop offs that came into existence due to difficulties completing a service. In theory though this question could help in discovering which services might be complicated to complete. A user might drop off because the service is too difficult to complete. Another alternative is that a drop off is due to the fact that a user was looking into something. For example, that a user is curious of what computers are available for order. This creates an issue as it is questionable whether something needs to be improved or not. Another indicator that drop offs are due to difficulty with a service is if Breakdown Question 1 (Time for a user to complete a task) shows a high completion time. However, it might be beneficial to perform further investigation to know if improvement is needed or not. This could be done with an open-ended question inside a survey where it is questioned if the specific service is too complex to use.

The previous paragraphs highlight some of the problem with using data from testers and not real end-users. There is a lower validity to it and no decisions regards to development can be based upon the result. However, if the implementation would be on a solution running live and having actual end-users the result would most probably be beneficial. This has been proven to be correct by applying the correlation on the web survey and user testing.

Analysis shows that some results are not fully satisfying. Breakdown Question 1.2 (Outliers regards to completion time for services) shows some outliers but from investigation of the raw data it seems that the algorithm is missing some expected outliers. Further investigation point toward the fact that the number of completions greatly differ between services. This becomes a problem when ranking completions, done in step 2 (a) for algorithm of Breakdown Question 1.2 in section 5.2. As an example, let service A have 50 completions and service B have 5 completions. Furthermore, let completion times for all completions in service B be greater than those of service A. Then service A will get a rank of $\sum_{i=1}^{50} = 1275$ and service B will get a rank of $\sum_{i=51}^{55} = 265$. Hence, service A will be significantly larger even though all completion times are lower than for service B. And, if the number of services with few completions are greater than the threshold then service A will determined to be an outlier even though in regards to completion time it is not. It seems that the algorithm that was used, in this study, for finding significant outliers provide incorrect results when there are large differences for the different groups of observation.

Even though data from testers were used the result of the software solution in chapter 5 still show high potential in providing a depth of knowledge for the gathered data. For instance Breakdown Question 1.1 (Time to complete service) provide clear results of which services take long time to complete.

An interesting discovery is the result of the clustering algorithm for Breakdown Question 2 (Time of day that tasks are carried out). The algorithm was able to pinpoint where usage of the system is frequent and where it is infrequent. By using clusters, which radius is the calculated height, it is possible to see at what point in time activity that is high. Moreover, due to the "exponential" growth of height function, discussed in section 2.2.1, the activity differences will be easier to for humans to distinguish. The testers, of the case product, often have meeting in the beginning of the day and start testing activities about an hour before lunch. This behavior can be noticed in the clustering chart, see figure 5.6, were a small cluster can be seen at around 8 and the clusters then increase in size until time gets close to 12. Trend analysis could also be used to find the activity levels throughout the day and easily be implemented using any chart tool. However, the clusters provide results which focus on the points where a lot of activity is close to each other, while the trend analysis focus on how the activity changes over time.

Noticeable is that Breakdown Question 8.1.1 (Visit to service) which in its nature is quite simple, provides very valuable information. The chart for this question, see figure 5.10, show the ten most used services and these results can be used to determine which services should be improved to give the most value, probably satisfying most end-users.

Breakdown Question 8.3.1 (General use of features), in the chapter 5, show interesting results for the use of the features; GoToNextIndex, GoToPreviousIndex, Submit. The chart for this question, see figure 5.11, shows the difference between next, previous and submit steps in different services. This can be used as an indicator of services where users fail to do a step right before proceeding or become uncertain of whether the last step was done right. In this case the data is not valid as the result is based on data gathered from testers.

7.3.2 User testing

The target group for the case product is working men and female with daily interaction with computers. For the user testing selection of subject that covered both male and female within possible age range was achieved. The subjects were perceived to have basic to advanced computer skills as they had a variety of line of work. This provides a scenario that is close to the real world scenario and a valid discussion can therefore be made.

A challenge when letting subjects test a system from task descriptions is that the descriptions might provide too much guidance. This was accounted for and the description went through several reviews in order to makes sure that the task description did not reveal which service the subjects ought to use for a task. However, the target group for these tasks is quite diverse and the subjects' computer experience affect how easy a task descriptions is interpreted and how they approach it. The problem is to avoid providence of clues that solves the task but still provide enough clues to understand and not get stuck.

There is a risk for some completions of tasks, that the data mining determined as easy was for the subject perceived as difficult and vice versa. For example, if a user has a very extensive computer experience, user1, then she will most probably seem to be fast in comparison with an end-user with very little computer experience, user2. The collected usage data, while using the application, from user1 will by the system be perceived as good results, and thereby easy, even if some of them are perceived by user1 as difficult. This is due to the fact that in comparison with user2, that have a general difficult and does everything quite slow, everything user1 does will seem as easy. However, for the execution regarding the correlation of Breakdown Question 1 and Breakdown Question 7, this is not an issue. The correlation is performed for services on individual basis which mean that the perception differences will not affect the ranking of services in each correlation.

An issue that was discovered while running data mining algorithm on data collected from subjects testing was that some users had looked around in the system after completion. This was handled by removing data that was not captured during the performance of a task. By filtering out the gathered data it was possible to use it to get accurate results for the user testing which was then provided to the correlation.

Even though the tasks in the user testing were randomly ordered, so that no test subject would do the tasks in the same order. It might be an issue that it was not monitored if a task order was more similar to another, i.e. if the task Order Hardware more often was set to be done before the task Order Software. This could create false results as the tasks are performed in unique order but might still give a majority that have used the application more before the task Order Software than the task Order

Hardware. And, they therefore approach with different experience and confidence of how to perform the tasks.

7.3.3 Web Survey

The web survey, conducted on the subjects participating in the user testing, gave good results that were possible to use for a later correlation. The response rate for the survey was high and only one test subject did not answer it. This subject was therefore removed from the correlation so that all subjects used in the correlation had answered the survey and performed the given tasks.

It is unknown if it was possible to avoid the boredom-effect [42], which could have resulted in a case where subjects answered as fast as possible without being sure of giving their correct reflections. But as there were so few questions it is most probable that this did not happen. Another possible scenario is that test subjects had difficulties in remembering all the different services and tasks while they were ranking them. If a web survey was sent out to actual end-users, which most probably have more experience with the software, they would know the services better and might provide a more correct answer, reflecting the reality.

Even though there could be an issue with the responses due to boredom-effect it is possible to argue that more questions would provide better result for the study. It would perhaps be possible to answer more of the Breakdown Questions. Suggestively, if there is a larger group of participants in a web survey they could be split into sub groups and be handed different web surveys. This would solve the boredom-effect issue and still provide a greater and wider variety of data.

7.3.4 Correlation

As mentioned in section 7.2 there was a great interest by employees at the software development organization, for this study, of quantitative data collection questions, which however are not feasible to answer through a web survey. This mean that the correlation is not possible for these questions, as it only has the data mining results. But due to the low complexity of implementation it is most certain that the extracted data is valid and that there is not a need to perform correlation to notice the benefits in this case. The correlation was executed on four Breakdown Questions, all of them showed a weak to moderate positive correlation.

Breakdown Question 1 (Time for a user to complete a task) shows a correlation average of $tau = 0.28$, which suggest a weak relationship between the survey and user testing data. Comparing the survey result with the user testing result it can be seen that both point to Manage Account as being the most time consuming service. Both point at the same as the most time consuming service but the relation is weak for the other services.

Breakdown Question 3 (Product Bounce rate) shows a positive correlation of, $tau = 0.333$. This can be strengthen by comparing the values from the survey result with the user testing result. The survey shows that 57% always find what they are looking for while the user testing shows that about 60% of the subjects have a bounce rate of less than three. Note that a bounce rate of three is assumed to be a threshold were subject no longer think that they "always find what they are looking for". A risk is that with only three data points the data is easily skewed when there are ties. If two data points are tied the result of the correlation will only reflect how the third data point relate to the other two. Hence, the correlation is easily skewed too a negative of positive correlation. However, for this correlation this has not been the case.

For Breakdown Question 5 (Utilization of internal search) both survey result and user testing result shows that search is the most preferred option, between the search and catalog option. However, the data mining miss that some subjects prefer the catalog option because no subject use the catalog for more than 60%. The correlation for this question show that their is a moderate relationship, $tau = 0.333$. The fact that the data mining miss subjects that prefer the catalog suggest that there is an issue with the data mining implementation, exactly what causes this has not been determined. A likely explanation is that the issue is due to that users are, for some reason, forced to use search and the implementation is unable to discover that.

For Breakdown Question 7 (Most difficult service to complete) the survey result and user testing result both indicate that Manage Account is the most difficult service to complete. The performed correlation shows an average correlation of $tau = 0.495$ which suggest a moderate relationship between what users perceived and what the data mining extracted. The result is similar as for Breakdown Question 1, mentioned above. It has found a worst case but the correlation is weaken due to discordant pairs between the other services.

The correlation presented in this paper used survey answers and usage tracing for six services. Four of these services were possible to use for correlation of Breakdown Question 1 and 7, which both required a workflow completion. A higher number of service and test subjects would have provided more strength to the correlation. Using more services would however have made the user testing too time consuming and the survey too complex. 14 subjects were used for the correlation, as one subject did not answer the survey. More subjects would most probably make the correlation result stronger.

7.4 Threats to Validity

This section discuss the validity threats to the result of this study, the structure and discussion is based on the book by Wohlin et al. [45].

Threats to Conclusion Validity is concerned with how the relation between treatment and outcome affect the ability to make correct conclusions. For the evaluation of the solution presented in this paper 15 subjects were used, of which 14 could be used for correlation. Since so few number of subjects were used it was not reasonable to test the solution with more than six services. The low number of subjects and service mean it cannot be rejected that the solution is unable to replace the use of web surveys. To mitigate the issue of low number of subjects all correlations were performed using a non-parametric correlation coefficient, Kendall's tau. Nevertheless, for the correlation of Breakdown Question 3 (Product Bounce rate) a normal distribution was assumed when creating the frequency list for user testing data, see 6.3.2.

Threats to Internal Validity is concerned with how external factors may have affected the result. A problem that was discovered while running data mining on user testing traces was that one had a very high bounce rate. After investigation it was determined that the subject had clicked around after having finished all tasks. For this subject's session, the traces after the completion of all tasks were removed. Since bounce rate for the other subjects were not exceptional, no effort was put into investigating if more data needed to be removed. Therefore, there is a slight risk that the data mining of Breakdown Question 3 for the user testing gave a false result which affected the correlation.

The subject might have gotten bored or tired during the user testing and the survey. It was previously mentioned that it is unlikely that subject got bored when taking the web survey, as it had just four questions. However, the combination of having the web survey directly after the execution of tasks performed during user testing might have caused the subjects to get bored or tired while taking the survey. As only one subject dropped out, did not complete the survey, there is little risk of mortality effect. However, the risk for maturation effect is of greater concern and it has not been established to what extent the subjects were affected of this.

Maturation might also have been present in the form of learning. As the subject proceed with the testing of different tasks there is a risk that they during the process get more familiar with the system. In order to mitigate learning effect the user testing design was made so that subjects would not perform the test in the same order. However, since only 15 subjects were used only a few ordering combinations of the possible combinations were utilized. This might have lead to some tasks often being performed later in the test orders and thereby that task might have gotten better performance from the test-subjects due to the learning.

Others potential issues which might have affected the correlation result is that subjects might had difficulties in ranking the different alternative in the web survey. Some result suggest that subjects are able to point out the most time consuming or most difficult service but the other services are difficult for them to rank for which they might then just have selected randomly.

Threats to Construct Validity is concerned with the resemblance between observation and theory. The subject used in the evaluation knew that they would be given a survey after completing the user testing tasks. Because of this there is a risk that subjects might have put extra effort into remembering the experience of each task.

Another risk is that some humans tend to perform better when they know they are being test, due to anxiousness of being evaluated. This might lead to the effect of a stronger relation between subject's data mining result and their answers to the survey.

Threats to External Validity is concerned with the generalizability of the result. In this study the evaluation was performed using test subjects that were acquaintances to the authors of this paper. However, all subjects represents the population when it comes to age range and computer experience.

A deviation can be seen in the setting of the evaluation. In an industrial setting the solution would collect traces for end-users over an extended period of time. The end-users would receive the survey while using the system as a work tool. It is then not certain that the end-user would have recently used the services asked about in the survey. Which for an industrial setting might mean a lower correlation between data mining result and a survey as end-users might not remember how she experienced a specific service the last time she used it.

8 Conclusion

In this paper we have a presented a solution, called AUTA, which is a two part approach. The first part, usage tracing that gathers usage data and is implemented with AOP. The second part, data mining of the data gathered by the usage tracing implementation. A comparison has been conducted with web surveys to evaluate the solution. This chapter present the conclusion for each research question and then discuss potential future work.

Is the use of AOP suitable for usage tracing in a large business system?
The implementation of usage tracing with AOP was proven to be much suitable when the software was developed in such a way that all request are executed on the server-side. As we see it the main reason for this is that cross-cutting concern of tracing could be coded without manipulating the rest of the code base. This led to several benefits. Firstly, the implementation could be performed as an activity that was not affecting the rest of the development. Secondly, this approach allowed for a rapid implementation since aspects could easily be applied to several execution points. Thirdly the separations of concerns were kept in the object-oriented programming language. It was also found that this is only applicable to the case where the software code follows standards of an object-oriented language and has clear structure with a naming convention that is followed.

Is it beneficial to conduct a workshop to gain deeper insight of a software product, and use that information to form questions that provide guidance for the implementation of data analysis?
A workshop was held to extract information from employees about potential issues and potential analysis that would be of value for improving the case product. The result of this workshop, eleven questions, proved invaluable for the later implementation of data mining. For example, it was shown that there was a great interest in summary statistics which guided the authors when determining what kind of data mining was needed. It was also found essential to have a dialog with the employees at the case organization in order to gain needed background knowledge so these eleven questions could be defined and later broken down. It is therefore concluded that the conduction of a workshop in for this purpose is highly beneficial.

Does AUTA provide equivalent results as a web survey and can it therefore

be used as a substitution for the process of gathering information about users interaction with a software product?

It was discovered that some information that was obtained using the approach presented in this paper could not be obtain using a web survey. Or, it would very unlikely that a survey asking questions to answer that information would be responded to. For example, asking which of 140 services a user uses would require the user to make 140 decision and would be a violation to all guidelines that exist for web surveys.

It was also found that the correlations show a positive relation between the web survey and the data mining. The analyzed relationships are between weak and moderate and it has therefore not been proven that AUTA can replace the use of web surveys. However, it is proven that AUTA adds an added value and answer parts that are not feasible for a web survey to answer. Therefore it is recommended to use AUTA in a software solution and to conduct web surveys to follow up or to get deeper insight when needed. Suggestively, the survey has open-ended questions so that end-users input provide more qualitative data and not only overlap with AUTA.

8.1 Future Work

Future research may study the use of different algorithms and which data mining techniques are most suitable when trying to improve a software product. Especially such research could investigate what would be an appropriate approach for finding which groups of observations are outliers among a set of groups of observations. The algorithm used in this study was discovered to give false results when the difference in number of observations is orders of magnitude.

Another potential future work is to try to replicate the correlation presented in this paper with larger sample and with a higher number of services. With a larger sample size it would be possible to run correlation on more services without increased risk of boredom-effect. It would also be of value if the sample was selected from actual end-users of the analyzed product.

Usage trends between software versions are an interesting part that can help to follow up the changes that are implemented and see that they change the usage as expected. It is therefore suggested that a future work could analyze the possibility and benefits that comes from the analysis of trends.

Efforts have not been put on researching appropriate visualization techniques for data mining. It is likely that the data mining will provide even more value if the right visualization techniques are used. Therefore, guidelines for what visualization technique is appropriate for a certain data mining result would be of interest.

Bibliography

[1] M. Denscombe, The good research guide: for small-scale social research projects, Open University Press, 2010.

[2] A. Pinsonneault, K. Kraemer, Survey research methodology in management information systems: an assessment, Journal of Management Information Systems 10 (2) (1993) 75–105.

[3] G. Kiczales, J. Lamping, A. Mendhekar, Aspect-oriented programming, ECOOP'97 — Object-Oriented Programming 1241 (1997) 220–242.

[4] Y. Tao, Capturing user interface events with aspects, in: Human-Computer Interaction. HCI Applications and Services, Springer, 2007, pp. 1170–1179.

[5] J. Tarby, H. Ezzedine, C. Kolski, Trace-Based Usability Evaluation Using Aspect-Oriented Programming and Agent-Based Software Architecture, Human-Centered Software Engineering (2009) 257–276.

[6] A. Tarta, G. Moldovan, Automatic usability evaluation using aop, in: Automation, Quality and Testing, Robotics, 2006 IEEE International Conference on, Vol. 2, IEEE, 2006, pp. 84–89.

[7] L. R. Oded Maimon, Data Mining and Knowledge Discovery Handbook, 2nd Edition, Springer US, 2010.

[8] W. L. Hursch, C. V. Lopes, Separation of Concerns, Tech. rep., College of Computer Science , Northeastern University, Boston, MA (1995).

[9] H. van der Schuur, S. Jansen, S. Brinkkemper, Becoming responsive to service usage and performance changes by applying service feedback metrics to software maintenance, 2008 23rd IEEE/ACM International Conference on Automated Software Engineering - Workshops (2008) 53–62.

[10] D. S. Kerr, G. K. W. K. Chung, Using cluster analysis to extend usability testing to instructional content.

[11] D. L. Parnas, On the criteria to be used in decomposing systems into modules, Communications of the ACM 15 (12) (1972) 1053–1058.

[12] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, W. G. Griswold, An overview of aspectj, in: ECOOP 2001—Object-Oriented Programming, Springer, 2001, pp. 327–354.

[13] S. Endrikat, S. Hanenberg, Is Aspect-Oriented Programming a Rewarding Investment into Future Code Changes? A Socio-technical Study on Development and Maintenance Time, 2011 IEEE 19th International Conference on Program Comprehension (2011) 51–60.

[14] S. Hanenberg, S. Kleinschmager, M. Josupeit-Walter, Does aspect-oriented programming increase the development speed for crosscutting code? An empirical study, 2009 3rd International Symposium on Empirical Software Engineering and Measurement (2009) 156–167.

[15] R. Burrows, A. Garcia, F. Taïani, Coupling metrics for aspect-oriented programming: a systematic review of maintainability studies, in: Evaluation of Novel Approaches to Software Engineering, Springer, 2010, pp. 277–290.

[16] M. S. Ali, M. Ali Babar, L. Chen, K.-J. Stol, A systematic review of comparative evidence of aspect-oriented programming, Information and Software Technology 52 (9) (2010) 871–887.

[17] Sharpcrafters, PostSharp (2013).
URL http://www.sharpcrafters.com/

[18] Eclipse AspectJ Project, AspectJ (2013).
URL http://eclipse.org/aspectj/

[19] SpringSource, Spring .NET Framework (2013).
URL http://www.springframework.net/

[20] Oodesign, Proxy-pattern (2013).
URL http://www.oodesign.com/proxy-pattern.html

[21] O. Maimon, L. Rokach, Introduction to Knowledge Discovery in Databases, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, Springer US, 2005, pp. 1–17.

[22] B. Mirkin, Clustering: A Data Recovery Approach, Vol. 19, CRC Press, 2012.

[23] J. R. Kettenring, The Practice of Cluster Analysis, Journal of Classification 23 (1) (2006) 3–30.

[24] Z. Huang, J. Ng, D. W. Cheung, M. K. Ng, W.-K. Ching, A cube model for web access sessions and cluster analysis, in: Proc. of WEBKDD, Vol. 2001, 2001, pp. 47–57.

[25] I. Ben-Gal, Outlier Detection, in: Data Mining and Knowledge Discovery Handbook, 2010, pp. 117–130.

[26] E. M. Knorr, R. T. Ng, V. Tucakov, Distance-based outliers: algorithms and applications, The VLDB Journal The International Journal on Very Large Data Bases 8 (3-4) (2000) 237–253.

[27] R. R. Wilcox, Fundamentals of Modern Statistical Methods, Vol. 76, Springer, 2010.

[28] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics bulletin 1 (6) (1945) 80–83.

[29] S. Garcıa, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.

[30] J. Shaffer, Modified sequentially rejective multiple test procedures, Journal of the American Statistical Association 81 (395) (1986) 826–831.

[31] J. M. Converse, S. Presser, Survey questions: Handcrafting the standardized questionnaire, Vol. 63, SAGE Publications, Incorporated, 1986.

[32] W. Fan, Z. Yan, Factors affecting response rates of the web survey: A systematic review, Computers in Human Behavior 26 (2) (2010) 132–139.

[33] M. D. Kaplowitz, F. Lupi, M. P. Couper, L. Thorp, The Effect of Invitation Design on Web Survey Response Rates, Social Science Computer Review 30 (3) (2012) 339–349.

[34] H. Sauermann, M. Roach, Increasing web survey response rates in innovation research: An experimental study of static and dynamic contact design features, Research Policy 42 (1) (2013) 273–286.

[35] T. A. Mahon-Haft, D. A. Dillman, Does visual appeal matter? Effects of web survey aesthetics on survey quality, in: Survey Research Methods, Vol. 4, 2010, pp. 43–59.

[36] A. F. Osborn, Applied imagination: Principles and procedures of creative problem-solving, C. Scribner's Sons; Rev. ed edition, 1957.

[37] D. W. Taylor, P. C. Berry, C. H. Block, Does Group Participation When Using Brainstorming Facilitate or Inhibit Creative Thinking?, Administrative Science Quarterly 3 (1) (1958) 23–47.

[38] M. Diehl, W. Stroebe, Productivity loss in brainstorming groups: Toward the solution of a riddle, Journal of Personality and Social Psychology 53 (3) (1987) 497–509.

[39] BuiltWith, Top in Analytics and Tracking (2013).
URL `http://trends.builtwith.com/analytics/top`

[40] Microsoft, ASP:NET framework (2013).
URL `http://www.asp.net/`

[41] M. Fowler, Repository, in: Patterns of Enterprise Application Architecture, Addison-Wesley Professional, 2003, pp. 322–326.

[42] N. Juristo, A. M. Moreno, Basics of software engineering experimentation, Springer Publishing Company, Incorporated, 2010.

[43] R. Newson, Parameters behind "nonparametric" statistics: Kendall's tau, Somers' D and median differences, Stata Journal 2 (1) (2002) 45–64.

[44] L. Adler, A modification of Kendall's tau for the case of arbitrary ties in both rankings, Journal of the American Statistical Association 52 (277) (1957) 33–35.

[45] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, Springer Publishing Company, Incorporated, 2012.

A Correlation

A.1 Breakdown Question 1

Correlation for Breakdown Question 1 (Time for a user to complete a task) using the Kendall Package in R programming language.

```
> surveydata <- read.table("survey_986853_R_data_file_ordered.csv",
>                            header=TRUE, sep=",")
> usertestingdata <- read.table("question1.csv", header=TRUE, sep=",")
>
>
> # Calculate number of tied ranks
> a <- array(1, dim=(c(14,6)))
>
> for(i in 22:27){
+   for(j in 2:15){
+     index <- 27-i
+     n <- surveydata[j,i]
+     if(n == "A1"){
+       a[j-1, 1] = index+1}
+     else if(n == "A2")
+       a[j-1, 2] = index+1
+     else if(n == "A3")
+       a[j-1, 3] = index+1
+     else if(n == "A4")
+       a[j-1, 4] = index+1
+     else if(n == "A5")
+       a[j-1, 5] = index+1
+     else if(n == "A6")
+       a[j-1, 6] = index+1
+   }
+ }
>
> #usertestingdata1a <- usertestingdata[2, 2:5]
> #surveyquestion1a <- a[2-1, 1]
```

```
>
> correlation_sum <- 0;
> tau <- "tau"
>
> for(k in c(2:13, 15)){ # for each subject
+   number <- toString(k-1)
+   print(paste("----- subject", number, "-----"), sep=" ")
+
+   usertesting <- usertestingdata[k, 2:5]
+   survey <- a[k-1, 1:4]
+
+
+   correlation <- Kendall(survey, usertesting)
+   summary(correlation)
+
+   correlation_sum = correlation_sum + correlation[[tau]]
+ }
[1] "----- subject 1 -----"
Score =  6 , Var(Score) = 8.666667
denominator =  6
tau = 1, 2-sided pvalue =0.089429
[1] "----- subject 2 -----"
Score =  4 , Var(Score) = 8.666667
denominator =  6
tau = 0.667, 2-sided pvalue =0.30818
[1] "----- subject 3 -----"
Score =  0 , Var(Score) = 8.666667
denominator =  6
tau = 0, 2-sided pvalue =1
[1] "----- subject 4 -----"
Score =  0 , Var(Score) = 8.666667
denominator =  6
tau = 0, 2-sided pvalue =1
[1] "----- subject 5 -----"
Score =  4 , Var(Score) = 8.666667
denominator =  6
tau = 0.667, 2-sided pvalue =0.30818
[1] "----- subject 6 -----"
Score =  0 , Var(Score) = 8.666667
denominator =  6
tau = 0, 2-sided pvalue =1
[1] "----- subject 7 -----"
Score =  0 , Var(Score) = 8.666667
```

```
denominator =  6
tau = 0, 2-sided pvalue =1
[1] "----- subject 8 -----"
Score =  2 , Var(Score) = 8.666667
denominator =  6
tau = 0.333, 2-sided pvalue =0.7341
[1] "----- subject 9 -----"
Score =  -2 , Var(Score) = 8.666667
denominator =  6
tau = -0.333, 2-sided pvalue =0.7341
[1] "----- subject 10 -----"
Score =  2 , Var(Score) = 8.666667
denominator =  6
tau = 0.333, 2-sided pvalue =0.7341
[1] "----- subject 11 -----"
Score =  2 , Var(Score) = 8.666667
denominator =  6
tau = 0.333, 2-sided pvalue =0.7341
[1] "----- subject 12 -----"
Score =  0 , Var(Score) = 8.666667
denominator =  6
tau = 0, 2-sided pvalue =1
[1] "----- subject 14 -----"
Score =  4 , Var(Score) = 8.666667
denominator =  6
tau = 0.667, 2-sided pvalue =0.30818
>
>
> print("----- Correlation average -----")
[1] "----- Correlation average -----"
> correlation_sum
[1] 3.666666
attr(,"Csingle")
[1] TRUE
> correlation_avg <- correlation_sum / 13
> print(correlation_avg)
[1] 0.2820513
attr(,"Csingle")
[1] TRUE
```

A.2 Breakdown Question 3

Correlation for Breakdown Question 3 using the Kendall Package in R programming language.

```
> getValue <- function(v){
+   if(length(v) > 0)
+     return(v)
+   else
+     return(0)
+ }
>
>
> getBounceRateRankings <- function(x, y){
+   always <- 0
+   sometimes <- 0
+   never <- 0
+
+   for(u in x){
+     if(u < y)
+       always = always + 1
+     else if(u > y * 2)
+       never = never + 1
+     else
+       sometimes = sometimes + 1
+   }
+
+   return(c(always, sometimes, never))
+ }
>
> surveydata <- read.table("survey_986853_R_data_file_ordered.csv", header=TRUE, sep=","
> questiondata <- read.table("question3.csv", header=TRUE, sep=",")
>
> #  Rank the numbers for question1
> surveyquestion3 <- as.numeric(factor(surveydata[2:15, 30], levels=c("A1","A2","A3")))
>
> usertestingndata <- questiondata[2:15, 2]
> max_usertesting = max(usertestingndata)
> min_usertesting = min(usertestingndata)
>
> thrid <- (max_usertesting - min_usertesting) / 3
>
> usertesting <- getBounceRateRankings(usertestingndata, thrid)
>
```

```
>
>
> survey_freq <- table(surveydata[2:15, 30])
> survey <- c(getValue(survey_freq[names(survey_freq)=="A1"]),
+             getValue(survey_freq[names(survey_freq)=="A2"]),
+             getValue(survey_freq[names(survey_freq)=="A3"]))
>
>
> summary(Kendall(survey, usertesting))
WARNING: Error exit, tauk2. IFAULT =  12
Score =  1 , Var(Score) = 3.666667
denominator =  3
tau = 0.333, 2-sided pvalue =1
```

A.3 Breakdown Question 5

Correlation for Breakdown Question 5 (Utilization of internal search) using the Kendall Package in R programming language. Note that the warnings that are displayed for some of the subject correlations is related to that it was not possible to calculate a correct p-value.

```
> getValue <- function(v){
+   if(length(v) > 0)
+     return(v)
+   else
+     return(0)
+ }
>
> surveydata <- read.table("survey_986853_R_data_file_ordered.csv",
> header=TRUE, sep=",")
> questiondata <- read.table("question5_mod.csv", header=TRUE, sep=",")
>
> survey_freq <- table(surveydata[2:15, 29])
> survey <- c(getValue(survey_freq[names(survey_freq)=="A1"]),
+             getValue(survey_freq[names(survey_freq)=="A2"]),
+             getValue(survey_freq[names(survey_freq)=="A3"]))
>
> usertesting_freq <- table(questiondata[2:15, 2])
> usertesting <- c(getValue(usertesting_freq[names(usertesting_freq)=="A1"]),
+                  getValue(usertesting_freq[names(usertesting_freq)=="A2"]),
+                  getValue(usertesting_freq[names(usertesting_freq)=="A3"]))
>
```

```
>
> summary(Kendall(survey, usertesting))
WARNING: Error exit, tauk2. IFAULT =  12
Score =  1 , Var(Score) = 3.666667
denominator =  3
tau = 0.333, 2-sided pvalue =1
```

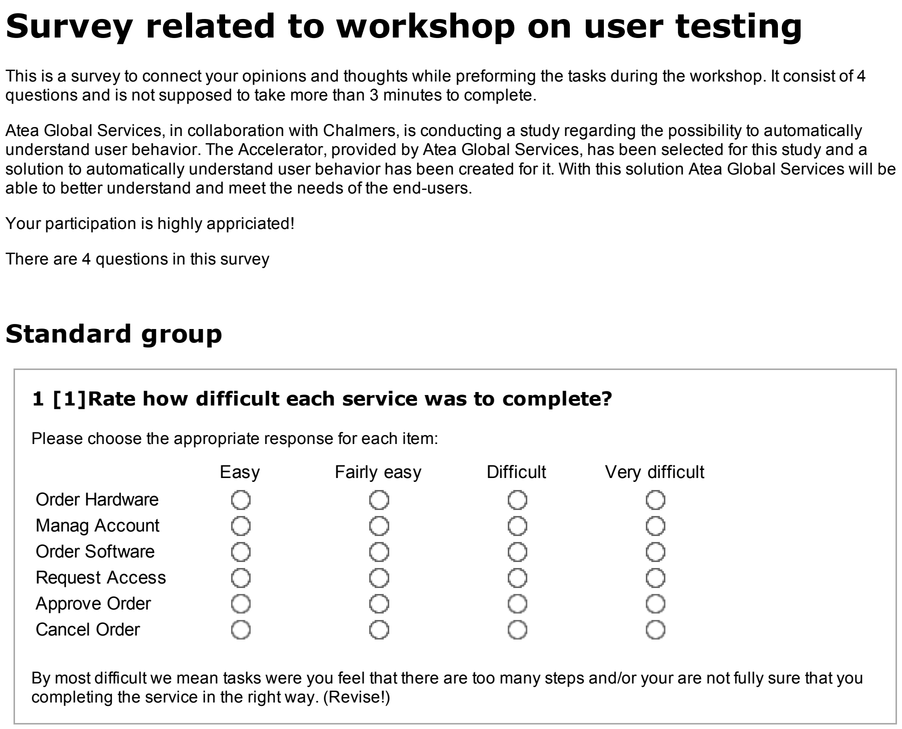
A.4 Breakdown Question 7

Correlation for Breakdown Question 7 (Most difficult service to complete) using the Kendall Package in R programming language. Note that the warnings that are displayed for some of the subject correlations is related to that it was not possible to calculate a correct p-value.

```
> surveydata <- read.table("survey_986853_R_data_file_ordered.csv",
> header=FALSE, sep=",")
> usertestingdata <- read.table("question7.csv", header=TRUE, sep=",")
>
>
> correlation_sum <- 0;
> tau <- "tau"
>
> for(k in c(2:4,6:13, 15)){ # for each subject
+   number <- toString(k-1)
+   print(paste("----- subject", number, "-----"), sep=" ")
+
+   usertesting <- as.numeric(usertestingdata[k, 2:5])
+
+   as.numeric(factor(surveydata[k, 16], levels=c("A1","A2","A3","A4")))
+
+   survey <- list()
+   survey = cbind(survey, as.numeric(factor(surveydata[k, 16],
+                                    levels=c("A1","A2","A3","A4"))))
+   survey = cbind(survey, as.numeric(factor(surveydata[k, 17],
+                                    levels=c("A1","A2","A3","A4"))))
+   survey = cbind(survey, as.numeric(factor(surveydata[k, 18],
+                                    levels=c("A1","A2","A3","A4"))))
+   survey = cbind(survey, as.numeric(factor(surveydata[k, 19],
+                                    levels=c("A1","A2","A3","A4"))))
+
+   correlation <- Kendall(survey, usertesting)
+   summary(correlation)
+
+   correlation_sum = correlation_sum + correlation[[tau]]
+ }
[1] "----- subject 1 -----"
WARNING: Error exit, tauk2. IFAULT =  10
Score =  0 , Var(Score) = 0
denominator =  0
tau = 1, 2-sided pvalue =1
[1] "----- subject 2 -----"
Score =  3 , Var(Score) = 5
denominator =  4.242641
tau = 0.707, 2-sided pvalue =0.37109
[1] "----- subject 3 -----"
Score =  0 , Var(Score) = 6.666667
denominator =  4.89898
tau = 0, 2-sided pvalue =1
[1] "----- subject 5 -----"
WARNING: Error exit, tauk2. IFAULT =  10
Score =  0 , Var(Score) = 0
denominator =  0
tau = 1, 2-sided pvalue =1
[1] "----- subject 6 -----"
Score =  4 , Var(Score) = 6.666667
denominator =  4.89898
tau = 0.816, 2-sided pvalue =0.24528
[1] "----- subject 7 -----"
WARNING: Error exit, tauk2. IFAULT =  12
Score =  0 , Var(Score) = 2.666667
denominator =  2.44949
tau = 0, 2-sided pvalue =1
[1] "----- subject 8 -----"
WARNING: Error exit, tauk2. IFAULT =  12
Score =  0 , Var(Score) = 0
denominator =  0
tau = 1, 2-sided pvalue =1
[1] "----- subject 9 -----"
WARNING: Error exit, tauk2. IFAULT =  12
Score =  0 , Var(Score) = 0
denominator =  0
tau = 1, 2-sided pvalue =1
[1] "----- subject 10 -----"
Score =  1 , Var(Score) = 5
denominator =  4.242641
tau = 0.236, 2-sided pvalue =1
```

```
[1] "----- subject 11 -----"
Score =  5 , Var(Score) = 7.666667
denominator =  5.477226
tau = 0.913, 2-sided pvalue =0.14856
[1] "----- subject 12 -----"
Score =  -1 , Var(Score) = 5
denominator =  4.242641
tau = -0.236, 2-sided pvalue =1
[1] "----- subject 14 -----"
Score =  0 , Var(Score) = 6.666667
denominator =  4.89898
tau = 0, 2-sided pvalue =1
>
>
> print("----- Correlation average -----")
[1] "----- Correlation average -----"
> correlation_sum
[1] 6.436474
attr(,"Csingle")
[1] TRUE
> correlation_avg <- correlation_sum / 13
> print(correlation_avg)
[1] 0.4951134
attr(,"Csingle")
[1] TRUE
```

B Survey Questions



Survey related to workshop on user testing

This is a survey to connect your opinions and thoughts while preforming the tasks during the workshop. It consist of 4 questions and is not supposed to take more than 3 minutes to complete.

Atea Global Services, in collaboration with Chalmers, is conducting a study regarding the possibility to automatically understand user behavior. The Accelerator, provided by Atea Global Services, has been selected for this study and a solution to automatically understand user behavior has been created for it. With this solution Atea Global Services will be able to better understand and meet the needs of the end-users.

Your participation is highly appriciated!

There are 4 questions in this survey

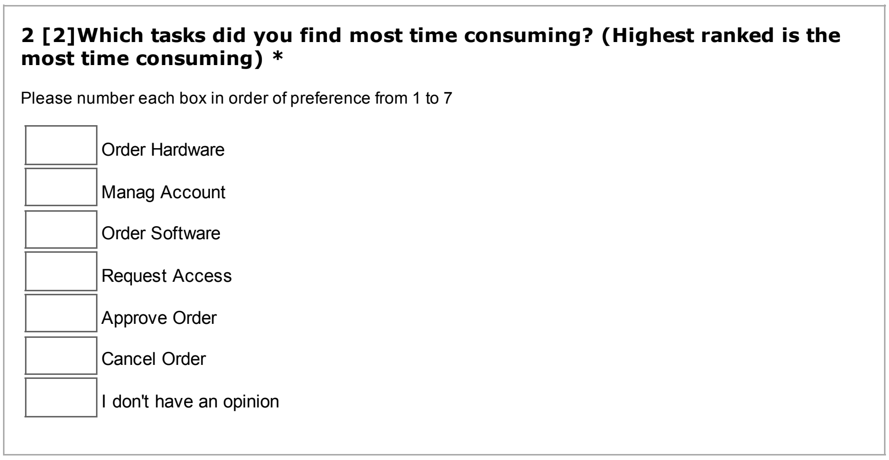
Standard group

1 [1]Rate how difficult each service was to complete?

Please choose the appropriate response for each item:

	Easy	Fairly easy	Difficult	Very difficult
Order Hardware	○	○	○	○
Manag Account	○	○	○	○
Order Software	○	○	○	○
Request Access	○	○	○	○
Approve Order	○	○	○	○
Cancel Order	○	○	○	○

By most difficult we mean tasks were you feel that there are too many steps and/or your are not fully sure that you completing the service in the right way. (Revise!)



2 [2]Which tasks did you find most time consuming? (Highest ranked is the most time consuming) *

Please number each box in order of preference from 1 to 7

□ Order Hardware

□ Manag Account

□ Order Software

□ Request Access

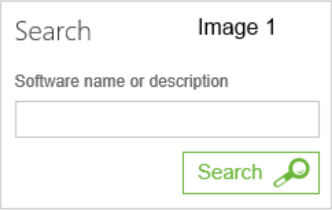
□ Approve Order

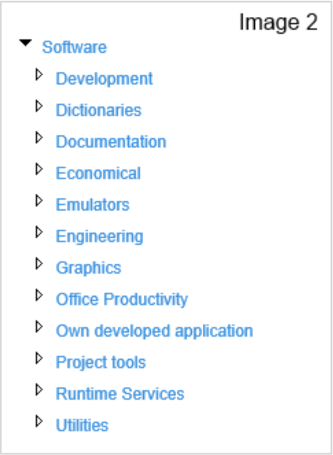
□ Cancel Order

□ I don't have an opinion

3 [3]

When you order products do you prefer to use the search box (image 1) or to use the catalog tree (image 2)?





*

Please choose **only one** of the following:

○ Catalog

○ Search box

○ Don't care

4 [4]What is your general experience navigating and finding the correct service for each task? *

Please choose **only one** of the following:

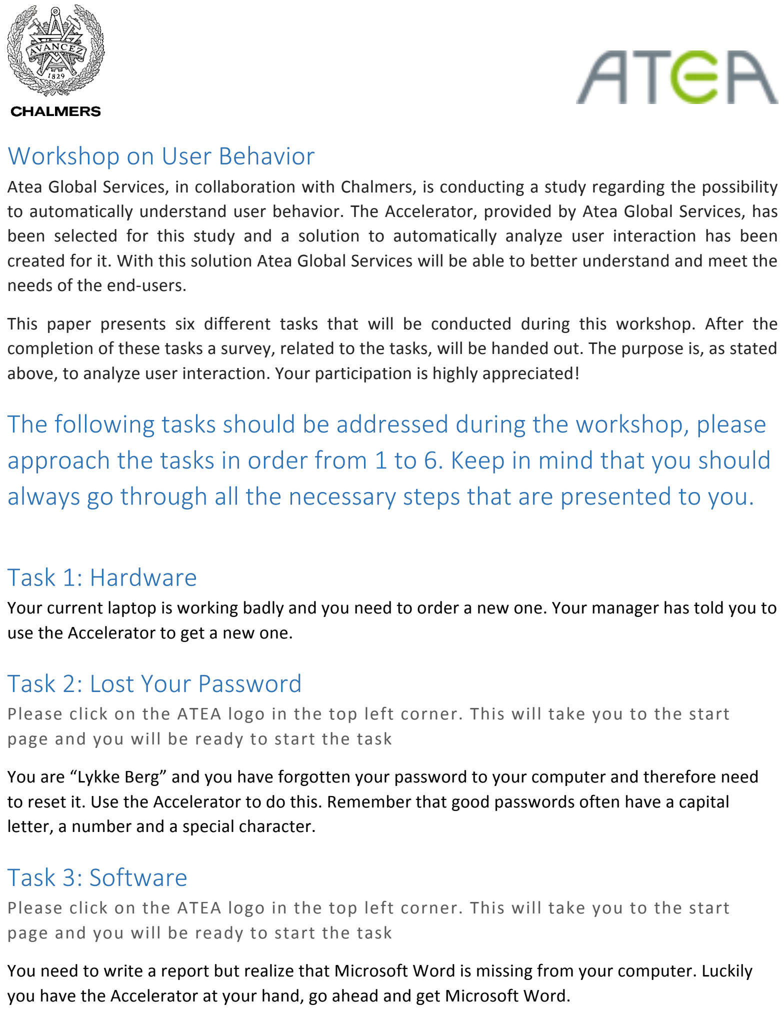
○ I always find what I'm looking for

○ I sometimes find what I'm looking for

○ I often have difficulties finding what I'm looking for

○ Other ______

C Workshop Tasks



CHALMERS

ATEA

Workshop on User Behavior

Atea Global Services, in collaboration with Chalmers, is conducting a study regarding the possibility to automatically understand user behavior. The Accelerator, provided by Atea Global Services, has been selected for this study and a solution to automatically analyze user interaction has been created for it. With this solution Atea Global Services will be able to better understand and meet the needs of the end-users.

This paper presents six different tasks that will be conducted during this workshop. After the completion of these tasks a survey, related to the tasks, will be handed out. The purpose is, as stated above, to analyze user interaction. Your participation is highly appreciated!

The following tasks should be addressed during the workshop, please approach the tasks in order from 1 to 6. Keep in mind that you should always go through all the necessary steps that are presented to you.

Task 1: Hardware

Your current laptop is working badly and you need to order a new one. Your manager has told you to use the Accelerator to get a new one.

Task 2: Lost Your Password

Please click on the ATEA logo in the top left corner. This will take you to the start page and you will be ready to start the task

You are "Lykke Berg" and you have forgotten your password to your computer and therefore need to reset it. Use the Accelerator to do this. Remember that good passwords often have a capital letter, a number and a special character.

Task 3: Software

Please click on the ATEA logo in the top left corner. This will take you to the start page and you will be ready to start the task

You need to write a report but realize that Microsoft Word is missing from your computer. Luckily you have the Accelerator at your hand, go ahead and get Microsoft Word.

Task 4: Membership

Please click on the ATEA logo in the top left corner. This will take you to the start page and you will be ready to start the task

It is possible to use the Accelerator to gain access to different groups, folders, etc. Your task is to request membership to an access group called “Test group”.

Task 5: You are the Manager

Please click on the ATEA logo in the top left corner. This will take you to the start page and you will be ready to start the task

The accelerator allows everyone to request both hardware and software. However, usually before anything is sent out a person with higher authority needs to approve it. There are several requests by “bolle” waiting to be approved and your task is to approve one of these, which one you choose to approve doesn’t matter. Go ahead and approve an order.

Task 6: You did Something Wrong

Please click on the ATEA logo in the top left corner. This will take you to the start page and you will be ready to start the task

After an order is requested it will wait for approval. If something was wrong with the order one placed it is possible to cancel it. Your task is to cancel the request you placed in “Task 1”, the new computer.