Investigating the impact on subjective satisfaction and learnability when adopting cloud in an SME

Erica Gavefalk
Elina Lundberg

Supervisor : Rouhollah Mahfouzi
Examiner : Cyrille Berger
Upphovsrätt

Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

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

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

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

Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years starting from the date of publication barring exceptional circumstances.

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

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

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

© Erica Gavefalk
Elina Lundberg
Abstract

Cloud services and solutions have served as a shift in the computer industry and create new opportunities for users. Clouds have been described as easily usable and fluid in terms of expansion and contraction depending on the real-time needs. Although the cloud is promoted with several benefits, it is not always apparent for the users that this is the case. Understanding both the benefits and challenges that exist is substantial for a successful adoption to cloud. This master’s thesis is conducted in collaboration with Exsitec AB and aims to investigate how the adoption of the cloud service Microsoft Azure will affect the development process. Also, it aims to provide a best practice for potentially needed updated working procedures, in terms of satisfaction and learnability. The investigation was performed through interviews and the System Usability Scale, to assess how the end-users experienced development in a cloud environment. The thesis revealed that the Azure portal has low overall usability, but that there also exists an inconsistency of that perception. Two major factors that contributed to the satisfaction and learnability was the lack of documentation and that the Azure portal was considered hard to master. The SUS score revealed that the mean value was below an acceptable level, and thus changes in the company’s working procedures need to be implemented. Internal documentation regarding how the company should use both cloud in general, as well as the portal in particular, are required in order to increase the learnability and subjective satisfaction.
Acknowledgments

We would like to thank various people for their help and contribution to this thesis: everyone at Exsitec that have participated in the interviews, Martin Jaensson, for being our supervisor and helping us with contacts and questions, our test respondent Per Lindström, for giving us relevant and useful feedback, Cyrille Berger and Rouhollah Mahfouzi, for being our examiner and supervisor at the University, and our seminar group during the Scientific Method course, for helping us in the early stages of the thesis. Also, we would like to give a special thanks to our opponents Fredrik Bengtsson and Adam Combler for providing us with valuable feedback during the mid-thesis review and to Filip Cornell and Richard Wigren for the same, during the opposition seminar and the final presentation.

Erica Gavefalk and Elina Lundberg
Linköping, June 2019
## Contents

Abstract iii  
Acknowledgments iv  
Contents v  
List of Figures viii  
List of Tables ix  

1 Introduction 1  
1.1 Motivation .............................................. 1  
1.2 Aim ...................................................... 2  
1.3 Research Question ...................................... 2  
1.4 Delimitations .......................................... 3  

2 Background 4  
2.1 Exsitec AB ............................................... 4  
2.2 Microsoft Azure ........................................ 4  
2.2.1 Azure Portal ......................................... 5  
2.2.2 Azure DevOps ....................................... 5  
2.3 Exsitec’s Integration Platform ......................... 5  
2.4 Exsitec’s On-Premise Solutions ....................... 6  

3 Theory 7  
3.1 Related work ........................................... 7  
3.2 Cloud Computing ....................................... 10  
3.2.1 Service models ...................................... 10  
3.2.2 Opportunities and Challenges ...................... 12  
3.2.3 Best Practice Within Cloud Adoption .............. 13  
3.2.4 Cloud Provider Lock-in ............................ 14  
3.3 Usability ................................................ 14  
3.3.1 Definition ........................................... 14  
3.3.2 Learnability ......................................... 15  
3.3.3 Satisfaction .......................................... 16  
3.4 Usability Evaluation .................................... 17  
3.4.1 Questionnaire ....................................... 19  
3.4.2 Interview ............................................ 20  
3.4.3 Validity and Reliability ............................ 22  
3.5 Requirements Elicitation .............................. 23  
3.5.1 Interviews .......................................... 25  
3.5.2 Scenarios ............................................ 25  
3.5.3 Sustainability ....................................... 26
## List of Figures

2.1 An overview of the Azure Portal ........................................ 6
3.1 Separation of responsibilities between On-premise, IaaS, PaaS and SaaS. .... 11
3.2 Learning curves for a hypothetical system. ................................ 16
3.3 Learnability subattributes .................................................. 17
3.4 Satisfaction subattributes .................................................... 18
6.1 Results from the SUS questionnaire ...................................... 42
6.2 Respondents view on the SUS questionnaire ............................ 44
6.3 The results regarding Deployment Effort ............................... 45
6.4 The results regarding Customisability ................................... 46
6.5 The results regarding Error Solving ..................................... 47
6.6 The results regarding scenario Client Approach ....................... 48
6.7 The results regarding Ease of Use ...................................... 49
6.8 The results regarding the learning curve ............................... 50
6.9 The result regarding set up a new environment ....................... 50
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Features enabled when using Cloud Computing.</td>
<td>12</td>
</tr>
<tr>
<td>3.2</td>
<td>Description of the five usability attributes.</td>
<td>15</td>
</tr>
<tr>
<td>4.1</td>
<td>Respondents in the questionnaire and the main interviews</td>
<td>33</td>
</tr>
<tr>
<td>5.1</td>
<td>Curved Grading Scale Interpretation of SUS Scores</td>
<td>37</td>
</tr>
<tr>
<td>6.1</td>
<td>The respondents’ experience, both as total years in the field of software development as well as from a cloud and on-premise perspective.</td>
<td>41</td>
</tr>
<tr>
<td>6.2</td>
<td>The individual SUS scores for each question and respondent.</td>
<td>43</td>
</tr>
</tbody>
</table>
1 Introduction

This chapter introduces the reader to terms regarding cloud and on-premise development and provides the aim of this master’s thesis. Furthermore, the problem is motivated and described from a general point of view, the research question is provided, and the delimitations are presented.

1.1 Motivation

Cloud computing serves as a shift in the computer industry, creating more attractive software and shapes how hardware is both designed and purchased. It reduces the need for upfront capital and thus the time to market, which creates unexplored opportunities for innovative developers whom will not need to be concerned about being overwhelmed by popularity not meeting their predictions [1]. Cloud computing makes it possible to outsource computing power, data storage, and services and is possible to make available as a commodity to the clients [2].

Furthermore, in different workshops, papers and conferences cloud computing has been a popular topic over the last years. The definition of cloud computing can be divided to include both the application delivered as a service over the Internet and as the hardware and system software in the data centres which provide this service [1]. The service can be delivered as a Software as a Service (SaaS), for example, Google Docs, a Platform as a Service (PaaS), for example, Google App Engine, or Infrastructure as a Service (IaaS), for example, Amazon’s Elastic Compute Cloud [3].

A cloud provider assembles virtualised services in large networks, and clouds are fluid in terms of that it is possible to expand and contract them depending on the needs of the client. There exist several different cloud providers who are, among others, Microsoft Azure, Amazon Web Services (AWS) and Rackspace, that enable the client to deploy their application to a pool of virtual resources, seemingly infinite [4].

Additionally, apart from being labelled a “buzz word”, clouds have been described as easily usable, due to deployment details being hidden from the user [5]. Thus, the attention paid to clouds can be characterised by its simple and externally managed environment.

The question regarding usability in software systems has been assigned an important role and have been found to have a close relationship to users’ satisfaction with the system in
question [6]. Furthermore, usability factors have been shown to affect the user acceptance of the systems [7]. The different cloud services will provide the client with an interactive system, which success is based on the usability of the product [6]. According to Zheng et al. [8] the quality of cloud services is important today and will stay so in the future. The authors divide quality into five dimensions out of which usability is one. Usability is described as how easy, efficient, and enjoyable the interface to a cloud service is. Moreover, usability is the only dimension which is subjective and non-negotiable. Therefore, it is reasonable to investigate how usability is altered by a shift from an on-premise solution to a cloud solution.

Johansson and Ruivo [9] have conducted interviews about cloud services, and one of the aspects discussed was usability. Usability was mentioned by 17 of the 20 experts as an essential aspect when discussing whether to use a SaaS provider instead of the on-premise provider. One expert specified that managers are not likely to choose a cloud solution over an on-premise solution if it not, at least, performs at the same level. Some other experts mentioned that their clients often ask about the user interface and whether it is similar or not to an on-premise interface. Thoss et al. [10] mention that the evaluation of non-functional properties, such as usability, is up to the cloud service user. They have done an investigation of cloud quality models, and out of seven models, six include usability in some way. Thus, it is reasonable to say that usability is considered a central aspect when discussing the quality of cloud computing. A system that is not usable and hard to learn will more likely be abandoned, and thus it is essential to understand which features of the system that makes it valuable to the organisation [11].

1.2 Aim

This thesis will investigate how the subjective usability, in terms of learnability and subjective satisfaction, may be altered when utilising the cloud service Microsoft Azure, instead of an on-premise solution. In order to investigate this, learnability and satisfaction will be explored using different methodologies. As a result, this thesis aims to try to understand the effects that the adoption of cloud services will have on the company and their development process and to provide a best practice for potentially needed updated working procedures.

1.3 Research Question

Usability is usually divided into five attributes, were learnability and subjective satisfaction are two of them. Learnability can be described as the most fundamental attribute as well as the most natural to measure [12]. Subjective satisfaction can be used to create an overview of the system’s overall pleasantness and assess whether users like the system or not. With this in mind, this master’s thesis will answer the following research question:

How are the learnability and subjective satisfaction altered by utilising the cloud computing service Microsoft Azure instead of an on-premise solution, from an end-user’s perspective?

In the case provided by this master’s thesis, an end-user is a developer with experience within on-premise solutions and the cloud service Microsoft Azure, working at the company or a partner.
1.4 Delimitations

This master’s thesis was conducted to investigate how the learnability and subjective satisfaction of the end-users at Exsitec were altered when comparing cloud development with on-premise development, and are therefore not general examples. This thesis was conducted as a collaboration with Exsitec, meaning that the data and resources used are specific for that company. Furthermore, Exsitec mainly utilises Microsoft Azure as a cloud provider, and thus this thesis will not consider other cloud providers. Additionally, the time constraint of 20 weeks sets a reasonable scope for this thesis.
2 Background

This chapter describes the company in which this thesis is in collaboration with and the background of the investigation described from the perspective of the company. The chapter aims to provide a deeper understanding of the context of the investigation.

2.1 Exsitec AB

This master’s thesis is conducted at Exsitec at their office in Linköping. The company has approximately 150 employees in total, and about 50 of them are located in Linköping, and is, according to the requirements presented in the EU recommendation 2003/361 [13], a Small and Medium-sized Enterprise (SME). Exsitec is an IT-consulting company working with and distributing Enterprise Resource Planning systems (ERP), management information, digitalisation and business applications. They are, among other things, one of Sweden’s largest retailer of Visma Business\(^1\) [14]. They provide the service to add and integrate applications and functionality to existing ERP systems. All applications generally need to be customised to fit the client’s existing system, as well as the chosen ERP systems. These different applications are hosted in both on-premise and cloud environments. Both environments imposes different limitations.

As for today, there is no coherent way of how to utilise the Azure Platform, and an investigation regarding which experiences different end-users have could benefit the company. Furthermore, the cloud providers promise many benefits with their solutions, but as with all systems there exist challenges that should be considered by the user.

2.2 Microsoft Azure

There exists a large number of different cloud providers, out of which Microsoft Azure\(^2\) holds a significant part of the market as for the fourth quarter of 2018 [15]. Other cloud providers who have a significant part of the market is AWS Lambda, Google Cloud, Alibaba Cloud and IBM Cloud. Microsoft states that Azure can be provided as SaaS, PaaS and IaaS solution [16],

\(^1\)https://www.visma.com/
\(^2\)https://azure.microsoft.com/en-gb/
and has been chosen as a cloud provider mainly based on the fact that Exsitec has a history of using different Microsoft products. Microsoft Azure will henceforth throughout the thesis be referred to as Azure.

Collier and Shahan [16], in an article published by Microsoft, discuss the differences between on-premise solutions and Azure in their book *Microsoft Azure Essentials: Fundamentals of Azure*. An on-premise infrastructure provides complete control over the hardware and the deployed software. This has historically meant that when to scale up, and thus, to purchase new servers to satisfy the performance need, is a challenging and essential question for companies. However, by using Azure, Microsoft will instead provide the hardware and infrastructure used by the companies.

The promise of more effortless scalability can be significant for especially smaller companies, according to Collier and Shahan, since Azure allows a company to start with a low cost and scale rapidly when more clients are gained. Azure also provides a pricing calculator, where a company can estimate their costs based on different selections, for example, the number of virtual machines or the storage size.

2.2.1 Azure Portal

The Azure portal is a way to build, manage, and monitor different types of applications, ranging from simple web apps to more complex cloud applications, in a single, unified console, according to Microsoft [17, 18]. Additionally, Microsoft [17] claims that it is easy-to-use and could be used to significantly simplify building, deploying and managing a company’s resources. Furthermore, the portal also includes a customisable dashboard and accessibility options, according to the Microsoft Azure documentation [18]. Figure 2.1 displays an overview of how the portal looks when initialised for the first time. Visible in the picture are several Azure services provided by the portal, such as creating virtual machines, handling storage accounts, creating databases and setting up continuous integration and continuous delivery. Azure also provides monitoring services, which analyses the applications performance, availability and usage, and identifies issues, according to Microsoft [19]. Additionally, the portal contains guidelines and online courses created by Microsoft to ease the learning period.

2.2.2 Azure DevOps

Exsitec makes use of a service from Microsoft called Azure DevOps, which is a service providing development collaboration tools. These tools include, among others, pipelines, Git repositories, continuous integration and development and automated testing. DevOps is, according to Microsoft, compatible with both cloud and on-premise solutions [20]. Exsitec utilises Azure DevOps when developing applications hosted in the cloud.

2.3 Exsitec’s Integration Platform

Today, Exsitec has developed an Integration Platform hosted in Azure. The integration platform is hosted in Azure as a PaaS model. The Azure PaaS solution is designed to serve all steps in the web application development life cycle, from building and testing the application to deployment, management and upgrading [21].

The role of the platform is to ease the integration of new client applications to the systems provided by Exsitec. If a new client wishes to utilise a service, the integration platform can ease the start-up process. The integration with the chosen service has already been set up, and thus it simplifies the procedure to connect a new client application. The concept may be compared to the use of adapters and provides the possibility to reuse already existing functionality.
2.4. Exsitec’s On-Premise Solutions

Exsitec also hosts its solutions in on-premise environments. The main difference is that Exsitec needs to communicate with a third-party provider, who provides the on-premise infrastructure. Often this includes; setting up authorisation, firewalls, virtual private networks, configure IP-addresses and decide on the dimensions of the projects in terms of storage, capacity and similar. It is also possible to utilise the Integration Platform when working with application hosted in an on-premise environment in order to integrate the application to the connected services.
The following chapter introduces the theoretical framework and the concepts of cloud computing, its opportunities and challenges, best practice within cloud adoption, as well as the quality attribute usability, and its sub-attributes. The chapter is introduced with a section regarding previous research in the area.

3.1 Related work

Stanton et al. [22] have studied the usability of cloud and discuss that different cloud service providers utilise different models and deployment types, and thus there exists no consistency of the user experience. Moreover, the authors mention that the cloud user community requires developing cloud usability standards in order to ensure more coherence.

Stanton et al. propose a framework for evaluating the user experience of cloud, to provide a structure so that vital areas of evaluation and client needs are not left out. The framework has been constructed upon five capabilities which are: Capable, Personal, Reliable, Secure and Valuable. Each of these capabilities has several sub-capabilities.

The authors discuss that the capabilities include user satisfaction and the importance of a high such in order for the organisation to continue using the service. They add on to that discussion that the cloud service provider should allow for a client to change the user interface’s look and feel. This includes identity management (access), and that it will ensure the ease of use through multiple access authentication. Additionally, it includes that the client should have a sense of control over the functionality in the application and be able to customise it to fit their use case.

Stanton et al. conclude that the proposed framework, in general, describes the client’s expectation of the cloud. Furthermore, the authors state that the framework can work as a foundation in the development of usability metrics for organisations adopting the cloud.

The interest in the possible benefits of migrating to a cloud service has been investigated in multiple articles and from different perspectives. Khajeh-Hosseini et al. [23] describe migration from an in-house data centre to Amazon EC2. The article study three companies A, B and C.
Khajeh-Hosseini et al. describe the different potential benefits and risks associated with migration. The study investigated how the migration would affect the time, resources, capabilities, values, status and satisfaction of the stakeholders. Further investigated aspects were the relations between stakeholder individuals and groups would be affected, and if the stakeholders would perceive the change as unjust. The largest source of benefits would be the ability to manage income and outgoings, but also the opportunity to offer new products and services. The largest source of risk is deterioration of customer care and service quality, increased dependence on external third party, and a decrease of satisfying work for support engineers, sales and marketing staff, and customer care staff.

Furthermore, the article shows that 21% of the support calls could potentially be eliminated if a cloud service was utilised instead since Amazon then would be responsible for the hardware. This would make it possible for a support department to focus on the issues necessary for the end-users, more precisely the software related issues.

In conclusion, Khajeh-Hosseini et al. summarise that the most significant decision point is the decrease in cost since it would be sufficient enough in order to decide to conduct a migration. However, they also remark that the above-mentioned impact on stakeholder satisfaction and service quality due to the use of a third party should be considered as well. The authors recommend that the risks should be adapted to a risk register and monitored.

In the article Cloud Migration Research: A Systematic Review [24] authors Jamshidi et al. have selected 23 studies and conducted a systematic literature review (SLR). Additionally, a cloud migration reference model is introduced. Jamshidi et al. define the primary objective of the SLR as follows:

We conducted an SLR with the primary objective to identify, taxonomically classify, and systematically compare the existing research focused on planning, executing, and validating migration of legacy systems toward cloud-based software [24, p. 142].

Jamshidi et al. describe cloud computing as the recent focus of attention in computing, both as academic research as well as industrial initiatives. Both dominant IT companies, as well as startups, see cloud computing as an opportunistic business strategy, in order to remain competitive and meet business objectives.

Some of the variables which make cloud environments so appealing are scalability, reliability, interoperability and cost flexibility. The incentives for conducting a migration to a cloud service are the promise of easier infrastructure provisioning, cost saving and avoiding over-provisioning as well as increased elasticity to a fluctuation of service requests. Additionally, the cloud uses a shared pool of infrastructure resources and cloud services are mainly used to expose services to clients.

According to Jamshidi et al. a cloud migration usually starts in order to meet a specific project or program needs. One of the fundamental design principles when conducting a migration is transparency in computing utilities. Additional critical principles are encapsulation and isolation from computing technologies. The migration should also promote flexible service delivery and should act like a single instance of software that serves multiple clients.

The introduced model can help to demonstrate current research at a conceptual level as well as identify trends and research directions. When deriving the reference model, the previous research for established reference models and frameworks for SOA migrations, as well as the commonalities in the primary studies, were used.

In conclusion, Jamshidi et al. define the model by describing the core migration processes with specific tasks. The model is defined as follows [24, pp. 155–156]:
3.1. Related work

- Planning: feasibility study, requirements analysis, decisions of providers and services, migration strategies.
- Execution: code modification, architecture extraction, data extraction, and transformation.
- Evaluation: deployment, testing, validation.
- Crosscutting concerns: governance, security, training, effort estimation, organisational change, multitenancy.

Roy et al. conduct a usability evaluation in their article *Usability Evaluation Of Some Popular PaaS Providers In Cloud Computing Environment* [25]. The article aims to estimate the usability of PaaS providers and measure user satisfaction based on specific usability attributes. The two PaaS providers investigated in the article are Microsoft Azure and Appharbor. These two were chosen based on their popularity and preference.

The authors have conducted a study regarding the usability of PaaS, using both interviews and questionnaires with 100 expert users. During the investigation, both a Pre-test Questionnaire and a Post-test Questionnaire was used. The Pre-test Questionnaire includes questions regarding how satisfied the user was with the cloud provider and questions regarding measuring functionality and usage. The answers were given on the form: Highly Agreed, Agreed, Not sure, Disagreed, Strongly Disagreed. The authors motivate that one of the most popular usability testing methods is the Questionnaire and Interviews method, and argue the following regarding that method:

*The usability testing method [The Questionnaire and Interviews method] is used in order to collect users performance and satisfaction level while performing a task in the cloud environment. Depending upon certain questionnaires the satisfaction level is measured based on user feedback while performing a task in the cloud platform* [25, p. 317].

The results showed that the users were more satisfied with Microsoft Azure and that it provided better performance and efficiency when compared to Appharbor. Also, the authors could conclude that Azure had a higher percentage of highly agreed level in a majority of the attributes investigate.

Johansson and Ruvio [9] have investigated companies value propositions for delivering ERP systems as SaaS-solutions. The idea for the ERP vendor’s is that the ERP-system can be delivered as a SaaS and through that be accessed over the internet instead of as an application installed locally. The method used was to interview several different experts and explore which perceived benefits that are realised by providing the ERP as a SaaS.

20 experts were interviewed, which all worked at Microsoft and has experience in the ERP and SaaS domain. The interview results were grouped into ten factors were Microsoft could bring value to the customer and gain market share in comparison to on-premise solutions. All 20 experts mentioned costs, security and availability, and 17 mentioned usability as an important factor. Regarding usability of the solutions, the experts pointed out factors such as how easy the application is to use, how it performs, how much control the user has, and how the user interface is different compared to on-premise.

In conclusion, some of the most important factors, according to Johansson and Ruvio, are flexibility, how simple it is to use, cost concerns and how efficient you can use it.

Armbrust et al. [26] discuss the concept of elasticity and that it can positively affect the economic aspects of the decision whether or not to host the service in the cloud. The authors claimed that the costs of resource over-provisioning and the risks of under-provisioning are
transferred to the cloud providers when using the cloud. Walker [27] also discuss cloud computing economics. By using cloud services, IT organisations can, instead of purchasing a server cluster, lease the required compute time for their computational needs. However, it can be hard to understand whether or not the prices are fair, and the IT organisations need to be able to understand this tradeoff to justify their purchase decisions.

Ardagna et al. [28] observe that the expectations on scalability could differ between cloud users, depending on which service they use. Scalability issues were furthermore considered an essential requirement for a PaaS infrastructure.

### 3.2 Cloud Computing

Arvanitis and Kyriakou [29] describe Cloud Computing (CC) as one of the most influential and important changes in how organisations access and use information and communication technologies (ICT) in their daily work. The US National Institute for Standards and Technology (NIST) [30] define CC as:

> Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [30, p. 1].

The main idea of CC can be described as that some parts of the organisations ICT are delivered externally as part of an on-demand service [29]. The organisation can pay for the service as an operating expense, and the cost is calculated based on the actual use. Furthermore, the organisation does not need to make initial hardware and software investments, keep space for the hardware or support for the cost of the maintenance.

Lehrig et al. [31] discuss and provide definitions for important concepts regarding CC, adding capacity to the previously mentioned three concepts. **Scalability** is defined as “the ability of a cloud layer to increase its capacity by expanding its quantity of consumed lower-layer services” [31, p. 84], **elasticity** as “the degree a cloud layer autonomously adapts capacity to workload over time” [31, p. 84], **efficiency** as “a measure relating demanded capacity to consumed services over time” [31, p. 84], and **capacity** as “the maximum workload a cloud layer can handle as bound by its SLOs” [31, p. 84].

The deployment of cloud solutions can be divided into three main models: Public Clouds, Private Clouds and Hybrid Cloud [32]. In a public cloud, all resources are shared between all users. On the other hand, in a private cloud, the cloud is deployed when the user needs to enhance the security of their data and thus need a cloud catered exclusively for them. A hybrid cloud is a combination of a public and private cloud. The cloud provider Azure is a public cloud [16]. However, a company can deploy several core Azure services in their data-center with the help of Microsoft Azure Stack, as well as use a self-service portal experience. A hybrid cloud can then be created through the use of a virtual private network in order to integrate Azure and Azure Stack.

### 3.2.1 Service models

Duan et al. [33] describe how CC has served as a part in the trend of providing everything as a service, referred to as XaaS or EaaS. Usually, CC is classified into three categories. These three are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [16, 32, 34]. The differences in responsibilities between on-premise, IaaS, PaaS and SaaS are shown in Figure 3.1 [35]. The following sections will cover the basic concepts of SaaS and IaaS, and more in depth, PaaS.
3.2. Cloud Computing

Software as a Service

SaaS can be described as a method of licensing and delivering software on-demand, based on a centralised hosting and managing solution [36, 16]. This kind of solution dissolves the hardware costs from the server hosted solutions along with the maintenance and the continuous renewal cost for the clients. According to Resceanu et al. one of the most sought-after benefits of SaaS models is the possibility of technical support for the software.

Some examples of existing SaaS applications are Dropbox¹, WordPress², Amazon Kindle³ and Microsoft Office 365⁴.

Platform as a Service

A PaaS vendor provides the ability to deploy and run any application provided by the developer company [16]. By using a PaaS vendor, the developers are freed from infrastructure management, which subsequently allows the developers to focus solely on developing. In addition to that description, PaaS models also allow the creation of applications with supported programming languages and tools [34]. By using PaaS models, the applications are allowed to be deployed onto the cloud infrastructure, and the users are given control over the deployed applications as well as the hosting environment configurations. When comparing SaaS, PaaS and IaaS, Chengtong et al. [37] state that PaaS plays a vital role throughout the whole system of cloud computing. PaaS models can provide rich APIs for the upper layer, as the system software role, to develop different SaaS applications. However, Chengong et al. argue that the primary goal of PaaS is not just like traditional software to provide some essential APIs, but more advanced service-oriented APIs. Thus, the above application can make use of these advanced services for end-users to quickly build a specific application.

¹https://www.dropbox.com
²https://wordpress.com
³https://www.amazon.com/kindle-dbs/fd/kcp
⁴https://www.office.com/?omkt=en-GB
3.2. Cloud Computing

Some examples of existing PaaS applications are AWS Elastic Beanstalk\(^5\), Microsoft Azure\(^6\), Heroku\(^7\), and Google App Engine\(^8\).

**Infrastructure as a Service**

An IaaS vendor both runs, as well as manages server farms running virtualisation software [16]. By using this server, the clients are enabled to create virtual machines running on the vendor’s infrastructure. IaaS models provide several vital computing resources, such as processing, storage and networks, and allow them to deploy and run the software, including operating systems and applications. The users do not manage or control the underlying cloud infrastructure when choosing an IaaS vendor [34].

Some examples of existing IaaS applications are Amazon EC2\(^9\), Microsoft Azure\(^10\), Rackspace\(^11\), and Google Compute Engine\(^12\).

3.2.2 Opportunities and Challenges

The decision to utilise cloud solutions instead of on-premise solutions should be adjusted depending on the requirements of the organisation using the system in question [23]. CC comes with several different advantages over an on-premise solution, but also challenges. Li et al. [38] discuss how CC will change how organisations adapt their business models. A business model is described as how an organisation runs its daily business and create value for their client. Several features are brought up which are enabled by the utilising of a cloud environment. These features have been adopted in Table 3.1.

<table>
<thead>
<tr>
<th>Features enabled by the cloud</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Flexibility</td>
<td>The use of CC reduce the amount of capital required by the organisation to operate and grow. Clients can be charged based on their consumption, for example “pay-per-use” or “pay-as-you-go”.</td>
</tr>
<tr>
<td>Broad Network Access</td>
<td>With network access services are made available at any time and place. Moreover, every user may share their real-time information for resource sharing and collaboration.</td>
</tr>
<tr>
<td>Resource Pooling and Rapid Elasticity</td>
<td>Every resource can be assigned and reassigned depending on the demand of the client. Additionally, the resources can rapidly be elastically provided and released in order to scale immediately.</td>
</tr>
<tr>
<td>Service Monitoring</td>
<td>The implementation may be monitored and controlled, and there is transparency of information for both provider and client.</td>
</tr>
</tbody>
</table>

CC environments enable business agility and allow enterprises to adapt quickly to changes in the market, competition, technology and the operational environment [39]. IT organisations

\(^5\)https://aws.amazon.com/elasticbeanstalk/
\(^6\)https://azure.microsoft.com/en-gb/
\(^7\)https://www.heroku.com/
\(^8\)https://cloud.google.com/appengine/
\(^9\)https://aws.amazon.com/ec2/
\(^10\)https://azure.microsoft.com/en-gb/
\(^11\)https://www.rackspace.com/
\(^12\)https://cloud.google.com/compute/
can suffer disastrous consequences if they are unable to adapt to these changes. Some struggles include a lack of interoperability and operational standards. Additionally, cloud-based solutions may not always result in reduced costs, so it is, therefore, essential to investigate whether or not the specific application is suitable for the cloud.

One advantage raised about the cloud is that a user is able to concentrate more on the core of their business since they can focus less on their IT infrastructure [40]. However, in the case of using a public cloud deployment model, the organisation needs to apply relevant best practices. Also, with the fast deployment of services, the cloud provides a way for organisations to investigate new and innovative cloud-based technology. Activities regarding the IT infrastructure does not disappear when using the cloud but rather is moved from the organisation to the cloud provider. Furthermore, the environmental aspect of utilising the cloud and the creation of energy efficient CC solutions are essential to acknowledge, since it will reduce the organisation’s carbon footprint.

Kandil and El-Deeb [41] state that although CC has shown to possess several benefits, it still brings some challenges, such as automated service provisioning, server consolidation, traffic management and analysis as well as storage technologies and data management. Furthermore, when switching from an on-premise to a cloud solution, the value network of enterprise software solutions changes [40]. Due to the standardisation of enterprise software, the amount of available customisability suffers. Additionally, the lack of transparency of licenses can discourage potential clients from entering a contract.

The cost flexibility of the cloud, mentioned in Table 3.1, is an important feature, but the cost model can also be described as a challenge [42]. Adoption of the cloud can significantly reduce the infrastructure cost, but increase the cost of data communication. If the client uses a hybrid cloud deployment model, this problem can be particularly prominent. Additionally, the cost of data integration can be substantial, since confidential data can be forced to be split up into pieces and distributed onto different clouds. This will add substantial extra financial cost, and seriously affect the system performance, i.e. the time cost [42]. Interoperability and a lack of standards are also mentioned in several articles as a struggle with cloud adoption [42, 43]. Furthermore, latency, the delay from a request for data to its final delivery, is a problem for some applications [43]. All of these concerns or challenges can slow down the adoption of CC.

3.2.3 Best Practice Within Cloud Adoption

Kandil and El-Deeb [41] describe that some important quality attributes when it comes to cloud adoption is the portability of solutions, scalability, modularity as well as a consistent view of the system for users have been identified. Furthermore, software migration can be considered to be a special case of adaptive maintenance that deals with modification for a new environment [24]. Adaptive maintenance is defined by ISO/IEC 14764 as:

*The modification of a software product, performed after delivery, to keep a software product usable in a changed or changing environment [44, p. 2].*

A company needs to consider multiple factors when evaluating the suitability of an application and infrastructure for cloud adoption [45]. These are business factors, technical factors and the ease of implementation. Due to the shared nature of the cloud, it is essential to monitor the cloud application regarding terms of performance, availability and security. As previously stated, cloud migrations can also be divided into several subprocesses [24]. The subprocesses can be used to create an overview of the necessary steps to perform a successful migration process, touching on the subjects of planning, execution, evaluation and crosscutting concerns.
3.3. Usability

3.2.4 Cloud Provider Lock-in

Cloud providers offer business models which are attractive to companies in multiple ways [46]. However, as the cloud providers offer tools on how to easily adopt cloud and migrate their solutions onto the platform, users have raised the question regarding how to move data from one cloud to another. The central problem is that each cloud provider develops its solutions, APIs, or similar [47], and thus the users become dependent, or locked-in, on a certain provider’s platform [48]. This means that they are unable to change the provider because of technical incompatibilities which generate substantial costs. Zhu and Zhou [49] argue that it is not possible to shift between providers with considerably the same product, without the need of paying substantial switching costs. This means that if the user chooses a cloud service provider who has built their platform on proprietary formats, the user might be locked in, and have difficulties to change provider in the future [46]. In turn, this means that the growth of cloud ecosystems will be constrained due to that the choice of service providers is limited since the data and applications will be locked in cloud environments. However, with knowledge and research, planning, strategy, technical awareness and selection of provider, such lock-ins can be mitigated.

3.3 Usability

Several cloud producers prioritise technology first and practical usability second [50]. There exist several essential problems regarding usability, from the service and the clients’ point of view, which needs to be solved. The cloud producers need to offer user-oriented controls, especially to those that directly affect sensitive user data.

Furthermore, Alonso-Ríos et al. [51] discuss definitions of the term usability and claim that the definitions are usually brief and vague, and there exists no consensus regarding the concept of usability among researchers or standard bodies.

3.3.1 Definition

The concept of Usability has been defined in several different ways over the years, and the definitions most commonly used will be defined in this section. The International Organization for Standardization provides several definitions of usability, the most recent given in ISO 9241-11, is seen below:

[The] extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use [52, p. 2].

Another definition is given by the IEEE Standards Board [53], where usability is defined as follows:

An attribute that bears on the effort needed for use (including preparation for use and evaluation of results) and on the individual assessment of such use by users [53, p. 19].

ISO 9241-11 [52] also states that the attribute usability is relevant both for regular ongoing use, as well as occasional use. Usability is also relevant when it comes to learning, minimising the risk and consequences of use errors as well as to enable maintenance task to be completed with satisfaction. Both definitions highlight the fact that the usability of a system can be determined by its users and their required effort to perform tasks.

The book Usability Engineering, by Nielsen [12], divides usability into the five attributes: Efficiency, Learnability, Satisfaction, Memorability, and Errors. Each attribute is defined in Table
3.2. Alonso-Ríos et al. [51] state that several researchers consider Nielsen's attributes to be widely accepted.

Table 3.2: Definitions of the five usability attributes proposed by Nielsen. Adapted from [12].

<table>
<thead>
<tr>
<th>Usability attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficiency</td>
<td>Once a user has learned to use the system, it should be possible to be efficient and productive</td>
</tr>
<tr>
<td>Learnability</td>
<td>How easy the system is to understand for a new user and be able to become productive quickly</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>The user should be content and thus subjectively satisfied</td>
</tr>
<tr>
<td>Memorability</td>
<td>The system is easy for a user to remember and can return to the system without the need of learning it from the start again</td>
</tr>
<tr>
<td>Errors</td>
<td>The system should have a low error rate and that it should be easy to recover from an error; furthermore, no catastrophic error should occur</td>
</tr>
</tbody>
</table>

Alonso-Ríos et al. constructed a model which contain the five attributes Knowability, Operability, Efficiency, Robustness, Safety and Subjective satisfaction. Some of the usability attributes in Nielsen's model [12] and the model provided by Alonso-Ríos et al. [51] are comparable with one another. The terms efficiency and satisfaction have an equivalent attribute in both models. Nielsen's terms learnability and memorability, are combined into the attribute knowability by Alonso-Ríos et al. It is also possible to draw parallels between Nielsen's attribute errors and the attribute robustness from the model created by Alonso-Ríos et al. Regarding usability of a cloud service, for an end-user without expertise within cloud services a Graphical User Interface, commonly referred to as a GUI, increases the usability [8]. A Web User Interface could perform even better. However, the interface should not cause too much cognitive pain for the end-users, and if necessary, some additional information could abet the user in the interaction.

As stated in Section 1.3, and in the Research Question, the attributes investigated in this thesis are learnability and satisfaction. These attributes will now be defined and discussed further.

3.3.2 Learnability

Learnability is described by some as the most fundamental usability attribute since systems generally need to be easy to learn [12]. There exist two main learning curves that are commonly used, and that will be used in this thesis, one with a focus on novice users and one that focuses on expert users. Figure 3.2 compare the learning curves for a hypothetical system between the novice and the expert, whereas the novice’s system is easy to master but less efficient to use and the expert’s system is harder to master but highly efficient. In the contrary, the expert’s system is easier to get started with and to complete basic tasks in, compared to the novice’s system. However, these standard learning curves do not apply if the user is transferring skills from previously used systems. In this thesis, the novice’s learning curve are referred to as a high learning curve, and the expert’s as a low learning curve, mainly referring to how easy it is to get started on a basic level.

Furthermore, the initial learnability is probably the most natural usability attribute to measure [12]. Additionally, it is essential to remember that users usually do not take their time to learn a system before they start using it thoroughly. Most of the time, users begin using the system after they have learned a part of the interface. Therefore, when investigating learnability, the researcher should try to understand not only how long it took the users to master
3.3. Usability

the system entirely but also how long it took to achieve a sufficient level of knowledge to do useful work.

The subattributes related to learnability in the model created by Alonso-Ríos et al. [51] can be seen in Figure 3.3. Clarity and Consistency are similarly divided in terms of elements, structure and functioning. The term functioning contains both user and system tasks, whereas elements and structure can be seen both on a formal and a conceptual level. Additionally, the subattribute Suitability of documentation relates to the fact that the content, which includes definitions, descriptions and examples, should be useful and sufficient. The other subattribute related to Helpfulness can be explained as to what extent the help provided by the system reacts to the user’s actions.

In conclusion, in order to ensure that a system is easy to learn, and thus have high learnability, it should have a clear and consistent structure. Additionally, each element and function should have the ability to be respectively subjective and performed correctly. As mentioned, since users often start using the system after learning only some of the interface, the importance of sufficient documentation and a system that responds in correlation to the user’s actions increases.

3.3.3 Satisfaction

The usability attribute subjective satisfaction refers to how pleasant it is to use a system [12]. Satisfaction can be measured by asking users for their subjective opinions. If they come from a single user, these opinions do not have a significant value, but when replies from several users are averaged together an objective measure of the system’s pleasantness is created. Interviews are recommended as the best way of measuring subjective satisfaction since the purpose of the attribute is to assess whether users like the system or not. Furthermore, an important aspect to consider is the fact that even if users have previous experience using a system, their subjective ratings of difficulty are more closely connected to the experienced peak difficulty rather than the low difficulty.

Nielsen [12] states that if the evaluation measures several systems, the right idea is to ask the users which system they prefer, or how strongly they prefer different systems over others, in order to measure the subjective satisfaction.
3.4 Usability Evaluation

In the model created by Alonso-Ríos et al. [51], satisfaction consist of the two subattributes Interest and Aesthetics, as shown in Figure 3.4. Interest is defined by the authors as the system’s capacity to both capture and maintain the attention and intellectual curiosity of the user. Furthermore, aesthetics can be explained as to what extent the system can please its user in sensorial terms. This attribute can also be subdivided into categories of visual (sight), acoustic (sound), tactile (touch), olfactory (smell) and gustatory (taste) aesthetics.

To summarise, subjective satisfaction can be viewed from both an intellectual perspective as well as a sensorial perspective. Thus, it is vital to pick the user’s interest and challenge them intellectually, without increasing the peak difficulty. A system does not need to have an extremely low rate of difficulty for it to be considered subjectively satisfactory. The system also needs to please the users sensorially, enforcing that the GUI needs to be both well functioning and aesthetically pleasing.

3.4 Usability Evaluation

As usability has grown to become a more important aspect in software engineering, the number of methods evaluating it has also grown. Fernandez, Insfran and Abrahão [54] define a usability evaluation method as:

A procedure which is composed of a set of well-defined activities for collecting usage data related to end-user interaction with a software product and/or how the specific properties of this software product contribute to achieving a certain degree of usability [54, p. 790].
Each user evaluation method has its advantages and disadvantages and should be used in its appropriate setting [55]. Some examples of usability evaluation methods are Interview, Questionnaire, Cognitive Walk-Through, Heuristic Evaluation, and Scenario-Based Method [54, 55, 56]. A qualitative approach to usability evaluation often includes fewer respondents, and their primary purpose is to identify the main usability problems of the system under evaluation and find solutions for them. A quantitative study often includes a more significant number of respondents and correctly reported it would provide information regarding the results statistical significance, to support the reliability of the study [57].

Paz and Pow-Sang [56] selected 228 journals and conference papers, and analysed and reviewed the methods used for investigating usability in their article *Usability Evaluation Methods for Software Development: A Systematic Mapping Review*. To be able to cover all aspects of usability, the journals and conference papers often combined several different methods, according to Paz and Pow-Sang.

Hartson, Andre and Williges [58] describe that there exists a general lack of understanding of the advantages and disadvantages of different usability evaluation methods. Researchers need to have a better understanding of each method’s effectiveness and application. Furthermore, because of the lack of standardised criteria for evaluation, it is not possible to make a reliable comparison of different usability evaluation methods. Additionally, the authors mention that one challenge is that the methods are changing continuously. As Paz and Pow-Sang [56] mention, it is preferable to combine different methods to be able to investigate different aspects of usability.
3.4. Usability Evaluation

In conclusion, several appropriate methods exist for investigating the usability of a system. Nielsen [12] brings forward several and one is thinking aloud test, where the test subject uses the system while continuously explaining and verbalising their thoughts. One disadvantage with this method is that it may give too much credit to the users’ own theories on why the system is not usable. The main advantage is that it will collect a vast amount of qualitative data. Another method is observation, where the researcher observe the users while they work. This is one of the simplest methods but it demands that the researcher can observe without disturbing the user. A third method is to perform user tests, where the user in different ways are asked to test the system. This method has the advantage of enabling a more quantitative measurement in terms of, for example, how long time it took for a user to perform a certain task. Performing user tests require immense preparation from the researcher and the users should be divided and tested based on their experience. Nielsen also draws special attention to the ethical aspects of tests with human testers. He acknowledges the fact that the users feel a great pressure to perform well and that users will inevitably make errors and be slow at learning the system.

However, Nielsen suggests combining questionnaires and interviews when investigating subjective satisfaction. Other researchers also recommend the combination of interviews with questionnaires during an investigation of subjective satisfaction [25, 59]. If the research is to be conducted on real users’ knowledge of a product, interviews are a common method to use [60]. Regarding the investigation of the learnability of a system, several different methods can be considered suitable, interviews and questionnaires included [12, 61]. It was convenient to choose interviews and questionnaires because the investigation of learnability could then be combined with the investigation of subjective satisfaction. Thus, demanding less time and focus from the respondents, which could potentially have removed possible respondents as a result of lack of time. Interviews are also suitable when conducting an exploratory study where the researcher does not know precisely what he or she is looking for. Therefore, interviews and questionnaires were deemed to be the most suitable methods for this thesis. Nielsen remarks on that when measuring usability with questionnaires or interviews, or both, they are considered indirect methods for all aspects except subjective satisfaction. In that case, they are considered direct methods. For the other aspect investigated in this thesis, learnability, it is not the learnability of the user interface itself that is being studied but rather the users’ opinions about how easy the user interface was to learn.

3.4.1 Questionnaire

Questionnaires are essential in the evaluation process and can be described as subjective aspects subjective by users, which cannot be objectively quantified [62]. Also, their purpose is to provide a quick overview of the usability of a system [63].

Lewis [63] has investigated questionnaires regarding subjective usability published from the early 1980s until the present. During the late 1980s, several standardised usability questionnaires were created by researchers, and the ones still used today, in an updated version, include The Questionnaire for User Interaction Satisfaction (QUIS), The software Usability Measurement Inventory (SUMI) and The Post-Study System Usability Questionnaire (PSSUQ). However, the most widely used standardised questionnaire for the assessment of subjective usability is the System Usability Scale (SUS). Lewis additionally claims that the SUS will probably be the most used questionnaire in the future as well. One argument made by Lewis supporting the statement is that the pace of research regarding SUS is accelerating.

Furthermore, Nielsen [12] states that it is also important to remember that during an interview the researcher can continuously evaluate the user’s replies, allowing for rephrasing misunderstood questions. A questionnaire is forced to stand on its own, and it is therefore vital for questionnaires to have been through significant testing before use. An already published questionnaire, as the SUS, has already been through multiple iterations of testing [64]. Lewis
and Sauro [61] state that the subjective usability measured by the SUS includes the areas satisfaction and learnability.

From the above-mentioned reasons, the SUS was therefore considered the most suitable questionnaire to use in this thesis.

SUS contains a set of 10 questions which should be answered on a scale from one to five, where one equals strongly disagree and five equals strongly agree. The scale is a Likert scale, which means that the statements included in the questionnaire should lead to extreme expressions of the attitude that is being captured. First, a large number of questions are constructed and then the ones who generated the most extreme responses are chosen to the final questionnaire [64].

Bangor et al. [65] has conducted extensive research studying the usability evaluation of various products and services using SUS. The questionnaire will generate a single score on a scale from 0-100 and is easily understood by a wide range of people. The authors discuss that both the mean and the median value of the SUS score are usable in an analysis. A fair system should have a score of at least 70, while better products score between the high 70s to the upper 80s. Truly superior products have a score better than 90. If a product scores less than 70, it should be considered a candidate for increased scrutiny.

A high SUS score is achieved if the respondent answers strongly agree on questions with an odd number, and strongly disagree for questions with an even number.

Al-Sumayti and Umar [66] evaluate the usability of a system using the SUS with the argument that it is a reliable scale. Furthermore, they mention that it is also considered as one of the easiest questionnaires to use and that when using small sample sizes it yields among the most reliable results compared to other questionnaires.

Sauro and Lewis [61] report that several non-native English speakers found that the word cumbersome in the original version of the SUS confused the respondents. Native English speakers claimed to understand the term, but non-native English speakers required clarification. In the questionnaire given to the respondents in this thesis, a translation of cumbersome to Swedish was given. Furthermore, the demographic information, such as educational background and job experience affect how the respondents rate usability [67]. Higher education and a more extensive experience had a positive effect on the usability rating, in terms of a higher rating. Thus, it is useful to ask them for this information.

3.4.2 Interview

As mentioned by Paz and Pow-Sang [56] interviews are one of the most commonly used usability evaluation methods. Interviews are also acknowledged to be the most commonly recognised form of qualitative research method [68]. The term qualitative interviewing most often refers to in-depth, semi-structured or loosely structured forms of interviewing. Mahrin et al. [60] mention that interviews are a common method for data collection if the research is conducted with real users responding based on their knowledge of the product.

Open-ended questions and semi-structured interviews are more commonly used when the researcher is performing a discovery-oriented investigation, compared to a confirmatory type of study where the researcher is more likely to use closed-ended questions [69]. Furthermore, extensive interviews are a fundamental element of usability testing [59].

The demands on the researcher includes that the wording of the questions can affect the results since the researcher can change the way he or she phrases them [70]. Furthermore, McNamara [71] has provided some guidelines for how to write interview questions. The list has been adapted from McNamara, stating that the questions should:

- have open-ended wording
3.4. Usability Evaluation

- be as neutral as possible
- be asked one at a time
- be worded clearly
- be careful against asking “why”

Furthermore, McNamara provided guidelines for how to start the interview session, the list has been adapted from McNamara [71]:

- Choose a setting with little distraction
- Explain the purpose of the interview
- Address terms of confidentiality
- Explain the format of the interview
- Indicate how long the interview usually takes.
- Provide them with contact information on how to get in touch with you later if they want to
- Ask them if they have any questions before you both get started with the interview
- Don’t count on your memory to recall their answers. Ask for permission to record the interview or bring along someone to take notes

There exist some trouble with collecting and analysing information from interviews with open-ended responses [70]. Since the open-ended questions aim for the respondent to express their answers in detail, it can be difficult for the researcher to compare the respondents’ answers. However, this approach can reduce researcher biases within the study, especially when the investigation is conducted with several respondents.

Interviews can also be used in collaboration with questionnaires, as a way of ensuring that the researchers understand the respondent’s answers correctly or if additional aspects need to be clarified [59]. The authors suggest that the questionnaire can be used as a checklist for managing a more conversational approach to the interview, in order for the respondent to volunteer information about themselves. This can make it harder for the respondent to guess the “right” answer, and the researcher might instead learn new useful things. Using a less structured approach of interviewing allows the researcher to ask follow-up questions which could help to decide if the respondent is close enough to the visualised profile in order to provide valid data.

The interviews should be recorded and transcribed immediately after conducted to ease the collection and evaluation of them [59]. Furthermore, the analysis of data falls into two distinct processes, with two different deliverables. Firstly, a preliminary analysis should be completed as soon as feasible after the interviews. The preliminary analysis should contain more significant trends and patterns. Secondly, a comprehensive analysis should be conducted two to four weeks after the interview. This analysis should include all the findings from the preliminary analysis, plus all the other analysis and findings that were not covered in the initial report. Finally, instead of seeking solutions when conducting qualitative research; the researcher should instead direct their efforts towards the making of arguments, thus constructing an interpretation or a line of reasoning regarding the collected data [68]. In order to verify the interview questions and ensure that it will give the desired outcome, a test interview should be conducted [72].
Moreover, it is crucial to decide whether or not the data is supposed to be read literally, interpretively or reflexively [68]. If the data is intended to be read \textit{literally}, the main interest lays in their literal form, content, structure, style, layout and so forth. Regardless of the view of literal readings, it is also important to decide to what extent an interpretive reading of the data will be made. An \textit{interpretive} reading will involve constructing or documenting a version of what the data is supposed to mean or represent. This implies \textit{reading through or beyond} the data in some way, whether or not this refers to texts, artefacts, visual images or something else.

It is also essential to decide how far a \textit{reflexive} reading of the data will be made [68]. A reflexive reading will locate the researcher as part of the generated data, and thus explore the researchers’ role and perspective in the data generation and interpretation process. Several qualitative researchers make a reading of their data on all three of these levels. Furthermore, the possibility of using diagrams and charts for displaying and organising the data can ease the process of reading, help with analytic thinking, and improve the presentation of the data. Visual diagrams or charts can merely be described as more “eye-catching” since they also can make complex material easier to understand.

Wilson describes three different types of interviews in the article \textit{Research Methods: Interviews} [73], structured interviews, semi-structured interviews and unstructured interviews. \textit{Structured} interviews include asking each research participant the same set of questions, with no room to move beyond that particular set of questions. Structured interviews are most suitable when conducting surveys, and can be quite limiting if used for qualitative research. Wilson also states that the use of \textit{semi-structured} interviews can offer more flexibility. Semi-structured interviews often contain some pre-defined questions but offer more room for discussion. \textit{Unstructured} interviews can be described as more of a conversation. The researcher should have an idea of what topics he/she would like to explore but should be more flexible and unrestricted when performing the interviews. It can be more challenging to compare data between interviews when using unstructured interviews.

\subsection*{3.4.3 Validity and Reliability}

Karlsson et al. [74] discuss validity when using interviews as a research method, by claiming that it is almost impossible to achieve an unbiased interview analysis. All interviewers will have their expectations and interpretations, which affect the analysis of the respondents’ answers. Furthermore, transcripts may be an insufficient way of analysing the results since accentuations and gestures are difficult to reproduce. By using several interviewers, the risk of bias and misinterpretations could be reduced.

In the article \textit{Foundations for Designing User-Centered Systems} the authors Ritter et al. [75] discuss the validity, reliability and sensitivity of usability evaluation. The terms relate to measuring the intended, which effects are measurable and to what extent it is possible to generalise the results. A study can have reliability and not viability, and vice versa, since they do not depend on each other. However, the terms need to be considered in both qualitative and quantitative studies, and they will all differ depending on the context in which the evaluation has been conducted. The validity of the study will stand for how accurate the results are [76]. Furthermore, it is essential to consider the validity, including the reliability, in all stages of the evaluation.

Ritter et al. [75] present different types of validity which need to be considered in evaluation. These have been divided into two categories: \textit{Instrument validity} and \textit{Experimental validity}. First, the instrument validity is shown, which relates to the instruments or measures used in the study [75]:
• **Construct validity:** refers to measuring what you think that you are measuring. It is not possible to measure the total usability of an application with one single metric since there exist different dimensions: efficiency, effectiveness and satisfaction, and these need to be measured separately.

• **Content validity:** refers to that the content of a measure correlates with that it was designed for. In terms of a usability study, this means that all the relevant aspects of the artefact have been covered by the survey, through a systematic review.

• **Face validity (or surface validity):** refers to if a test seems to measure some certain criteria. Also, it is closely related to content validity, but with the difference that it does not make use of a systematic review. Instead, people assess and judge the test based on its surface appearance.

Second, the experimental validity is shown below; this relates to the generalisability of the results [75]:

• **Internal validity:** refers to the ability to conclude the causal effect relationships based on the design of the study. Besides, this includes which measures that were used and the context in which the study was conducted. To be able to achieve a high internal validity there is a need to be able to have control over effects that could have an impact on the obtained results.

• **External validity:** refers to in which extent the results may be generalised to other contexts. It is crucial to consider who participate in the study. In order to achieve a high external validity, it is important to be aware of the effects which may affect the obtained results.

• **Ecological validity:** refers to, to which extent the results can be applied to a real-world context. To achieve a high ecological validity the method, material, and setting of the study must be similar to the studied real-life situation.

There will exist trade-offs between internal and external validity since a high external validity will reduce the internal validity and vice versa [75]. For example, the evaluation conducted in a laboratory setting will produce high internal validity but lose external or ecological validity or both. On the other hand, if the evaluation is carried out in a real-world setting, as an observation, for example, you will have high external and ecological validity but a decreased internal validity. The trade-off made need to be adjusted to the study in question since it depends on the research strategy.

Furthermore, as for reliability it is concerned with the ability of the measure to produce a consistent result if the same study is conducted again but under different conditions [75, 76]. More precisely, this means that a reliable study will produce the same results if the same test is conducted under the same conditions and with a similar set of participants, but on another day.

### 3.5 Requirements Elicitation

Zowghi and Coulin [77] make an effort to try to define the concepts and process of **requirement elicitation**. The authors define it as an approach to understanding the needs of different stakeholders and communicating that to the system developers. A significant part of this approach is to rigidly uncover, extract and surface the needs and desires of the stakeholders and probably expose more requirements than expected. Furthermore, the needs and desires of the stakeholders will change during the project and affect the quality of the requirements.
Zowghi and Coulin point out that several different elicitation techniques should be used in conjunction in order for the elicitation to be successful.

In *The guide to SWEBOK* [78] one of the fundamental principles for a successful requirement elicitation is effective communication between the stakeholders and that the communication continues through the entire software development process. Furthermore, the book also states that the scope of the project needs to be included in the elicitation and that the needs and deliverables of the stakeholders have been prioritised accordingly.

SWEBOK also brings forward different techniques for requirements elicitation, since the requirements seldom can be elicited directly, but rather the engineer elicits information from which requirements are specified. The principal techniques for elicitation are:

- **Interviews:** A traditional way of eliciting requirements, which include interviewing stakeholders.
- **Scenarios:** Creating a scenario which provides the engineer with a framework for questions regarding user tasks.
- **Prototypes:** A tool often used for clarifying opaque requirements. Prototyping can include both paper mock-ups and beta-versions of software.
- **Facilitated meetings:** A group of people brainstorm or similar to create a better insight into the software requirements, than what an individual could achieve. If it succeeds, it often results in a more precise set of requirements.
- **Observation:** Includes techniques such as ethnography, to elicit requirements. This is often expensive.
- **User Stories:** Includes construction of short and high-level descriptions of the required functionality, which have been expressed in client terms.

Kassab [79] has written an article in which he summarises three surveys conducted in 2003, 2008, and 2013 which investigated the state of practice of requirements elicitation. In the latest survey, the results from the study showed that the most used techniques for requirements elicitation were, in descending order: interviews, user stories, prototyping, and scenarios. The respondents were able to select multiple different techniques. The article also concludes that according to the surveys, the overall satisfaction of the practices has increased from 67% in 2003 to 87% in 2013. Saiedian and Dale [80], mention that the most important with requirements elicitation is not the method used, but instead that the client participates in the process of defining requirements.

Bano et al. [81] claim that interviews are the most widely used elicitation technique. The combination of technical and soft skills required for requirements analysts can make it challenging to conduct effective interviews for requirements elicitation. Karlsson et al. [74] also praise the interview approach, and state that the qualitative interviewing approach is useful when it comes to exploring an area of interest, obtain an overview of a complex area or discover diversities rather than similarities.

Creating scenarios is also a common technique in requirements elicitation [77]. Furthermore, a scenario provides a narrative and specific description of both the current and future processes. These processes can include actions as well as interactions between the users and the system. Creating scenarios during the requirements elicitation phase can also prove to be useful when it comes to both understanding and validating requirements, as well as during test case development. Furthermore, scenarios are suitable for a process of requirements elicitation that collects examples and descriptions from respondents and then tries to generalise them [82].
The three methods prototypes, facilitated meetings and observation were not chosen since they were deemed to be too time-consuming. Observation was also considered to be too expensive and facilitated meetings to require too much effort and time from the respondents. The goal of prototypes was also decided to be too far away from the aim of the requirement’s elicitation, since it focused more on application’s functionality rather than its usability. User stories were compared to scenarios, and the scenarios’ goal of describing processes was deemed to be more suitable than the description of the required functionality.

Respondents can be collected through a method referred to as purposive sampling. Guest et al. [83] describe purposive sampling as a way of selecting participants according to some predetermined criteria to a specific research objective. Ritchie and Lewis [84] similarly describe purposive sampling as a way of choosing respondents with a “purpose”, in order to lay down sound bases for the success of the investigation. If compared, the “purpose” in Ritchie and Lewis’s definition can be seen as the predetermined criteria in the definition provided by Guest et al. In conclusion, both definitions highlight the importance of choosing the respondents meticulously.

In conclusion, purposive sampling was used to decide on the pool of respondents. Interviews and scenarios were deemed to be most beneficial and suitable for the requirements elicitation process of this thesis. The reasons for choosing interviews were that this is the most common elicitation method and for the possibility to collect a wide range of unstructured opinions in a fast manner. Scenarios were deemed most suitable because it would allow for collection of examples and descriptions of the views and uses of cloud today at the company. These methods will now be discussed further.

3.5.1 Interviews

Bano et al. [81] describe several common mistakes when conducting interviews, were the most common ones were question formulation, question omission, the order of interview questions and communication skills. Dick, Hull and Jackson [85] mention that the researcher needs to be a good communicator in order to retrieve real requirements from the stakeholder interviews. Furthermore, the researcher is required to take a comprehensive set of notes and not to be judgemental [85]. The set of notes can be used later to organise a structured set of requirements, and after that returned to the stakeholder. Additionally, the researcher’s experience in interviewing, as well as the format of the interview, affect the results from the interviews. Mistakes made during the design and execution of the interview can severely impact the results requirements [81].

Goguen and Linde [86] also discuss several different requirements elicitation methods, and mention interviews as one of the most successful. Close-ended questions can, however, turn out to be problematic when conducting interviews for requirements elicitation if the respondent does not share the same views/systems as the set of answers the interviewer has selected. This could lead to several answers being marked as “do not know” or “other”. Goguen and Linde state that categories and concept that are transparent to one community can be entirely opaque to members of another set of people. Furthermore, the open-ended interview avoids several problems related to the close-ended method.

3.5.2 Scenarios

Scenarios can be described as the actions and interactions of a user and the system [77]. Sutcliffe [87] agrees that scenarios have become a popular method in requirements elicitation, but claims that unfortunately the term has been interpreted by far too many authors to have a commonly accepted definition. Scenarios can be used in both business systems analysis and requirements elicitation, and Sutcliffe discusses their different advantages and disad-
vantages. One advantage, which also provides a challenge, is that it provides a good set for argument and reasoning in specific details. The challenge is that the general view is lost.

Furthermore, Sutcliffe discusses that since scenarios are examples closely related to the real world and it is common to tend to seek only positive examples which correspond to our preconceptions, scenarios need to be chosen carefully. Scenarios should also be chosen in such a manner that they include errors and exceptions as well as the norm. In the requirements elicitation process, multiple nonhomogeneous scenarios should be gathered.

A scenario may contain information of both actors, roles, and their organisation settings [82]. Additionally, it is easier for a stakeholder to relate to a scenario than to a more abstract description of the needs of the system. It is both possible to use more informal representations, as well as more formal models, such as event scenarios or use cases. Furthermore, a scenario is described to contain the following characteristic elements:

- Setting (context of the environment),
- Actors or agents,
- Sequences of actions, events, and goals

A description of the scenario has to include at least one actor, who completes a set of tasks in order for a specific goal to be fulfilled in the selected setting, and at least one goal.

3.5.3 Sustainability

Sustainability can be described as the capacity to preserve the functionality of a given system over an extended time [88]. García-Mireles [89] analyses sustainability from a software quality perspective by stating that sustainability can be discussed in terms of both different entities as well as different abstraction levels. Furthermore, there exists some confusion and misunderstanding when it comes to software sustainability terms and concepts, and the consensus about the meaning of software sustainability in Software Engineering is still emerging. The concept of sustainability is complicated as well as multifaceted since it can be viewed from different perspectives and it might include several dimensions and factors. A lack of information regarding how sustainability can be decomposed in other qualities exists when sustainability is treated as a nonfunctional requirement, or as a software attribute.

Furthermore, García-Mireles also discusses different perspectives of sustainability, where some only consider one dimension, environmental, and some divide sustainability in several different dimensions. Regarding sustainability dimensions, the classification considers the following dimensions: environmental, technical, social, economic and individual. Environmental dimensions are focused on protecting more natural resources, whereas the technical dimension aims at evolving software for a long time. Social dimensions are more concerned with creating communities and supporting each other, and the economic dimension is more related to wealth creation. Lastly, the individual dimension concerns personal development.
To elicitate the requirements for the respondents and the interview guide for this thesis, a pre-study was conducted. The purpose of a pre-study is to search, describe and choose different solutions and assess the value and feasibility of the concept. This chapter starts with a presentation of how the pre-study was conducted, which include the methods used during the literature study and the requirements elicitation process. It continues by describing the obtained results from these processes.

4.1 Method

This thesis was initialised with a comprehensive literature study in order to form a theoretical framework. The literature study additionally helped to precise the scope and direction of the thesis. Additionally, the pre-study phase contained the elicitation of requirements. The methods used during the literature study and the requirements elicitation are described in this section.

4.1.1 Research Methodology

Kothari [90] describes two different approaches to research, quantitative and qualitative approach. A quantitative approach includes the collection of data in a quantitative format which can be analysed formally and rigidly. On the other hand, the qualitative approach collects data related to the subjective assessment of attitudes, opinions and behaviour. It will produce soft values and data which can not be subject to quantitative analysis. The approach taken in this master’s thesis was, therefore, a qualitative approach since the data was mainly collected through interviews which are not suitable for quantitative analysis.

4.1.2 Requirements Elicitation

The requirements elicitation was performed in the pre-study to define an initial set of requirements for the main interviews regarding subjective satisfaction and learnability. As described in Section 3.5, the method used for requirements elicitation was primarily interviewing since it is the most common elicitation method [78, 79] and the possibility to collect a wide range of unstructured opinions in a fast manner. Additionally, the elicitation technique scenarios was used during the process, to validate and better understand the requirements [77]. The
selection of respondents for the main interviews were conducted during the requirements elicitation phase.

**Interviews**

Several unstructured qualitative interviews were conducted with knowledgeable employees at the company. The interviews were conducted to gather information about CC and how the company uses the cloud in their daily business today. Unstructured interviews were used to gather the requirements since they will allow for a more loose and informal conversation [73]. A more conversational form was desired since it allowed for creating a relaxed setting, so the respondent felt that every piece of information regarding the chosen topic was significant. However, as stated by Turner [70] it is critical to understand that the flexibility in the questioning is the key to success when conducting more conversational interviews.

During the interviews, one person asked the questions, and the other took notes. This approach was chosen so that the focus of the interviews was divided between the interviewers. The person taking notes was allowed to ask clarifying questions or request additional information if it was deemed necessary. The interviews were loosely structured, and no interview guide was formed, instead the respondents were mainly asked to describe their relations to cloud services, and how they and Exsitec have worked with it. Furthermore, the respondents were asked to elaborate on their view of how cloud services affect their work.

In total, four interviews were held during the requirements elicitation, with personnel of knowledge regarding how Exsitec utilises cloud in their business. The respondents had together extensive knowledge of the different aspect of how they and other end-users at Exsitec, as well as how their clients, utilised cloud. They aimed at creating a broad knowledge about the structures and problems that they faced.

**Scenarios**

The data collected from the interviews were used to create six scenarios. All scenario descriptions contained a setting, actors and a sequence of actions, as according to the instructions from Alawairdhi and Aleisa [82]. By using the results from the pre-study interviews during the creation of the scenarios, the end result was as closely related to the real world as possible. The scenarios were created by going over the collected results from the pre-study interviews and highlighting different themes, or aspects, that were regularly discussed. More than six scenarios were created during the first iteration, but several were removed or merged into other scenarios in the end of the process.

**Respondents**

The respondents in the study were chosen based on recommendations from the pre-study interviews and followed the method purposive sampling. All respondents needed to have competence in both cloud services and solutions, on-premise solutions and be able to compare them, which restricted the amounts of available respondents at Exsitec.

Ritchie and Lewis [84] also state that the decisions about the criteria used during selection should be completed during the early stages of the research, which gathering of requirements during the pre-study of this investigation. They also suggest that the selection of participants should be as diverse as possible, within the boundaries of the defined requirements. Diversity is needed to optimise the chance of identifying all underlying factors and circumstances. Because of this, the location of the respondents was not included as a requirement.

Furthermore, the recommendation regarding the number of respondents when using purposive sampling differed between researchers, according to Guest et al. [83]. Some recommendations ranged from five to twenty-five respondents, some from thirty to fifty and some from
six to twelve. Guest et al. stated that if the aim is to understand common perceptions, and experiences, in a relatively homogeneous group of individuals, twelve respondents should suffice. Since Exsitec is a relatively small company, and that a set of requirements existed, the number of possibly suitable respondents decreased quickly. Furthermore, the respondents with suitable experience worked within one division of about 21 employees, which further reduced the available suitable respondents. The target amount of conducted interviews for this investigation was set to somewhere between 10 to 15 interviews. Furthermore, the research conducted by Guest et al. show that a small sample group can be highly sufficient, as long as the participants possess a certain degree of expertise in the investigated domain.

The respondents were contacted over email, and the interview was held in person if they were situated in Linköping. If the respondent was in another location the interview was held through Google Hangouts, and thus all interviews were conducted as face-to-face meetings.

4.2 Result

This section includes the results obtained from the pre-study, which includes the pre-study interviews, the interview guide, the respondent requirements and the test interview. From these results, both the problem formulation and the theoretical framework were established.

4.2.1 Interviews

Four interviews were held during the requirements elicitation process; these interviews will be referred to as the pre-study interviews. The pre-study interviews had two different purposes. Firstly, they were used to gather information about how Exsitec uses CC and the problems the company is facing. The resulting information was later used to create the interview questions used during the investigation phase. Secondly, the interviews were used to collect suggestions and requirements for suitable interview subjects to interview during the investigation phase. In order for someone to be considered an interesting interview subject the person needed to have knowledge and experience from on-premise applications as well as cloud services and solutions.

The respondents from the pre-study interviews will henceforth be referred to as Respondent A, Respondent B, Respondent C and Respondent D. First and foremost, all respondents in the pre-study interviews had some form of knowledge or competence regarding cloud and on-premise solutions. The respondents began by describing how Exsitec utilise the cloud to help their clients.

Exsitec provides their client with the possibility to customise integrations if the provided software solutions lack any of the clients’ desired or necessary functionality. Several of the respondents discussed these integrations and mentioned several benefits as well as challenges when using cloud services and solutions. Only the parts of the interviews that fit the aim of this thesis will be presented.

Respondent A

Respondent A said that Exsitec’s clients pay Exsitec to be able to host their applications in Exsitec’s integration platform. The clients often have their systems and applications hosted in the cloud, which makes it easier to host the integrations developed by Exsitec in the cloud as well. According to Respondent A, Exsitec has traditionally used other Microsoft solutions and applications, which simplified the choice of picking Azure as a cloud provider.

Furthermore, Respondent A explained that the solutions were previously hosted at the clients’ locations using an on-premise solution. Exsitec then needed a third party company
4.2. Result

handling the on-premise solution. By hosting the applications in the cloud, instead of on-premise, Exsitec does not need to use this third-party company anymore, which is beneficial for them.

Respondent A also discussed a client, who built a system from scratch and hosted it in the cloud. The client wanted to provide a SaaS solution to their clients and asked Exsitec to help them. Exsitec constructed a PaaS solution which the client then could use as a SaaS. A significant advantage with the cloud environment, for Exsitec, is that it is more standardised, which makes it easier to start new client projects. Furthermore, it simplifies logging, since Microsoft offers error messages in a more standardised way. This makes it easier to discover errors and to adjust them. It also makes it possible to search through history and statistics.

Respondent A continued by describing that the deployment pipeline can be reused between several clients. This makes the take-off distance shorter since they do not have to set up accounts, virtual private networks, and similar. Exsitec can instead focus on creating value for the clients. Another benefit for Exsitec, when it comes to cloud, is that the client does not have to update their application when new software arrives since this is done by Exsitec instead. Respondent A stated that working with cloud services also makes continuous delivery of software easier.

Respondent B

Respondent B has worked a lot with comparing on-premise to cloud solutions, regarding how to manage information. The respondent often works with Microsoft Power BI\(^1\). A great deal of the development process regarding how to manage information lies in the underlying architecture. Exsitec needs to continually discuss and compare how it will affect them whether they choose to continue developing the application locally, an on-premise solution, compared to a cloud solution, using Azure. Respondent B shared that this is regularly discussed by Exsitec, to try to decide which solution will provide the best opportunities.

Respondent B claimed that one of the advantages of using the cloud is that it makes it possible to use several different types of data storage. It is also relatively easy to build architecture for information management in a cloud service. The cost versus the need is something that always requires to be considered when discussing whether or not it would be beneficial to use a cloud service instead. Furthermore, regarding usability, Respondent B thought that it should always be best to have an application stored in the cloud, strictly theoretically. However, due to lack of competence and an increase of complexity, it still needs to be discussed whether or not it would be the best choice. Even though the portals are functional and user-friendly in theory, it may be more complicated in real life. One problem Respondent B brought up, is that there is no possibility to dump and restart a database.

Respondent B discussed that there are several knowledge areas on how to set up or change the cloud environment, in which the developers lack competence. Thus, it is harder to estimate and realise the cost when providing the client with an offer. Every extra customisation and integration makes the application cost increase for the clients. This can sometimes confuse the clients if they expect that the bought service should provide the functionality in the standard version, but Exsitec instead needs to develop the integration, thus increasing the cost.

Respondent B also discussed some benefits of using the cloud, and mainly mentioned the increased scalability and the fact that the applications can be shut down or altered during the evenings, to decrease unnecessary cost when the product is not in use. However, cloud applications often require more effort, which leads to more hours and a higher cost for the clients.

\(^1\)https://powerbi.microsoft.com/en-us/


Respondent C

Respondent C discussed how cloud services are used in several client projects at Exsitec. Some clients have, for example, changed to a third-party cloud-based economic system. These systems deliver products and provides the system against a fee, based on a specific set of parameters. Several of Exsitec’s clients are not that knowledgeable when it comes to cloud services and cloud-based solutions, which can cause some communications issues between Exsitec and the client.

Exsitec is not allowed to make changes directly in the third-party system’s cloud environment, and the clients usually want the integrations to be placed in the same cloud environment as these systems. Sometimes, the client can argue that some functionality is missing from the third-party system and that it should not be their responsibility to pay for it. These types of problems mainly revolve around the fact that the client, Exsitec and the system provider have different expectations on one another, according to Respondent C.

Respondent D

During the final pre-study interview, Respondent D discussed that the clients might not notice too much if Exsitec uses on-premise or cloud solutions, except perhaps when it comes to availability and speed. It can be more beneficial for Exsitec to have cloud-based solutions since they do not have to teach the clients as much for them to be able to use the applications. This is mainly because Exsitec, instead of the client, will be receiving error messages when a cloud service is used. Respondent D stated that the clients are more pleased when they do not have to acknowledge any error messages.

Furthermore, the competence of both the developers and the clients, as well as the economic aspect, need to be taken into consideration when choosing whether a cloud solution would be more beneficial than an on-premise solution. If Exsitec is hosting their solutions on-premise, they need to consider the server capacity and optimise it in order for it to be financially sustainable, according to Respondent D.

Respondent D discussed that there is a higher need for surrounding infrastructure when implementing cloud solutions and it takes more time for the developers. Respondent D experiences that it takes more effort and help to set up a new application hosted in the cloud. Exsitec’s integration platform could host multiple streams together and control these through a set of parameters. The integrations platform also enables Exsitec to reuse code and functionality between different clients and products.

Furthermore, Respondent D stated that if everything that Exsitec works with is cloud-based, everything can be updated regularly and installed quickly, making version control more straightforward and more fluently. For on-premise solutions, they need to push out an update and sometimes help the client to install it. Moreover, if all Exsitec’s integrations were cloud-based solutions, it would simplify the error handling and versioning. Cloud also eases the work for the developers when using continuous integration and continuous delivery, which also enable smaller regular updates. Smaller regular updates also increased the availability since it does not require the entire application to be shut down in order for an update to be completed.

Regarding if cloud or on-premise solutions is preferable mainly depends on the size and growth of the company, according to Respondent D. A high growth might make it easier to use the cloud, because of the scalability. Furthermore, Respondent D explained that a larger developing team working on a more critical application probably should choose cloud, since the developers do not need to sync their work as much with each other. The developers can instead do smaller updates on their respective module continuously.
4.2. Scenarios

The requirements elicitation resulted in six scenarios, which covers the subject brought up in the pre-study interviews. The scenarios are presented below:

1. **Deployment Effort**
   Cloud services enable a shorter start-up process of new projects since it is easier to reuse some code or settings. Several existing services in Azure make it possible to reuse the deployment pipeline between different clients. By reusing existing resources the development process is simplified.

2. **Customisability**
   The ability to customise client solutions is a required functionality in order to be able to make the end product client-specific. The resources also need to have high elasticity, to simplify the process of scaling client projects up and out.

3. **Error Solving**
   Azure provides a standardised way of logging error messages, and it is easier to discover and adjust them. It is also possible to search through error history and statistics.

4. **Software Updates**
   Cloud enables the user always to be aware of the version run by the client and the possibility to update the software. Cloud services also empower continuous integration and delivery and perform smaller updates on the application.

5. **Client Approach**
   When trying to sell a development project to a client, it is necessary to provide cost budgets in the offers. Using cloud services make it harder to compose the cost estimate. Every customisation and integration along with the elasticity and resource pooling will affect the cost. Moreover, the clients have different IT maturity, which means that they have different expectations on the functions and performance of cloud services.

6. **Ease of use**
   It is important that the systems a developer utilise daily is easy to understand, learn and use in order to ensure high productivity. Sufficient documentation and guidelines are therefore required. A development environment with low usability will affect the end-users’ subjective satisfaction negatively.

4.2.3 Respondent Requirements

The requirements elicitation mainly focused on getting information and knowledge on how the main interviews should be conducted. The gathered requirements were based on both how to find suitable respondents for the SUS and the main interviews as well as input regarding the content and direction of the main interviews.

A suitable respondent should be an end-user with experience in both developing for on-premise solutions as well as for cloud solutions, in order to be able to compare the two. The respondent does not need to be an employee of Exsitec but should, in that case, be a client to Exsitec, a partner of Exsitec or similar.

Table 4.1 show the respondents who participated in the questionnaire and the main interview. The table includes their role at Exsitec, if they had participated in the study earlier and where they are situated.
### Table 4.1: Respondents in the questionnaire and the main interviews

<table>
<thead>
<tr>
<th>Role</th>
<th>Location</th>
<th>Participated earlier</th>
<th>Working at Exsitec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer (Consultant)</td>
<td>Uppsala</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Developer (Consultant)</td>
<td>Linköping</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Developer (Subcontractor)</td>
<td>Linköping</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Responsible for cloud delivery</td>
<td>Göteborg</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Developer (Consultant)</td>
<td>Linköping</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Developer (Consultant)</td>
<td>Linköping</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Developer (Consultant)</td>
<td>Linköping</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Developer (Consultant)</td>
<td>Stockholm</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Competence area responsible (Consultant)</td>
<td>Stockholm</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
5 Method

This chapter will describe and motivate the chosen methods during the investigation and evaluation. The three main sections Study Structure, Investigation and Evaluation, constitutes the layout of the chapter. The first section, Study Structure, presents an overview of the study. The investigation chapter includes a detailed description of the methods used during that phase. The chapter also presents some additional theory related to the research methodology and describes how the collected results will be evaluated.

5.1 Study Structure

The structure of the study consists of two main phases that together provided the overall conclusion. The research conducted during the investigation phase examined the subjective satisfaction and learnability of the Azure portal and Exsitec’s integration platform, compared to the same of on-premise services. By conducting interviews with personnel at Exsitec, both in person and over Google Hangouts, the subjective satisfaction and learnability has been researched. The respondents were asked to fill out a standardised pre-made questionnaire at the beginning of the interview to measure the subjective subjective satisfaction and learnability. The interviews were completed, transcribed and then analysed twice to assert the truthfulness of the results.

During the evaluation phase, the results from the questionnaire and the interviews were analysed according to the selected methods and guidelines. Finally, based on the outcomes of the investigation, the research question was answered. The method and the results were discussed, from which the conclusion regarding the research question could be drawn.

5.2 Investigation

This section describes the two chosen methods for investigating the subjective satisfaction and learnability: questionnaire and interview. As mentioned in Section 3.4, in terms of learnability the questionnaire and the interviews are considered an indirect method and thus they measure the users’ opinions on the learnability of the interface. Regarding subjective satisfaction, questionnaire and interviews are considered a direct method. Furthermore, the section describes the interview guide and the test interview.
5.2. Investigation

5.2.1 Questionnaire

One part of investigating the subjective satisfaction and learnability was by using the pre-made questionnaire The System Usability Scale (SUS), presented by Brooke [64] in the article SUS - A quick and dirty usability scale. SUS is, as previously mentioned, the most widely used standardised questionnaire for assessing subjective usability, and the original paper by Brooke has been cited over 5500 times, as of March 2018 [63]. It is also considered to be one of the easiest questionnaires to use and to yield among the most reliable results compared to other questionnaires when conducted with few respondents [66]. The usability attributes learnability and satisfaction, provided by Nielsen [12] was evaluated with the SUS. Question four and ten from the questionnaire are more related to learnability whereas the remaining questions are more related to satisfaction [61].

According to the guide on how to use SUS the respondent needs to answer all questions, if the respondent is not able to do so, the centre point of the scale should be marked. When a respondent has filled out the questionnaire, it will yield a single number, which represents a composite measure of the overall usability, of the system which is studied. The SUS score is then calculated as below:

\[ \text{To calculate the SUS score, first sum the score contributions from each item. Each item’s score contribution will range from 0 to 4. For items 1,3,5,7, and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is five minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU. [64, p. 5]} \]

The questionnaire published by Brooke, and used in this thesis, can be found in Appendix A. The questionnaire contains the term the system, which in the case of this thesis refer to the Azure portal or the Integration Platform, or both, depending on the exact experience of the respondent. The term was clarified with the respondent before filling out the questionnaire. Thus, the questionnaire will not create an explicit comparison of the respondent’s thoughts on cloud and on-premise. Every respondent answered the questionnaire once. Additionally, since the SUS is a standardised questionnaire and the questions are quite general and non-specific, the respondents were asked to rate the questionnaire. The rating was based on a scale from one to five regarding how well the questionnaire investigated the usability of a system, in terms of satisfaction and learnability.

5.2.2 Interview

The answers to the SUS questions can only indicate that there exist a problem, and one goal with the interviews was to investigate the more concrete problem. Another goal was to understand how cloud services could ease the work of the end-users, and thus questions regarding the differences between cloud and on-premise solutions were asked. The interview was conducted directly after the completion of the questionnaire with the respondent, in order for the subjective satisfaction and the learnability to be evaluated to its fullest. A qualitative approach, as described by Kothari [90], was selected for the collection of data, since interviews are not suitable for quantitative analysis. The interview will be used both to clarify the respondent’s answers, as well as ask other questions created from the requirements elicitation process.

Additionally, the interviews were chosen to be semi-structured. Although this approach is more structured than conversational interviews, there is still some flexibility when composing the interview guide. Using standardised open-ended questions allows the respondents to be asked identical questions, with open-ended responses that allow the respondents to share
5.2. Investigation

as much detailed information as desired. Furthermore, it also allows the researcher to ask follow-up questions.

Another reason for choosing this approach is, as described earlier that the semi-structured approach allows the interviews to start from a shared space but can take different directions based on the discussion with the respondent [73]. The interviews were, if possible, held face to face, since this also provides the ability to see facial expressions and body language, which could be useful in the data analysis of the results. Video calls over Google Hangouts was used if it was not possible to conduct the interviews in person.

Interview Guide

The construction of the interview questions is the most crucial part of the interview design [70] and followed the principles described in Section 3.4.2. The interview questions were chosen carefully and revised during several iterations. Many questions had subquestions to be as understandable as possible. Furthermore, the questions in the interview guide were formed from the information presented in the Pre-Study Chapter. Some of the questions were asked to investigate questions from the questionnaire more in-depth and to understand which attributes the respondent thought of when completing the questionnaire.

The interview guide was written in several drafts and revised with the help of a test interview. The questions were based on both the SUS questionnaire, on the pre-study interviews and the scenarios, and on the theory presented.

The SUS questionnaire was used in order to be able to investigate deeper which attributes that affected the subjective satisfaction and learnability. The result of the pre-study interviews and the scenarios was created to provide the foundation for the interview guide and to generate a deeper understanding of how Exsitec utilises cloud services today. The presented theory included benefits of utilising cloud services and solutions; therefore some questions were derived from it. These questions mostly related to how cloud services and solutions could ease the effort of the end-user and their subjective subjective satisfaction.

As mentioned earlier, the respondents’ experience can affect the SUS score [67], and thus the interview guide starts with a section of questions regarding the respondents experience in the field and their knowledge about cloud and on-premise solutions. These questions were asked in order to receive a common understanding of the different respondents’ knowledge. Furthermore, this section includes a question regarding the respondents’ thoughts of the questionnaire, and how well it investigated the subjective satisfaction and learnability. Also, it includes some information regarding that the interviews were recorded, with the respondent’s approval, that the answers were anonymous in the thesis, and that the questionnaire should be completed before the start of the interview and that the respondent needs to answer all questions in the questionnaire.

Furthermore, the guide has been divided into two themes, satisfaction and learnability. The themes have been defined in order for the interviewer to ease the understanding of them for the respondent. Moreover, the respondent can answer the questions with that definition in mind. Therefore, some questions may be similar between the two different themes, since the purpose of them is similar, but they are investigating from two different perspectives.

The respondent was asked to compare cloud and on-premise solutions, as well as discuss how it has affected their satisfaction or learnability. A majority of the questions, therefore, includes one or more subquestions, which have the aim to give a deeper understanding of the respondents’ reflections. The subquestions might also include different attributes which the interviewer can use to clarify the question to the respondent.
5.3 Evaluation

Test Interview

The test interview was conducted with a respondent outside the project and the company, who had experience in working with both cloud services and solutions, as well as with on-premise solutions. The reason for choosing a respondent outside the project was to receive a new view on the questions and the relation to the aim and research question of this thesis. Another reason was that it was desirable to conduct a test interview with a person with knowledge in both cloud and on-premise services. People possessing this knowledge within the company, and who were also able to participate in the thesis, were already scheduled for the primary interviews. Thus, it was not preferable to conduct the test interview with one of them if other possibilities existed. Lastly, since one of the respondents participating in the interviews does not work at Exsitec, it was important that the questions were suitable regardless.

The interview was performed similarly to the main interviews, and the test respondent answered all the questions, including the questionnaire. However, unlike the main interviews, the test interview was not recorded, and thus not transcribed. Moreover, the test interview was held more informally than the other interviews. The reason why the test interview was neither recorded nor kept as formal was that the purpose was to investigate the suitability and understandability of the questions and not the answers to them.

5.3 Evaluation

This section describes the methods used for evaluating the collected results, including how to evaluate the SUS score and the interviews.

5.3.1 Evaluating the SUS Score

As mentioned previously, the subjective satisfaction and learnability were investigated using two different approaches: a questionnaire (SUS) and a set of interviews.

The completion of the SUS generated a SUS score, which was evaluated with the Sauro-Lewis Curved Grading Scale [91] seen in Table 5.1. Additionally, the evaluation followed the recommended guidelines by Bangor, Kortum and Miller [65].

Table 5.1: Curved Grading Scale Interpretation of SUS Scores Adapted from [91].

<table>
<thead>
<tr>
<th>SUS Score Range</th>
<th>Grade</th>
<th>Percentile Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>84.1-100</td>
<td>A+</td>
<td>96-100</td>
</tr>
<tr>
<td>80.8-84</td>
<td>A</td>
<td>90-95</td>
</tr>
<tr>
<td>78.9-80.7</td>
<td>A-</td>
<td>85-89</td>
</tr>
<tr>
<td>77.2-78.8</td>
<td>B+</td>
<td>80-84</td>
</tr>
<tr>
<td>74.1-77.1</td>
<td>B</td>
<td>70-79</td>
</tr>
<tr>
<td>72.6-74</td>
<td>B-</td>
<td>65-69</td>
</tr>
<tr>
<td>71.1-72.5</td>
<td>C+</td>
<td>60-64</td>
</tr>
<tr>
<td>65-71</td>
<td>C</td>
<td>41-59</td>
</tr>
<tr>
<td>62.7-64.9</td>
<td>C-</td>
<td>35-40</td>
</tr>
<tr>
<td>51.7-62.6</td>
<td>D</td>
<td>15-34</td>
</tr>
<tr>
<td>0-51.7</td>
<td>F</td>
<td>0-14</td>
</tr>
</tbody>
</table>
5.3. Evaluation

5.3.2 Evaluating the Interview

As mentioned in Section 3.4.2 the collecting and analysing of results may provide some difficulties when using interviews with open-ended responses [70]. Therefore, a preliminary analysis was conducted immediately after the interviews, containing thoughts and impressions as well as general guidelines of the interview’s direction. Similarly, a more comprehensive analysis was conducted approximately two weeks after each interview, according to the guidelines provided by Rubin and Chisnell [59].

Rubin and Chisnell also state that similar answers should be grouped into meaningful categories when summarising performance data from qualitative interviews. The categories will ease the process of scanning the results for a general indication of their content. All questions were listed, and similar answers were grouped and labelled. In this thesis, the respondents’ opinions have, as far as possible, been grouped into the scenarios.

The resulting data from the investigation will be read literally, mainly regarding its content. An interpretive reading will be conducted, to a small extent, in order to read beyond the data and retrieve the core of the respondents’ opinions. Similarly to the interpretive reading, a smaller reflexive reading will be conducted to explore the researchers’ role in the resulting data lightly.
This chapter presents the obtained results from the creation of the interview guide, the conduction of the test interview, the experience of the respondents who participated, and the investigation phase. All obtained results during the investigation have been evaluated according to the described methods. The chapter is structured in the order of how the stages were conducted, e.g., first the interview guide, then the test interview, the respondents experience, followed by the SUS questionnaire and lastly the results from the interviews, grouped by scenario.

6.1 Interview Guide

The complete interview guide with questions can be found in Appendix B. As mentioned in Section 5.2.2, the pre-study interviews were a foundation for the interview guide. An example of a question based on the pre-study is:

If we discuss the start-up of new projects, what differences do you experience when starting a new project that will be hosted on-premise compared to whether it should be hosted in the cloud? Please be as concrete as possible.

The question above also included subquestions, regarding attributes the respondent’s answer may revolve around. Several of the respondents in the pre-study interviews discussed this matter and that their experience showed that it was more accessible to start up new projects when utilising a cloud instead of an on-premise solution. Therefore, it was considered an important aspect to investigate both in terms of if it is considered more natural or not, and which aspects that are considered essential. Other matters discussed in the pre-study interviews which showed similar interest as the one above, were also included in the interview guide. These questions were formulated similarly, and often with subquestions.

Also, some of the questions were based on the presented theory. One example of this is the following question:

When it comes to the concepts of resource pooling and the elasticity of resources, do you notice any difference if you compare cloud solutions with on-premise solutions?
This question was based on the theory presented in Section 3.2.2 and that CC should enable resource pooling and rapid elasticity. As with the other questions, these often included subquestions.

Lastly, the interview guide contained questions based on the SUS questionnaire. An example of this type of question is:

Based on the question: *I thought the system was easy to use*, which attributes did you think of when answering that question?

Multiple of these questions revolved around that the respondent should elaborate on which attributes that they considered when filling out the form. Moreover, thus, provide a deeper understanding of which aspects are essential.

### 6.2 Test Interview

The respondent worked outside the company and thus did not know how Exsitec works with Azure or the integration platform. Instead, the answers were from the respondent’s perspective and experiences. Furthermore, the respondent had experience working with external clients, but on a smaller scale compared to Exsitec.

The test interview showed that some of the questions were quite long and that the respondent had trouble remembering them. Since multiple questions refer to the questionnaire, it was decided that the questionnaire should be available to the respondent during the whole interview, in order for the respondent to be able to read and see the answers. The interviewer should also be prepared to repeat the question when needed.

Additionally, the test interview showed that some questions could be formulated in a better way to make them more precise. One question, regarding the systems the respondent usually works in, was removed because it was too broad and other questions fulfilled the purpose of that question.

The respondent mentioned that some questions were repetitive. This was, however, not changed since the questions in question belonged to different themes, e.g. learnability and satisfaction. Nonetheless, the questions have been noted to resemble each other, and during the primary interviews, the interviewer clarified for the respondent that he or she should answer with the different themes in mind. The reason for keeping the repetitive question was that the purpose was to investigate deeper which attributes that contribute to increasing the subjective usability of the end-user and thus it is preferable that the respondent elaborates on their answers.

Lastly, the respondent said that the questions had a proper close relation to the aim and the purpose of this thesis and that it covered the respondent’s thoughts on the matter. The test respondent suggested ending the interview with a question regarding if the respondents wanted to add or change any information. Such a question was therefore added at the end of the interview guide.
6.3 Respondent Experience

The respondents’ experience is shown in Table 6.1, both in total years in the field of software development as well as what they have worked with. The respondents have been numbered in order to keep their anonymity.

Table 6.1: The respondents’ experience, both as total years in the field of software development as well as from a cloud and on-premise perspective.

<table>
<thead>
<tr>
<th>Respondent</th>
<th>Experience</th>
<th>Cloud experience</th>
<th>On-premise experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Respondent 1</td>
<td>3 years</td>
<td>Has worked a lot with the Azure portal as well as SSAS, SSRS, and SSIS.</td>
<td>Quite similar with SSAS, SSRS and SSIS. Also extensive knowledge in Microsoft BI and backend development.</td>
</tr>
<tr>
<td>Respondent 2</td>
<td>22 years</td>
<td>Much knowledge within the Azure portal and hosting of different web applications. Was responsible for the installation in one of those projects.</td>
<td>Has installed several applications on-premise. Worked on many client projects, but not with infrastructure.</td>
</tr>
<tr>
<td>Respondent 3</td>
<td>20 years</td>
<td>Various cloud solutions within Azure. Has worked the last five years worked predominately with the cloud.</td>
<td>Extensive knowledge throughout many various projects.</td>
</tr>
<tr>
<td>Respondent 4</td>
<td>20 years</td>
<td>Construction and hosting of several client projects. Has worked with cloud solutions for nearly 20 years.</td>
<td>Essentially the same. Many infrastructures, working with the underlying layer of applications.</td>
</tr>
<tr>
<td>Respondent 5</td>
<td>2 years</td>
<td>Has worked a lot with Exsitec’s integration platform, and the Azure portal.</td>
<td>Has worked with several client solutions, and on-premise based ERP systems.</td>
</tr>
<tr>
<td>Respondent 6</td>
<td>9 years</td>
<td>Not a lot of experience. Has created a testing environment in the Azure portal and worked with cloud-based ERP systems.</td>
<td>Has worked with various on-premise based client applications.</td>
</tr>
<tr>
<td>Respondent 7</td>
<td>4 years</td>
<td>Has worked with Azure and various cloud-based ERP systems and smaller APIs of various kinds.</td>
<td>Most solutions worked with has been on-premise based client applications.</td>
</tr>
<tr>
<td>Respondent 8</td>
<td>15 years</td>
<td>Has hosted several web applications in the cloud. Has worked with the Azure portal to some extent during the last four years.</td>
<td>Extensive knowledge working with several client applications.</td>
</tr>
<tr>
<td>Respondent 9</td>
<td>4 years</td>
<td>Has worked on several smaller and middle scale projects hosted in the cloud. And, on one large cloud-based client project and several hybrid solutions. Has also worked with several cloud-based ERP systems.</td>
<td>A lot of experience working with on-premise based client applications.</td>
</tr>
</tbody>
</table>
6.4 System Usability Scale

This subsection will present both each respondents score from the SUS and their view on how well SUS investigated the usability of a system.

6.4.1 SUS Score

The SUS scores have been calculated according to the description in Brooks article [64], and are shown individually in Figure 6.1; the mean value was 60.8, and the median was 62.5. By comparing both the mean value and the median to the grading scale, the overall grade D was retrieved, in agreement with the scale presented by Sauro and Lewis [91]. Bangor et al. [65] explains that a product that scores less than 70 should be considered a candidate for further investigation, which is true for the retrieved value. The figure displays an inconsistency in thoughts from the respondents, and their individual SUS score resulted in a grade varying from an A+ to an F. The deviant (orange) dot represents the value from the respondent that did not work at Exsitec.

![Figure 6.1: The individual scores from the SUS questionnaire. The deviant (orange) dot represents the respondent working outside Exsitec.](image)

Table 6.2 shows the individual SUS scores from each question. The values without parenthesis are the actual selected values; the values in parenthesis for odd numbers are the selected value minus one and values in parenthesis for even numbers are five minus the selected value. As seen in the table, the two questions with the lowest mean score were question three (1.9) and seven (1.6). Question three asked the respondent if they thought that the system was easy to use, and question seven if they thought that most people could learn to use the system very quickly. The question with the highest mean score was question one (3.4), which asked the respondents if they would like to use this system frequently. Several respondents commented that they did not have a choice, and that they needed to use the system regularly whether they liked it or not. Question four had the second highest mean score (3) and revolved around whether or not the respondents needed support from a technical person to use this system. Almost every respondent considered themselves to be the "technical person", and said that there were sometimes trouble with finding anyone else to ask. The
6.5. Interviews

The table also shows the standard deviation\(^1\) for the different questions. The highest standard deviation was calculated for question two.

Table 6.2: The individual SUS scores for each question (Q), respondent (R), Mean and Standard Deviation (Std. Dev.)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5(4)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>4(3)</td>
<td>3(2)</td>
<td>5(4)</td>
<td>3(2)</td>
<td>5(4)</td>
<td>4.4(3.4)</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>4(1)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>2(3)</td>
<td>2(3)</td>
<td>5(0)</td>
<td>3(2)</td>
<td>4(1)</td>
<td>3(2)</td>
<td>2.8(2.2)</td>
<td>1.4</td>
</tr>
<tr>
<td>3</td>
<td>2(1)</td>
<td>4(3)</td>
<td>3(2)</td>
<td>5(4)</td>
<td>3(2)</td>
<td>1(0)</td>
<td>2(1)</td>
<td>2(1)</td>
<td>4(3)</td>
<td>2.9(1.9)</td>
<td>1.3</td>
</tr>
<tr>
<td>4</td>
<td>2(3)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>2(3)</td>
<td>2(3)</td>
<td>3(2)</td>
<td>1(4)</td>
<td>2(3)</td>
<td>4(1)</td>
<td>2(3)</td>
<td>1.0</td>
</tr>
<tr>
<td>5</td>
<td>4(3)</td>
<td>4(3)</td>
<td>4(3)</td>
<td>3(2)</td>
<td>3(2)</td>
<td>2(1)</td>
<td>4(3)</td>
<td>2(1)</td>
<td>5(4)</td>
<td>3.4(2.4)</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>3(2)</td>
<td>1(4)</td>
<td>2(3)</td>
<td>2(3)</td>
<td>2(3)</td>
<td>3(2)</td>
<td>4(1)</td>
<td>4(1)</td>
<td>2(3)</td>
<td>2.6(2.4)</td>
<td>1.0</td>
</tr>
<tr>
<td>7</td>
<td>2(1)</td>
<td>3(2)</td>
<td>3(2)</td>
<td>5(4)</td>
<td>1(0)</td>
<td>2(1)</td>
<td>3(2)</td>
<td>1(0)</td>
<td>3(2)</td>
<td>2.6(1.6)</td>
<td>1.2</td>
</tr>
<tr>
<td>8</td>
<td>4(1)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>2(3)</td>
<td>3(2)</td>
<td>4(1)</td>
<td>3(2)</td>
<td>4(1)</td>
<td>2(3)</td>
<td>2.7(2.3)</td>
<td>1.2</td>
</tr>
<tr>
<td>9</td>
<td>3(2)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>3(2)</td>
<td>2(1)</td>
<td>4(3)</td>
<td>3(2)</td>
<td>4(3)</td>
<td>3.8(2.8)</td>
<td>1.1</td>
</tr>
<tr>
<td>10</td>
<td>4(1)</td>
<td>2(3)</td>
<td>2(3)</td>
<td>2(3)</td>
<td>4(1)</td>
<td>3(2)</td>
<td>2(3)</td>
<td>3(2)</td>
<td>3(2)</td>
<td>2.8(2.2)</td>
<td>0.8</td>
</tr>
<tr>
<td>SUS</td>
<td>47.5</td>
<td>87.5</td>
<td>82.5</td>
<td>82.5</td>
<td>52.5</td>
<td>30</td>
<td>62.5</td>
<td>35</td>
<td>67.5</td>
<td>60.8</td>
<td>21.1</td>
</tr>
</tbody>
</table>

6.4.2 Respondents view on the SUS

All respondents were also asked to share how well they thought that the SUS investigated the usability of the system on a scale from one to five. The mean rating, 3.4, and respondents’ individual rating can be seen in Figure 6.2. The median rating was 4.

Overall comments from the respondents regarding the SUS included that the questions were vague, too broad and non-specific. Several respondents appreciated that the SUS covered several different areas and aspects of the system in question. One respondent’s comment, which multiple other respondents touched upon, was:

*There is much room for interpretation in the questionnaire. You can perform several complex procedures, without it being too difficult in the portal itself.*

6.5 Interviews

This section includes the results obtained from the interviews during the investigation phase. All interviews were conducted in Swedish, and thus any presented quotes are translated to English by the authors of this thesis. The results will be presented in collaboration with the six scenarios created during the pre-study and have been visualised in graphs. Furthermore, each SUS score obtained from the questionnaires will be displayed together with the mean and the median. Scenario 4 Software updates, was not mentioned by any respondent during the interviews and has therefore been excluded from the rest of this thesis. The graphs show how many respondents have mentioned one aspect out of the total amount of nine respondents. The same respondent might be represented in multiple answers in one graph, as long as their answers are not contradictory.

---

\(^1\)Standard deviation is a concept that shows how much different measurement values are spread around a mean value.
6.5. Interviews

The respondents were also asked if they felt that their performance depended on how satisfied they were with their development environment. One respondent stated that their performance got somewhat affected, saying:

*You get annoyed if you work with environments that do not work the way you think it should work, you get slower.*

Although this statement can be seen as a very obvious opinion, it is important to note that it is also applied for this thesis and to keep it in mind during the rest of the investigation. Four other respondents also agreed with the statement, saying that their productivity was affected and that it is very frustrating to work in systems that do not cooperate. Only one respondent stated, blankly, that they are not affected by the state of their development environment.

### 6.5.1 Deployment Effort

The scenario Deployment Effort was described as: cloud services enable a shorter start-up process of new projects since it is easier to reuse some code or settings. Several existing services in Azure make it possible to reuse the deployment pipeline between different clients. By reusing existing resources, the development process is simplified.

The interview guide included a subquestion regarding reusability during the start-up of new projects and if it was eased by utilising a cloud service. As seen in Figure 6.3, four respondents mentioned Exsitec’s use of Azure DevOps and how it eases the deployment and development of applications. Also, two respondents mentioned that it is easier to streamline the deployment pipeline in a cloud environment than in an on-premise environment. Five respondents claimed that they did not experience any significant differences in reusability. Three respondents did not agree and claimed that the reusability process is more comfortable in the cloud.

One of the respondents said that:
6.5 Interviews

Figure 6.3: The aspects associated with the scenario Deployment Effort

It [Azure DevOps] works very well, and it is easy to manage code and back-log. Maybe it is not as adapted to Scrum and development processes as it could be. But regarding manage code, deployment and test it works really well.

The other four respondents also mentioned these concepts brought up by the respondents in some way. The respondents who had experience of working with Azure DevOps thought that it was a helpful tool and that even if it is possible to use it with an on-premise solution, it is much easier to utilise it in the cloud. Regarding that DevOps provides a way of reusing a deployment pipeline and raises the level of satisfaction, one respondent mentioned that: It is easier to streamline in the cloud, and thus become more satisfied.

6.5.2 Customisability

The scenario Customisability was described as: the ability to customise client solutions is a required functionality in order to be able to make the end product client-specific. The resources also need to have high elasticity, to simplify the process of scaling client projects up and out.

As seen in Figure 6.4 seven different aspects related to the customisability arose during the interviews. Although scalability, elasticity and reusability was explicitly mentioned in the interview guide, other relevant aspects were discussed as well.

Although one respondent claimed that there is no difference in elasticity and scalability when comparing cloud and on-premise, seven respondents said that the scalability in the cloud is superior. One respondent stated that:

It is super easy to scale out in a cloud environment, stupidly easy. Comparing the scalability of an on-premise solution to a cloud solution is like night and day. I am much more satisfied when working with cloud. Increasing the number of resources would take much longer in an on-premise solution.

Several respondents also discussed the elasticity of the cloud, and one stated that it is the very foundation to why cloud is so good. Four respondents discussed the modular aspect of
6.5. Interviews

Figure 6.4: The aspects associated with the scenario Customisability

cloud and claimed that it makes it easier to add new modules and functionalities to already existing applications.

Overall, the respondents were favourable to the customisability of cloud environments when compared to on-premise environments.

6.5.3 Error Solving

The scenario Error Solving was described as: Azure provides a standardised way of logging error messages, and it is easier to discover and adjust them. It is also possible to search through error history and statistics.

As can be seen in Figure 6.5 most of the respondents thought that there was no difference regarding how easy it was to detect or solve problems when comparing a cloud environment to an on-premise environment. One respondent said that: *Hardly any difference in how well it works to solve problems on-premise versus cloud, it more depends on your skills.* Two respondents thought that it was easier to solve problems in a cloud environment, and mainly based this on the fact that in an on-premise solution it is often hard to receive all the privileges to conduct the necessary actions.

Some respondents argued that debugging was more difficult in a cloud environment because the error could occur in more places. A comment from one respondent, which was mentioned by multiple others, was: *Cloud solutions are tricky, when we discuss architecture, there exist more aspects to take into consideration, and it becomes more difficult to troubleshoot.*

Furthermore, four respondents discussed the built-in and standardised monitoring services provided by Azure in the portal and how it can streamline the process of detecting errors. One respondent reasoned regarding the differences in monitoring in a cloud and an on-premise solution:

*Much easier in the cloud. On-premise it is monitored very differently per customer. Usually, it [an error] is written to a log, which someone must have set up monitoring on,*
Figure 6.5: The aspects associated with the scenario Error Solving

which, in its turn, differs from client to client. In Azure, you can set up alerts, and it is possible to see when an error occurred and recreate the setting it occurred in.

Four other respondents did not think that it was neither easier nor harder to detect errors in a cloud environment. Multiple of these respondents mentioned that it somewhat depends on the system, how it has been set up and if there has been given time to set up monitoring during the development process. They concluded that it does not depend on the environment, but on time dedicated to the project.

### 6.5.4 Client Approach

The scenario Client Approach was described as: when trying to sell a development project to a client, it is necessary to provide cost budgets in the offers. Using cloud services make it harder to compose the cost estimate. Every customisation and integration along with the elasticity and resource pooling will affect the cost. Moreover, the clients have different IT maturity, which means that they have different expectations on the functions and performance of cloud services.

Although no question in the interview directly asked the respondents on their thoughts on the cloud, from a cost perspective, this scenario was represented in the interview guide through various questions. Figure 6.6 shows the results related to the cost aspect.

Several respondents discussed the effect cost has on their clients, both positive and negative. Four respondents stated that the cost image in the cloud is quite complex and difficult to understand. One respondent said:

*It is very hard to know what the cost will be since it is hard to know which functions to use. And since you should only pay for what you use, but you do not know what to use, it can be hard to give an exact price to the client. It can be hard to determine which functionality that actually will be used, and how much it will cost in the end.*
Respondents also raised the fact that it is hard to estimate future costs, which creates uncertainty. Some respondents, however, appreciate the dynamic cost model that cloud offers and the fact that it can be scaled so it is possible to either decrease or increase the number of resources. Two of the respondents discussed the ability to schedule the use of resources. One respondent stated that the function of shutting down resources is developed to minimise the cost, and stated that: *It is probably best for the application to be up and running all the time, so to shut it down during evenings and nights is strictly from a cost perspective.*

Four respondents discussed that cloud creates a lock-in effect, which will bind the clients to the chosen providers. One respondent discussed the risks of this effect, stating that:

*It is tricky and cost ineffective to change cloud provider. If Microsoft raises their prices, should you then change provider? How much does it cost to switch and what do you earn from it?*

Based on the views of the respondents that mentioned the cost aspect during the interviews the cloud cost model was mostly seen as complicated, and difficult to understand.

Several respondents discussed their clients’ attitude towards the cloud, stating that it can differ a lot from client to client. However, most of them agreed that the clients are getting more and more positive towards the cloud. Their mindset usually depends on the size of the client company, their IT-maturity or the type of integration. If the client company does not have any servers of their own, they will most likely not choose an on-premise solution. Similarly, if the client company would like an integrate for two existing cloud-based solutions, there exists no reason to install the integration on-premise. The type of integration can also affect the choice of how it should be hosted. For some clients, it is unthinkable not to own the parts themselves, and it makes them uncomfortable to not be able to pull the plug if they want to. Several respondents also mentioned that they feel that more communication is required when working with cloud since more parties are involved. This can slow down the process and make it inefficient.
6.5.5 Ease of Use

The scenario Ease of Use was described as: It is important that the systems a developer utilise daily is easy to understand, learn and use in order to ensure high productivity. Sufficient documentation and guidelines are therefore required. A development environment with low usability will affect the end-users’ subjective satisfaction negatively.

Figure 6.7 shows the results regarding the ease of use of the system. The respondents had polarised opinions regarding the Azure portal’s GUI. Most respondents agreed that the features of the portal are complex, and two respondents claimed that the high amount of functionality made it challenging to choose the most suitable one. However, four respondents argued that the visual aspect of using the GUI simplified the portal and made it easier to use, despite its complexity. Two of the respondents praised PaaS and said that it simplifies and abstracts away many of the intricate parts of a project and that it allowed them to not focus as much on infrastructure or other underlying functionality. Four respondents discussed the fact that the Azure documentation is often limited, outdated or both, which makes it harder to understand the portal’s functionalities. Two respondents mentioned that it is easier to work concurrently on a project or to take over another colleague’s project.

The respondents were asked about their experience regarding learning the Azure portal. All respondents elaborated on the matter concerning the learning curve, and the results can be seen in Figure 6.8. It shows that a significant part of the respondents thought that the portal was easy to learn but hard to understand fully. Only one respondent thought that the Azure portal was hard to get started with, that respondent did however not expatiate on how difficult it was later on.
The respondents also elaborated on the question of receiving some education of how the portal works, and that could ease the process of becoming productive. One respondent said, regarding how to conduct specific assignments in the portal; *This is the way to do it, it is not obvious, but you just have to learn that this is the way.* Several respondents stated similar thought regarding understanding how to use certain functionality.

Six of the respondents mentioned that their performance was affected if the development environment has low usability. The respondents mainly based this on that even if you are to perform a simple task, there exist many unforeseen steps which makes it harder. Also, they mentioned that it becomes frustrating and is a source of irritation.

Furthermore, all respondents discussed the matter regarding how easy or hard they subjective it was to set up new environments in the Azure portal. One respondent did not have an opinion, but out of the remaining eight, seven thought that it was more comfortable in a cloud environment, as seen in Figure 6.9. The respondents who thought it was easier to set up a new environment said that it required less knowledge and less communication with the provider. One respondent said that: *As a developer who does not work very much with infrastructure, it gives much greater opportunities to take a project from start to finish.*

The one respondent who thought that it was harder, based it that it was due to getting stuck in the platform and not knowing which function to use.
This chapter presents the discussion regarding the obtained results and the used methods in this thesis. Moreover, the work is put in a broader context, and the reliability, validity and replicability of this thesis are discussed, as well as ethical considerations. The chapter ends with a section regarding future work.

### 7.1 Results

This section discusses the results obtained during the investigation phase of this thesis, i.e. the SUS questionnaire and the main interviews. The results are compared to the material presented in the theory chapter. Especially, the section discusses the results that either was as expected in regards to theory or diverged from it.

#### 7.1.1 System Usability Scale

The mean score for the SUS questionnaire was assessed to 60.8, which shows that the system should be considered to be scrutinised further [65]. As described in the theory, the experience of the respondent affects the SUS score [67]. Table 6.1, show the respondents experience and can be compared to the results presented in Figure 6.1. Respondent two, three and four can be considered to have the most extensive knowledge in the cloud compared to the remaining respondents. Respondent one and nine can be considered to have less than the previously mentioned three, but more than the remaining. Respondent two, three and four had high scores in the SUS evaluation, all above 80, while the results of respondent one and nine were somewhat equal to the answers of the less knowledgeable respondents. To conclude it is possible to see that the respondents with a pervasive knowledge had a much higher SUS score, thus confirming the theory presented. However, there was not as much difference when the respondents had just some more experience. For some questions, it was possible to see a pattern between the respondents’ answers and their experience, these questions will now be discussed.

In Table 6.2 the standard deviation has been calculated, and the second question yields the highest value, and was thus the question where the respondents were the most non-cohesive. The second question was: *I found the system unnecessarily complex.* The ones with the lowest
7.1. Results

total SUS score, were also the ones that answered Strongly agree to question two. Also, the respondents with the highest amount of experience answered Strongly disagree to the question. This could indicate that the system is, before it has been mastered, found to be complex and hard to work in. Thus implying that the system has a low learning curve.

Furthermore, questions three, seven and eight has yield similar standard deviation and the next highest value of it. These were (in ascending order): I thought the system was easy to use, I would imagine that most people would learn to use this system very quickly, I found the system very cumbersome to use. The same pattern for which of the respondents that have answered Strongly agree/disagree can be seen here as for question two. Similar conclusions as for question two could be stated for these questions.

Regarding the respondents’ thoughts on SUS, the biggest concern was that it was general. This might however not be entirely negative, in the sense that SUS aims to provide a general picture. It will also create an indication on how to continue the investigation of the subjective satisfaction and learnability, in order to be able to find the most crucial parts of the system to improve. Since the questions do not include any specific terms or similar, the experience of the respondent will not affect their possibility to answer all questions. But, as mentioned earlier, the respondents’ experience may still affect their scores. Nonetheless, the median and mean value was 3.4 and 4 respectively, meaning that the respondents overall considered SUS as a valid measurement of the subjective satisfaction and learnability of a system.

7.1.2 Interviews

Elasticity and scalability are considered important concepts regarding CC and is considered to be significantly superior in cloud solutions compared to on-premise solutions [24, 31]. Most respondents agreed with the presented theory and argued that cloud both makes it much easier to scale up and out as well as provides an elasticity that is not available when working with on-premise solutions, thus increasing their satisfaction.

The impact of the GUI was also discussed in the theory chapter, stating that it needs to be both well functioning and aesthetically pleasing in order to increase satisfaction and learnability [8, 51]. The respondents were divided regarding the suitability of the GUI in the Azure portal. Four respondents appreciated it and stated that even though the features are complex, the GUI is transparent and quite easy to navigate. Three respondents thought instead that the GUI was hard to find and navigate through, based on the sheer amount of features. A majority of the respondents that appreciated the GUI had more extensive knowledge and had worked more with it, which indicates that it takes some time to become thoroughly familiar with it. Furthermore, the respondents thought that some kind of education or walkthrough of the portal could be beneficial. However, nearly all respondents still thought that it was easier to set up a new cloud environment, compared to in an on-premise environment.

Several respondents mentioned that the Azure documentation often is limited or outdated, making it harder to understand the portal’s functionalities. As stated by Alonso-Ríos [51] documentation regarding definitions, descriptions and examples need to be both useful and sufficient. Thus, the lack of sufficient documentation can negatively affect the learnability of the system. Furthermore, the satisfaction can also be affected, since several respondents stated that they get annoyed and frustrated when working in a system that does not cooperate, or work the way they expect it to. The previously mentioned problems with the GUI might also be triggered by the lack of sufficient documentation since it can be hard to find information about functionality that is hard to understand or work with. It will furthermore increase the need of creating an internal walkthrough or documentation of the portal to accommodate for the insufficient official documentation. Another important aspect regarding the need for more solid documentation in the cloud is that the knowledge is not as comprehensive as for on-premise solutions. The interviews indicated that fewer employees have
extensive knowledge within cloud compared to on-premise. This means that there exist benefits in formalising that knowledge and increase the knowledge sharing.

Continuing with the learnability, Nielsen [12] has described two different learning curves, the one focusing on expert users and the one focusing on novice users. By dividing it into two sections, two interesting questions can be investigated: How easy is it to get started? Moreover, how easy is it to master the system? By having a low learning curve, a system is generally easy to get started with on a basic level, but that does not mean that it is necessarily easy to master, and the opposite for a high learning curve. However, it is also worth noting that not all learning curves will look exactly as pictured in Figure 3.2. Eight of the respondents experienced more or less the same learning curve, by agreeing that it was easy to get started and perform basic functionality in the portal, but that it was hard to master. One respondent thought that it was difficult to both get started and master the portal, and thus experienced a different learning curve. Overall, a clear majority of the respondent concluded that the portal has the learning curve more focusing on expert users.

Four of the respondents mentioned Azure DevOps and stated that it eases the development process. This is consistent with Microsoft’s views on the service, stating that it significantly eases the building, deployment and management of resources [20]. The respondents that had used Azure DevOps thought that it increased their satisfaction.

Cloud enables several features, including service monitoring [38]. By monitoring and controlling the implementation, errors can be found and dealt with earlier. The respondents during the pre-study interviews mentioned the monitoring services provided by Azure. The monitoring services can help to analyse applications performance and availability, as well as identify issues to fix them, according to Microsoft [19]. Respondent A and D in the pre-study interviews, seen in Section 4.2.1, discussed that using cloud-based solutions can simplify the error handling and versioning of applications. By making the error messages more standardised, it is easier to discover and adjust errors in time. In the primary interviews, however, respondents were not entirely convinced that the cloud makes detecting and solving errors simpler. Seven respondents saw no difference regarding how difficult it was to solve errors, compared to two respondents stating that it was easier to solve errors in a cloud environment. Four respondents thought that it was easier to monitor and detect errors, but the same amount of respondents saw no difference, and two respondents thought that it was more difficult to debug in a cloud environment. It was not possible to see a pattern that the four respondents that found it easier all had more extensive knowledge. It rather depended on if the respondents hade used the built-in monitoring services that Azure provides or not. The impact these features had on the satisfaction and learnability was thus not entirely clear, since there was no agreement whether or not cloud was superior in regards to error monitoring and solving.

Another feature enabled by the cloud is cost flexibility [23, 24, 35]. Cost flexibility is described as one of the variables that make the cloud so appealing, since it can generate lower costs compared to on-premise solutions. However, several respondents thought that the cost image in the cloud is intricate and that it is hard to estimate future costs for clients, thus negatively affecting the satisfaction. The respondents pointed out that it is hard to understand which functions that is required by the application and the client. Therefore, it is hard to understand if the calculated cost is actually correct and valid. Furthermore, clients are generally getting more and more positive towards the cloud, but it can vary depending on the company’s IT-maturity, their size and the type of integration. One benefit for the clients, when it comes to their IT-maturity, is that they will not receive as many error messages when using cloud services since all of them will be handled by Exsitec and the cloud provider.

The lock-in effect has been raised as a problem by several researchers and is stated to make it hard for a client to change provider without facing significant costs [46, 47, 48, 49]. Clients with a lack of IT-maturity might have trouble understanding and mitigating the concept since
7.2 Method

This section critically discusses the methods used during this thesis to highlight the potential consequences for the results. The replicability, reliability, and validity of this thesis will also be discussed. The section ends with a subsection regarding source criticism.

7.2.1 Pre-Study

The pre-study was mostly concerned with interviewing respondents in order to create an overview of how cloud is utilised at Exsitec and perform the requirements elicitation. Few requirements needed to be fulfilled in order to be chosen as a respondent in the pre-study interviews, and it could thus have been conducted with a significantly higher number of respondents. However, although respondents were allowed to participate in both the pre-study interviews as well as the main interviews, the overlap of respondents was desired to be as low as possible. The aim to have two groups of respondents without significant overlapping came from the desire to gather a wide selection of individual opinions. The timeframe also aggravated the process and forced it to be completed quite early on.

Another method could have been used to validate the elicited requirements, other than interviews and scenarios. Several popular methods exist and are described in this thesis, and researchers have claimed that they should be used in conjunction in order for the elicitation to be successful [77]. Also, interviewing is considered to be the most common elicitation method [79]. The reason why two methods, and not more, were chosen was since the authors of this thesis did not want to depend too much on the ability to schedule more meetings with respondents. Most of the existing methods rely on other respondents and the risk of not being able to find and schedule enough meetings was deemed too high.

One problem that needs to be mentioned is the fact that the study was conducted with few respondents. The desired number of respondents for the investigation phase was somewhere between 10 to 15, circling the recommended amount of 12 respondents by Guest et al. [83], but the purposive sampling only resulted in nine respondents. One solution to this problem could have been to divide the respondents into two groups with roughly ten respondents each, one group with knowledge of cloud and one with knowledge in on-premise solutions. The results from the two groups could then have been compared, instead of letting the respondents themselves compare the two different environments. However, there exist other problems with this solution since the working procedures can differ quite a lot. By using two groups and letting the researchers compare the respondents’ results, the researchers’ bias will probably further affect the comparison.

It can also be difficult for a respondent to answer a particular question if there is no specific common baseline to compare against. As stated previously, Nielsen [12] recommends asking comparative questions when investigating the users’ subjective satisfaction towards several systems. Furthermore, a qualitative approach to usability evaluation is usually conducted with a smaller set of respondents [57].
7.2.2 Interview Guide

The interview guide was created by using the pre-study interviews and the scenarios, the SUS questionnaire and the presented theory. All these aspects provided a broader foundation for the investigation. In the theory chapter, more possible benefits were mentioned than those discussed during the pre-study or in the questionnaire. Therefore, questions related to this was added, but it could have been further extended. This could have further connected the questions with the promised benefits of cloud from the theory but would have made them less connected to the real-world setting. The questions were formulated according to the recommendations from McNamara [71] which aided the work with compiling the questions and create a well thought through guide. Furthermore, McNamara’s guidelines regarding how to start the interview session created a good flow for the beginning of the session and that the respondent felt confident with the format.

7.2.3 Test Interview

The test interview was conducted to verify that the questions were understandable and that they provided valuable answers [72]. To verify this further, more test interviews could have been conducted with other respondents. This would have increased that certainty that the questions were of sufficient quality. Other respondents could have been someone at the company with some knowledge in the cloud, but not enough to be a part of the investigation phase. Some knowledge of the cloud was desired during the test interview to make sure that the interview guide was related to the aim and research question of this thesis.

7.2.4 Investigation

The investigation was conducted with two separate methods; questionnaires and interviews. There exist several other usability evaluation methods, each with its own set of advantages and disadvantages [55, 58]. Some of the other evaluation methods could have been used, but that would have created another outcome from the investigation since different usability aspects could have been investigated. However, it is likely that similar problems would have been discovered, but from other angles, since the various usability attributes have diverging focus points. It was deemed that the respondents’ thoughts and view of the system were the primary investigation point, to try and understand if Exsitec could change their working procedures to increase their satisfaction and learnability. Furthermore, the investigation is only related to the respondents’ opinions regarding the user interface, and not the user interface itself.

During the interviews, the respondents were asked about their opinion of SUS. This resulted in a mean score of 3.4, and most of the critique was regarding the generality of the questions. One way to raise the score could have been to adapt the SUS to the investigated situation. Examples of changes could have been to switch out the used term “this system” towards the Azure Platform or the Integration platform or to make some questions a bit more specific. However, the questionnaire was decided to be kept at its original form to keep its reliability and trustfulness.

One issue regarding the SUS was that it was only answered in consideration to Microsoft Azure, and not their existing on-premise solution. The questionnaire could have been distributed to two different groups of respondents, were one group answers the questionnaire with Azure in mind and one group with the on-premise solution in mind. Another option could have been to let the respondents fill out two versions of the same questionnaire, one answering with the Azure portal in mind, and one with the on-premise solution. This solution would have removed the necessity of having two different groups of respondents but would have increased the required time for the respondents.
The interviews were recorded and listened to twice, in order to make sure that the material was collected properly. Since the interviews were decided to be semi-structured, the respondents did not discuss precisely the same things. The semi-structured approach provides an ample space to start from but does not prohibit interviews from taking slightly different directions based on the respondent’s discussion [73]. If a respondent started to discuss matters that were not relevant to the questions, the question was clarified, and the respondent was steered back in the right direction.

However, one disadvantage with using qualitative interviews with open-ended questions is that it is harder to collect and analyse the information [70]. To ease this process it was decided that the results should be grouped based on the closest related scenario. The most commonly stated answers were combined and given a label. The combining of answers required an interpretive read [68] of the data since the respondents did not give the same answers word for word, and it was thus needed to read through and beyond the data. Therefore, there exists a possibility that some of the answers could have been wrongly interpreted and consequently gathered within the wrong group of answers. To try and avoid errors the grouping of the answers, and their label, was done twice and heavily discussed during the process.

If a quantitative evaluation method would have been used to investigate the learnability, another angle of the attribute could have been investigated, possibly improving the results. Some possible areas that could have been investigated are which functions that take longer time to learn, or are harder to remember between sessions, how frequently the users use the manuals, and how often they solved the users’ problem. User tests, for example, would have provided quantifiable measurements, but also introduced ethical aspects and required immense preparation from the researchers.

As previously mentioned, the number of respondents were a bit lower than what was desired. However, it was also found that after a couple of interviews nearly no new information or viewpoint was obtained. This can be referred to as saturation and shows when sufficient interviews have been conducted [76]. Thus, more interviews might not have led to other results but instead confirmed already stated opinions.

7.2.5 Replicability, Reliability, and Validity

To achieve high replicability, meaning that it is possible for someone else to follow the method and compare if the gathered results are the similar [76], this thesis has aimed to present the method with a high level of details. Both the interview guide and the questionnaire used during the investigation have been attached in the appendix. Since the authors of this thesis wrote the interview guide, their bias will affect the results. To reduce the risk of bias and misinterpretations, as suggested by Karlsson et al. [74], both authors of this thesis participated in all interviews. It was apparent during the interviews that the respondents did not talk with each other regarding the interviews and the subjects. When the respondents arrived to the interview they were not familiar with the interview subject or approach.

The number of respondents in this thesis will affect the generalisability due to low external validity. However, it is likely that the results would be similar if the investigation were conducted again on a different day, since the investigation was performed over a timespan of two weeks. A similar set of respondents would probably also culminate in similar results, due to the fact that saturation was reached. This means that the results can be considered reliable [75, 76]. Both the reliability and validity have been affected by the number of respondents, and would have increased if the sample set was more extensive. However, even if the results might not be generalisable to such a vast extent, it can be used as an indication.
7.3. The work in a wider context

7.2.6 Source Criticism

The sources used in this thesis is a variety of peer-reviewed journal articles, books and websites. As far as possible, peer-reviewed journal articles have been chosen over books and websites. The information provided regarding Microsoft Azure has sometimes been referenced to Microsoft themselves and is therefore biased. When these sources have been used it has been stated in the text that it is from Microsoft to reduce the risk of misunderstandings.

The aim has been to use as recent information as possible since the research within the field of software engineering is always developing. However, older sources and publications have also been used. An example of this is the book written by Mason [68], published in 2002, which has a later edition published in 2017 but it was not accessible for the authors of this thesis.

A smaller amount of information has been collected from secondary sources, which might not have been collected for the same purpose as this thesis. Therefore, this thesis has applied triangulation to reduce the risk of misinterpreting information. Triangulation provides a more comprehensive picture and is an indispensable part of a qualitative study to help improve the validity and reliability [76].

7.3 The work in a wider context

The ethical and societal aspects should always be considered when conducting a scientific study [76]. When conducting applied research, i.e. trying to find a solution for a problem by analysing and critically evaluating existing data, in a study that will be published, it is even more critical. To minimise the risk of revealing any information that might be confidential or similar, for the company, the supervisor at the company has reviewed the thesis. When conducting the interviews, the respondents were informed of the aim and purpose of the thesis. The respondents were also informed that their answers would be anonymised. The full job title of the respondent has been simplified, and the gender and age have been left out to preserve their identity. Since the experience of the respondents affects the results, it was included. However, the information has been simplified and reduced to minimise the risk of revealing the identity of the respondents.

It is also worth noting that the respondents’ attitude towards the cloud itself can affect their opinions regarding a cloud provider, for example, Microsoft Azure. However, it is extremely difficult to distinguish their bias of cloud compared to their opinions of the portal, unless the change in attitude is extreme. In the investigation conducted for this thesis, no extreme bias was discovered and it was clear when the respondents discussed cloud in general or the portal in particular. It is, therefore, reasonable to believe that the respondents’ opinions are mostly related to the portal itself, and not the cloud.

7.4 Future Work

Since this thesis was conducted in collaboration with Exsitec all data and resources are specific to that company. Meaning that since Microsoft Azure is utilised, this thesis was mainly concerned with the satisfaction and learnability of the Azure portal. A wider study investigating a more generalised view on the usability of cloud services would provide a complete picture. Another interesting continuation of the study would be to compare different companies opinions on cloud providers and their usability.

Furthermore, the usability of cloud in terms of the remaining usability attributes should be considered as an important topic, in order to understand clouds usability to the fullest. It would also be interesting to understand how usability is a part of the development of the portal, from the provider’s point of view. Lastly, it would be engaging with a study about
organisations strategy to avoid the lock-in effect. Such study could also include which methods that could prohibit that effect. All this would be helpful for an organisation adopting to cloud. Notably, in terms of how to decide the provider and how to develop an application which is platform independent.
Conclusion

This chapter presents the conclusions made in this master’s thesis and answer to the research question provided in the Introduction chapter.

This thesis aimed to investigate how the learnability and subjective satisfaction may be altered when utilising the cloud service Microsoft Azure instead of an on-premise solution. This thesis has tried to understand the effects of adopting cloud in a small and medium-sized enterprise and, if possible, provide a best practice for potentially needed updated working procedures. To fulfil this aim, this thesis intended to answer the following research question: **How are the learnability and subjective satisfaction altered by utilising the cloud computing service Microsoft Azure instead of an on-premise solution, from an end-user’s perspective?**

To answer the research question, in terms of elasticity, customisability, scalability, starting up new environments and reusability, cloud performs better. But, the wider knowledge regarding on-premise development is not the same for cloud, and therefore it decreases the learnability and satisfaction of cloud. Additionally, the documentation for on-premise development is often more suitable and up-to-date, enforcing the previous statement of a decreased learnability and satisfaction. On the other hand, results regarding cost flexibility, the GUI, service monitoring and resource pooling divided the respondents. Therefore, it is hard to draw any definite conclusions regarding these aspects, in terms of how they have been altered by using CC instead of an on-premise solution.

The two significant factors that can be improved to increase the learnability and subjective satisfaction were derived. These were the lack of documentation or insufficient documentation, and that the Azure portal is hard to master. The documentation provided by Microsoft is often seen as outdated or not extensive enough. Multiple respondents thought that the vast amount of features enabled in the portal were confusing, but that internal documentation regarding which features that are most commonly used at the company, and how to use them, would aid that. This could also subsidise the second major factor, that it is hard to master the portal. Many respondents commented that the competence of on-premise solutions was much more comprehensive, and thus the knowledge regarding the portal should be shared to a wider extent. Moreover, there was no coherent picture of how the portal, or cloud in general, were used by different teams at the company, or to what extent it was used.
A majority of the respondents stated that their performance was negatively affected if they were not satisfied with their development environment. The results of this thesis showed that multiple of the respondents were not satisfied with the portal, and thus it is reasonable to believe that their performance is affected negatively. The documentation and knowledge sharing has been shown to have a significant effect on the users, and it is reasonable to believe that it is true regardless of the used cloud provider and the organisation. The SUS score showed that the mean value was below an acceptable level and how the portal is used should, therefore, be changed in order to raise this level. The conclusion from this is that there exists a need for updated working procedures and increased knowledge sharing, in order to raise the learnability and subjective satisfaction of working with the cloud.
Bibliography


[38] Li, T. and He, T. and Zhang, Y. “Service-centric Business Model in Cloud Environment”. In: 2015 IEEE International Conference on Services Computing. 2015, pp. 530–537. DOI: 10.1109/ICSCC.2014.6982405.


Bibliography


Appendices
A The System Usability Scale
System Usability Scale

The System Usability Scale includes 10 questions and is used to measure the overall usability of a system.

*Obligatorisk

1. I think that I would like to use this system frequently *
   Markera endast en oval.

   1 2 3 4 5
   Strongly disagree Strongly agree

2. I found the system unnecessarily complex *
   Markera endast en oval.

   1 2 3 4 5
   Strongly disagree Strongly agree

3. I thought the system was easy to use *
   Markera endast en oval.

   1 2 3 4 5
   Strongly disagree Strongly agree

4. I think that I would need the support of a technical person to be able to use this system *
   Markera endast en oval.

   1 2 3 4 5
   Strongly disagree Strongly agree

5. I found the various functions in this system were well integrated *
   Markera endast en oval.

   1 2 3 4 5
   Strongly disagree Strongly agree

6. I thought there was too much inconsistency in this system *
   Markera endast en oval.

   1 2 3 4 5
   Strongly disagree Strongly agree

https://docs.google.com/forms/d/1iN52FpARlOcYg1s-X-DG9ljL5fSw-cOz2BESTGC8F8Bg/edit
7. I would imagine that most people would learn to use this system very quickly *

*Markera endast en oval.*

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. I found the system very cumbersome (besvärligt) to use *

*Markera endast en oval.*

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9. I felt very confident using the system *

*Markera endast en oval.*

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10. I needed to learn a lot of things before I could get going with this system *

*Markera endast en oval.*

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly disagree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
All questions should be answered from the respondents perspective. The answers will be anonymous, and the interviews will be recorded.

**Questionnaire:** All questions in the questionnaire need to be answered. If the respondents during completion do not feel that he or she can answer one question, a 3 should be marked, since it is a neutral answer. It can also be noted by the interviewer.

**Interview:** The interview should be completed after the questionnaire has been completed. The interview should start with the questions regarding highest level of completed studies and so forth.

The system equals the internal integration platform system set-up at Exsitec.

**Role:**
- **Years of experience in the field:**
- **Experience working with cloud services and solutions:**
- **Experience working with on-premise solutions:**
- **How do you think the questionnaire investigated the usability (in terms of subjective satisfaction and learnability) of the system? On a scale from 1-5.**

Based on the question: *I felt very confident using the system*, do you find that your performance at work depends on how satisfied or comfortable you feel with your development environment?

**Theme: Satisfaction**

This theme relates to how pleasant the user finds the system, as well as how enjoyable it is for the user to work with it.

Based on the question: *I think I would need the support of a technical person to be able to use this system.* If you have a problem while working, how do you go about solving it?

- How does your approach differ when you work with an application hosted in a cloud environment compared to if it is hosted on-prem?
- Do you think that the technology you use affects how quickly you can solve a problem?
• How do you feel that problems affect your productivity?

Based on the question: *I thought the system was easy to use*, which attributes did you think of when answering that question?

Based on the question: *I thought there was too much inconsistency in this system*, how does it affect your attitude towards using cloud?
  • Does it affect how satisfied you feel with your work?

If we discuss the start-up of new projects, what differences do you experience when starting a new project that will be hosted on-premise compared to whether it should be hosted in the cloud? Please be as concrete as possible.
  • How do you see the question of whether you speak in terms such as
    – Reusing
    – Account set-up
    – Start-up of new environments

Based on the question: *I think that I would like to use this system frequently*, what is behind your decision?
  • Which parts of the system/application is it that you appreciate more/less?

Based on the question: *I found the various functions in this system were well integrated*, what is behind your decision?
  • Which functions need to be well integrated in order for it to affect your satisfaction?

If you take one of your typical projects/ client dilemmas, what challenges do you see when it comes to using a cloud environment?

How do you perceive your clients’ attitude to the choice between a cloud or on-premise solution?
  • Is that question being discussed?

Based on the question: *I found the system very cumbersome to use*, do you feel that your performance is significantly affected if the development environment has low usability?

When it comes to the concepts of resource pooling and the elasticity of resources, do you notice any difference if you compare cloud solutions with on-premise solutions?
  • If so, what are the differences?
  • Does these abilities and the differences between the solutions affect your satisfaction with the system? How?

If something happens in the application, that there is an error, how is it monitored on-premise compared to cloud? Do you feel that there is any difference in how easy it is to detect errors? If so, what differences do you experience and satisfaction?
Theme: Learnability

This theme relates to how easy the system is to understand for a new user, and how long it takes before the user becomes productive in their work.

Starting from the question: *I needed to learn a lot of things before I could get going with this system*, which parts did you experience as very difficult to learn? Is it possible to estimate the regarding how long it took to get started?

Do you experience any difference based on how fast it went for you to get started and become productive in development towards cloud environment compared to on-premise?

Do you experience any difference to working with Azure as a supplier, compared to working with an on-premise supplier? What, in your experiencing, differs?

Based on the question: *I found the system unnecessarily complex*, what attributes do you think of?

- How are you affected by the attributes when using the cloud?

Based on the question: *I would imagine that most people would learn to use this system very quickly*, can you develop what thoughts and reasoning lies behind your decision? Is there any way you can change the system so that it becomes easier to learn?

Based on the question: *I found the system very cumbersome to use*, how have your views changed since you initially used the system? What has affected your views?

Guided by the question: *I think that I would need the support of a technical person to be able to use this system*, did you feel that you needed to ask for help often during the learning period/first time you used the system?

- Have you been helping other colleagues during their learning period? How did that work out?

Would you like to add something? Or would you like to change something in a previous question?