JÖNKÖPING UNIVERSITY
School of Engineering

Technology stack selection

Guidelines for organisations with multiple development teams

Main Subject area: *Computer Engineering*
Author: *Hugo Martinsson, Victor Svanqvist*
Supervisor: *Rob Day*
JÖNKÖPING 2022 June

This final thesis has been carried out at the School of Engineering at Jönköping University within Computer Engineering. The authors are responsible for the presented opinions, conclusions, and results.

Examiner:	Johannes Schmidt
Supervisor:	Rob Day
Scope:	15 hp
Date:	2022-06-02

Abstract

When starting a new software project, selecting what technology stack to use is one of the most important decisions to make. Selecting a technology stack is a large part of the software architecture design, and the choice of the technology stack is crucial to get right since it can make or break a project and is usually hard and expensive to change in the future.

This thesis was conducted to develop guidelines for organisations to use during the technology stack selection process by identifying the essential steps of the technology stack selection process at private sector organisations with multiple development teams that perform in-house development. As well as identifying scenarios where it is reasonable to choose similar technology stacks for different development teams, and scenarios where it is reasonable to select different technology stacks for different development teams. The guidelines aim to help organisations evaluate different solutions and help organisations decide whether it is worth it to choose different technology stacks for different development teams.

The guidelines provide the essential steps of technology stack selection, control questions that can be used to evaluate whether a given technology stack would work for an organisation, as well as scenarios where it is reasonable to select similar or different technology stacks for multiple development teams.

The guidelines are developed using Design Science Research and semi-structured interviews with software developers, software architects and managers at Husqvarna and other organisations that agreed to participate. The interviews were analysed using thematic content analysis to develop a draft of the guidelines, which was then attached to a survey that was sent out to gather feedback which helped further improve the guidelines and validate that they apply to a broad audience.

This thesis does not cover technological aspects of technology stack selection, such as performance or efficiency, nor does it cover programs or tools used to aid the development, like integrated development environments (IDE:s), code-sharing software or team communication tools.

Keywords:

Technology stack, software architecture, multiple development teams

Acknowledgement

The authors of this report would like to start by recognising and showing our appreciation towards the people that have supported us during our thesis work. We could not have undertaken this journey without the help of Andreas Rowell at Husqvarna, and we would also like to express our gratitude to our supervisor Rob Day for his help throughout the thesis. Thanks should also go to all interview participants and survey respondents who made this thesis possible.

Table of content

Figures

Tables

1 Introduction

The information technologies used in software development are commonly called a technology stack and choosing what information technologies to use in a technology stack is a large part of software architecture design. Defining a technology stack is no easy feat since it can consist of many different technologies and might be developed over time by adding more technologies. A technology stack often refers to a set of programming languages, frameworks, libraries, databases, and many other technologies (Nikulchev et al., 2021), as illustrated in Figure 1 below.

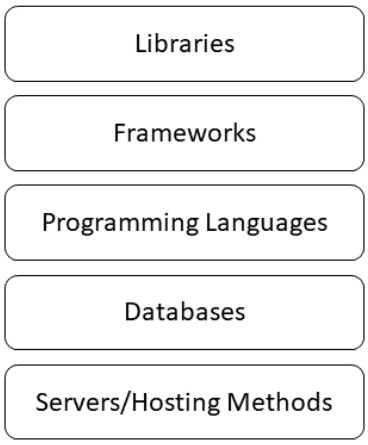


Figure 1: Example of technology stack parts

An essential attribute of a technology stack is the replaceability of its components. Each information technology used in a technology stack can most often be replaced with an alternative, which can impact the performance of the software, the hardship of development, and the compatibility with other technologies (Nikulchev et al., 2021).

When choosing a technology stack, various aspects can be taken into consideration. Firstly, there are technological aspects like what technologies work well together and what technologies are most performant. Secondly, there are procedural aspects to have in mind, such as who should be responsible for choosing and how the size and number of development teams should affect different technology choices. The guidelines developed in this thesis focuses on the procedural aspects of technology stack selection.

1.1 Problem statement

When starting a software development project, there are a lot of technology choices that must be made. The technologies that are chosen for use in a project are often combined and called the technology stack. Since the technology stack is the building block for an entire project, much thought and consideration need to go into choosing the right technologies. The technologies chosen at the beginning of the project can make or break the software developed. According to Falatiuk et al. (2019):

> …the main goal of software architecture is to define which components the system should consist of, how those components are going to communicate with each other and how they must be deployed in order to fulfill these requirements. That is the hardest and the most important part of the application construction because all the mistakes made on this stage are the most expensive to fix in future, so you always need to put the right amount of effort on this task.

Defining which components the system should consist of also includes choosing the technology stack. Falatiuk et al. (2019) reinforce the idea that choosing a solid technology stack can make or break a project and is very hard and expensive to fix in the future. This is because changing parts of a technology stack most often requires rewriting large parts of the code, which can be time consuming and therefore expensive. Choosing the right technology stack is no easy feat since it involves figuring out what is suitable for a specific project based on many factors and aspects. This does not only apply to beginners but can be a challenge to even the most experienced users and teams. Currently, no guideline covers most software projects. There are general guidelines for choosing a technology stack online, but these guidelines are rarely supported by any research. They are most commonly written in blogs and website articles based on the writer or writers' opinions and not actual data from research.

Whilst there are previous research papers about technology stack selection, earlier research only covers the selection of technology stacks in specific cases or for a specific part of the software field. For example, Aggarwal and Verma (2018) compare two specific technology stack solutions, Falatiuk et al. (2019) investigate technology stacks for e-Archive systems, and Rao (2009) focuses on the selection of geographic information system (GIS) frameworks for an informed enterprise. There is also previous research which covers the performance of technology stacks, such as (Nikulchev et al., 2021). They developed a mathematical model for evaluating the effectiveness of different technology stack solutions. Whilst these results most likely are valuable in their specific topic, they are not of much use if the specific topic is not the one of interest. Not to mention that many specific topics are not covered by such research.

The most complete technology stack selection guidelines that were found were by Zhu and Shao (2015) who created guidelines for technology stack decisions in web development. Whilst their guidelines (WeTS) sound reasonable and useful, they provide a framework for comparing and evaluating different technology stack solutions to find a technology stack that fits the purpose of the product being developed. Zhu and Shao do not cover the procedural aspects of technology stack selection and neither do they cover the selection of technology stacks for multiple development teams or organisations working on multiple products. The guidelines developed in this thesis does therefore not build on WeTS, but there could be situations where it would be of value to combine the guidelines created in this thesis with WeTS. The WeTS guidelines

would then be used to establish which technology stack that is best for the product being produced, while the guidelines produced in this thesis would be used to guide the entire technology stack selection process.

To conclude, there are no previous research papers that cover the procedural aspects or the selection of technology stacks in organisations with multiple development teams.

This shows that there is a lack of research on technology stacks in general and that most of the guidelines that exist for the selection of technology stacks today are not based on research. There is a large need for guidance in the process of choosing technology stacks, especially when choosing for multiple development teams.

1.2 Purpose and research questions

Drawing on the problem statement, it is evident that the need for guidance in the technology stack selection process is significant. It is also evident that previous research mainly focuses on selecting technology stacks in specific areas. There is no previous research covering the procedural aspects of technology stack selection for organisations that work with various products or with multiple development teams. Consequently, the purpose of this study is:

> *To develop guidelines for organisations to use during the technology stack selection process. The developed guidelines aim to help organisations in their evaluation of different solutions and help organisations decide whether it is worth it to choose different technology stacks for different development teams.*

The guidelines can lead to organisations saving time, money and resources as well as providing them with a streamlined approach to selecting effective technology stacks.

For the guidelines to be useful to a broad audience, it was important to study which factors are the most important when selecting technology stacks and to establish the essential steps of the selection process. Therefore, the study's first research question is:

> [1] What are the essential steps in the technology stack selection process?

Another goal of the guidelines is to make it easier for organisations to decide whether it is best to use similar technology stacks for different development teams or let the teams decide for themselves and therefore have different technology stacks. For the guidelines to do that, factors that influence this decision-making have been studied. Hence, the second and third research questions are:

> [2] When is it reasonable to choose similar technology stacks between different development teams?
>
> [3] When is it reasonable to choose different technology stacks between different development teams?

1.3 Scope and delimitations

The following scope and delimitations have been defined to make the research feasible as a bachelor's thesis:

1.3.1 Scope

This research covers the selection of technology stacks in private sector organisations that perform in-house software development (developing software using the competence available in the organisation). The study focuses on the procedural aspects of selecting technology stacks in organisations with multiple development teams.

In the scope of this research, the following elements are considered as parts of a technology stack:

- Programming languages
- Frameworks
- Libraries
- Databases and Database Management Systems
- Servers and Hosting

1.3.2 Delimitations

The following delimitations have been established:

- Technical aspects of technology stack selection, such as performance or efficiency, are not studied
- Consultancy organisations are not studied
- The study does not cover programs, tools, and processes used to develop the system, like code editors, team collaboration software or code-sharing services.

1.4 Disposition

This section describes the structure and disposition of the remainder of the report. Section 2 describes the methodology and methods that have been used and how the research has been carried out. Section 3 provides a theoretical background about technology stacks and software architecture, as well as established key concepts of previous research. Section 4 firstly presents the collected data to later present the produced guidelines and the justifications for them. The results presented in section 4 are then discussed in section 5. Section 6 concludes the thesis and provides the implications and possible further research that could be carried out.

2 Method and implementation

This chapter describes the scientific methods that were applied during this thesis. The research method used has been Design Science Research (hereafter DSR), as described by Hevner et al. (2004). In short, DSR is a research method used to identify and solve a problem by developing an artefact. Iiavari and Venable (2009) describe DSR as constructing a new reality rather than explaining an existing one.

Data has been collected using semi-structured interviews and a mixed data survey. The collected data has been analysed using thematic content analysis.

2.1 Design science research

This section describes DSR, why it was chosen as the research method and how it was used during this thesis. DSR was chosen as the research method due to its problem-solving capabilities and ways to improve or change existing solutions (Baskerville et al., 2015). DSR allowed a deep understanding of the topic to be gained, a solution to be developed and then validation and improvement of the solution. Figure 2 below provides a simple overview of the DSR process.

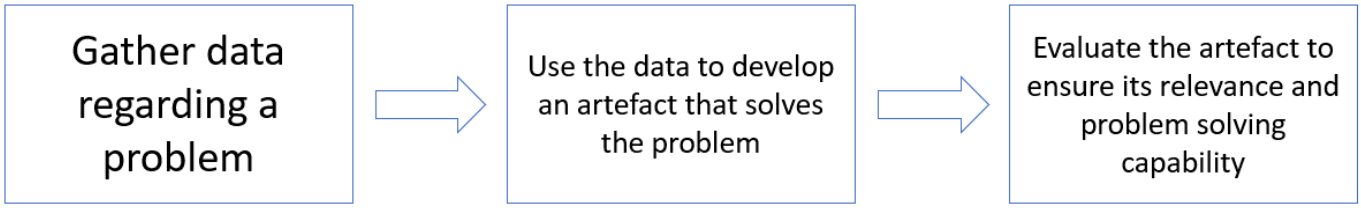


Figure 2: Design Science Research process

Hevner et al. (2004) provide seven guidelines for performing DSR in Information System Research. Sections 2.1.1 to 2.1.7 below describe each of the unique design science research guidelines (DSRG) and how they have been used in the thesis.

2.1.1 Design as an Artifact

The first DSRG defines an artefact that is purposeful and that addresses an organisational problem. It states that artefacts can be both hardware and software, but also constructs, methods or models applied in the development of information systems. This research creates guidelines to be used during technology stack selection, which can be seen as either a construct, model, or methods.

2.1.2 Problem Relevance

The second DSRG defines a problem as "the difference between a goal state and the current state of a system" (Hevner et al., 2004, p. 85). This DSRG states that solving a business problem often can lead to increased revenue or decreased costs and that for a problem to be relevant, it must address problems faced by organisations. As discussed

in chapter 1, the selection of technology stacks is a problem that can have a large financial impact on organisations and therefore is relevant to study.

2.1.3 Design Evaluation

The third DSRG portrays the importance of evaluating the newly designed artefact. It states that the requirements upon which the evaluation of the artefact are based should be established by the business environment and that artefacts can be evaluated with different metrics depending on the type of artefact. The guidelines produced in this thesis were validated in terms of usefulness and relevance by surveying prospective users of the guidelines.

2.1.4 Research Contributions

The fourth DSRG mentions three diverse types of research contributions that can be made using DSR. The research contribution of this thesis falls under the *Foundations* category, meaning that it provides methods for the selection of technology stacks. *Research Contributions* also describe how the contributions must be *implementable* to be of value. During the development of the guidelines, a large focus has been put on making the guidelines easy to understand and relevant in different scenarios for the guidelines to be implementable for many organisations.

2.1.5 Research Rigor

The fifth DSRG regards how the research is conducted. It states that rigorous methods must be applied both during the construction and evaluation of the artefact. This thesis applies rigorous methods as described in sections 2.2 and 2.3 below.

2.1.6 Design as a Search Process

DSRG number six states that DSR is an iterative process and that design can be seen as a search process for discovering an effective solution to a problem. It also introduces the *Generate/Test Cycle* which means generating alternatives of the artefact and iteratively testing different alternatives against requirements. Due to time and resource constraints, the guidelines produced in this thesis were tested and evaluated once. If possible the guidelines could have been tested and evaluated multiple times and in diverse ways as further discussed in section 5.3.

2.1.7 Communication of Research

The last DSRG, *Communication of Research*, handles the process of presenting research to different audiences. The way of communication must enable practitioners to wreak the benefits offered by the artefact, as well as enable researchers to use the research as a knowledge base to further extend and evaluate. The guidelines created in this thesis are presented in a document, easily understandable for practitioners.

Researchers can apart from the document also take part in the more detailed description of the findings available in chapter 4.

2.2 Data collection

This section explains the methods of data collection that have been used. The primary method of data collection has been semi-structured interviews that were conducted with software developers, software architects, and managers at Husqvarna and other organisations that perform in-house development. The secondary method of data collection has been a mixed data survey. The interview questions were constructed after initial research into software architecture, technology stacks and previous research papers regarding aspects to consider when taking architecture or technology decisions. During the construction of the questions, the focus was put on making them clearly worded, participant focused and non-leading for them to provide the richest possible data (Kallio et al., 2016). A pilot interview was conducted to test and further improve the questions. When conducting semi-structured interviews, it is important to have a good interview plan and to be careful when carrying out the interviews so that the way the interview is carried out does not lower the quality of the collected data (Hove & Anda, 2005).

2.2.1 Interview participants

Interviews were conducted with six employees at Husqvarna and three employees from other organisations. As shown in Table 1 below, the participants were software developers, software architects and diverse types of managers in software development departments. These three categories were chosen to capture a wide organisational perspective. The interview participants from Husqvarna were made up of two experienced developers who are now managers in development departments; three software architects; and one product manager. The participants were chosen based on their experience in the area for the collected data to be as valuable and useful as possible.

To find interview participants from other organisations, six small to medium-sized organisations in the Jönköping area were contacted via email. Three of the six organisations responded and agreed to participate. Participants from other organisations were one technology manager, one software architect and one software architect/developer.

Table 1: Interview participants

Role	Husqvarna	Other organisations
Software Architect	1	-
Software Architect/Developer	2	1
Line Manager	2	1
Product Manager	1	-
Developer/Line Manager	-	1

2.2.2 Conducting interviews

The interviews were conducted in a semi-structured way, with both specific questions and open-ended questions (Hove & Anda, 2005), to make sure that the appropriate data was collected and to minimise the risk of introducing bias. To further minimise the risk of introducing bias, an interview questionnaire and script were created to make sure all participants got the same base questions phrased in the same way. Before the real interviews were conducted, a pilot interview was conducted to test the questionnaire. After performing the pilot interview, minor changes were made to the questionnaire to increase the quality of the collected data in the real interviews (Kallio et al., 2016). The pilot questionnaire can be seen in Appendix 1 and the improved interview questionnaire can be seen in Appendix 2. The interviews were conducted digitally using video conference calls, both due to the coronavirus-pandemic but also due to digital interviews being more time-efficient for both participants and researchers when removing the need of travelling to the interview destination. This resulted in participants not being distracted from their regular work more than necessary. Participants were asked beforehand whether they preferred to do the interview in English or Swedish to make sure interview participants were as comfortable as possible during the interviews. This resulted in six interviews being conducted in English and three interviews in Swedish. Both authors participated in the interviews since that most often leads to more follow-up questions being asked and therefore more data being collected (Hove & Anda, 2005). Audio recordings were taken during the interviews to ease and improve the analysis. The data collected during the interviews was then

analysed and used to develop the draft guidelines, which is explained more in detail in section 2.3.

2.2.3 Ethics

All interview participants were asked beforehand via email whether they consented to an audio recording being taken. All participants consented after being informed that the recorded material and other sensitive information only would be used for the purpose of this research and that all recorded material would be deleted once the research is finished. At the beginning of each interview, participants were informed that they would always be anonymous in the report, and they were then asked for consent to be quoted in the report. All participants gave consent to be quoted.

When answering the survey, all respondents were informed that their answers always would be anonymized in the report.

2.2.4 Survey

The guidelines were then sent out together with a survey as the secondary method of data collection. The survey was sent both to people who have already been interviewed and to other developers, software architects, and managers at organisations that perform in-house development. The survey was created by taking screenshots of each section of the drafted guidelines, uploading it to an online survey tool and adding to each section a score slider from 1 to 10 depending on how useful or relevant the respondents thought each section was. Under each section, the respondents also had a chance to give more in-depth feedback in a free text answer so the respondents could explain their score in more detail and give their critique on the section in question. Scoring each section was mandatory while the free text answers were optional.

To make sure relevant data was collected, the first two questions of the survey were control questions to ensure that all respondents were within the relevant scope of the survey. The survey was sent out via email to the previous interview participants and some new people to get a broader dataset. The survey was sent to the organisations that had previously been interviewed. Respondents were also encouraged to share the survey within their organisation and with people they thought might be able to give some relevant feedback. This approach resulted in seven of the interview participants answering the survey and 13 new people. All respondents got the same set of questions in the survey. The data collected from the survey was both quantitative and qualitative in the form of free text questions and a score of each section and was used to improve and refine the guidelines, as well as validate that they apply to a broad audience. The full survey questionnaire can be found in Appendix 4.

2.3 Data analysis

This sub-chapter explains how the collected interview- and survey data has been analysed.

2.3.1 Interview data analysis

To analyse the collected data, the interview recordings were first transcribed and later analysed with thematic content analysis. Transcripts were written using Intelligent Verbatim transcription, meaning that repeated words, filler words and other things that are irrelevant and can distract the reader were removed. Intelligent Verbatim transcription was used to ease the analysis process and make sure the authors could focus on the meaning of the transcripts when analysing.

"Thematic content analysis is a method for identifying, analysing and reporting patterns (themes) within data" (Braun & Clarke, 2006, p. 79). Thematic content analysis is a widely used and flexible qualitative analytic method (Braun & Clarke, 2006). Braun and Clarke propose clear guidelines for performing thematic content analysis in psychology but also mention that it is a particularly useful analytic method beyond psychology. Thematic content analysis was chosen to properly establish patterns and common ideas between the different interviews.

The steps of thematic content analysis (Braun & Clarke, 2006):

1. Familiarising yourself with your data
2. Generating initial codes
3. Searching for themes
4. Reviewing themes
5. Defining and naming themes
6. Producing the report

By following this process of analysis, the interview transcripts were read through and things that were relevant to the study were highlighted. The transcripts were then gone through again to establish codes. Codes from all interviews were then gathered and themes were extracted. The themes were then reviewed to see how many of the participants that had mentioned each theme. These themes were then used to develop the draft guidelines.

2.3.2 Survey data analysis

When analysing the survey data, it was of high importance to first vet the data for consistency and completeness before starting the analysis (Kitchenham & Pleeger, 2003). After the survey had closed and all answers had been vetted for consistency and completeness, the analysis of the survey was started. The sections with the lowest score were looked at the most and the free-flowing text answers were used to improve the

draft guidelines. The other sections that got a higher score were also looked at, but those sections got less attention and changes due to the already good score.

2.4 Research process

This section presents the design science research process that has been used during this thesis. As shown in Figure 3 below, the main data collection method has been interviews. The data collected during the interviews was used to create the first version of the artefact: the draft guidelines. The draft guidelines were then evaluated in the survey by letting 20 respondents rank their usefulness and come up with suggestions for changes and improvements. With the feedback from the survey, the guidelines were modified slightly to further increase their usefulness and to create the second and final version of the artefact: the final guidelines.

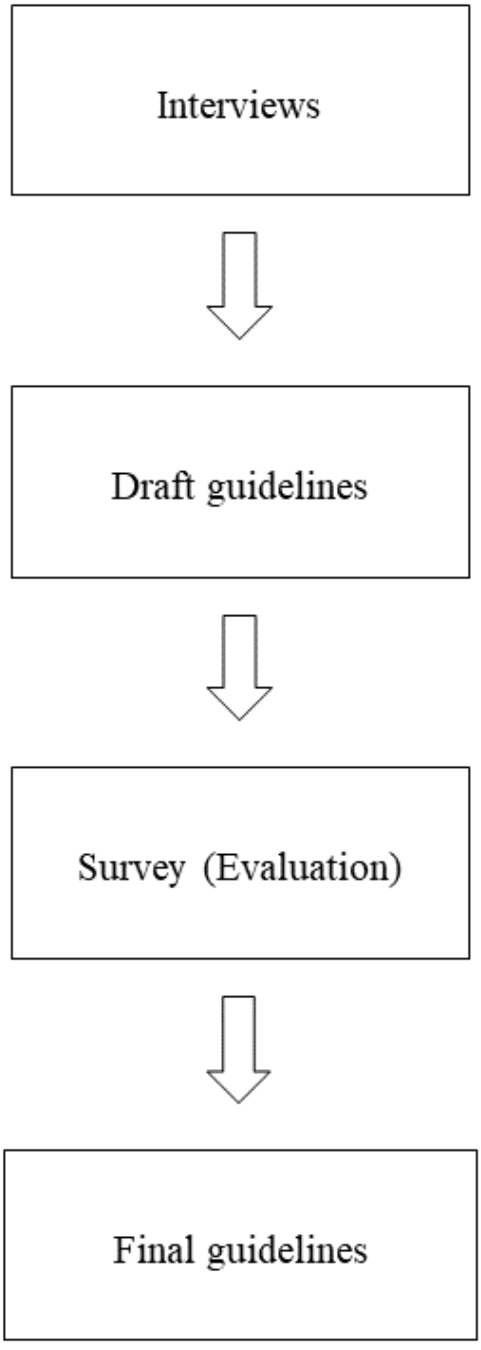


Figure 3: Design Science Research process of this thesis

2.5 Research element matrix

Figure 4 below describes which research elements have provided data to answer each of the research questions. As presented in Purpose and research questions (see section 1.2) the research questions are as follows:

> [1] What are the essential steps in the technology stack selection process?
>
> [2] When is it reasonable to choose similar technology stacks between different development teams?
>
> [3] When is it reasonable to choose different technology stacks between different development teams?

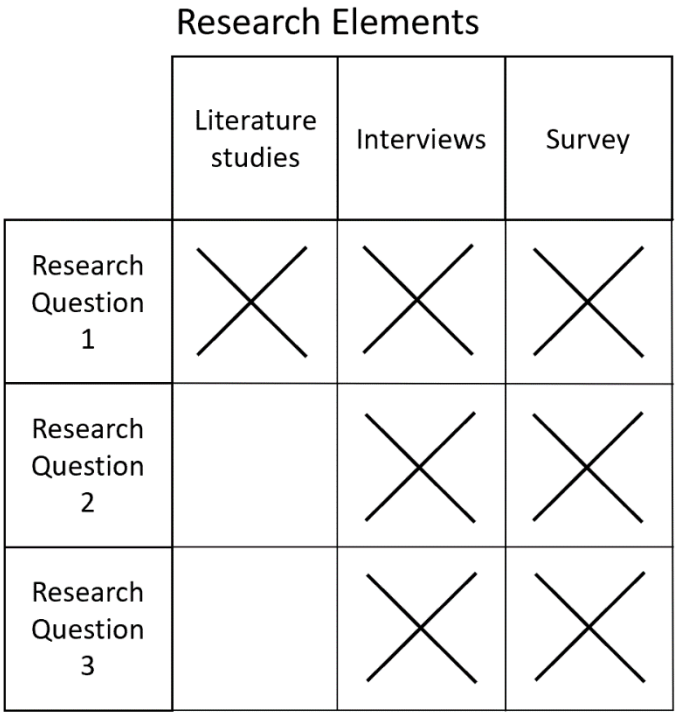


Figure 4: Research Element Matrix

2.6 Validity and reliability

Design science research was chosen as a research method to develop guidelines that are validated and improved by a larger number of respondents to make sure valid and reliable guidelines were produced. Compared to constructing guidelines from a few interviews only, the survey used as the method for validating and improving the produced artefact guaranteed more valid guidelines.

Semi-structured interviews were chosen as the main method for data collection to make sure detailed data could be collected. The interviews used a semi-structured format to ensure reliability while at the same time not limiting the data collected during the interviews. As mentioned in section 2.2.2 an interview questionnaire was created so

that all participants received the same base questions, but the semi-structured format allowed specific follow-up questions when necessary. To increase the validity and reliability of the interviews, a pilot interview was performed with a manager at Husqvarna to test the questionnaire. The pilot interview resulted in small changes to the questionnaire. The questionnaires can be seen in Appendix 1 and Appendix 2.

To ensure reliability in the performance of the survey, all respondents received the same questionnaire, and incomplete responses were not included as previously described in section 2.3.2. To ensure the validity of the survey, all respondents' email addresses were collected to enable reminders to be sent out to the people who might have missed the initial email and not answered the survey, as well as having the possibility to remove answers from unknown respondents whose knowledge of the subject is unclear.

2.7 Considerations

In this section, the considerations made throughout the thesis are outlined and explained.

The first consideration made was that interview participants might not be able to say everything they wanted if they had to speak English during the interviews. Because of that, it was decided that participants got to choose whether the interview should be performed in Swedish or English.

A pilot interview was conducted to make sure that the questions could generate good answers and that the time of the interview was not too long or too short.

Out of the nine interviews, none were conducted with a software developer only. All software developers interviewed also had a second role as a software architect or line manager. To get their input as well, the survey was sent to many software developers to get a broader audience to critique the guidelines. In the end, six software developers responded to the survey.

3 Theoretical framework

This chapter provides an overview of the current knowledge in the field, as well as established theories that have been kept in mind during the research. The Problem Statement (see section 1.1) identified that very few research papers addressed the topic of technology stack selection. Therefore, it was decided to widen the scope of the literature search to include software architecture in general. The theoretical framework first addresses software architecture in general, software architecture in agile development environments, the technology stack more specifically, and lastly, previous technology stack selection guidelines.

3.1 Software architecture

According to the ISO/IEC 42010 standard, the architecture of a system can be defined as the: "fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution" (ISO/IEC/IEEE, 2011). As seen from the definition, the architecture of a system is a wide term that could mean many different things. Software architecture, or the architecture of a software system, is the part of the system architecture which is relevant to this research.

3.1.1 The importance of a good architecture

As established in the Problem Statement, it is vital to select a good architecture from the start of a project. Rozanski and Woods (2012) argue that if the challenges of selecting a good architecture are not dealt with early, software projects will run over budget, be delivered late or with an unacceptably low level of quality. Falatiuk et al. (2019) reinforce this idea by saying that the architecture is the foundation of an application and that mistakes that are made at this stage of a project are the most expensive to fix. Considering the high importance of good architecture, more effort must be put into Software Systems Architecture research to make the process of selecting a good architecture easier.

3.1.2 Ways of handling architectural decisions

Rozanski and Woods provide a sort of guideline for how to handle software system architecture, and they argue that to establish a good architecture, it is essential to break down a system architecture into smaller pieces. Their method uses *viewpoints* and *perspectives* to break down an architecture into smaller and more understandable pieces. A viewpoint could be seen as a general solution to one part of an architecture. On the other hand, a perspective can be seen as a way of slightly modifying the selected viewpoints. One could, for example, apply the performance perspective if the system performance is of high importance.

Another way of making architectural decisions more manageable is to determine which architectural style to use. According to Falatiuk et al. (2019), the architectural style determines principles and a high-level paradigm for the architecture, which can help guide the more detailed decisions that have to be taken. They also mention six different architectural styles that can be used for different systems.

It is evident that to select good software system architecture, strategies must be applied to ease the process and make it easier to understand the otherwise overwhelming architecture. This shows that the guidelines developed in this research are relevant since they aim to guide the selection of one software architecture part, namely the technology stack.

3.1.3 Software architecture in agile development

With the introduction and increased use of agile methods in software development, the role of software architecture and software architects changed. Babar (2009) explored the challenges of architectural decisions when using agile development approaches. Babar claims that many practitioners of agile methods view software architecture as part of the plan-driven development paradigm and that, for them, it is usually not worth it to evaluate software architecture. Babar does however state that there is a growing interest in the architectural aspects of agile methods and increased recognition of the mechanics and prerequisites of integrating agile and architectural approaches.

Madison (2010) takes this one step further to provide a framework for incorporating architecture into common agile methods. Madison shows how software architects can help developers by integrating architectural decisions into agile user stories. However, this means that architectural changes could be left waiting if part of a non-priority user story.

Agile ways of working have changed the process of software architecture and it is clear that developers practising agile methods usually have a different view of architecture.

3.2 Technology stack

Whilst there is no universally accepted definition of a technology stack, it usually refers to a set of programming languages, frameworks, libraries, databases, and many other technologies (Nikulchev et al., 2021). There are therefore many unique technology stacks, making specific guidelines for when to select a specific technology stack over another hard to find. According to Bass et al. (2012, p. 19), there is no such thing as the best software architecture. Since the technology stack is a part of the architecture, this would also apply to the technology stack. If you would put two equally skilled software architects or software developers on the same project with the same requirements, they would most definitely design their architectures and technology stacks differently, but that does not mean that one is better than the other (Bass et al., 2012, p. 56). The

responsibilities of an architect for a product include many aspects and aims to please customers, testers, developers, stakeholders, and project managers and because of this it also affects the company or organisation the architecture is made for (Bass et al., 2012, p. 29/47). This is also supported by Van Der Linde (2013) who outlines what a software developer needs to think of to ensure system development, one of them is to follow the existing architecture but contrary to the Bass et al. (2012), Van Der Linde (2013) argue that the developers have a say in creating and following the system development plan and defining the software and requirement baseline.

3.2.1 Software architect's role in the technology stack selection

The term software architect is still quite a new term and occupation that has no one single accepted definition. According to Bass et al. (2012, p. 19), there is no such thing as good or bad software architecture. It all depends on the product being produced, but there are rules of thumb and one in particular that sticks out. It states that the architect that is building the software architecture, should never depend on a particular version of a commercial product or tool or in other words a part of the technology stack. This is so, that if said product or tool the architect picked out cannot be used by the software developers it would be more inexpensive and straightforward to change to another product or tool.

3.2.2 Software developer's role in choosing the technology stack

In a study where both interviews and surveys were conducted with software developers, developers were asked whether they just wrote code or if they had a say in how to improve functionality or what system components to use. All the interviewees answered that they did have a say in all those things. The survey also supports the idea that most developers do have a say in the development architecture, with 84.21% answering yes (Van Der Linde, 2013).

Developers were also asked if they thought using the same stack and re-using code from other projects from teams in the company was a good idea. Most of the participants did not agree that re-using components was a good idea and said that building new systems with other technology stacks would increase productivity and save time since the developers were not forced to customise already existing components that were not always guaranteed to work, resulting in workarounds and an all-around lessening in the quality of the product being developed (Van Der Linde, 2013).

Van Der Linde (2013) also concluded that if software developers only wrote code and did not come with any input on the architecture, more specifically the technology stack of the system, the impact of those developers' code on other parts of the system was not realised, which will most likely take time to find, affect the efficiency negatively and increase costs for the company producing the product.

3.3 Previous technology stack selection guidelines

This section presents and discusses previous technology stack selection guidelines that have been established in research. Zhu and Shao (2015) created Web Technology Stack Selection (WeTS) guidelines, a framework meant to help inexperienced developers in their selection of a web application technology stack. WeTS consists of three parts: WeTS Processes, WeTS Algorithm and WeTS Software Quality Characteristics. The WeTS Processes are a set of instructions that developers should follow in given order when using the WeTS guidelines. The WeTS algorithm provides an overall score on a technology stack and the WeTS Software Quality Characteristics are strategies by which a technology can be evaluated and ranked.

If WeTS were to be used to select a technology stack, the first step is to establish what software quality criteria that are important for the task at hand. Step two is to establish which evaluation criteria to use to assess and evaluate each of the selected quality criteria. Step 3 is to identify possible technology stack solutions, either by building stacks from scratch or by selecting common technology stacks. The WeTS algorithm should then be used to score the different technologies and technology stacks for each quality characteristic to then compare the scores between the different technology stacks.

While the WeTS guidelines are relevant, they do have some limitations. Zhu and Shao make it clear that WeTS is a framework for selecting Web Technology stacks, however, one could argue that WeTS might work as well for other technology stacks, such as stacks for mobile or desktop applications. Another weakness is that Zhu and Shao mention the fact that their approach builds on putting together candidate technology stacks before the different technologies have been evaluated. It could be hard to put together reasonable technology stacks before evaluating without first establishing some requirements and needs of the product, resulting in having to evaluate many technology stacks that might not work very well.

Compared to WeTS, Koder (2021) does not provide outright guidelines for technology stack selection, Koder investigated how productivity in full-stack web development could be increased by technology selection. Whilst the study presents interesting productivity factors and data on lots of different technologies, Koder clearly states that the research is focused on "…development of prototypes with a small team or even by a sole developer" (p. 8). Even if the productivity factors established by Koder are sufficient, it is unclear how they would apply to multiple development teams or organisations working on multiple products. Koder (2021) also states that factors such as process, teamwork and other non-technical factors are outside the scope which reinforces the idea that Koder´s data might not be relevant to this thesis.

4 Results

This section presents the results of the research. Section 4.1 and 4.2 objectively present the data that has been collected during the interviews and survey respectively. Section 4.3 then discusses the draft guidelines that were established from the interviews, and section 4.4 justifies the final guidelines that were established after the survey.

4.1 Interview data

This section presents the data that was collected during nine semi-structured interviews with software architects, software developers and managers. The main themes extracted from the interviews are presented in Table 2 below. Table 2 also shows how many of the participants mentioned each theme. The themes are discussed in more detail and linked to participant responses in the following sections 4.1.1 to 4.1.6:

Table 2: Interview data themes

Interview data theme	**Mentioned by**
Technology stack decision making should be done in collaboration with all affected stakeholders	9/9
Software architects should be responsible for technology stack decisions	8/9
Technology stack supporting the purpose of the product	8/9
Importance of good resources	7/9
Similar technology stacks within an organisation	5/9
Advantageous to have different technology stacks	7/9

4.1.1 Technology stack decision making should be done in collaboration with all affected stakeholders

All interview participants said that technology stack decisions should not be taken by one person or by people of a specific role. There should be a discussion regarding technology stack decisions, and it is important to create a consensus. All affected parties should be part of the discussion.

Participant 4 – *"I think the architect is in a position where he's involved in quite many most perhaps of these questions but taking the expertise from developers."*

4.1.2 Software architects should be responsible for technology stack decisions

Eight out of nine participants agreed that software architects should be responsible for the decisions but take input from the other people involved, and not take the decisions him/herself. Participants were clear that responsibility does not mean that architects should take the decisions. Only one participant disagreed and stated that a development team should have shared responsibility for all decisions regarding the team.

Participant 6 – *"So hopefully it could be a collaboration and you can come to a joint decision I would say because it's a collaboration all throughout the project between architects and developers. But I guess if the system has to be made, I would say perhaps the architect should have the final say then."*

A minority of the participants said that if developers would get to select the technology stack more freely, they would be more comfortable in their work. One participant also stated that developers selecting the technology stack would lead to more effective development. In contrast to that, participant one stated:

Participant 1 – *"If you only choose what the developer would like to have, you will never get any product ready, and it will be very costly."*

4.1.3 Technology stack supporting the purpose of the product

Eight out of nine interview participants said that when selecting a technology stack, it is most important to select a stack that is suitable for the product to be developed. The technology stack should provide the tools necessary to solve the problems that the product has and will have, and make sure that the product meets its requirements.

The most important aspect to consider when selecting a technology stack:

Participant 2 – *"It needs to solve the task, it must be adapted to the task you have."*

Participant 3 – *"If the technology stack should be suitable for that type of issue or thing you're trying to solve is very important."*

Participant 4 – *"It's of course a high-level suitability for the service that we put on the market."*

Participant 8 – *"It is partly what the purpose is, what you are going to build of course. And that must be the underlying thing."*

4.1.4 Importance of good resources

Seven out of nine interview participants mentioned that resources are very important when selecting technology stacks. Both what resources you have in the organisation,

and what resources you can find and recruit in the future. Five out of nine interview participants also stated that this is especially important when working on-site, and not as important when working remotely since the number of resources available increases sharply if resources do not have to be local. When working remotely it is more important that the technologies chosen are common worldwide or in the entire area where resources are recruited.

Participant 3 – *"And as I said, the resources that you have access to or will have access so that they know the technology stack and be able to work in it in an efficient way is very important."*

Participant 7 – *"Again, I said staffing is very important. You should be able to find people with that competence"*

4.1.5 Similar technology stacks within an organisation

Five out of nine interview participants preferred having as similar technology stacks as possible in an organisation, but also stated the importance of not limiting the technology stack if a product had specific requirements that could not be fulfilled with the same or a similar technology stack.

Participant 6 – *"If there are no real benefits to selecting different I would say just to stick with the same, if you need to perhaps move people around between squads, you need help between them, then I think it's easier if it´s the same stack, but yeah, it shouldn't limit specific use cases obviously."*

Participant 9 – *"I would try to look at the needs and select as I said, as few stacks as possible."*

The main reason mentioned for having similar technology stacks was that with a similar technology stack across multiple teams it is easier for team members to switch between different teams, which in most cases leads to positive effects for an organisation. Similar technology stacks also lead to more knowledge exchange between teams, regardless of if people are switching between teams or not. Secondly, similar technology stacks allow for patterns and common services to be developed, which can then be used by multiple different teams.

Participant 5 – *"If I were to generalise I would say you should have similar stacks to facilitate moving between different teams."*

Some participants also mentioned that having a similar technology stack does not explicitly make it easy to switch teams. There are other equally important factors. Such as the willingness to switch among team members, whether there is an open social culture that facilitates switching, and whether the overall architecture is similar (not only the technology stack).

Participant 6 – *"But it is easier to switch in my sort of experience if things look the same, and perhaps it's not only the technology stack, because you can have the same technology stack but totally different architectures, you know internally, and perhaps that is even harder to switch between I guess."*

4.1.6 Advantageous to have different technology stacks

Whilst no participant explicitly stated that they preferred having different technology stacks in an organisation, seven out of nine participants mentioned that there are cases where it can be advantageous to have different technology stacks. Mostly if specific products or services have very different needs and requirements that would be compromised by using a similar technology stack. One participant also mentioned that it can be good to select a different stack if the product needs to be delivered very quickly.

4.2 Survey data

This section presents the data that was collected from 20 survey responses. The survey was sent out to the previous interview participants and some more people to get broader insight. Table 3 below shows how many survey respondents that also were interview participants, as well as if they were employees at Husqvarna or another organisation.

Table 3: Survey respondents

Type of respondent	Husqvarna	Other organisations	Total
Interview participant	4	3	7
Non-interview participant	4	9	13

The roles of the survey respondents are presented in Figure 5 below.

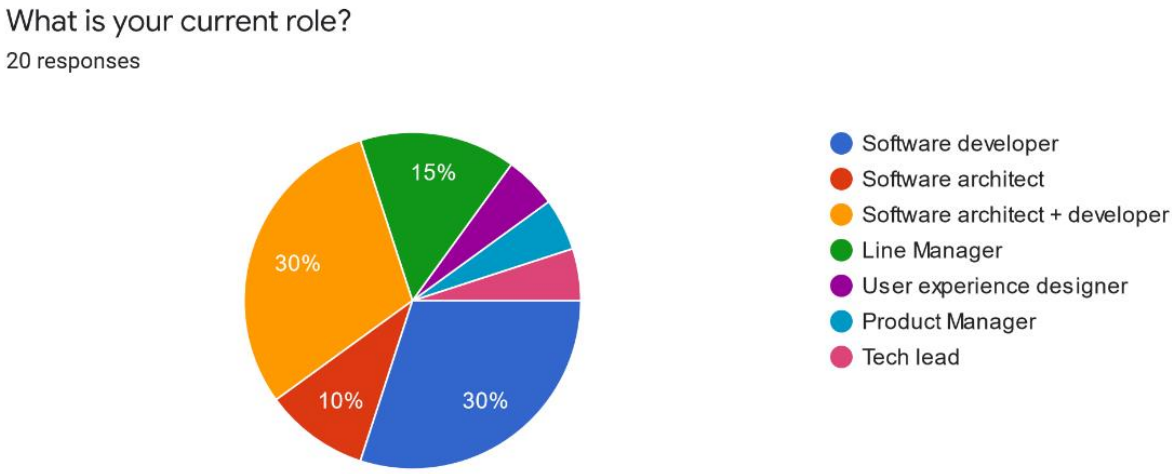


Figure 5: Roles of survey respondents

4.2.1 Score of survey sections

Each respondent was asked to score each section of the draft guidelines as well as their overall score for the entire draft guidelines with a score from 1 to 10 based on relevance and usefulness. All respondents scored all sections of the guidelines. The mean scores of the different sections are presented in Figure 6 below. All scores can be seen in Appendix 5 where respondents from Husqvarna are compared with respondents from other organisations. As shown in Appendix 5, there were no significant differences in scores between Husqvarna employees and others.

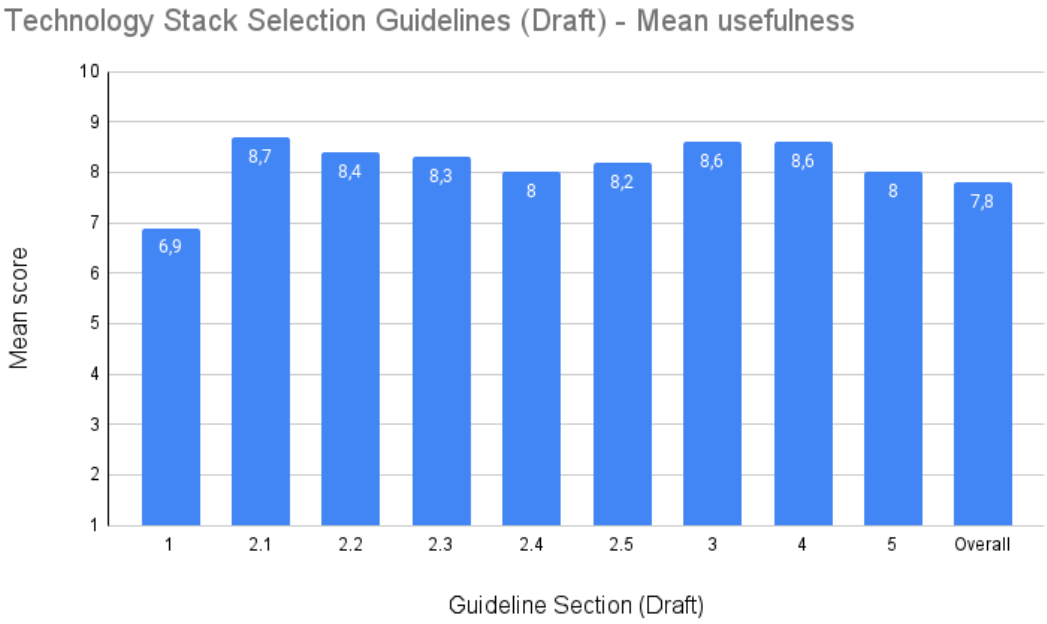


Figure 6: Technology Stack Selection Guidelines (Draft) – Mean usefulness
Scale: 10 = Very useful, 1 = Not useful at all

4.2.2 Survey free text answers

When asked about the guidelines as a whole, six out of eight survey respondents that answered the optional free text mentioned that they were very relevant, but also gave some good feedback on things than could be improved.

Respondent 2 – *"Short and to the point. It shouldn't be a complicated process."*

Respondent 5 – *"Relevant and good points/bullets to think about. Some are very valid! It is always hard to look into the future and to know where to place your bet. Requirements always changes, and we need to be able to quickly re-think the design/implementation/architecture."*

Respondent 15 – *"Good guidelines to consider. My experience is that the decision of selecting a tech stack is not always taken that seriously. Having these guidelines can be a good help in the process."*

Section two, part one of the draft guidelines received some free text answers both with suggestions for improvement and with positive feedback:

Respondent 2 – *"Not all requirements will be known when selecting a stack. Important to design for change as well."*

Respondent 5 – *"Sometimes you don't know the requirements when you start to build the service. So, also important is to think about that requirements will change, and design for that from the start."*

Respondent 8 – *"Yes, this is key: product needs should shape technology, not the other way around."*

4.3 Draft Guidelines

To answer research question one: *What are the essential steps in the technology stack selection process?* draft steps were derived from the themes presented in section 4.1. And to answer research questions two: *When is it reasonable to choose similar technology stacks between different development teams?* and three: *When is it reasonable to choose different technology stacks between different development teams?* different scenarios were derived from the themes presented in section 4.1. These essential steps and scenarios were then combined to form the draft guidelines (see Appendix 3). The draft guidelines were then sent out in the survey to be evaluated. The survey responses were then analysed as previously described in section 2.3.2. The analysis resulted in smaller improvements to the guidelines which formed the final guidelines (see Appendix 6). Section 4.4 justifies the final guidelines.

4.4 Final Guidelines - Justification

The conclusions presented and discussed in this chapter can also be seen in the Final Technology Stack Selection Guidelines (Appendix 6). The sections below present and

discusses the different sections of the guidelines to answer the research questions and fulfil the purpose of this thesis.

4.4.1 Essential steps of technology stack selection

Identify what needs and requirements the product will have for it to fulfil its purpose: As mentioned by all but one interview participant, it is most important to select a technology stack which is suitable for the product that will be developed. And before that can be done, the needs and requirements of the product must be known. This was strongly reinforced in the survey with a mean relevance score of 8.7/10.

Establish which stakeholders should be involved in technology stack discussions and decision-making: All interview participants stated that it is very important that all affected parties are involved in technology stack decision-making, however in a large organisation it might be hard to involve everybody, and it is, therefore, important to establish who should be involved so that everyone feels heard.

Establish technologies for which you already have the competence in the organisation: Seven out of nine interview participants mentioned resources as a vital thing to have in mind when selecting technology stacks, and firstly to establish for what technologies you already have competence in the organisation. However, three out of the five survey respondents that wrote a free-text answer mentioned that it´s most important to have competent resources in the organisation, and if you do, they will be able to learn new technologies and competencies.

Establish technologies for which you think you can find good resources in the future: The second part of resources regards future resources. This can be less crucial when working remotely due to the number of possible resources increasing sharply when they do not have to be local but is still important to consider.

Establish how important it will be for you to keep costs down: Five out of nine interview participants mentioned costs as something to consider when selecting a technology stack, but no one said that cost was the most important. One participant mentioned that it can be good to first establish whether costs are a problem or if it is more important that the product is good for the user and easy to maintain. Survey respondents mainly agreed with a mean relevance score of 8.2/10 and said that maintenance costs are often not thought about and that costs are very hard to predict and therefore should not weigh very heavily in the selection process.

4.4.2 Scenarios where to select similar technology stacks

This section answers research question two by presenting scenarios where it is reasonable to select similar technology stacks for different development teams. The scenarios are as follows:

When you want to facilitate easy competence moves between different development teams: Six out of nine interview participants stated that similar technology stacks make it easier for members to switch between teams, which in most cases has a beneficial impact on the organisation. Members bring with them other perspectives and experiences which can be beneficial for the new team.

When you want to have the possibility to develop common services: Common services could be things like internal code libraries, policies for how to handle security or operations setups. Six out of nine interview participants stated that similar technology stacks allow for common services to be developed, which can reduce the amount of work each team needs to undertake, and thus prevent having to tackle the same problem many times.

When you want to facilitate cross-team knowledge exchange: Even without members switching to a different team, knowledge sharing across teams can be powerful and have large positive effects.

When you want to be able to reuse parts of an application and therefore lower the development costs and future maintenance costs: If elements of two different applications are identical, it is natural to want to reuse those elements in both applications. If similar technology stacks are necessary to reuse these elements, it can be a good indication of how to select technology stacks.

Survey respondents confirmed that the scenarios presented above are useful and agreed that similar technology stacks facilitate common services to be developed and competence moves between teams. Some respondents did however mention that it is often trickier to use similar stacks in reality compared to what it might seem on paper. This of course makes the decision harder, but the scenarios above are still relevant.

4.4.3 Scenarios where to select different technology stacks

This section answers research question three by presenting scenarios where it is reasonable to select different technology stacks for different development teams. The scenarios are as follows:

When the result or user experience would be compromised by selecting similar technology stacks: As mentioned by all but one interview participant, it is most important to select a technology stack which is suitable for the product that will be developed. And therefore, if the result or user experience would be compromised by selecting similar technology stacks, it is reasonable to select different technology stacks.

When you want to optimise teams to deliver something with specific requirements or within a limited timeframe: If in a special scenario where one team is under high

pressure to perform, it can be reasonable to select a technology stack that is optimised for that team to ensure they have the best possible conditions to deliver.

5 Discussion

This chapter discusses the results of the study, what the results indicate and how the results relate to previous research. It then discussed the methods used, if the methods helped fulfil the purpose and answer the research questions, and if something could have been done differently. Lastly, the limitations of this study are presented and discussed.

5.1 Result discussion

This section discusses the results of the research and evaluates the results in relation to the purpose and research questions. As previously mentioned in Purpose and Research Questions (see section 1.2) the purpose of this research is:

> *To develop guidelines for organisations to use during the technology stack selection process. The developed guidelines aim to help organisations in their evaluation of different solutions and help organisations decide whether it is worth it to choose different technology stacks for different development teams.*

And the research questions are:

> [1] What are the essential steps in the technology stack selection process?
>
> [2] When is it reasonable to choose similar technology stacks between different development teams?
>
> [3] When is it reasonable to choose different technology stacks between different development teams?

To establish the essential steps in the technology stack selection process, data was collected through semi-structured interviews and a mixed data survey. The interviews provided solid answers for RQ1, whilst at the same time showing that the answer greatly depends on the situation. The interviews clearly showed that all affected parties (developers and architects) should be and are involved in technology stack decision-making, which confirms the results from Van Der Linde (2013). The survey then confirmed that the results of the interviews are valid and that the interviews do answer RQ1 and fulfil the first part of the purpose.

The results of this research also confirm that of Bass et al. (2012) who stated that a software architecture cannot be classified as good or bad because it all depends on the product which it is used for. It was made clear during both data collections that it is essential to design the technology stack (one part of the software architecture) for the product being produced.

An interesting question that came up in the survey is how to make sure that everyone affected gets to have a say in the selection process. Because while it can sound simple,

it can be hard in practice. In a large organisation, everyone probably cannot be involved in discussions, and one survey respondent even stated that it can be very hard to decide if there are too many people involved. And whilst it is outside the scope of this thesis to answer how to select representatives to be involved in the selection process, it is an interesting and hard question that organisations have to deal with which affects the selection process.

Contradictory to Van Der Linde (2013), five out of nine of our interview participants mainly saw pros of using similar technology stacks, reusing components, and developing common services. Participants in Van Der Linde (2013) stated that building new systems with other technology stacks would increase productivity, and whilst one participant in our interviews said that different technology stacks can lead to more effective development, the overall opinion was that similar technology stacks increase the efficiency of an entire organisation. This difference could be due to the fact that participants in Van Der Linde (2013) all were software developers working for the same organisation that might have had a culture of different technology stacks.

The second part in the draft guidelines (see Appendix 3) "Essential steps in the technology stack selection" got a very good overall score in the survey, but when reading the written feedback of bullet point three "Establish technologies for which you already have competence" it becomes quite clear that the respondents did not think this was as important as was thought before. Three out of five survey respondents that wrote in the free-text answer agreed that it was of utmost importance to have good competencies to rely on in a company but almost equally as important to have talent that could learn new competencies, so you do not have to rely on old technology.

RQ2 was also answered during the interviews, showing multiple scenarios when it is reasonable to select similar technology stacks, providing a guide to organisations on when to select similar technology stacks for multiple development teams, and therefore partly fulfilling the second part of the purpose. The survey assured that the scenarios extracted from the interviews are valid and applicable in the industry.

The results indicate that there are more scenarios where it is reasonable to select similar technology stacks for multiple development teams than there are scenarios where it is reasonable to select different technology stacks for multiple development teams. However, this could be a bit misrepresentative since there are many scenarios where it is simply not possible to select similar technology stacks. Naturally, scenarios where it is reasonable to select different technology stacks for multiple teams also include all scenarios where it is not possible to select similar technology stacks. Even though more scenarios benefit similar technology stacks, some people in both the interviews and survey mentioned that it could be good for both the employees and the company to sometimes dare to choose different technologies to learn something new and to keep the developers sharp.

All but one interview participant stated that they have had good experiences of team members switching between teams in an organisation and that it most likely leads to positive effects for an organisation. Six out of nine participants argued that there needs to be similar technology stacks for members to switch, whilst some interview participants and survey respondents stated that it is good for members to switch to other technologies to expand their knowledge base.

One interesting thing that was found out when discussing the architect's role in the technology stack decision making, was that three participants agreed that an architect that was also a software developer was way better and more reliable at taking technology stack decisions than an architect with little to no experience with software development. While this was not a question in the questionnaire, these 3 participants brought it up when asked who should have the responsibility when choosing the technology stack. Everyone who brought it up said that an architect that does not know software development should not be taking decisions on a software level.

When analysing the survey results, the mean score of overall usefulness stood out as relatively low compared to all individual section scores. When taking the mean of all individual scores, the overall score results in 8.2, higher than the actual overall score of 7.8. The low mean score of overall usefulness is mainly caused by respondent two who scored the individual sections from 4 to 8, but an overall usefulness score of 2. Respondent two did explain the answer and said that the guidelines felt very obvious and that he/she was unsure whether they would add any value to a mature organisation. It is however unclear why respondent two ranked all sections as more useful than the guidelines overall. All other respondents gave the guidelines an overall usefulness of 7-10 and the overall score from respondent two is the only real outlier that can be seen in the survey responses.

A factor that could have influenced the answers to research questions two and three is that six out of nine interview participants worked at Husqvarna who have relatively similar technology stacks for different teams and different products and have had a tradition of similar technology stacks for a long time. This could have affected the results as participants might like the setup at Husqvarna and therefore favour having similar technology stacks, but it could also be the other way around that those participants have worked with similar stacks for a long time and might therefore favour different technology stacks to get some change. However, during the interviews, there was no noticeable difference in the opinion of similar stacks vs different stacks between Husqvarna and the other organisations that took part.

5.2 Method discussion

The research method chosen for this thesis was design science research (DSR). The advantages of using DSR have been that the final guidelines were evaluated and

improved rather than just established after interviews. The evaluation process of design science research also confirmed that the final guidelines are applicable and useful to the industry. Without the evaluation stage of DSR, it would be very hard to know whether the final guidelines would be applicable and useful.

5.2.1 Interviews

All the interviews were conducted online via video conference calls to improve efficiency and reduce the environmental footprint of this thesis. Online interviews can however lead to interviewers missing participants´ body language, but since video conference calls were used the amount of body language missed was kept to a minimum. Only the audio was recorded for each interview. This was done to make participants feel more comfortable, and since it is hard to watch a video while transcribing the interviews, there would have been few benefits to recording the video as well. Since interviews were performed in the language preferred by the participant, this together with not recording video of the participant made sure that participants felt as comfortable as possible during the interviews.

The interviews were performed using a semi-structured format and an interview questionnaire. The execution of a pilot interview was good and resulted in smaller changes to the questionnaire which made questions clearer for the real interviews. Both authors participated in all interviews which most likely resulted in more follow up questions being asked compared to if interviews would be performed by only one author. The interviews fulfilled their purpose and provided data that was used to create the draft guidelines. The interviews provided rich data for answering research questions one and two but left a little to be desired for research question three. This could however mean that there are not as many scenarios where it is reasonable to select different technology stacks for different development teams, but that is hard to say from the interview data. The interviews did fulfil their purpose as they led to draft guidelines which were applicable and useful for organisations as shown by the survey.

5.2.2 Survey

The survey was sent to both the interview participants as well as others to make sure the conclusions drawn from the interviews were valid and that they apply to a wider audience. This was achieved since a good amount of both interview participants and others answered the survey. The results from the survey confirmed that the interview data answered the research questions, whilst also providing some new answers to research questions one and three. The execution of the survey was overall good, but some questions and descriptions could have been better clarified for respondents to better understand the draft guidelines and questions. This was most evident in section one, where four respondents wrote free-text answers indicating that they thought something was unclear. This could have affected scores for section one negatively.

If another survey was to be performed, the entire draft guidelines document should be available for viewing at the start of the survey, resulting in the respondents getting a better idea of the draft guidelines before ranking specific sections. Section scores now might be affected by respondents who thought something was missing when it was actually mentioned in a different section. Participants did however have the possibility to go back and change their answers if they realised something actually was not missing, which should have reduced the impact it could have had on the results.

5.2.3 Validity and reliability

Reliability and validity in the execution of interviews were ensured by using the interview questionnaire and formulating questions as non-leading as possible. The validity of the survey was increased by gathering the email addresses of respondents, meaning any respondents not working for an organisation performing in-house development could be excluded from the results. The reliability of the survey is ensured by the survey questionnaire available in Appendix 4.

5.3 Research limitations

This section discusses the limitations of this research which could have affected the results.

None of the participants in the interviews were solely software developers. The interviewed software developers all had a second role as either software architects or line managers. It would have been good to capture the perspective of a software developer who is not as involved in architecture work since they might see things from a different perspective. To minimise the adverse effects this limitation could have on the validity of our research, ten software developers were invited to answer the survey, and six software developers responded to the survey, ensuring that this perspective was not missed completely.

As previously discussed in Method and Implementation (see section 2), the draft guidelines created in this research were evaluated and improved once to produce the final guidelines. And as described by (Hevner et al., 2004), it would have increased the quality of the final guidelines if more iterations of evaluation or different types of evaluations had been performed. However, that was not possible in the timeframe of a bachelor´s thesis.

A possible limitation could be that six out of nine interview participants work at Husqvarna, an organisation which mainly uses similar technology stacks. This could be something that has affected the participants and made them biased, although throughout both the interviews and the survey no clear evidence of this was found.

6 Conclusions and further research

This chapter presents the conclusions of this thesis and suggests related topics that can or need further scientific investigations.

6.1 Conclusions

This section presents the conclusions of the research and the implications it can have for practitioners and the scientific community. The authors made the conclusions after performing nine semi-structured interviews and a mixed data survey with twenty responses.

The study has been focused on developing guidelines for organisations to use in order to save time, money, and resources in the technology stack selection process. The guidelines have proved during the survey phase to be both relevant and useful. The guidelines provide organisations with the essential steps of technology stack selection, which are as follows:

- Identify what needs and requirements the product will have for it to fulfil its purpose
- Establish which stakeholders should be involved in technology stack discussions and decisions
- Establish technologies for which you already have the competence in your organisation
- Establish technologies for which you think you can find good resources in the future
- Establish how important it will be for you to keep costs down

The guidelines also provide scenarios where it is reasonable to select similar or different technology stacks for multiple development teams:

Scenarios where it is reasonable to select similar technology stacks for different development teams:

- When you want to facilitate easy competence moves between development teams
    - Members switching between development teams most often provide positive effects for an organisation. When switching, the members bring other perspectives and experiences that can be shared within the new team.
- When you want to have the possibility to develop common services (internal libraries, policies for security handling, etc.)

- When you want to facilitate cross-team knowledge exchange.
    - Knowledge exchange between teams can be powerful and provide significant positive effects, even without members switching to a different team.
- When you want to be able to reuse parts of an application and therefore lower the development costs and the future maintenance costs.

Scenarios where it is reasonable to select different technology stacks for different development teams:

- When the result or user experience would be compromised by selecting similar technology stacks.
- When you want to optimise teams to deliver something with specific requirements or within a limited timeframe.

6.1.1 Practical implications

The guidelines produced in this thesis can lead to organisations saving time, money, and resources both during the selection process and during a project after selecting effective technology stacks, as shown by the high scores of relevance and usefulness achieved in the survey. In addition, the guidelines guide the technology stack selection process, making it easier for organisations to select technology stacks. One interview participant even mentioned after having answered the survey that he/she saw high value in section 3 of the draft guidelines and that he/she will use section 3 when selecting technology stacks in the future.

6.1.2 Scientific implication

Since the results and conclusions of this thesis are general, they can serve as a base for future research going more in-depth on specific parts of the technology stack. The results and conclusions can also be used together with technical research on specific technologies to provide more detailed guidelines for technology stack selection.

6.2 Further research

One interview participant asked whether interviews were going to be performed with any employees from government organisations, which was not done. The participant had previous experience with such organisations and mentioned that they often have more rules and regulations to comply with regarding their technologies, making the technology stack selection process different from private sector organisations. It would be interesting for further research to investigate how the technology stack selection process differs in an organisation with a lot of rules and regulations to comply with for technology choices.

It would also be very interesting to perform a study where the guidelines produced in this thesis are tested and evaluated in reality to see how much organisations can benefit from using the Technology Stack Selection Guidelines.

As previously mentioned in the Result Discussion (see section 5.1) survey respondents brought up the question of how to decide which representatives should be involved in technology stack decision-making. This is an important question since all stakeholders should have input in the process, but as mentioned by survey respondents, it can become hard to decide if there are too many people involved in the process. It would therefore be of importance to further research how to select representatives to represent all stakeholders.

7 References

Aggarwal, S., & Verma, J. (2018). Comparative analysis of MEAN stack and MERN stack. *International Journal of Recent Research Aspects, 5*(1), 127-132. https://www.ijrra.net/Vol5issue1/IJRRA-05-01-26.pdf

Babar, M. A. (2009). An exploratory study of architectural practices and challenges in using agile software development approaches. *2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture* (pp. 81-90). Cambridge, UK: IEEE. https:/doi.org/10.1109/WICSA.2009.5290794

Baskerville, R. L., Kaul, M., & Storey, V. C. (2015). Genres of Inquiry in Design-Science Research: Justification and Evaluation of Knowledge Production. *MIS Quarterly, 39*(3), 541. https://doi.org/10.25300/MISQ/2015/39.3.02

Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice* (3 ed.). Addision-Wesley.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77-101. https://doi.org/10.1191/1478088706qp063oa

Falatiuk, H., Shirokopetleva, M., & Dudar, Z. (2019). Investigation of Architecture and Technology Stack for e-Archive System. *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, (p. 7). https://doi.org/10.1109/PICST47496.2019.9061407

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in Information Systems Research. *MIS Quarterly, 28*(1), 75-105. https://doi.org/10.2307/25148625

Hove, S. E., & Anda, B. (2005). Experiences from conducting semi-structured interviews in empirical software engineering research. *11th IEEE International Software Metrics Symposium (METRICS'05)*, (pp. 10 pp.-23). https://doi.org/10.1109/METRICS.2005.24

Iiavari, J., & Venable, J. R. (2009). Action research and design science research - Seemingly similar but decisively dissimilar. *ECIS 2009 Proceedings*, (p. 73). http://aisel.aisnet.org/ecis2009/73

ISO/IEC/IEEE. (2011). *Systems and software engineering — Architecture Description.* IEEE.

Kallio, H., Pietilä, A.-M., Johnson, M., & Kangasniemi, M. (2016). Systematic methodological review: developing a framework for a qualitative semi-

structured interview guide. *Journal of advanced nursing, 72*(12), 2954-2965. https://doi.org/10.1111/jan.13031

Kitchenham, B., & Pleeger, S. L. (2003). Principles of survey research part 6: data analysis. *Software engineering notes, 28*(2). https://doi.org/10.1145/638750.638758

Koder, M. (2021). *Increasing Full Stack Development Productivity via Technology Selection.* Jyväskylä: Jyväskylä University of Applied Sciences. Retrieved from https://www.theseus.fi/bitstream/handle/10024/509674/Masters_Thesis_Koder.pdf?sequence=2&isAllowed=y

Madison, J. (2010). Agile Architecture Interactions. *IEEE Software, 27*(2), 41-48. https:/doi.org/10.1109/MS.2010.35

Nikulchev, E., Ilin, D., & Gusev, A. (2021). Technology Stack Selection Model for Software Design of Digital Platforms. *Mathematicas, 9*(4), 308. https://doi.org/10.3390/math9040308

Rao, S. (2009). *Choosing the right GIS framework for an informed Enterprise Web GIS Solution.* http://www.ciesin.org/documents/rao_gisframe-paper-jan2010.pdf

Rozanski, N., & Woods, E. (2012). *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives.* Addison-Wesley.

Van Der Linde, J. (2013). *An investigation into the impact of enterprise architecture decisions on the responsibilities of software developers in companies that develop software.* Master Thesis, University of South Africa. https://uir.unisa.ac.za/bitstream/handle/10500/10147/dissertation_van%20der%20linde_j.pdf?sequence=1&isAllowed=y

Zhu, C., & Shao, W. (2015). *WeTS - Web Technology Selection Guidelines.* KTH Royal Institute of Technology. http://kth.diva-portal.org/smash/get/diva2:928809/FULLTEXT01.pdf

8 Appendices

Appendix 1: Interview questionnaire - Pilot

Appendix 2: Interview questionnaire - Final

Appendix 3: Technology Stack Selection Guidelines - Draft

Appendix 4: Survey Questionnaire

Appendix 5: Survey Responses

Appendix 6: Technology Stack Selection Guidelines - Final

Appendix 1

Interview questionnaire - Pilot

Background Information:

The purpose of our study is to develop guidelines for organisations to use during the technology stack selection process. The developed guidelines aim to help organisations in their evaluation of different solutions and help organisations decide whether it is worth it to choose different technology stacks for different development teams. As previously discussed, an audio recording will be taken during the interview with the purpose of easing and improving the analysis of the collected data. The recorded material will only be used for the purpose of this study, will not be shared with anyone, and will be deleted as soon as the research is finished. You as an interviewee will always be anonymous in our report, and if we feel like we would like to quote from your interview, would you be okay with that?

Finally, we want to remind you that there are no right or wrong answers, and we encourage you to speak freely.

Background Questions:

1. What is your current job title and which tasks are involved in day-to-day business?
2. Approximately how many large-scale software projects have you been involved in?
3. How would you define a technology stack?
4. What are your experiences of technology stack decision-making?

Questions:

1. As an ("profession"), how would you say you contribute to technology stack changes and selection, and who are the others that contribute?

   Possible follow up:

   1.1. If having contributed: What do you think the impact of your contribution has been?

   1.2. If not having contributed: Is it because you have not had the chance, or it is because you are pleased with how things are running and think it's the optimal solution?

   1.2.1. If wasn´t given the chance to contribute: How do you think you could have contributed?

2. In your opinion, how much of the technology stack decision-making should the architect(s) and developers respectively be in charge of?
3. What do you think are the pros of having the developers in the team choose the technology stack without any or with little input from architects?
4. What do you think are the cons of having the developers in the team choose the technology stack without any or with little input from the architects?
5. What do you think are the pros of having the architects choose the technology stack without any or with little input from the developers?

6. What do you think the cons of having the architects choose the technology stack independently without any or with little input from the developers?
7. Current research states that software architects should take most responsibility for architectural decisions but with input from the developers. How do you feel this applies to the technology stack specifically?
8. Would you prefer having similar technology stacks for different teams, or having each team choose their own stack? Motivate why.

    Possible follow up:

    8.1. Would you think differently when choosing technology stacks for multiple development teams? If so, what would you do differently?
9. In your opinion, what are the most important aspects to have in mind when selecting a technology stack? Ranked from most important and dropping.

    Possible follow up:

    9.1. In your opinion, what is needed to enable developers to switch?
10. What are your experiences of developers switching development teams?

    Possible follow up:

    10.1. If yes: What are they?
11. Are there other advantages of using similar technology stacks than developers being able to switch teams?
12. How should the competence available in the close vicinity of the company affect technology stack decision-making (for example if the company is close to a school that teaches Java, how do you think that would affect the stack-decision process)?
13. What type of guidance would you appreciate in the technology stack selection process?

Interview questionnaire - Final

Background Information:

The purpose of our study is to develop guidelines for organisations to use during the technology stack selection process. The developed guidelines aim to help organisations in their evaluation of different solutions and help organisations decide whether it is worth it to choose different technology stacks for different development teams. As previously discussed, an audio recording will be taken during the interview with the purpose of easing and improving the analysis of the collected data. The recorded material will only be used for the purpose of this study, will not be shared with anyone, and will be deleted as soon as the research is finished. You as an interviewee will always be anonymous in our report, and if we feel like we would like to quote from your interview, would you be okay with that?

Some questions can require you to do some thinking, and there is no stress, feel free to take some time when needed. And if you think a question is confusing or there is something you do not understand, don´t hesitate to say so and we will try our best to clarify what we mean. Finally, we want to remind you that there are no right or wrong answers, and we encourage you to speak freely and take your experience into account. Meaning that you do not only have to see things from your Husqvarna perspective.

Background Questions:

1. What is your current job title and which tasks are involved in day-to-day business?
2. What roles have you had historically?
3. Approximately how many medium- to large-scale software projects have you been involved in?
4. How would you define a technology stack and has that definition evolved over the different roles you have had historically?
5. What are your experiences of technology stack decision-making?

Questions:

1. As an (“profession”), how would you say you contribute to technology stack changes and selection, and who are the others that contribute?

    Possible follow up:

    1.1. If having contributed: What do you think the impact of your contribution has been?

    1.2. If not having contributed: Is it because you have not had the chance, or it is because you are pleased with how things are running and think it’s the optimal solution?

        1.2.1. If wasn´t given the chance to contribute: How do you think you could have contributed?

2. In your opinion, how much of the technology stack decision-making should the architect(s) and developers respectively be in charge of?
3. What do you think are the pros of having the developers in the team choose the technology stack without any or with little input from architects?
4. What do you think are the cons of having the developers in the team choose the technology stack without any or with little input from the architects?
5. What do you think are the pros of having the architects choose the technology stack without any or with little input from the developers?
6. What do you think the cons of having the architects choose the technology stack without any or with little input from the developers?
7. Current research states that software architects should take most responsibility for architectural decisions but with input from the developers. How do you feel this applies to the technology stack specifically?
8. Would you prefer having similar technology stacks for different teams, or having each team choose their own stack? Motivate why.
9. In your opinion, what are the most important aspects to have in mind when selecting a technology stack? Ranked from most important and dropping.

    Possible follow up:

    9.1. Would you think differently when choosing technology stacks for multiple development teams? If so, what would you do differently?

10. What are your experiences of developers switching development teams?

    Possible follow up:

    10.1. In your opinion, what is needed to enable developers to switch?

    10.2. If you have experience of developers switching, how has it impacted the team and it´s development?

    10.3. What is your opinion on developers switching teams?

11. Are there other advantages of using similar technology stacks than developers being able to switch teams?

    Possible follow up:

    11.1. If yes: What are they?

12. How should the competence available in the close vicinity of the company affect technology stack decision-making (for example if the company is close to a school that teaches Java, how do you think that would affect the stack-decision process)?
13. What type of guidance would you appreciate in the technology stack selection process?

Swedish:

Syftet med vår studie är att ta fram riktlinjer för företag att använda när de väljer teknikstack. Riktlinjer syftar till att hjälpa företag i deras utvärdering av olika lösningar, och hjälpa dem bestämma om det är värt att välja olika teknikstackar för olika utvecklingsteam. Som tidigare nämnt kommer en ljudinspelning att tas under intervjun med syfte att underlätta och förbättra analysen av den insamlade datan. Det inspelade materialet kommer endast att användas för detta examensarbete, kommer inte delas med någon och det kommer raderas så fort arbetet är slutfört. Som intervjudeltagare kommer du alltid att vara anonym i vår rapport. Om vi känner att vi skulle vilja citera från denna intervju i vår rapport, hade du varit okej med det?

Vissa frågor kan kräva att du tänker efter lite, och det är ingen stress, ta din tid. Sist men inte minst vill vi påminna dig att det inte finns några rätta eller felaktiga svar, och vi vill uppmuntra dig att prata fritt.

Bakgrundsfrågor:

1. Vad är din nuvarande jobbtitel och vilka arbetsuppgifter innebär det på daglig basis?
2. Vad har du haft för tidigare jobbtitlar eller andra erfarenheter inom området?
3. Ungefär hur många stora/medelstora mjukvaruprojekt har du varit delaktig i?
4. Hur skulle du definiera en teknikstack och har den definitionen förändrats under din karriär?
5. Har du några erfarenheter av att välja teknikstack? Och i så fall hur skulle du beskriva dem?

Questions:

1. Som en (“Arbete”), hur skulle du säga att du är delaktig i val och ändring av teknikstack, och vilka är de andra som delaktiga?

    Eventuella följdfrågor:

    1.1. Ifall du har varit delaktig: Vad anser du att din delaktighet har haft för påverkan?

    1.2. Ifall du ej har varit delaktig: Är det på grund av att du ej fått möjlighet eller för att du är nöjd med hur processen fungerar och tror att det är optimalt?

        1.2.1. Ifall du ej fick chansen att vara delaktig: Hur tror du att du hade kunnat bidra?
2. Enligt dig, hur stor del av teknikstacksval skall arkitekter respektive utvecklare vara ansvariga för?
3. Vad tror du är fördelarna med att utvecklarna i teamet själva får välja teknikstack med ingen eller lite input från arkitekter?
4. Vad tror du är nackdelarna av att utvecklarna själva får välja utan eller med lite input från arkitekter?
5. Vad tror du är fördelarna med att arkitekter väljer teknikstacken utan eller med lite input från utvecklarna?

6. Vad tror du är nackdelarna med att arkitekter väljer teknikstacken utan eller med lite input från utvecklarna?
7. Tidigare forskning visar att arkitekter skall ta det stora ansvaret för arkitekturbeslut men ha en dialog med utvecklare. Hur väl tycker du det stämmer för teknikstacken specifik?
8. Föredrar du att ha liknande teknikstackar för olika utvecklingsteam, eller att varje team själva får välja sin stack? Motivera varför.
9. Enligt dig, vilka är de viktigaste aspekterna att ha I åtanke när man väljer teknikstack? Rangordnade från viktigast och nedåt.

    Eventuella följdfrågor:

    9.1. Hade du tänkt annorlunda om du skulle välja för flera utvecklingsteam? Och i så fall vad hade du gjort annorlunda?
10. Vad är dina erfarenheter av utvecklare som byter till andra utvecklingsteam?

    Eventuella följdfrågor:

    10.1. Vad tycker du krävs för att utvecklare ska kunna byta team?
    10.2. Ifall du har erfarenhet av utvecklare som byter team, hur har det påverkat teamen och deras utveckling?
    10.3. Vad tycker du personligen om att utvecklare byter team?
11. Finns det andra fördelar att användas liknande teknikstackar bortsett från att utvecklare byter team?

    Eventuella följdfrågor:

    11.1. Ifall det är så, vilka är fördelarna?
12. Hur bör kompetensen som finns i företagets närområde påverka valet av teknikstack? (Till exempel om företaget ligger nära ett universitet som lär ut ett Java, hur tror du att det skulle påverka valet av teknikstack?)
13. Vad för typ av riktlinjer eller hjälpmedel hade du uppskattat vid val av teknikstack?

Appendix 3

Technology Stack Selection Guidelines - Draft

Technology Stack Selection: Guidelines for organisations with multiple development teams – Draft

These guidelines have been created during a bachelor's thesis in Computer Engineering at Jönköping University. Nine qualitative interviews were performed with software architects, software developers and managers at Husqvarna and other organisations that perform in-house software development. The following guidelines contain our conclusions from the interview analysis.

The guidelines are meant to aid the process of technology stack selection. Section 1 mentions things that are important to know and consider before starting the selection process. Section 2 provides the essential steps of the selection process and a description of each step. Section 3 provides questions you can use to evaluate how well a certain technology stack would work for you. Section 4 and 5 mention different situations where it can be reasonable to select similar or different technology stacks for different development teams. For the remainder of these guidelines, a *product* can refer to either a product or a service.

1. Think of this before starting the technology stack selection process:

- Software architects should have responsibility for technology stack decision-making, but decisions should be taken together.
- All parties affected by the technology stack decisions should have a say in the process
- If your organisation already has an established technology stack, consider reading Sections 3 and 5 first and see if you need to make any changes in the technology stack or if you can continue with the established one.
- Generally speaking, there are more pros to choosing similar technology stacks compared to choosing different, however, it is very important not to limit the technology stack if a product has specific requirements.

2. Essential steps of technology stack selection:

- **Identify what needs and requirements the product will have for it to fulfil its purpose:** When choosing a technology stack, the most important step is to establish what requirements the product you are developing should have. The technology stack you choose must enable those requirements to be fulfilled.
- **Establish which people should be involved in technology stack discussions and decisions:** It is crucial that people from all affected parties are involved in discussion and decision-making. In smaller organisations, everyone affected can be involved, whilst you as a larger organisation might have to select representatives from different teams, roles etc.

- **Establish technologies for which you already have the competence in your organisation:** Does the organisation have competence in one or more technologies already? Can the same technologies be applied to the product in question?
- **Establish technologies for which you think you can find good resources in the future:** Are there technologies where you think you have a higher possibility of finding resources in the future? This is less crucial when working remote, but still important to consider.
- **Establish how important it will be for you to keep costs down?** It is important to consider costs from different perspectives and not only initial development costs. Depending on the product, the cost of operations and cost of maintenance can be as important if not even more important than development costs.

3. If you have established candidate technology stacks:

If you have established one technology stack which you think is an alternative, the questions below can be used to evaluate how well the technology stack would work. If you have established a couple of candidate technology stacks, the questions can be used to compare the different stacks.

- Is this technology stack suitable for the product?
- Do you have the required competence for this technology stack?
- Is it easy to find and recruit future resources for the technology stack? (More critical when working on-site)
- Will this technology stack facilitate effective development?
- Is there an established community that can provide support?
- Does the stack come with licensing fees or other costs? Are those costs an issue?

If you feel that the technology stack could be a good alternative after going through the questions, consider doing a Proof of Concept (POC) where you try out the technology stack for a small part of the product to properly establish your own opinion to be used as a base for the decision.

4. When is it reasonable to select similar technology stacks for multiple development teams?

- When you want to facilitate easy competence moves between development teams
    - Members switching between development teams most often provide positive effects for an organisation. When switching, the members bring with them other perspectives and experiences which can be shared within the new team.
- When you want to and have the possibility to develop common services (internal libraries, policies for security handling etc)
- When you want to facilitate cross-team knowledge exchange.

        - Knowledge exchange between teams can be powerful and provide large positive effects, even without members switching to a different team.
    - When you want to be able to reuse parts of an application and therefore lower the development costs and the future maintenance costs.

5. When is it reasonable to select different technology stacks for multiple development teams?

- When the result or user experience would be compromised by selecting similar technology stacks.
- When you want to optimise teams to deliver something with specific requirements or within a limited timeframe.

Survey Questionnaire

Bachelor´s thesis in Computer Engineering: Evaluation of draft Technology Stack Selection Guidelines

Goal of this survey:
To evaluate and refine the draft Technology Stack Selection Guidelines created during our Bachelor´s Thesis at Jönköping University.

The draft guidelines presented in this survey have been created from 9 interviews with software architects, software developers and managers in organisations that perform in-house software development.

The responses to this survey will be used to evaluate and refine the guidelines.

All questions are written in English, but if you prefer answering in Swedish that is accepted aswell.

* Required

Background information

What is your current role? *

- ◯ Software developer
- ◯ Software architect
- ◯ Software architect + developer
- ◯ Line Manager
- ◯ Other: _____

Do you work in an organization that has multiple development teams? *

- ◯ Yes
- ◯ No

Section 1: Think of this before starting the technology stack selection process:

Section 1 of the guidlines aim to give the person reading an overview of what is important to think about before starting the technology stack selection process. Please rate how useful you think this section is on a scale of 1 to 10? Do you think anything is missing or do you not agree with some of the bulletpoints? Please describe under "Section 1 feedback". Note that feedback is not necessary, but if given a low rating of usefulness, we appreciate an explanation of why.

1. Think of this before starting the technology stack selection process:
   - Software architects should have responsibility for technology stack decision-making, but decisions should be taken together.
   - All parties affected by the technology stack decisions should have a say in the process
   - If your organization already has an established technology stack, consider reading Sections 3 and 5 first and see if you need to make any changes in the technology stack or if you can continue with the established one.
   - Generally speaking, there are more pros to choosing similar technology stacks compared to choosing different, however, it is very important not to limit the technology stack if a product has specific requirements.

How useful is this section? *

	1	2	3	4	5	6	7	8	9	10	
Not useful at all	○	○	○	○	○	○	○	○	○	○	Useful

Section 1 feedback

Description of your section 1 answer. Do you think something is missing or is there something you do not agree with?

Your answer

Section 2: Essential steps of technology stack selection:

Section 2 of the guidelines presents the five most important steps during the technology stack selection process. The steps are not to be performed in a specific order. How useful do you think this section is on a scale of 1 to 10? Do you think anything is missing or do you not agree with some of the bulletpoints? Please describe under "Explain your answer". Note that an explanation is not nescessary, but if given a low relevance we appreciate an explanation of why.

Step 1

- **Identify what needs and requirements the product will have for it to fulfil its purpose:** When choosing a technology stack, the most important step is to establish what requirements the product you are developing should have. The technology stack you choose must enable those requirements to be fulfilled.

Rate how relevant Step 1 is? *

	1	2	3	4	5	6	7	8	9	10	
Not relevant at all	○	○	○	○	○	○	○	○	○	○	Very relevant

Explain your answer.

Is there something in step 1 that you do not agree with? Is there something that should be changed?

Your answer

Step 2

- **Establish which people should be involved in technology stack discussions and decisions:** It is crucial that people from all affected parties are involved in discussion and decision-making. In smaller organisations, everyone affected can be involved, whilst you as a larger organisation might have to select representatives from different teams, roles etc.

Rate how relevant step 2 is? *

	1	2	3	4	5	6	7	8	9	10	
Not relevant at all	○	○	○	○	○	○	○	○	○	○	Very relevant

Explain your answer.

Is there something in step 2 that you do not agree with? Is there something that should be changed?

Your answer

Step 3

- **Establish technologies for which you already have the competence in your organisation:** Does the organisation have competence in one or more technologies already? Can the same technologies be applied to the product in question?

Rate how relevant step 3 is? *

	1	2	3	4	5	6	7	8	9	10	
Not relevant at all	○	○	○	○	○	○	○	○	○	○	Very relevant

Explain your answer.

Is there something in step 3 that you do not agree with? Is there something that should be changed?

Your answer

Step 4

- **Establish technologies for which you think you can find good resources in the future:** Are there technologies where you think you have a higher possibility of finding resources in the future? This is less crucial when working remote, but still important to consider.

Rate how relevant step 4 is? *

	1	2	3	4	5	6	7	8	9	10	
Not relevant at all	○	○	○	○	○	○	○	○	○	○	Very relevant

Explain your answer.

Is there something in step 4 that you do not agree with? Is there something that should be changed?

Your answer

Step 5

- **Establish how important it will be for you to keep costs down?** It is important to consider costs from different perspectives and not only initial development costs. Depending on the product, the cost of operations and cost of maintenance can be as important if not even more important than development costs.

Rate how relevant step 5 is? *

	1	2	3	4	5	6	7	8	9	10	
Not relevant at all	○	○	○	○	○	○	○	○	○	○	Very relevant

Explain your answer.

Is there something in step 5 that you do not agree with? Is there something that should be changed?

Your answer

Section 3: If you have established candidate technology stacks:

Section 3 provides questions you can use to evaluate how well a technology stack would fit your organisation. How useful do you think this section is on a scale of 1 to 10? Do you think anything is missing or do you not agree with some of the bulletpoints? Please describe under "Section 3 feedback". Note that feedback is not necessary, but if given a low rating of usefulness, we appreciate an explanation of why.

3. If you have established candidate technology stacks:

If you have established one technology stack which you think is an alternative, the questions below can be used to evaluate how well the technology stack would work. If you have established a couple of candidate technology stacks, the questions can be used to compare the different stacks.

- Is this technology stack suitable for the product?
- Do you have the required competence for this technology stack?
- Is it easy to find and recruit future resources for the technology stack? (More critical when working on-site)
- Will this technology stack facilitate effective development?
- Is there an established community that can provide support?
- Does the stack come with licensing fees or other costs? Are those costs an issue?

If you feel that the technology stack could be a good alternative after going through the questions, consider doing a Proof of Concept (POC) where you try out the technology stack for a small part of the product to properly establish your own opinion to be used as a base for the decision.

How useful is this section? *

	1	2	3	4	5	6	7	8	9	10	
Not useful at all	○	○	○	○	○	○	○	○	○	○	Very useful

Section 3 feedback

Description of your section 3 answer. Do you think something is missing or is there something you do not agree with?

Your answer

Section 4: When is it reasonable to select similar technology stacks for multiple development teams?

Section 4 provides situations when it is reasonable to select similar technology stacks for multiple development teams. How useful do you think this section is on a scale of 1 to 10? Do you think anything is missing or do you not agree with some of the bulletpoints? Please describe under "Section 4 feedback". Note that feedback is not necessary, but if given a low rating of usefulness, we appreciate an explanation of why.

4. When is it reasonable to select similar technology stacks for multiple development teams?
   - When you want to facilitate easy competence moves between development teams
     - Members switching between development teams most often provide positive effects for an organization. When switching, the members bring with them other perspectives and experiences which can be shared within the new team.
   - When you want to and have the possibility to develop common services (internal libraries, policies for security handling etc)
   - When you want to facilitate cross-team knowledge exchange.
     - Knowledge exchange between teams can be powerful and provide large positive effects, even without members switching to a different team.
   - When you want to be able to reuse parts of an application and therefore lower the development costs and the future maintenance costs.

How useful is this section? *

	1	2	3	4	5	6	7	8	9	10	
Not useful at all	○	○	○	○	○	○	○	○	○	○	Very useful

Section 4 feedback

Description of your section 4 answer. Do you think something is missing or is there something you do not agree with?

Your answer

Section 5: When is it reasonable to select different technology stacks for multiple development teams?

Section 5 provides situations when it is reasonable to select different technology stacks for multiple development teams. How useful do you think this section is on a scale of 1 to 10? Do you think anything is missing or do you not agree with some of the bulletpoints? Please describe under "Section 5 feedback". Note that feedback is not necessary, but if given a low rating of usefulness, we appreciate an explanation of why.

5. When is it reasonable to select different technology stacks for multiple development teams?
   - When the result or user experience would be compromised by selecting similar technology stacks.
   - When you want to optimize teams to deliver something with specific requirements or within a limited timeframe.

How useful is this section? *

	1	2	3	4	5	6	7	8	9	10	
Not useful at all	○	○	○	○	○	○	○	○	○	○	Very useful

Section 5 feedback

Description of your section 5 answer. Do you think something is missing or is there something you do not agree with?

Your answer

Overall evaluation

Below are the entire draft Tecnology Stack Selection Guidelines. Please rate the overall relevance of the guidelines.

Page 1

Jönköping University
2022-04-29

Authors:
Hugo Martinsson
Victor Svanqvist

Technology Stack Selection: Guidelines for organizations with multiple development teams – Draft

These guidelines have been created during a bachelor's thesis in Computer Engineering at Jönköping University. Nine qualitative interviews were performed with software architects, software developers and managers at Husqvarna and other organisations that perform in-house software development. The following guidelines contain our conclusions from the interview analysis.

The guidelines are meant to aid the process of technology stack selection. Section 1 mentions things that are important to know and consider before starting the selection process. Section 2 provides the essential steps of the selection process and a description of each step. Section 3 provides questions you can use to evaluate how well a certain technology stack would work for you. Section 4 and 5 mention different situations where it can be reasonable to select similar or different technology stacks for different development teams. For the remainder of these guidelines, a *product* can refer to either a product or a service.

1. Think of this before starting the technology stack selection process:

- Software architects should have responsibility for technology stack decision-making, but decisions should be taken together.
- All parties affected by the technology stack decisions should have a say in the process
- If your organization already has an established technology stack, consider reading Sections 3 and 5 first and see if you need to make any changes in the technology stack or if you can continue with the established one.
- Generally speaking, there are more pros to choosing similar technology stacks compared to choosing different, however, it is very important not to limit the technology stack if a product has specific requirements.

2. Essential steps of technology stack selection:

- **Identify what needs and requirements the product will have for it to fulfil its purpose:** When choosing a technology stack, the most important step is to establish what requirements the product you are developing should have. The technology stack you choose must enable those requirements to be fulfilled.
- **Establish which people should be involved in technology stack discussions and decisions:** It is crucial that people from all affected parties are involved in discussion and decision-making. In smaller organisations, everyone affected can be involved, whilst you as a larger organisation might have to select representatives from different teams, roles etc.
- **Establish technologies for which you already have the competence in your organisation:** Does the organisation have competence in one or more technologies already? Can the same technologies be applied to the product in question?
- **Establish technologies for which you think you can find good resources in the future:** Are there technologies where you think you have a higher possibility of finding resources in the future? This is less crucial when working remote, but still important to consider.
- **Establish how important it will be for you to keep costs down?** It is important to consider costs from different perspectives and not only initial development costs. Depending on the product, the cost of operations and cost of maintenance can be as important if not even more important than development costs.

Page 2

Jönköping University
2022-04-29

Authors:
Hugo Martinsson
Victor Svanqvist

3. If you have established candidate technology stacks:

If you have established one technology stack which you think is an alternative, the questions below can be used to evaluate how well the technology stack would work. If you have established a couple of candidate technology stacks, the questions can be used to compare the different stacks.

- Is this technology stack suitable for the product?
- Do you have the required competence for this technology stack?
- Is it easy to find and recruit future resources for the technology stack? (More critical when working on-site)
- Will this technology stack facilitate effective development?
- Is there an established community that can provide support?
- Does the stack come with licensing fees or other costs? Are those costs an issue?

If you feel that the technology stack could be a good alternative after going through the questions, consider doing a Proof of Concept (POC) where you try out the technology stack for a small part of the product to properly establish your own opinion to be used as a base for the decision.

4. When is it reasonable to select similar technology stacks for multiple development teams?

- When you want to facilitate easy competence moves between development teams
  - Members switching between development teams most often provide positive effects for an organization. When switching, the members bring with them other perspectives and experiences which can be shared within the new team.
- When you want to and have the possibility to develop common services (internal libraries, policies for security handling etc)
- When you want to facilitate cross-team knowledge exchange.
  - Knowledge exchange between teams can be powerful and provide large positive effects, even without members switching to a different team.
- When you want to be able to reuse parts of an application and therefore lower the development costs and the future maintenance costs.

5. When is it reasonable to select different technology stacks for multiple development teams?

- When the result or user experience would be compromised by selecting similar technology stacks.
- When you want to optimize teams to deliver something with specific requirements or within a limited timeframe.

Rate the overall relevance of the guidelines? *

	1	2	3	4	5	6	7	8	9	10	
Not relevant at all	○	○	○	○	○	○	○	○	○	○	Very relevant

Overall feedback

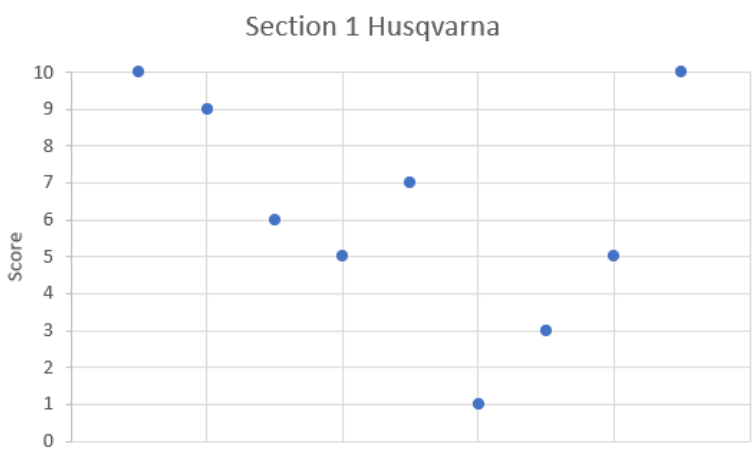
If you have overall feedback on our guidelines, please tell us here. If your feedback belongs to a specific section of the guidelines, please go back to the correct section.

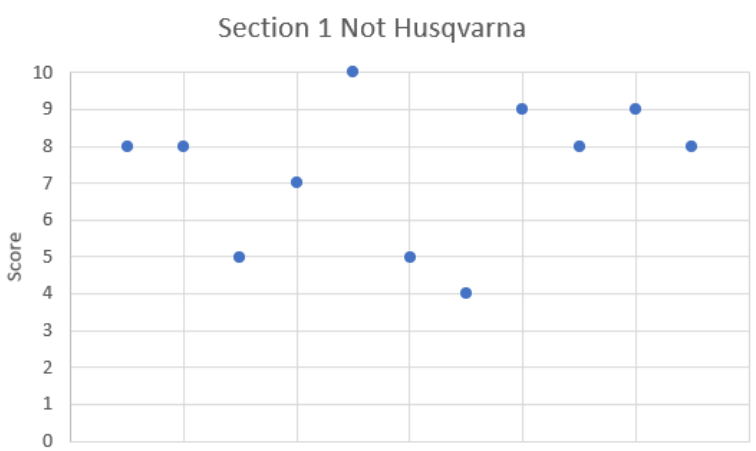
Your answer

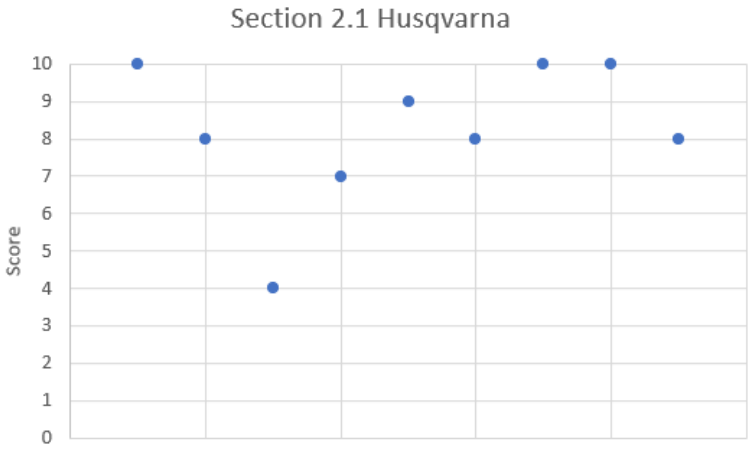
Appendix 5

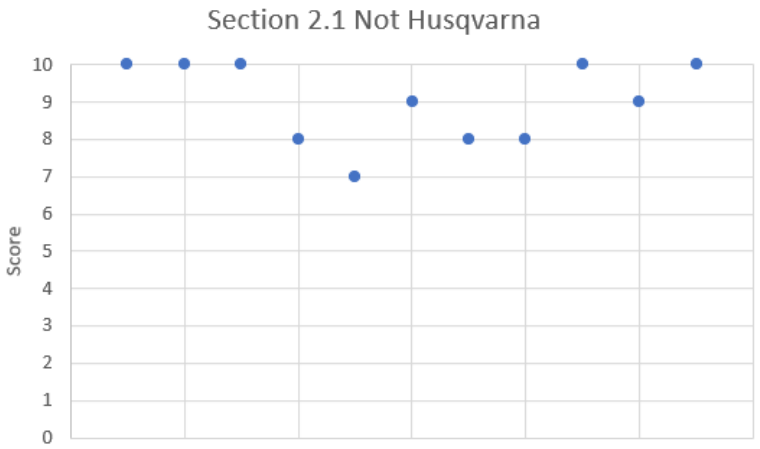
Survey Responses

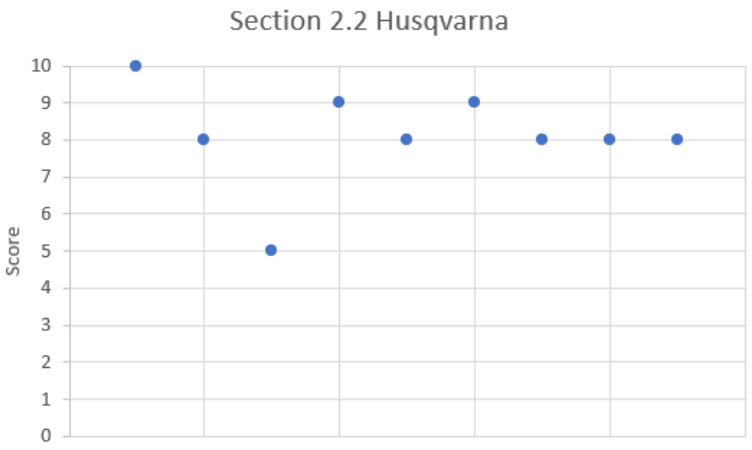
The following graphs show all scores of usefulness from each section of the survey. The responses are divided into two groups, responses from Husqvarna and responses from other organisations.

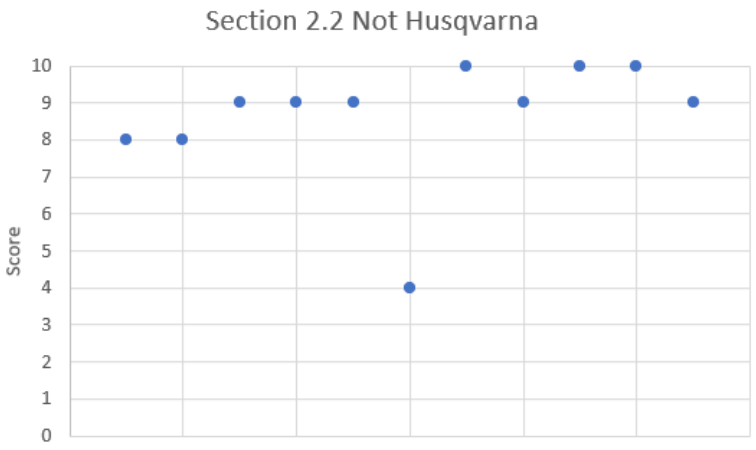


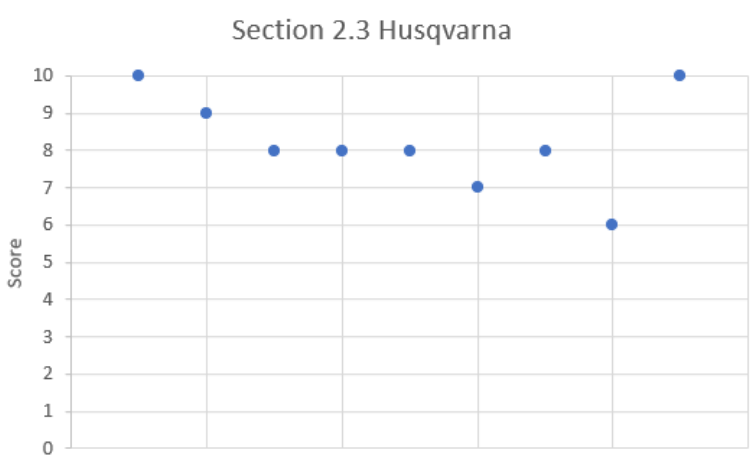


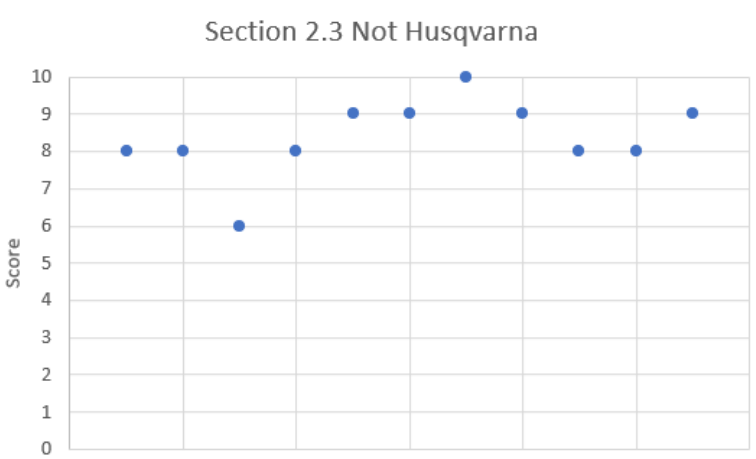


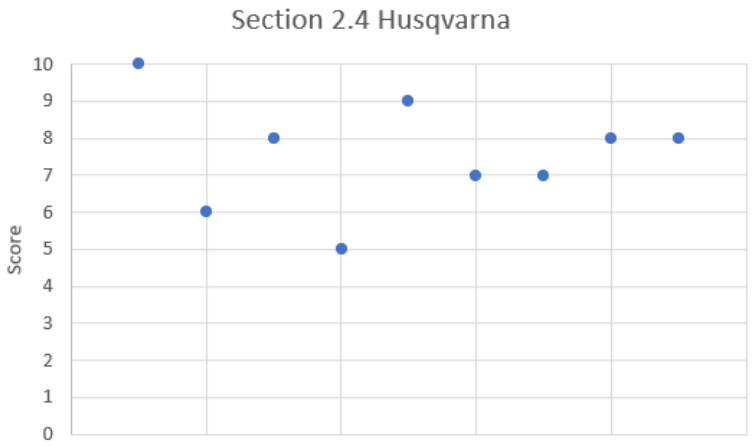


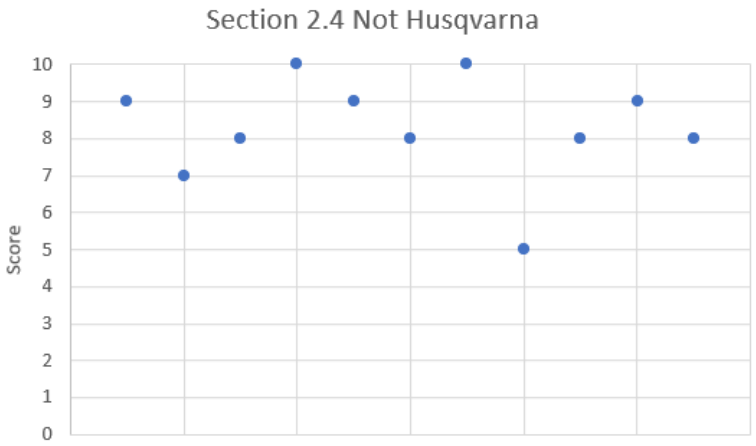


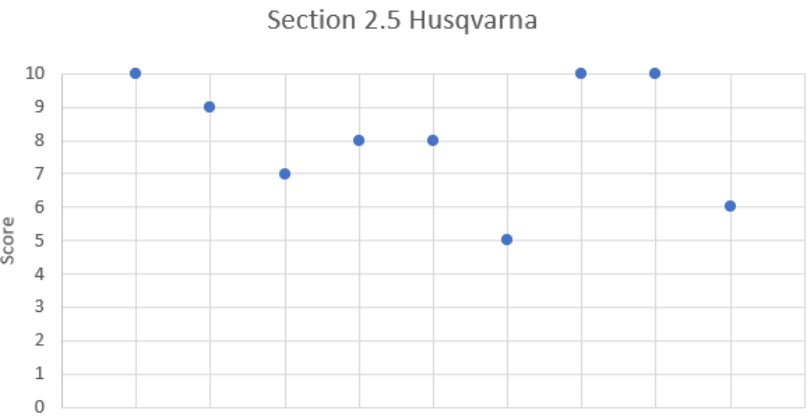


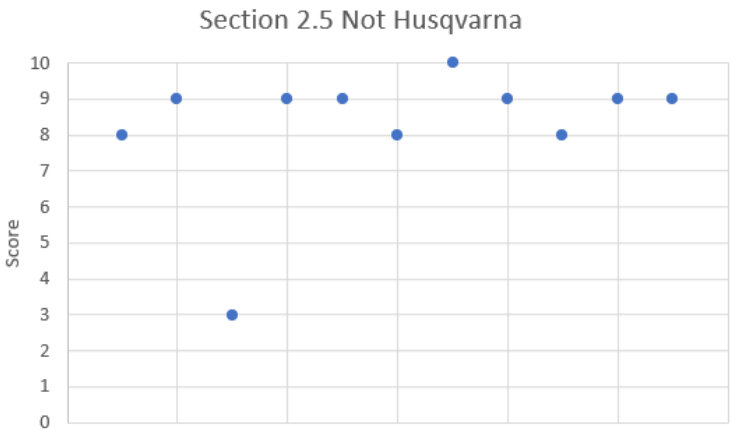

Section 2.3 Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

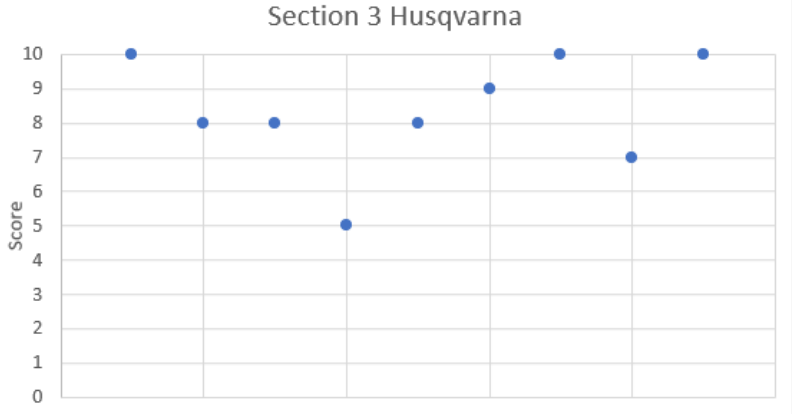

Section 2.3 Not Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

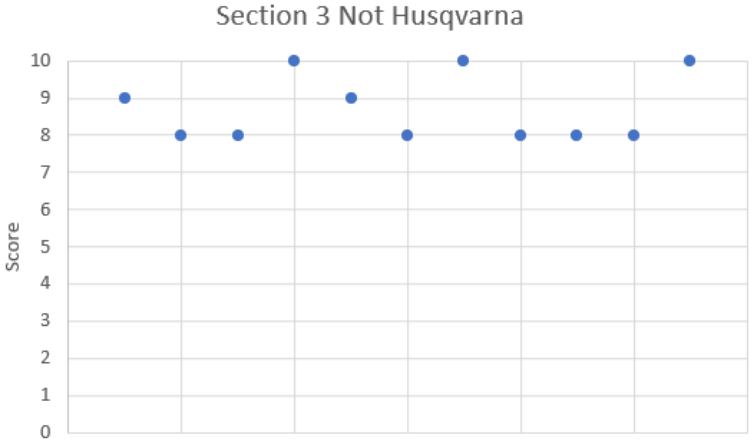

Section 2.4 Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

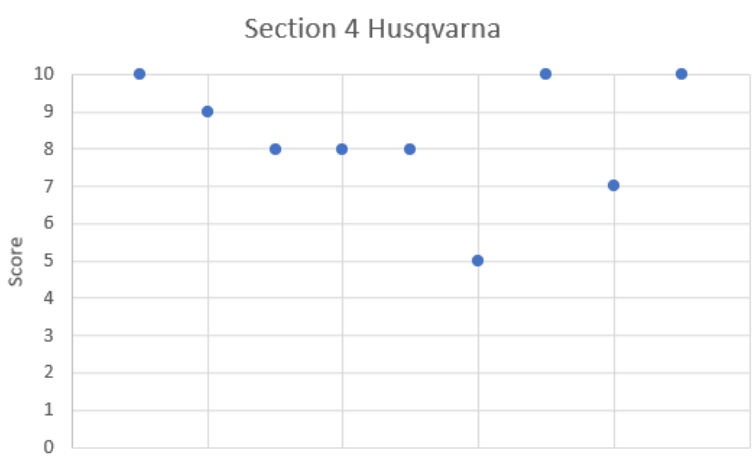

Section 2.4 Not Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

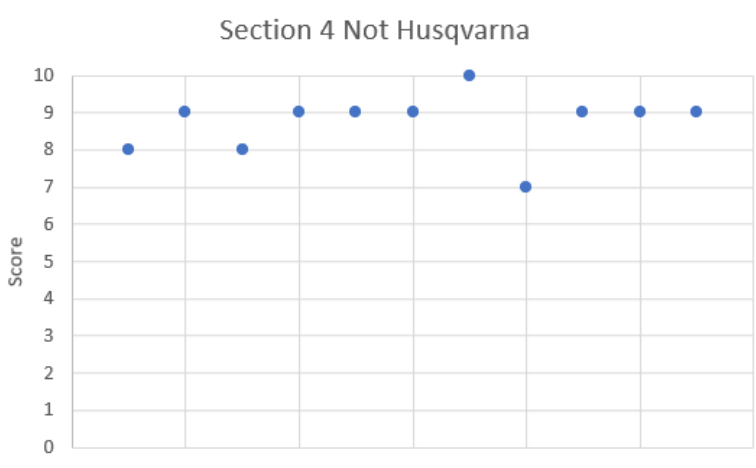

Section 2.5 Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

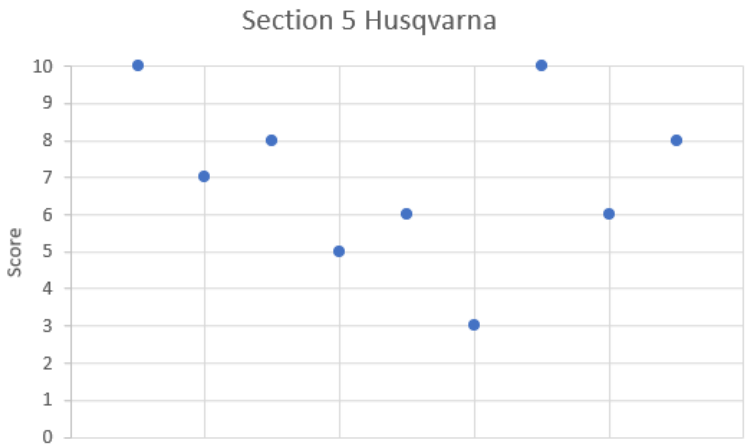

Section 2.5 Not Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

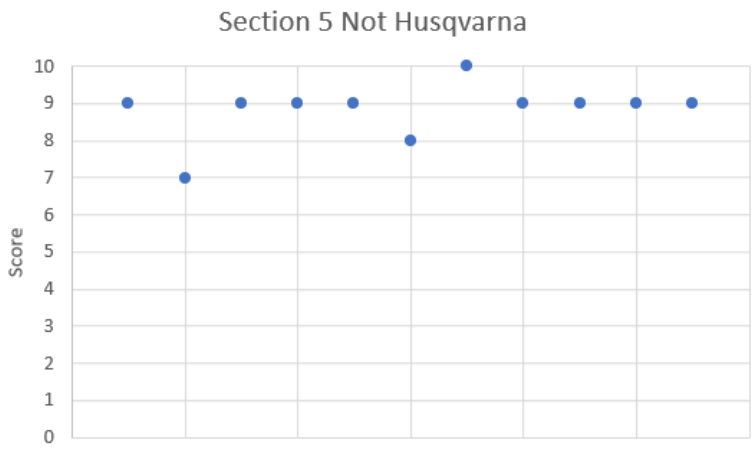

Section 3 Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

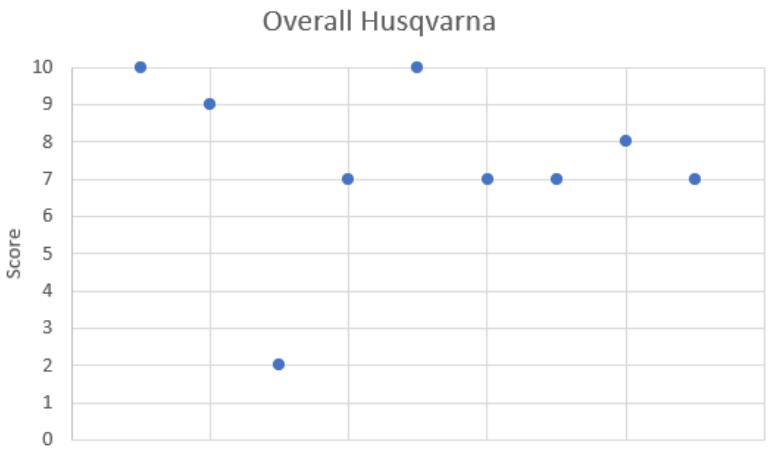

Section 3 Not Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

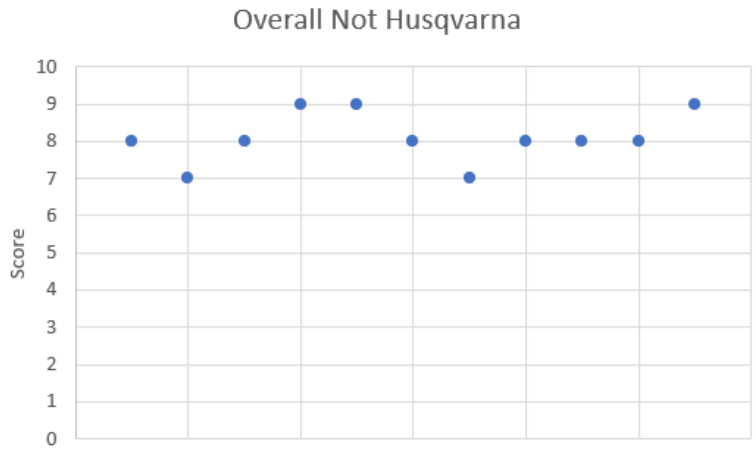

Section 4 Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0


Section 4 Not Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

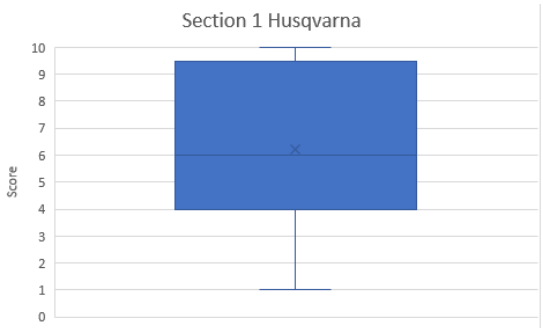

Section 5 Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

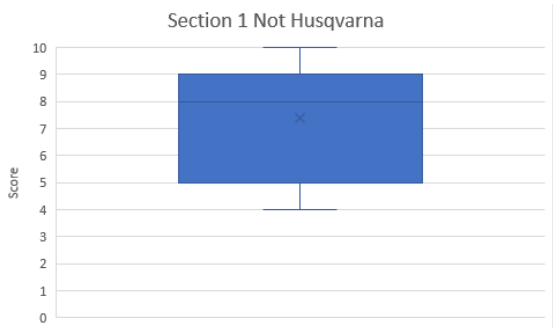

Section 5 Not Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

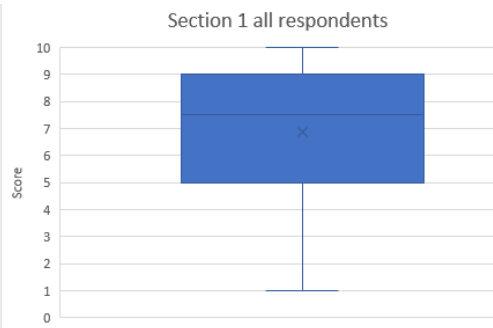

Overall Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

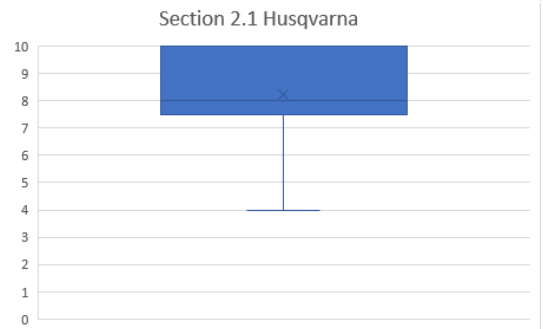

Overall Not Husqvarna
Score
10
9
8
7
6
5
4
3
2
1
0

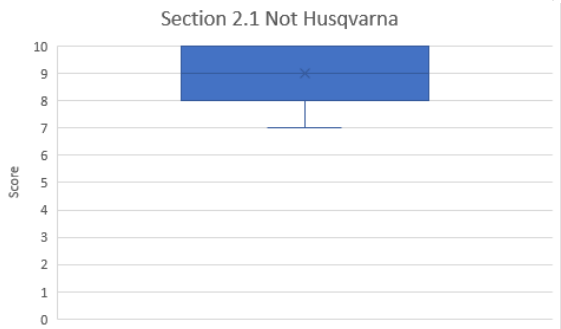
The following graphs show the usefulness scores from all sections of the survey with box graphs. The responses are divided into three groups: responses from Husqvarna, responses from the other organisations and all responses combined. The box graphs show the maximum and minimum value, the mean (the cross), the median (the line) and +/- one standard deviation (the blue box).

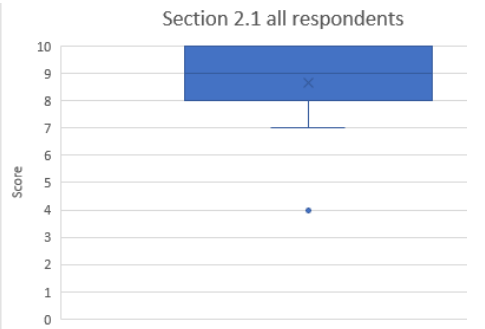


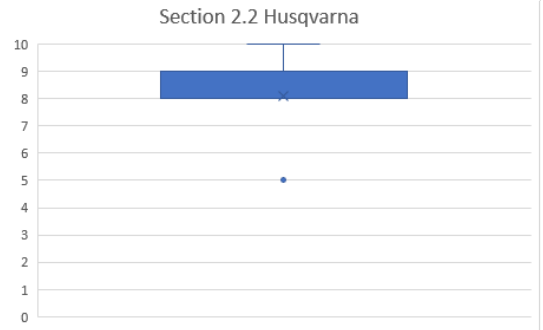


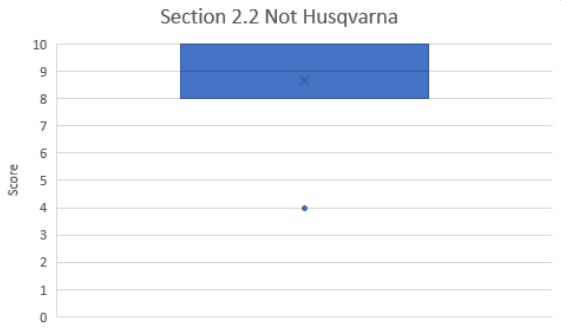


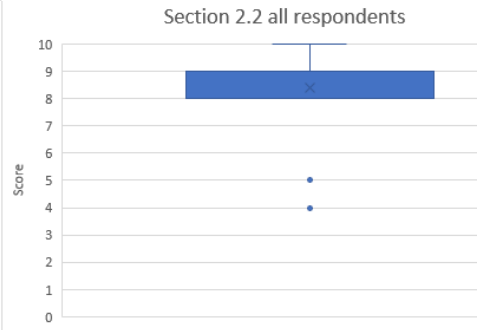


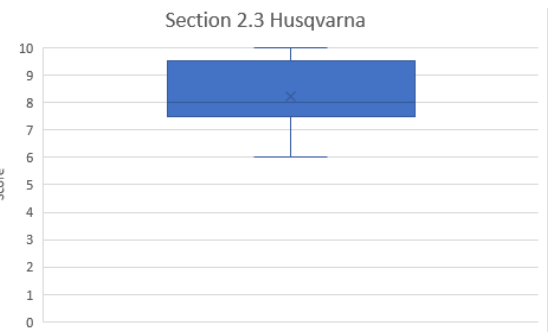


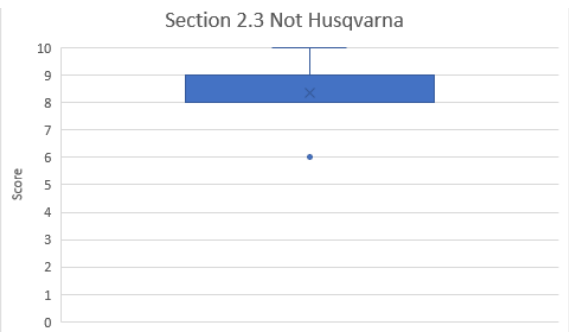


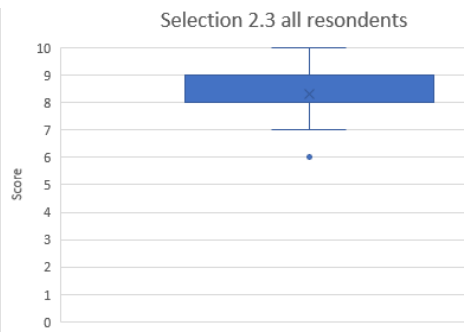


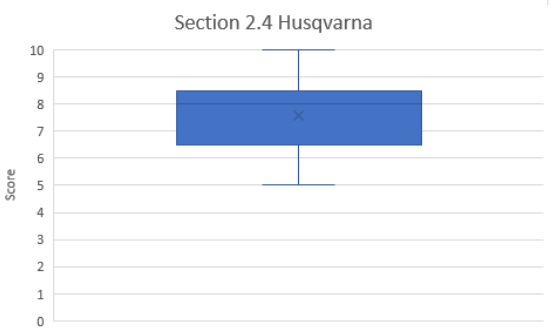


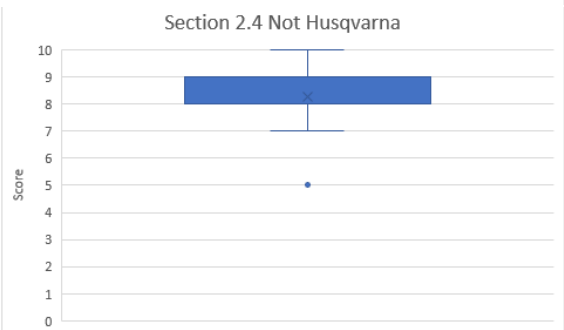


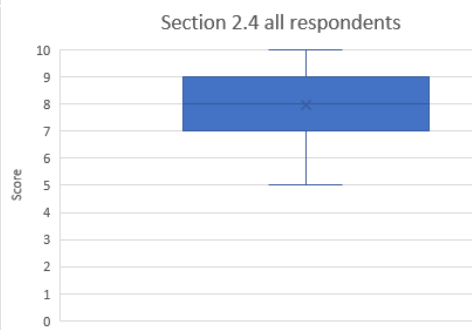


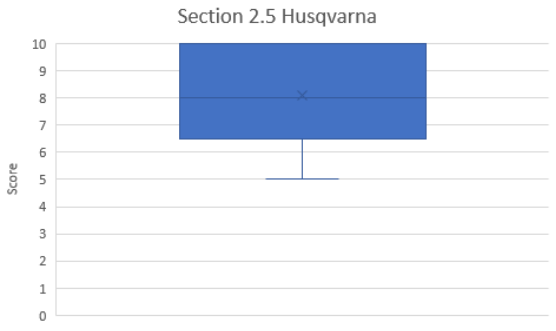


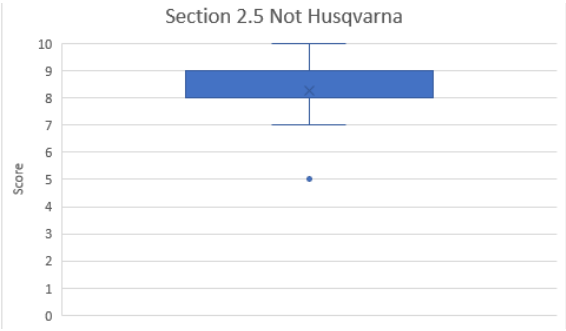


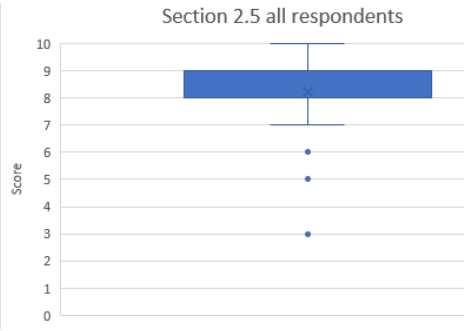


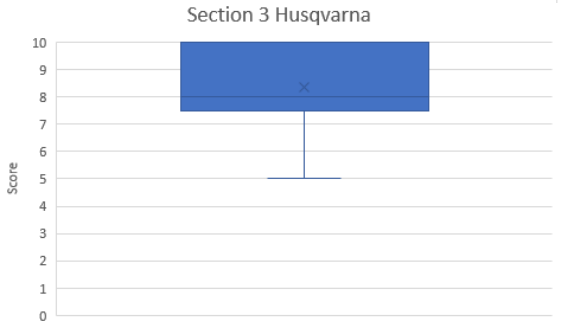


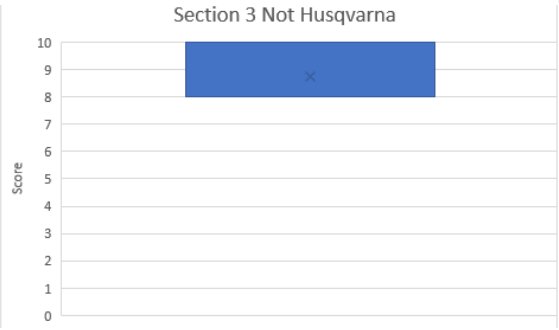


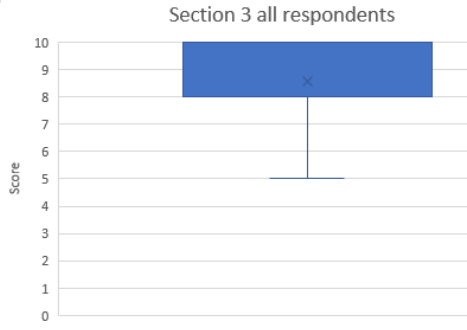

Section 2.5 Husqvarna
Score

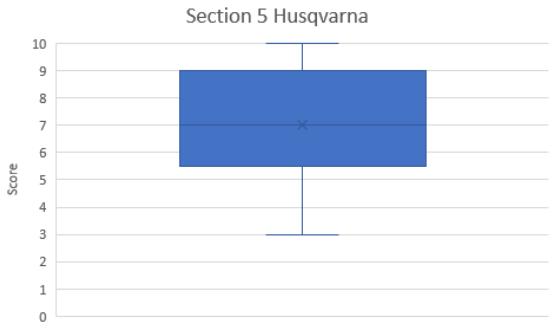

Section 2.5 Not Husqvarna
Score

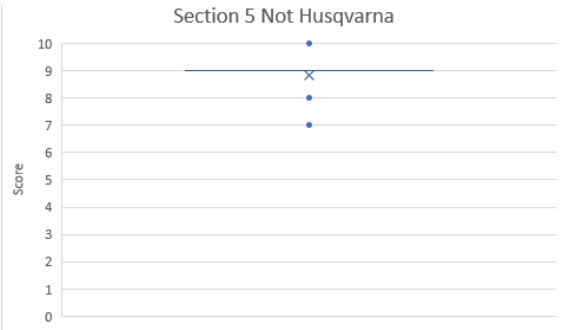

Section 2.5 all respondents
Score

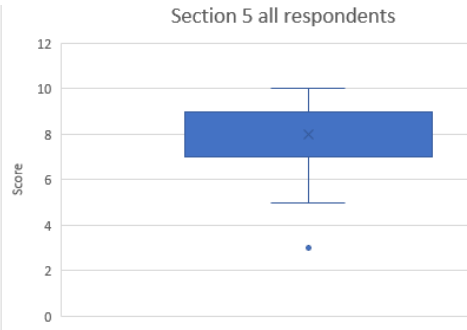

Section 3 Husqvarna
Score

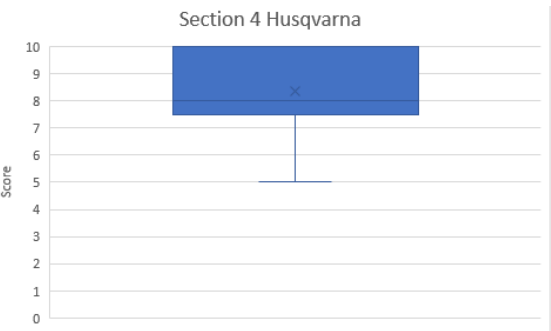

Section 3 Not Husqvarna
Score

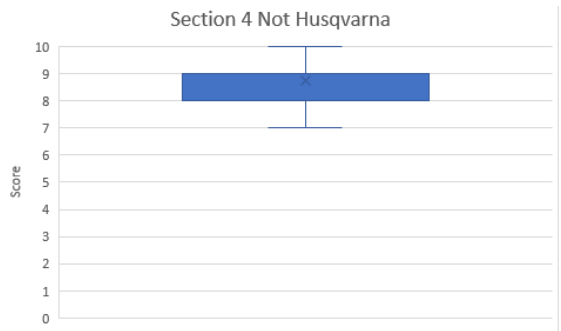

Section 3 all respondents
Score

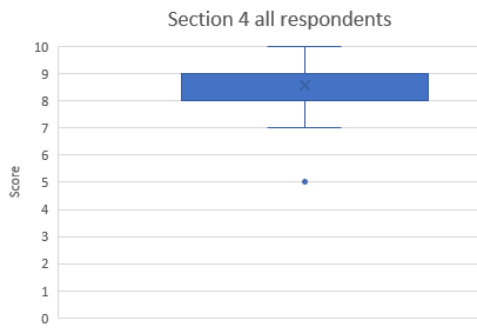

Section 5 Husqvarna
Score

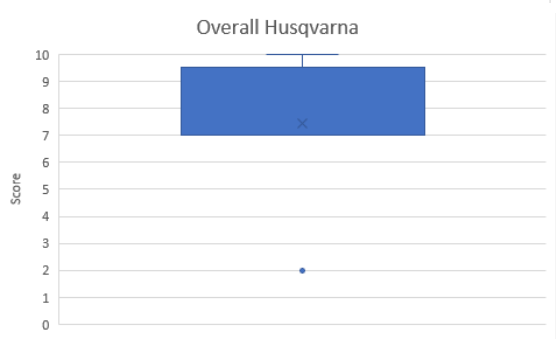

Section 5 Not Husqvarna
Score

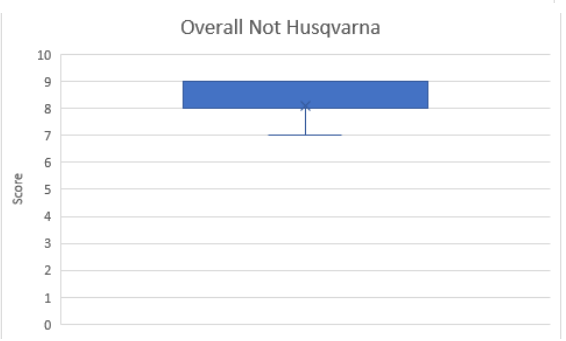

Section 5 all respondents
Score

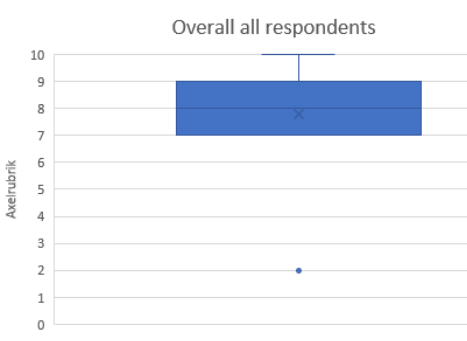

Section 4 Husqvarna
Score


Section 4 Not Husqvarna
Score


Section 4 all respondents
Score


Overall Husqvarna
Score


Overall Not Husqvarna
Score


Overall all respondents
Axelrubrik

Appendix 6

Technology Stack Selection Guidelines - Final

Technology Stack Selection: Guidelines for organisations with multiple development teams

These guidelines were created during a bachelor's thesis in Computer Engineering at Jönköping University. Nine qualitative interviews were performed with software architects, software developers and managers at organisations that perform in-house software development. The following guidelines contain the conclusions from the thesis work Technology stack selection: Guidelines for organisations with multiple development teams.

The guidelines serve to aid the process of technology stack selection, mainly for new software projects. But the guidelines can also be used when considering changing parts of or an entire technology stack during a project. Section 1 mentions important things to know and consider before starting the selection process. Section 2 provides the essential steps of the selection process and a description of each step. Section 3 provides questions you can use to evaluate how well a certain technology stack would work for you. Finally, sections 4 and 5 mention situations where selecting similar or different technology stacks for different development teams can be reasonable.

For the remainder of these guidelines, a *product* can refer to either a product or a service.

1. Think of this before starting the technology stack selection process:

- Software architects should have responsibility for technology stack decision-making, but decisions should be taken together with all affected stakeholders.
- All stakeholders affected by the technology stack decisions should have a say in the process.
- If your organisation already has an established technology stack, consider reading Sections 3 and 5 first and see if you need to make any changes in the technology stack or if you can continue with the established one.
- Generally speaking, there are more pros to choosing similar technology stacks for multiple teams in an organisation compared to choosing different, however, it is essential not to limit the technology stack if a product has specific requirements.
- All requirements of a product might not be known from the start, hence it is important to design the technology stack for future change.

2. Essential steps of technology stack selection:

- **Identify what needs and requirements the product will have for it to fulfil its purpose:** When choosing a technology stack, the most important step is to establish what requirements the product you are developing should satisfy.

The technology stack you select must enable those requirements to be fulfilled.

- **Establish which stakeholders should be involved in technology stack discussions and decisions:** People from all affected parties must be involved in discussion and decision-making. In smaller organisations, perhaps everyone affected can be involved. However, a larger organisation might have to select representatives from different teams, roles, etc. With too many people involved, it might get harder to decide.
- **Establish technologies for which you already have the competence in your organisation:** Does the organisation have competence in one or more technologies already? Can the same technologies be applied to the product in question?
- **Establish technologies for which you think you can find good resources in the future:** Are there technologies where you think you have a higher possibility of finding resources in the future? This is less crucial when working remotely due to the increased number of possible resources, but still important to consider.
- **Establish how important it will be for you to keep costs down?** It is important to consider costs from different perspectives and not only initial development costs. Depending on the product, the cost of operations and cost of maintenance can be as important, if not even more important, than development costs.

3. If you have established candidate technology stacks:

If you have identified one technology stack which you think is suitable, the questions below can be used to evaluate how well the technology stack would work.

If you have identified alternative candidate technology stacks, the questions can be used to compare the different stacks.

- Is this technology stack suitable for the product?
- Do you have the required competence for this technology stack?
- Is it easy to find and recruit future resources for the technology stack? (More critical when working on-site since the number of resources decreases to the ones available in the nearby area)
- Will this technology stack facilitate effective development now and in the future?
- Is there an established community that can provide support?
- Does the technology stack enable good scalability and maintainability in the future?
- Does the technology stack enable resource flexibility and knowledge exchange with other teams?
- Does the stack come with licensing fees or other costs? Are those costs an issue?

If you feel that the technology stack could be a good alternative after going through the questions, consider doing a Proof of Concept (POC) where you try out the technology

stack for a small part of the product to properly establish your own opinion to be used as a base for the decision.

4. **When is it reasonable to select similar technology stacks for multiple development teams?**
    - When you want to facilitate easy competence moves between development teams
        - Members switching between development teams most often provide positive effects for an organisation. When switching, the members bring other perspectives and experiences that can be shared within the new team.
    - When you want to have the possibility to develop common services (internal libraries, policies for security handling, etc.)
    - When you want to facilitate cross-team knowledge exchange.
        - Knowledge exchange between teams can be powerful and provide significant positive effects, even without members switching to a different team.
    - When you want to be able to reuse parts of an application and therefore lower the development costs and the future maintenance costs.
5. **When is it reasonable to select different technology stacks for multiple development teams?**

- When the result or user experience would be compromised by selecting similar technology stacks.
- When you want to optimise teams to deliver something with specific requirements or within a limited timeframe.