Sunniva Block and Hanne Olssen

Addressing Gender Inequality in Software Development Processes

Master's thesis in Computer Science
Supervisor: Letizia Jaccheri
Co-supervisor: J. David Paton-Romero
June 2022

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Sunniva Block and Hanne Olssen

Addressing Gender Inequality in Software Development Processes

Master's thesis in Computer Science
Supervisor: Letizia Jaccheri
Co-supervisor: J. David Paton-Romero
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Context: Retention problems among women in the Information Technology (IT) industry are prominent and complex, and it is essential to review whether the current processes aid women or persist as a barrier to their participation in software development. As long as women remain a minority in software development processes, the resulting repercussions on the software products must be reviewed.

Objective: The objective of this Master's thesis is to map out what actions can be taken to improve software development processes with unequal gender distribution and their resulting products. The Research Questions (RQs) set to scope the investigation are:

> **RQ1:** What actions can be taken to improve software development processes with unequal gender distribution?
>
> > **RQ1.1:** How can the artifacts resulting from software development processes with unequal gender distribution be optimized?
> >
> > **RQ1.2:** How can the challenges women meet in software development processes today be diminished?

Method: The study consisted of 12 semi-structured interviews to gather data from experienced representatives from the IT industry. These representatives consisted of both developers and project leaders. The interviews were transcribed before being further examined through thematic analysis, identifying recurring themes and ideas.

Result: The results indicate a division in the field on what actions to implement to decrease retention problems among women. However, the input on previously discovered barriers for women gave the basis for creating a set of modules that can be implemented into software development processes. These modules are suggested as add-ons to supplement already existing software development methodologies.

Conclusion: Our study suggests a set of modules the IT industry can adapt as add-ons to existing software development methodologies to make the processes more inclusive and increase the diversity of the resulting solutions. We hope these add-ons will work as a starting point and motivation for further research, proposals, and case studies into the fields of gender equality and software engineering.

Keywords: Gender, Gender Equality, Software Development Processes, Software Development Methodologies, Software Engineering, Systematic Literature Review, Case Study

Sammendrag

Kontekst: Turnover er et komplekst og fremtredende problem blant kvinner i IT-bransjen. Det er viktig å undersøke om dagens utviklingsprosesser er tilrettelagt for kvinner eller om de hindrer kvinners innflytelse i programvareutvikling. Så lenge kvinner er en minoritet i programvareutviklingsprosesser, så må konsekvensene dette har på den resulterende programvaren evalueres.

Formål: Formålet med denne masteroppgaven er å kartlegge hvilke tiltak som kan iverksettes for å forbedre programvareutviklingsprosesser med ujevn kjønnsbalanse og den resulterende programvaren. Forskningsspørsmålene satt for denne studien er:

> **RQ1:** Hvilke tiltak kan iverksettes for å forbedre programvareutviklingsprosesser med ujevn kjønnsbalanse?
>
> > **RQ1.1:** Hvordan kan de resulterende sluttproduktene fra programvare utviklingsprosesser med ujevn kjønnsbalanse optimaliseres?
> >
> > **RQ1.2:** Hvordan kan utfordringene kvinner møter i utviklingsprosesser i dag reduseres?

Metode: Studien besto av 12 semistrukturerte intervjuer, gjennomført med mål om å samle data fra representanter med relevant erfaring fra IT-bransjen. Disse representantene besto av både utviklere og teamledere. Alle intervjuene ble transkribert, før de videre ble undersøkt gjennom en tematisk analyse som identifiserte gjentakende tema og ideer.

Resultater: Resultatene fra denne casestudien indikerer splittelse innen fagfeltet når det kommer til hvilke tiltak som kan implementeres for å redusere turnover blant kvinner i IT-bransjen. Tilbakemeldinger på tidligere identifiserte barrierer kvinner møter i utviklingsprosesser, har imidlertid dannet grunnlaget for et sett med moduler som kan implementeres i programvareutviklingsprosesser. Disse modulene er foreslått som tillegg (add-ons) for å supplementere allerede eksisterende metodologier innen programvareutvikling.

Konklusjon: Vår studie foreslår et sett med moduler som skal kunne fungere som tillegg til eksisterende metodologier innen programvareutvikling i IT-bransjen. Dette med mål om å gjøre prosessene mer inkluderende og tilrettelegge for mer mangfoldige løsninger. Vi håper med dette at tilleggene vil kunne motivere og danne grunnlaget for videre forskning og studier innen programvareutvikling og ujevn kjønnsbalanse.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) as part of the course TDT4900 Computer Science, Master's thesis. The work has been performed at the Department of Computer Science, NTNU, Trondheim, under Professor Letizia Jaccheri as the principal supervisor and Doctor J. David Paton-Romero as co-supervisor.

The Master's thesis builds on a Systematic Mapping Study (SMS) from the course TDT4501 Computer Science, Specialization Project. In addition, *Gender Equality in Information Technology Processes: A Systematic Mapping Study* is a revised version of the specialization project authored by J. David Patón-Romero, Sunniva Block, Claudia Ayala, and Letizia Jaccheri that has been submitted to the ACM International Conference on Information Technology for Social Good (GoodIT 2022).

Acknowledgment

We want to thank everyone who has taken part in this project and helped us carry out the research for this Master's thesis. First of all, we would like to thank Professor Letizia Jaccheri for being our supervisor and providing us with the guidance and knowledge needed throughout this entire process. Secondly, our co-supervisor, Doctor J. David Paton-Romero, has been of great assistance. Thirdly, we want to express our gratitude to our fellow master students, PhD candidates, and others participating in our master community/group for tips, guidance, and inspiration. Our weekly meetings have been an excellent arena for communication and collaboration. We would further like to thank the Department of Computer Science at NTNU for providing us with the equipment, tools, and facilities needed to conduct our research.

This work has been partially supported by COST Action EUGAIN CA19122 - European Network for Gender Balance in Informatics and by NFR 295920 IDUN.

Last but not least, we would like to thank all of the interview subjects who have participated; without the views and experiences you shared, this thesis would not have been possible.

NTNU, June 2022
Sunniva Block and Hanne Olssen

Table of Contents

Abstract i

Sammendrag ii

Preface iii

Acknowledgment iv

Table of Contents v

List of Figures viii

List of Tables ix

Abbreviations x

1 Introduction **1**

1.1 Motivation . . . 2

1.2 Research Questions . . . 3

1.3 Research Scope . . . 4

1.4 Research Protocol . . . 4

2 Background **6**

2.1 SDG 5 Gender Equality . . . 6

2.2 Women in Software Development . . . 7

2.2.1 Retention problems . . . 8

2.2.2 Women Affecting Software Products . . . 11

2.3 Software Development Methodologies . . . 12

2.3.1 Traditional Approaches . . . 16

2.3.2 Agile Approaches . . . 19

2.3.3 DevOps . . . 28

2.3.4 Choice of Methodologies and Approaches . . . 30

3 Literature Review **35**

3.1 Motivation . . . 35

3.2 Research Method . . . 35

3.2.1 Research Questions . . . 36

3.2.2 Search Strategy 36
3.2.3 Selection Criteria 36
3.2.4 Data Extraction Strategy 37
3.2.5 Threats to Validity 37
3.3 Results 37
3.3.1 RQ1: What kind of studies exist on IT processes and gender equality? 39
3.3.2 RQ2: What gender equality targets are addressed by IT processes? . 39
3.3.3 RQ3: What are the main challenges to achieve gender equality in IT processes? 40
3.3.4 RQ4: What are the best practices established to address gender equality in IT processes? 41
3.4 Future work 41

4 Research Method **42**
4.1 Research Questions 42
4.2 Interviews 43
4.2.1 Selection Criteria 44
4.2.2 Personal data 45
4.2.3 Developing the Interview Guide 46
4.3 The Interview Process 50
4.4 Data Analysis 51
4.4.1 Transcription 51
4.4.2 Thematic Analysis 53
4.4.3 NVivo 55
4.4.4 Quality Assurance 56

5 Results **58**
5.1 Interview Subjects 58
5.2 Companies 59
5.2.1 Female Presence 60
5.2.2 Events or Activities for Women 60
5.3 Challenges for Women in Software Development 61
5.3.1 Gender Bias 61
5.3.2 Imposter Syndrome 63
5.3.3 Code Review 64
5.3.4 Gender Stereotypes 65
5.4 Adapting Software Development Processes 66
5.4.1 Womens Approach to Software Development 66
5.4.2 Suggested Improvements 67

5.5 Womens Impact on Software Artifacts . . . 68
5.5.1 Effect on Resulting Software Artifacts . . . 68
5.5.2 Ensuring Women's Impact on Software Artifacts . . . 69
5.6 Additional Input . . . 69

6 Discussion 71
6.1 General Discussion . . . 71
6.2 Gender Bias . . . 73
6.3 Imposter Syndrome . . . 74
6.4 Code Review . . . 75
6.5 Gender Stereotypes . . . 77
6.6 Women's Impact on Software Development . . . 79
6.7 Additional Proposals . . . 81
6.8 Threats to Validity . . . 82
6.8.1 Construct Validity . . . 83
6.8.2 Internal Validity . . . 83
6.8.3 External Validity . . . 83
6.8.4 Reliability . . . 84
6.8.5 On the Topic of Gender . . . 84
6.9 Implications for Research and Practice . . . 85

7 Conclusion 86

Bibliography a

Appendices g
A Information Letter . . . h
B Interview Guide . . . k

List of Figures

2.1 Division of men and women in the Norwegian IT industry from 2008 to 2021 [55] . . . 8
2.2 The design of the Waterfall Methodology [22] . . . 17
2.3 The Scrum Life Cycle [81] . . . 24
2.4 The XP Life Cycle and its practices [49] . . . 26

3.1 Results from data extraction of RQ1 fetched from [13] . . . 39
3.2 Results from data extraction of RQ2 fetched from [13] . . . 40
3.3 Results from data extraction of RQ3 fetched from [13] . . . 41

4.1 A screenshot from the NVivo workbench . . . 56

List of Tables

2.1 The components and techniques of PRINCE2 [12] 19

2.2 The software development methodologies reported used in projects sorted by percentage [78] 32

2.3 The software development approaches used in projects sorted by percentage [78] 33

3.1 The data extraction results for RQ1 (Type), RQ2 (Targets), RQ3 (Challenges), and RQ4 (Best practices) [13] 38

4.1 The thematic topics used for coding interviews 55

5.1 The general information on interview subjects. (PID=Person ID, CID=Company ID) 59

5.2 The presence of females and female developers in interview subjects' companies 60

6.1 The challenges and topics with suggested modules and their corresponding goals 73

6.2 The change for question 6 in the interview guide 76

Abbreviations

EIGE The European Institute for Gender Equality. 47, 48

IT Information Technology. i, vi, viii, 1–4, 6–12, 15, 19, 30, 32, 35–37, 39–43, 45–48, 50, 54, 58–66, 69, 71, 72, 74, 76, 77, 79, 81–87

NCWIT National Center for Women Information Technology. 9

NSD Norwegian Agency for Shared Services in Education and Research. 45, 46

NTNU Norwegian University of Science and Technology. iii, iv, 45, 55

RQ Research Question. i, 1–3, 5, 6, 36, 42, 43, 46–49, 58, 71

SDG Sustainable Development Goal. v, 2, 3, 6, 35, 36, 39, 41, 43

SMS Systematic Mapping Study. iii, 35, 42, 43, 75, 76, 83, 86

STEM Science, Technology, Engineering and Mathematics. 6, 8–10

UN United Nations. 6, 35

Chapter 1

Introduction

The retention problems among women in the IT industry are prominent and complex [35], [36], [70]. When the IT industry now attempts to tackle the issue of increasing women in IT, it is important to review whether the current processes aid women or persist as a barrier to their participation in software development. In the investigation of the retention problems among women in IT, it is imminent to consider and assess the software development processes that structure a developer's work environment, as the development methods used today were defined for an industry characterized by its persistent male majority.

This thesis's main objective is to study gender equality in software development processes. Further, the aim is to map out what actions can be taken to improve software development processes with unequal gender distribution, both optimizing the resulting artifacts and diminishing the challenges women encounter during these processes. A case study was conducted using semi-structured interviews to gather data from 12 representatives from the IT industry to reach the goal. The results give insight into the IT industry's disputed thoughts on gender equality and how to achieve it. The results implicate a set of add-ons for the software development process that will contribute to achieving gender equality in the field.

The following sections will briefly introduce the thesis and the research conducted. The present chapter aims to ground the motivation for the thesis and provide an overview of the structure. The first section will present the thesis's motivation, thus giving a brief introduction to different aspects that encouraged the authors and inspired this project. The designed RQs are outlined in the second section to introduce the purpose of the thesis before the brief presentation of the research scope. Finally, the research protocol is showcased in the last section to present the method and structure used throughout this project.

1.1 Motivation

The minority of women in the Norwegian IT industry is prominent. In 2019 only 15% of software developers were women [77]. The gender equality paradox is a term used to explain the gender imbalance of an industry such as IT in an otherwise prominent country concerning gender equality [15]. Brenda et al. propose persisting gender stereotypes and norms as sources for the division [15]. As long as these sources persist in the industry, they are a part of the narrative for the high retention. Of the women entering the IT industry, 50% leave it within 12 years [35], this despite that 4 out of 5 women are highly satisfied with their work [77] [36]. The need for labor force in the IT industry will be impossible to meet without recruiting and persisting women in the industry.

To gain insight on what barriers might exist for women in IT a literature review was conducted in the fall of 2021 by one of the authors [13]. The literature review introduced several challenges for women's participation in IT, and further research showed that there are no processes in place that directly aspire to eliminate the existing barriers for women in software development processes [13]. Further work from in the literature study implied that addressing these challenges for gender equality can aid Sustainable Development Goal (SDG) 5.5c as described in section 2.1 [13]. Adjacent to the established prospect of including more women in software development processes is what implication the lack of women can have on the products developed and how this can be solved while women remain a minority.

This thesis will investigate current existing software development methodologies in relation to the challenges identified for women in the IT industry and what implications the lack of female developers can have for software products.

The thematic for this Master's thesis concerning the industry's retention problems and gender inequality in software development processes is highly relevant to the authors. Mainly because they both are female and about to venture into software development and the IT industry themselves, this relevance is considered one of the most significant motivational factors for researching the topic. In addition to their natural connection to the issues addressed, their interest was further captivated through studying and uncovering a series of challenges that women face in IT processes.

The following section will introduce the RQs designed for this project. It will also include a more detailed grounding and motivation for formulating and including the specific sub-research questions.

1.2 Research Questions

The research conducted in this Master's thesis aims to investigate gender inequality in software development processes. The lack of women in the IT industry might have led to predicaments in the existing software development processes. We want to develop a set of modules that can be integrated into these development processes, working as add-ons to current software development methodologies. The plugins are implemented to decrease and compensate for the drawbacks of unequal gender distribution. To be able to develop these plugins, further research had to be done on the topic of software development processes with unequal gender distribution and its potential challenges, hence motivated the formulation of the following RQs:

> **RQ1:** What actions can be taken to improve software development processes with unequal gender distribution?
>
> > **RQ1.1:** How can the artifacts resulting from software development processes with unequal gender distribution be optimized?
> >
> > **RQ1.2:** How can the challenges women meet in software development processes today be diminished?

In the literature study conducted prior to this thesis, the goal was to investigate the state of the art on gender equality in software development processes [13]. The study identified a series of challenges for achieving gender equality, such as gender bias, stereotypes, and code acceptance, and that there was no process directed specifically at IT that addressed these challenges as further described in section 3.3 [13]. Further, the study implied that addressing these challenges in software development processes would aid SDG 5. These findings are what inspired the formulation of RQ1.2.

Further, research shows that the current retention problems in the IT industry can cause a deficiency in female input in software development for several years to come. There are several examples of software development processes where the resulting artifact has been inadequate due to the lack of female involvement. One example of this is the Apple Health App [29], where Apple launched an app set to cover all health metrics that did not take the female reproductive system into account. This is what inspired the formulation of RQ1.1.

1.3 Research Scope

The scope of this Master's thesis comprehends the boundaries of the project. The scope was reduced to the Norwegian IT industry to contribute to the state of the art with research robust enough to create an overview with the limited time and resources of a Master's thesis. The study was conducted using semi-structured interviews to gain insight into the IT industry. Therefore, it was considered necessary that the representatives had gained some experience from the IT industry themselves, whether they were still working in the field or not. Further, it was regarded as essential that they had some prior experience with software development methodologies, which made for seeking out representatives with roles directly involved in software development processes. Likewise, the pool of candidates included representatives of distinct genders and from different companies in the research to gain a broad insight and provide diversity. Finally, the representatives could not previously have a personal or close relation to any of the authors, as it might affect their answers.

These requirements resulted in inclusion and exclusion criteria to select representatives from the industry. The representatives had to fulfill all criteria to participate in the research. The inclusion criteria are listed below:

Inclusion Criteria

i1 The interviewee must have at least five years of experience in a software development team. Either as a project leader or developer.

i2 The interviewee must have experience in using one or more development methodologies.

i3 Minimum half of the interview subjects must be female.

i4 The total group of interviewees must represent at least six different companies.

Exclusion Criteria

e1 The interviewee can not have a close connection to one or both of the authors.

1.4 Research Protocol

The chosen method of this Master's thesis is based on a strategy presented by Oates [56], as case studies. Interviews were selected as the data generation method for the case studies to utilize the natural setting around the case. The interviews were performed in a semi-structured manner, and the data collected from the interviews are considered qualitative. An interview guide, including the interview questions, was developed based on the stated RQs, following the five phases suggested by Kallio et al. [42].

All interviews were conducted and transcribed in Norwegian and recorded digitally through Teams, except for one interview conducted in English. A thematic analysis was performed on the transcribed interviews, following the six steps presented by Kiger and Varpio [44]. The thematic analysis was conducted using NVivo and was implemented from the second step. The transcriptions were coded into seven main themes with 18 sub-codes. The themes were created by uniting both a deductive and inductive approach, meaning that the themes are based on both preliminary theories and new themes revealed by the data itself.

Chapter 2

Background

The following chapter will introduce the theory and background literature relevant to the scope of this Master's thesis and its stated RQs. The content provides the reader with knowledge on the topics addressed throughout the thesis and makes ground for the results and further discussion.

First, the Sustainable Development Goal (SDG) on Gender Equality will be presented along with its given targets. Secondly, there will be a general introduction to software development methodologies and what concepts they bring into software development processes. This section also mentions the two main approaches to software development and lists some of the methodologies that are most commonly used in the IT industry today.

2.1 SDG 5 Gender Equality

The United Nations (UN) introduced 17 goals for a more sustainable future in 2015 [54]. The fifth goal is to promote gender equality in all public spheres and covers all aspects, from diminishing gender-based violence to promoting women in leadership. In the past decades, progress has been made, but the gender difference is still prominent, i.e., only 28% of managerial positions are held by women as of 2019 [54]. As women in the past decades have gradually entered the industry to be men's equals, women are still paid significantly less and spend, on average, three times as many hours on domestic housecare as men [28]. The UN's 2020 gender demographics [28] show that gender disparities endure in women's career choices. Only 20% of IT professionals are women, and 35% of Science, Technology, Engineering and Mathematics (STEM) students are women.

SDG Targets

In order to achieve gender equality the UN implemented nine targets that address the current obstacles for gender equality. The targets of SDG 5 fetched from United Nations [61] are:

5.1 *End all forms of discrimination against all women and girls everywhere.*

5.2 *Eliminate all forms of violence against all women and girls in the public and private spheres, including trafficking and sexual and other types of exploitation.*

5.3 *Eliminate all harmful practices, such as child, early and forced marriage and female genital mutilation*

5.4 *Recognize and value unpaid care and domestic work through the provision of public services, infrastructure and social protection policies and the promotion of shared responsibility within the household and the family as nationally appropriate.*

5.5 *Ensure women's full and effective participation and equal opportunities for leadership at all levels of decision making in political, economic and public life.*

5.6 *Ensure universal access to sexual and reproductive health and reproductive rights as agreed in accordance with the Programme of Action of the International Conference on Population and Development and the Beijing Platform for Action and the outcome documents of their review conferences*

5.5a *Undertake reforms to give women equal rights to economic resources, as well as access to ownership and control over land and other forms of property, financial services, inheritance and natural resources, in accordance with national laws.*

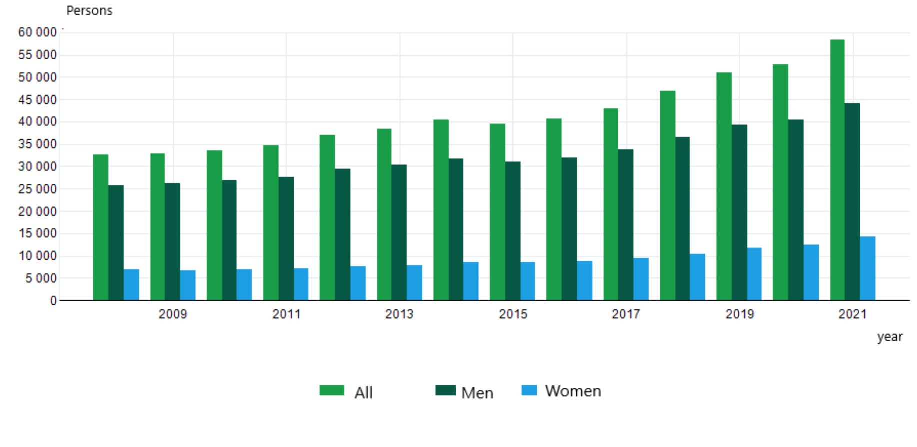
5.5b *Enhance the use of enabling technology, in particular information and communications technology, to promote the empowerment of women.*

5.5c *Adopt and strengthen sound policies and enforceable legislation for the promotion of gender equality and the empowerment of all women and girls at all levels.*

For gender equality in software development processes target 5.5, 5.1, 5.4 and 5.5c are the most relevant as further described in the findings in Figure 3.2.

2.2 Women in Software Development

Research performed by Kantar TNS [77] in 2018 shows that 28% of employees in IT companies in Norway are women. This is a significant raise from 2013 when women in IT were at a meager 19% [77]. Kantar TNS precise that regardless of the increase, women are still to a large degree underrepresented in IT. The report further states that women account for 15% of employees working with IT development or programming, and 22% of IT leaders are women [77]. A similar study from the US in 2019 [26] shows that women represented 26% of the employees in computing and mathematical occupations, with female computer programmers at approximately 20%. 79% of women in Norway say that they are highly satisfied with their work [77] compared to 80% of women in the US science, engineering, and technology industry [36]. This sets the US and Norway at a comparable level.



Source: ssb.no

Figure 2.1: Division of men and women in the Norwegian IT industry from 2008 to 2021 [55]

Figure 2.1 shows the total of all employees, total men and total of women working in the IT industry in Norway from 2008 to 2021 [55]. In 13 years, the industry has almost doubled, going from approximately 32 000 to 58 000 employees. However, the percentage of women has had a slow increase in comparison, having only slightly more than doubled[55].

2.2.1 Retention problems

For engineering occupations, Singh et al. [70] point out retention problems as a bottleneck for achieving gender balance and further argues the importance of identifying the aspects that make women leave. Their takeaway from current turnover theories highlights job satisfaction and organizational commitment as important for persisting in a job. Sing et al.'s [70] results emphasize the positive impact self-efficacy and outcome expectations have on an engineer's job attitude. The study implies that among women, lack of support and environmental barriers are essential factors that can lead to higher turnover, thus highlighting the importance of an encouraging work environment.

Glass et al. [35] performed an investigation looking into the differences between women in STEM where most were in IT or engineering, in comparison to women in other professional occupations such as managers and administrators. The results show that 50% of the women in STEM usually leave STEM within 12 years. In comparison, 20% of women in other professional occupations exit their fields. Further, the study shows that the women who leave STEM usually move to another field rather than out of the labor force entirely. A study by Coqual shows that 32% of women in the US, 22% in Brazil, and 30% of women in China state that they are considering leaving their field in science, technology, or engineering within a year [36]. The annual study by National Center for Women Information Technology (NCWIT) also highlights that women leave the Technology industry faster than other fields and suggests that women who leave IT often move to the public sector leaving the private sector behind [26].

An article written by Seternes [69] explores the retention tendencies among women in a Norwegian company. In the company, 66% of the women in tech positions have moved to other non-tech roles in the past three years in comparison to 13% of the men. Seternes [69] further questions what makes more women than men shift their focus away from tech and indicates three possible solutions. Firstly, the soft skills women often possess are not valued enough in technical positions. Secondly, women often spend more time on office housework, such as arranging social events or performing non-profitable tasks. Finally, it is presented that women can be too critical of their technical competence [69].

The Athena Factor 2.0 report by Coqual [36] identifies four main challenges for women in science, engineering, and technology. The hostile macho culture in the industry is the first obstacle that can prove to exclude women and impose bias. The second challenge is isolation which is presented through the lack of role models for women and the exclusion from social networks. Further, women get less effective sponsors, which hinders their career advancement. Finally, women have difficulties with performing in leadership positions [36]. Another interesting statistic from Coqual is that only 30% of development suggestions presented by women get approval compared to 37% of men's [36]. A study conducted by Accenture points out that leaders are more than twice as likely to think that their company support women to prosper in technology roles as the women in technology roles themselves experience the support [1].

Fouad and Santana [32] conclude that more research is needed to identify the aspects causing current retention problems and boost the persistence of women in STEM.

Imposter Syndrome

Imposter syndrome causes a person to feel like a fraud due to distrust in own abilities despite high competence [18]. Chrousos and Mentis [18] write that imposter syndrome is more widespread among women, high achievers as well as minorities, where many women in IT fit into all mentioned categories. Acknowledging the mentioned vulnerable groups for imposter syndrome also acknowledges the diversity threat it poses. A way to combat imposter syndrome is through encouragement to acknowledge own high performance, advising to have acquirable prospects, as well as open dialogues on the topic within the organization [18].

Gender Bias

Gender bias can present as implicit or explicit and lead to a hostile environment as well as marginalizing behavior [40]. Moss et al. [53] present reluctant aspirations and attitudes in addition to less sense of belonging among women experiencing gender bias in STEM. Further, the study shows that gender bias affecting women's welfare, equal treatment, and career progress contributes to STEM gender gaps. One way to tackle gender bias is through diversity training. Jackson et al. found that diversity training could reduce the occurrence of implicit bias among men and thus create a more welcoming environment for women[40].

Gender Stereotypes

Smeding [71] presents stereotypes as a way to link a group of people with a set of characteristics or a concept, exemplifying that men can be associated with STEM and thus reducing women's affiliation to STEM. Further, Cundiff and Vescio [20] present that stereotypes can consist of explanations for justifying a gender being unsuccessful in a field such as IT. When gender stereotypes contribute to explaining positions as appropriate for the group holding them, they assist in persisting gender inequality [20].

In many of the most equitable and advanced countries, gender segregation in some industries is still prominent, which has led to the introduction of the term gender-equality paradox [15]. The reason behind the paradox has been explained as gender preferences. Still, Brenda et al. [15] presents that it might be due to gender stereotypes and norms about suitable career choices and math aptitudes that are still introduced to children at a young age.

Code Acceptance

Terrell et al. [75] investigated differences in pull request acceptance for women compared to men in open source. Overall the study showed that women got their code more frequently accepted than men, but only when their names could not identify that they were women. Some possible explanations for this are that women are assessed by higher standards that make them deliver better seen in addition to gender bias [75].

Gender Quotas

Gender quotas are a concept implemented to positively promote gender-balanced involvement and inclusion in a field [82]. In Norway, gender quotas are implemented in the admission for some university studies where the gender imbalance is prominent and gives the minority gender one to two extra points for their grade point average. For many IT engineering degrees such as computer science, women get two additional ”gender quota” points [59].

Zehnter and Kirchler [82] studied students’ associations with women’s quotas and men’s quotas. They highlight that, in general, women’s quotas are in place to aid an inferior group compared to men’s quotas. However, they also present the paradox that support for women is perceived as less legitimate and justified than similar support for men. The study showed that there were more negative associations with women quotas; in particular, they were perceived as ”counterproductive, derogatory, and unfair, whereas they perceived men quotas as beneficial and fair” [82, p. 1]. Introducing gender quotas can lead to rapid improvement in gender imbalance in a field. Still, Wroblewski [80] emphasize that gender quotas do not initiate cultural changes or remove known obstacles to women’s careers.

2.2.2 Women Affecting Software Products

Software products define how we communicate, work and conduct our everyday life. It is a large part of the economic and social future. Albusays et al. [3] describe that a lack of diversity can negatively affect products by imposing constraints on the users and highlights the value of striving for diversity in the development of products to create inclusive products.

In the 1940s, women pioneered computer programming, but by the 1960s, men claimed the field and replaced the women already present in the industry [37]. The common idea is that when men take over a domain, it is because it becomes more complex; however, in this case, men entered computing as it improved and became easier to use [37]. As men entered the field, the status of computer programming as an occupation also rose drastically.

The Apple Health App was announced back in 2014 and introduced to monitor vital health aspects of the human body [29]. Apple’s Senior Vice President of Software Engineering stated that it would help you ”monitor all of your metrics that you’re most interested in” [29]. Some of the metrics included were tracking sodium intake or height if you are still growing. However, a neglected metric was women’s reproductive system. The lack of attention to gender differences in the development of the app put focus on the general absence of women, and thereby their influence in software development [29].

Tatman [74] explored how gender bias affects voice recognition on YouTube. The results showed that women were affected by lower recognition accuracy, making it more difficult for women to use the feature. Voice recognition learns through the datasets it is exposed to, and the results indicate a lack of gender diversity in the datasets used for testing [74].

In a hackathon hosted by TechCrunch in New York, a group of three teenage girls outcompeeted 700 coders and designers with their app that tests children for ADHD [76]. The app was developed in 24 hours and exceedingly accelerated the detection of ADHD, which usually takes at least six months to diagnose. Their considerable achievement resulted in positive feedback, but they also received several comments that they were the diversity pick and had only won because they were young women of color [76].

A study performed by Huang and Yuan [39] show how usability is perceived differently by women and men on tourism websites. The study highlights the gender difference in perception, preferences, and expectations in information systems. When evaluating usability in a website, men value efficiency, freedom, and user control, while women value help, documentation, security, privacy, and pleasing design [39].

The full picture of how the lack of women affects the software industry and the products available will be hard to comprehend fully. However, Judy [41] draws out some important aspects. He points out that the industry needs a labor force and that replacing experienced women lost to turnover is a very costly problem. Further, women stand for more than half of online purchases, and other sectors have grown markets from adapting products and services to fit women. "Failure to adequately understand and address the needs of women may harm them in ways we cannot anticipate. Software systems are a fundamentally complex product that can produce unintended consequences." [41, p. 5281]

2.3 Software Development Methodologies

In the industry of IT and software development, the developers strive to deliver software of high quality to their customers. A non-planned and non-systematic approach to software development will result in a software product with a high cost and low quality if applied to software with more complex requirements [50]. In other words: How the software development process evolves impacts the resulting quality of the delivered end product and necessitates the need for a structured and planned approach to software development.

The process surrounding the development of software artifacts can be divided into different stages. All the distinct stages relate to one another but contribute to the overall process by providing its output or result [22]. According to a study on software development methodologies done by Despa [22], there is a general agreement by the software development community that there are seven stages that form the foundation of every software development project. These seven stages and their characteristics are:

- **Research:** At this stage, the product owner, the development team, and the manager of the project get together to exchange information about and discuss the project. The product owner is responsible for formulating a set of requirements to be delivered to the project manager. Once the manager receives the requirements, they need to be evaluated and further passed on to the development team to address more technical specifications and requirements. Hence, the project manager needs to oversee the project from two perspectives, both a business-related and a technical one. The development team receives the technical requirements from the manager and evaluates these from a technical perspective. The team should also do further research on how to fulfill these requirements through investigating what technologies and tools, among other things, are suitable for the given project [22] [64].

- **Planning:** The goal of the planning stage is to set everything to start the development of a software product and involves outlining the project's scope by defining the desired overall flow of the product. This flow is further broken down into smaller parts of required functionalities, making it more comprehensive and easy to work with and making ground for deciding upon and designing a database structure. Considering the resulting functionality and database structure, the project manager and the development team should agree on the technologies they want to develop the software product. Another critical aspect of the planning stage is for the project manager to figure out what methodology is best suited for the given project [22].

- **Design:** In the design stage, the overall structure and layout for the software product are created. Depending on the nature of the product being developed, this can involve everything from a more high-level graphic design focused on functionality to a more detail-oriented and artistic creation. This stage is considered a crucial stage in the development process, as the configuration is presented to the product owner, providing the opportunity for feedback before the development of the product starts. The feedback potentially results in new requirements from the project owner that need to be supplemented with preliminary research and planning. The design stage often tends to overlap with its previous and following stages, namely the planning stage and the development stage [22] [64].

- **Development:** The development stage is focused on the development of the software product, which usually implies writing the code. The step typically starts with setting up the project's development and testing environments. The code is initially written in the development environment before uploading it to the testing environment. These environments should be synchronized using the same synchronization protocol. The development team's primary responsibility throughout this stage is developing new functionality. Simultaneously they are responsible for continually debugging and conducting tests to ensure their code has no bugs or errors before uploading it to the testing environment. The development team should also implement coding standards and comment on their code, to make it less complicated for other developers to understand. The project manager is responsible for monitoring and keeping track of the team's progress according to the predetermined plan. This responsibility also involves keeping the product owner updated on the overall advancement in the project [22] [64].

- **Testing:** This stage is quite self-explanatory, as this is the stage where the central part of all the testing takes place. The phase involves the identification of potential errors in the software product, followed by fixing these errors. The uncovered errors can be either programming or design errors. If the system behaves unexpectedly or inconsistently, not compared to plan, this would typically be due to a programming error. Programming errors also comprise faults concerning the usability and security of the software product. An example of this is vulnerabilities in the product that potential attackers could easily exploit. On the other hand, design errors originate from the planning stage and are usually not as easily fixed as the programming errors. These errors occur from inconsistency between the implementation by the development team and the original requirements from the product owner. Due to this, the design errors are easier to identify through collaboration and discussions with the product owner [22].

- **Setup:** The setup stage is where the software is installed in the live environment and enables utilization of the developed product, as it is now made available in a customer environment. The setup stage is often also referred to as the configuration stage, as it comprises the configuration of software, hardware, and resources on security. The setup itself potentially involves the implementation of source code and database, installing third-party applications and cron-jobs, and configuration of APIs. After the software product is fully configured and the setup is complete, it will undergo another testing cycle [22].

- **Maintenance:** The main goal of the maintenance stage is to name, maintain, and keep the software running. Monitoring error logs from involved components, like the firewall, SHH and HTTP, is an excellent approach to oversee that the software works correctly. Potential errors affecting the performance of the software can also be supervised by monitoring traffic and input data. Not all mistakes can be detected by monitoring error logs and traffic, and further testing of the resulting product is considered necessary. Therefore, the maintenance stage includes systematically testing the functionality to uncover potential errors. In addition to keeping the software running, this stage opens to implementing new features or changes in existing functionality. The addition of, or change in, functionality leads to new code that needs to be embedded into the preceding stages [22] [64].

The stages of a software development process are generally restricted through given time limits on when the result is expected to be delivered. The exact content of each stage and how they are weighted in the overall process might vary and depends on the given project and the chosen approach to development, meaning what software development methodology is decided upon during the planning stage [64].

A software development methodology can be defined as "a set of rules and guidelines that are used in the process of researching, planning, designing, developing, testing, setup, and maintaining a software product" [22, p. 41]. Each development methodology incorporates its core values adopted by the team and introduces principles that put these values into practice. The methodologies also suggest a set of tools that can be implemented to guide and assist its different stages. Depending on the methodology implemented, the stages can be named differently compared to those mentioned above, and in some cases, they might be overlapping or merged/blended. Some methodologies might even exclude one or several of the stages. Regardless of the outlines, the overall goal of a software development methodology is to assist the software development process by providing a set of rules and guidelines throughout its different stages.

The industry today can choose from a series of software development methodologies. These methodologies are divided into two distinct main approaches based on their characteristics. These are the traditional and agile approaches, which bring their respective advantages and disadvantages into a software development process. According to Vijayasarathy and Butler [78] the more traditional approaches are still commonly implemented in the industry, but the agile practices have also shown a significant increase in popularity over the last decade. Another trend that has arisen in the IT industry is combining different methodologies within the approaches together, making a more hybrid and customized approach to software development. Due to this, it is often easier to classify a project's software development approach rather than explicitly naming the individual methodologies implemented [78]. The following sections will elaborate further on the principles and characteristics of the two main approaches and introduce some of the most commonly known methodologies from each of them.

2.3.1 Traditional Approaches

The traditional approaches to software development are often referred to as the group of heavyweight methodologies. As the name implies, these methodologies are characterized by including a lot of overhead in the process, consisting of heavy and detailed documentation and extensive planning, amongst other things [43]. Another common feature within the traditional approaches is the sequential flow in development, where each phase consecutively is conducted.

Waterfall

The Waterfall methodology is known to be the very first software development methodology [64] and its fundamental principles were first introduced by Royce in 1970 [62]. Even though Royce does not use the specific expression "Waterfall" in this publication, his article on *Managing the Development of Large Software Systems* [62] describes the pillars and ground principles of what we today know as the waterfall methodology.

The waterfall methodology is well known for its sequential design, where one stage follows the other back-to-back. The stages are precise and unambiguous and do not allow for overlapping, making the process easy to maintain. Further, they are each introduced along with their respective set of criteria, both in terms of requirements for output and milestones. The resulting output from a given stage serves as input for the consecutive stage, which necessitates that the current stage is fully completed before the next can begin [48].

Projects that implement the waterfall methodology continue to move forward and generally do not go back to previous stages after completing the current one [48]. However, choosing to go back is allowed, but it would give rise to high-cost consequences. These consequences comprise loss in terms of both time and money, as it requires changes in documentation and further development of the software product. Delays like this would affect the entire timeline of the project and the project itself [48]. Due to this, the waterfall methodology is known to fit better with smaller software projects. That is, where the project requirements are clear and well defined by the product owner, enabling the creation of a detailed plan for the development team to follow throughout the process [64]. Figure 2.2 visualizes the design of the methodology and its different stages. It also shows how the methodology suggests a process structure that is progressively flowing downwards, just like a waterfall.

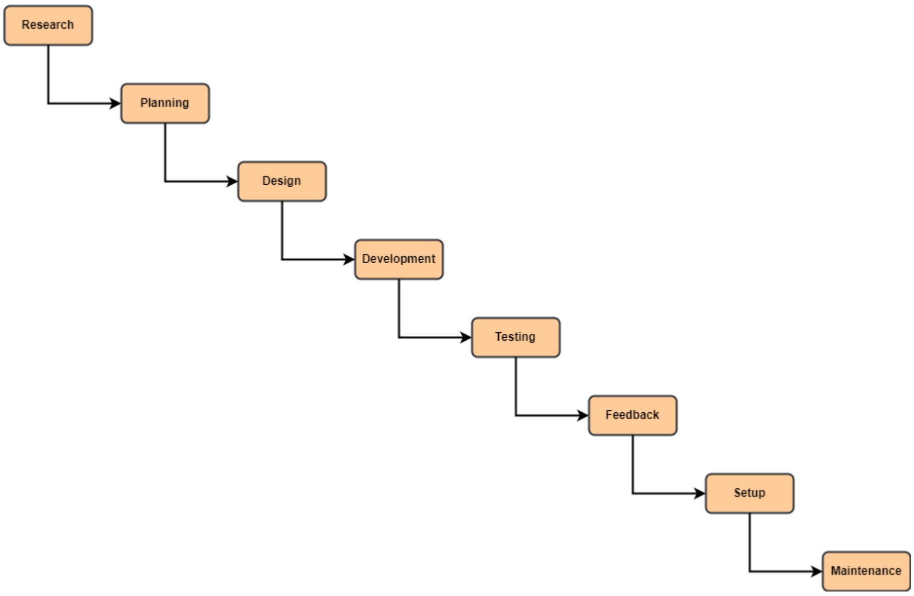


Figure 2.2: The design of the Waterfall Methodology [22]

PRINCE2

PRINCE is a project management methodology, and the name stands for *Projects In Controlled Environments*. It originates from another methodology named PROMPT (*Project Resource Organisation Management Planning Technique*), developed by Simpact Systems, a company in the private sector. Software development projects struggled with keeping within their time limits and budget, calling for the development of PRINCE2 as a response to this. The framework was supposed to make the projects more manageable [60].

In 1990, PRINCE was released into the public domain by the UK Government. The methodology became recognized among companies in the private sector, both used in projects related to the government but also for internal projects [14]. From PRINCE, a second version of the methodology was developed in 1996, namely PRINCE2. The methodology is now the standard methodology for managing projects used by the UK government [60]. The new version introduced a more process-driven approach, focusing more on ”what” and ”why” rather than the ”how.” Consequently, PRINCE2 is adapted to fit a more extensive range of projects, both in regards to type and size [14].

PRINCE2 is, as mentioned, a process-driven methodology divided into eight different processes. These processes make sure the project at hand is off to a controlled start, has controlled progress, and ends up with a controlled closing [12]. They also help manage what should be done within the project and when. The eight processes are listed below [60]:

Processes

1. **SU:** Starting Up a Project
2. **PL:** Planning
3. **IP:** Initiating a Project
4. **DP:** Directing a Project
5. **CS:** Controlling a Stage
6. **MP:** Managing Product Delivery
7. **MSB:** Managing Stage Boundaries
8. **CP:** Closing a Project

All projects implementing PRINCE2 as a methodology need to consider each of these processes. However, how these processes are addressed and to what extent depends on the task at hand. Each process should be considered up against the specific project, tailoring them to its given needs [12].

The PRINCE2 methodology consists of three key attributes, and the eight processes mentioned above make up one of these attributes. The remaining two are components and techniques. The set of components introduced in PRINCE2 aims to aid the different processes by explaining how each of them can be used and why they are all needed in the overall project structure. On the other hand, the techniques are only suggestions, meaning they are optional to implement. The technique concerning product-based planning is, however, considered essential and should be implemented in every project running under PRINCE2 [12]. The reason is that planning is an essential aspect of the methodology, which is also emphasized by the fact that one of the listed processes, process 2, exists for this purpose only. Table 2.1 lists the distinct components and techniques of PRINCE2 [12].

Components	**Techniques**
• Business Case • Organisation • Plans • Controls • Management of Risks • Quality in a Project Environment • Configuration Management • Change Control	• Product-Based Planning • Change Control • Quality Reviews

Table 2.1: The components and techniques of PRINCE2 [12]

According to Pincemaille [60] managing quality is one of the most significant aspects of the PRINCE2 methodology, as all of the three key attributes, the processes, components, and techniques, address it. This forces the project to emphasize and ensure quality in the overall process, which is an important aspect, especially for software development projects, as software quality assurance is complex and challenging to practice appropriately.

PRINCE2, being a project management methodology, is primarily made to fit a wide range of projects. It contributes to a well-defined process through a clear framework. However, the framework is defined rather generically to make it adjustable to different projects of any type and size. The methodology itself may therefore be less explicitly adapted to software development processes compared to the already existing software development methodologies out there [60].

2.3.2 Agile Approaches

The IT industry is constantly evolving, and it can be hard to keep up with the never ending changes in technology. The software industry is also concerned with this issue, as the user requirements continuously get more dynamic, and the resulting software artifacts continuously get more complex. In response to this ever-increasing demand, the industry shifted away from the more traditional approaches to software development. Instead, there were used approaches based on agile principles to support the development processes [50]. These approaches are also known as the lightweight methodologies [43]. In contrast to the already existing heavyweight methodologies and their predictive approach to software development, the lightweight methodologies provide a more adaptive approach through iterative development cycles and more frequent delivery[50].

Today, there are many different agile software methodologies, and some vary quite a lot from each other in given aspects. There is, however, one thing these agile approaches have in common, namely what values and principles that establish their foundation. These originate from the Agile Manifesto [21], a manifesto for software development written in 2001, describing the values and principles of agile software development. This manifest gives rise to the main characteristics of the agile approaches and is what makes the methodologies agile in the first place. The list below presents the four key values from The Agile Manifesto [5]:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

Each of the four fundamental values of the Agile Manifesto represents a change the agile approach was supposed to introduce to improve the traditional approaches to software development. Looking at the above-listed values, one can see that the first part of each value states what the new agile methodology desires. In contrast, the last part of each value represents what the traditional software methodologies would emphasize. Other common characteristics of today's agile approaches arise from these values. Incremental delivery, more frequent contact with the customer, and shorter development cycles are examples of such [21].

The key values are the pillars of agile methodology and are what further lays the foundation for agile principles. Both the values and principles are reflected in various software development methodologies and frameworks today. Scrum, Extreme Programming, Kanban, and Lean are some examples.

Scrum

Scrum is a lightweight framework that presents one of the most popular approaches among the agile software development methodologies today. For the first time, the process was introduced in public by Sutherland and Schwaber in 1995 through their publication of ”The Scrum Development process” at a conference in Austin, Texas [81]. Later that same year, Schwaber came up with another publication on Scrum that demonstrated the new methodology. The methodology was presented as an ”enhancement of the commonly used iterative/incremental object-oriented development cycle” [66, p. 117].

Scrum is well known for its incremental design, splitting development into short iterations known as sprints. Each of the respective sprints includes all the different phases of software development, meaning each sprint represents one life cycle of the development process. As a result of this, a working version of the product, an Increment, should be ready and delivered to the user by the end of each sprint [38] [50]. The sprints are carried out by the Scrum Team, which is a self-organized team that consists of three predefined roles. These roles are the Product Owner, the Scrum Master, and the Development Team [73] [81]:

Roles

- **The Product Owner:** The Product Owner's main responsibility is to make sure that the Development Team utilizes all of its potential and thereby delivers a high-quality product. The Product Owner is also responsible for managing the Product Backlog by identifying the items for the Product Backlog, prioritizing them, and ensuring they are transparent and understood by everyone involved.
- **The Scrum Master:** The Scrum Master is the leader of the Scrum Team and is responsible for the team to follow the rules of Scrum and its underlying Agile values and principles. Another part of this role is working as a facilitator within the team. This involves overseeing that the practices agreed upon by the team are, in fact, carried out to ensure progress in the development process.
- **The Development Team:** The Development Team is a cross-functional team, meaning that they collectively have all the competence needed to fulfill their tasks and achieve their goals. The team is also, as aforementioned, self-organized, as they work independently on turning the Backlog Items into functionality on the new Increment of the product. All members of the Development Team are considered to be developers, regardless of what work they do.

In addition to the predefined roles, the Scrum framework brings along a set of artifacts to be implemented into the development process. The framework provides these artifacts to assist the team with inspecting and adapting the development process throughout each life cycle. They are also supposed to make sure that the team has a common understanding of the project at hand [73]. The artifacts are listed and further explained below [46] [73] [81]:

Artifacts

- **Product Backlog:** The product backlog represents what the customer wants, formulated as a list of requirements, stories, or features. These backlog items are often called User Stories, which the team needs to prioritize according to the customer's wishes. The prioritizing consists of identifying what User Stories are the most important to implement in the next Increment of the product. The development team should describe all User Stories using terminology that the customer is familiar with.
- **Sprint Backlog:** The Sprint Backlog has to be finalized before the beginning of a new Sprint. This involves for the team to select User Stories from the Product Backlog, representing the work the team aims to complete within the upcoming Sprint. Once the User Stories for the Sprint Backlog are picked out, they are often broken down into smaller issues or tasks. These issues give a more detailed description of what features that need to be implemented to complete the respective User Stories.
- **Product Increment:** The Product Increment comes as a result of the Sprint Backlog, that is, the resulting product at the end of each Sprint. The Increment from each Sprint represents a step forward in the development process and is added to all previously finished increments. If the Increment meets the team's Definition of Done' (the paragraph below will provide a further description), this means a shippable and functioning product is delivered.
- **Definition of Done:** A Definition of Done is something the Development Team and the Product Owner must discuss and agree upon. This definition holds the acceptance criteria for when a User Story, and the Increment of the product, can be considered finished. If an item from the Sprint Backlog does not satisfy the conditions set in the Definition of Done, it should not be released as a part of the current Increment. The definition provides transparency between the Development Team and the Product Owner through establishing a common understanding of what to expect from the resulting Increment.

In addition to the Artifacts, the Scrum frameworks suggest a set of events that should be conducted as part of every Sprint to supplement the development process further. The prime objective of each event is to provide transparency within the Scrum Team. In addition to this, the events supply the team with the possibility to review and modify the product [81]. Each of the events has its respective predefined maximum duration. There are five distinct events in total, which are all listed and further elaborated below [46] [47] [73]:

Events

- **Sprint:** As previously mentioned, a Sprint represents one life cycle of the development process and results in a new Increment of the software product. The Development Team implements functionality corresponding to the User Stories from the current Sprint Backlog during these Sprints. They are conducted consecutively, meaning the Development Team can only begin a new Sprint after completing the previous one. All Sprints should have their own objective, called the Sprint Goal, that gives a short explanation of what the team wants to accomplish throughout the upcoming Sprint. No changes or adjustments that potentially could affect the Sprint Goal are allowed during the Sprint. This means that the Sprint Backlog should not be modified, and its refinement should be conducted before/in between Sprints. The duration of the sprints might vary, but they usually last somewhere between two to four weeks and never exceed more than a month.
- **Sprint Planning:** Sprint planning is a meeting held before the beginning of each Sprint to define a Sprint Goal and prepare the Sprint Backlog for the upcoming iteration. The User Stories picked out from the Product Backlog should be based on the priorities set by the Product Owner, as well as how much work the Development Team think they will manage within the given Sprint [47]. Therefore, all the Scrum Team members should take part in this meeting to discuss and agree upon what they want to achieve and what User Stories they want to include. The Sprint Planning meeting results in the Sprint Backlog.
- **Daily Scrum:** The Daily Scrum is a short stand-up meeting arranged daily to synchronize work and make sure the development proceeds towards the Sprint Goal. The meeting includes all of the Development Team members sharing what they have been doing since the last meeting and what they will be doing until the next meeting. The team members should also address any potential problems or obstacles keeping them from progressing in their work. The meeting is supposed to happen at the same time every day, and the meeting is limited to last no longer than 15 minutes to ensure efficiency [5].
- **Sprint Review:** The Sprint Review is a meeting that is held after the end of each Sprint to, namely, review the forgoing iteration and the resulting Increment. The entire Scrum Team and the product stakeholders are invited to participate. The meeting involves evaluating the progress made during the Sprint and addressing the

work that has not yet been completed, whether it is work left over from previous Sprints or the current Sprint. The Increment is demonstrated to the stakeholders to receive feedback and discuss the current status of the product. As a result, if the Scrum Team finds it necessary, the Product Backlog is modified according to the information obtained through this meeting.

- **Sprint Retrospective:** The Sprint Retrospective is a meeting held after the Sprint Review and before the Sprint Planning for the upcoming Sprint. The purpose of this meeting is for the team to discuss and reflect upon the work practices and the process so far in terms of what has been working well and what has not been working well. The result of these discussions is further used to make adjustments and improve the process for upcoming Sprints [47]. The Sprint Retrospective meeting often results in one or more tangible action items, which are proposals for improvements the team would like to focus on progressing forward.

Figure 2.3 illustrates how the Scrum Life Cycle unfolds. The figure visualizes how the events mentioned above and artifacts are connected to each other and where they fit in during an iteration of the development process.

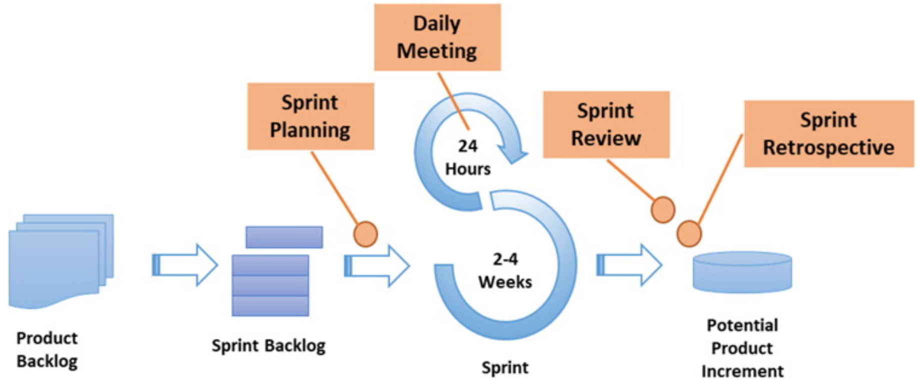


Figure 2.3: The Scrum Life Cycle [81]

Extreme Programming

Extreme Programming (XP) is another lightweight methodology commonly implemented in the industry today. XP reforms the traditional development process structure, like the Waterfall Model, by dividing the process into smaller blocks. That is, the methodology focuses on planning, analyzing, and designing little at a time throughout the entire process, rather than handling the whole future of the project at once [11]. This iterative approach, involving shorter development cycles, came as a response to the high cost of changing the software product. Hence, the goal was to reduce these costs and improve efficiency by replacing the long development cycles, as they could not adapt to the rapid changes in requirements [22] [50].

Extreme Programming is concentrated around four core values: courage, communication, feedback, and simplicity [49]. As XP is a lightweight methodology, it shares the same underlying Agile values and principles as the other Agile approaches to software development. However, XP differs from many methodologies by introducing its own set of practices that foster these values. Specifically, concerning simplicity, as the practices aim to make the overall process less complex. Where other methodologies want to answer the question of ”What are the practices I might ever need on a Software Project?” XP aims to provide the simplest set of practices that possibly could be needed [49]. Kniberg also states that where ”Scrum focuses on management and organization practices, while XP focuses mostly on actual programming practices” [46, p. 104]. The result of this is 12 simple and independent software development practices that define XP [49]:

Principles of XP:

- Pair Programming
- Test-Driven Development (Test-First Design)
- Refactoring (Design Improvements)
- Simple Design
- Customer Tests (Acceptance Tests)
- Whole Team
- Planning Game
- Small Releases
- Continuous Integration
- Collective Ownership
- Coding Standard
- Metaphor
- Sustainable Pace

The core practices of XP, as mentioned earlier, do not necessarily comprise all the practices needed throughout a software development process. Different teams might find different principles helpful, and what combination of concepts leads to success might also depend upon the project at hand. On top of the 12 core practices, there exist some additional ones. Some commonly implemented ones are Open Workspace and Retrospectives [22] [49]. However, it is important to emphasize that neither the core principles nor the additional principles are new concepts and were not invented as part of the XP methodology [11].

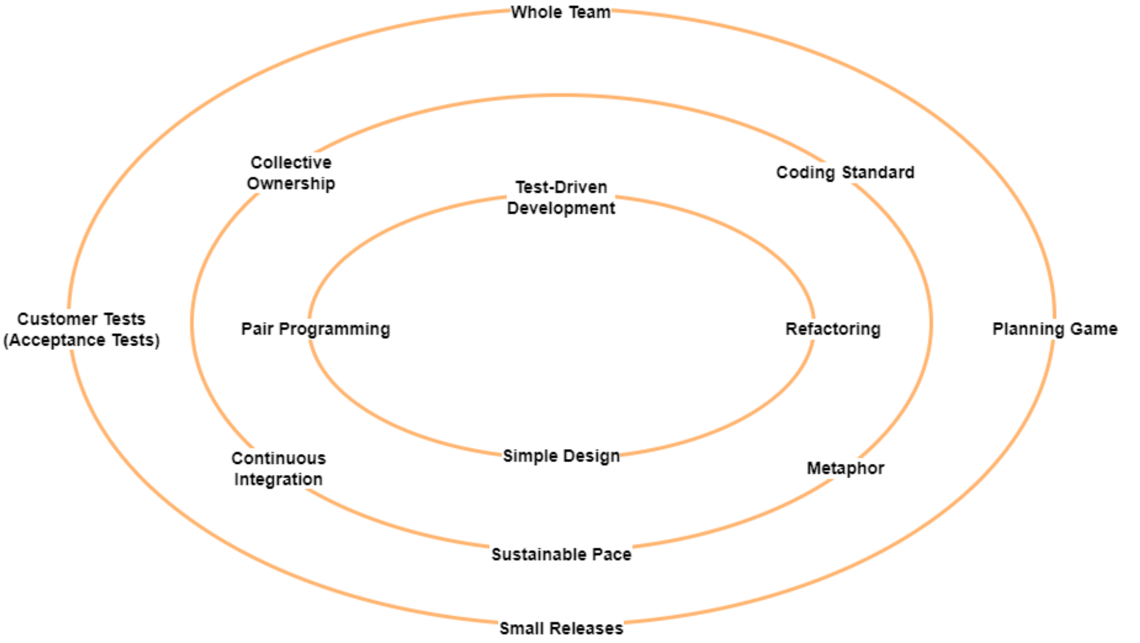


Figure 2.4: The XP Life Cycle and its practices [49]

Figure 2.4 visualizes the life cycle of XP and how its practices are used to form a flow of activities. The figure consists of three circles that groups the distinct principles based on whom or what part of the development process they are relevant to [49] [11]:

- **Inner Circle:** The inner and smallest circle lists practices applicable to the programmers during development. Pair Programming is one of them and is also one of the main characteristics of XP. This practice involves all developers working together in pairs (or smaller groups), coding on a single screen and keyboard. Another essential aspect of XP is Test-Driven Development, which provides good test coverage throughout the development process. In accordance with one of the core values of the methodology, this will also ensure valuable feedback. The remaining practices in the inner circle encourage the team to maintain a simple design and to continuously improve it through refactoring the code according to current needs.

- **Middle Circle:** The middle circle contains practices that concern the entire development team. These are provided to ensure good communication and coordination among all team members, resulting in a high-quality software product. Continuous Integration is one of the principles of the middle circle, which requires the team to integrate newly produced code into the current system pretty much straight away and keep the system running continuously. In XP, all the programmers can improve any code segment when they find it beneficial, no matter where within the system it is located. This introduces the principle of Collective Ownership and underlies the importance of the principle of Coding Standards. The last two practices in the middle circle are named Metaphor and Sustainable Pace. These comprise for the entire team to have a common understanding of what the system looks like and that they consistently work at a pace that possibly could be sustainable for the whole future.

- **Outer Circle:** The largest and outer circle represents the communication between the development team and the customer. The practices mainly address the planning stage of development and are concerned with aiding the communication throughout this particular stage. Three of the listed practices are Customer Tests/Acceptance Tests, Whole Team, and Planning Game. These practices involve the team figuring out what features to implement next, predicting when these features will be finished and designing and building acceptance tests for these features in collaboration with the customer. Subsequently, the team will develop the features through short iterations, producing frequent and smaller deliveries. This is in accordance with the last practice of the outer circle, named Small Releases.

Kanban

Scrum is, aforementioned, one of the most popular approaches to agile software development today. Alongside Scrum, Kanban has also shown a marked increase in popularity in the software industry as an agile methodology, building on agile principles to achieve agile goals [50] [81]. Kanban means 'visual signal'. As the name initiates, the methodology aims to aid software development processes by visualizing and limiting the work in progress. The methodology originates from a system developed in the late 1940s for the Toyota company. The system was based on the concept of filling up stocks in accordance with the vendor's needs rather than their supply and was used by Toyota's employees to track processes [81].

Kanban is highly focused on the workflow during the software development processes, and work scheduling is an essential aspect of the methodology. This additionally includes eliminating activities considered unimportant throughout the process and minimizing the duration of the iterations as much as possible. Kanban is related to Lean and JIT (Just-In-Time) development and a lot of their principles are reflected in the Kanban framework [19]. According to Ahmad [2] it is clear how principles from Kanban have a high degree of overlap with the principles of Lean Software Development. Even though there are a lot of similarities between Kanban and other Agile methodologies, the Kanban Framework has some characteristics that distinguish it from the rest [50] [81]:

- **Kanban Board:** The Kanban Board is a tool for visualization used to manage and guide the workflow of the development process. The idea is to divide the work into different categories (or columns on the actual board), representing the status of the current task. The three categories are To-Do, In-Progress, and Done, and a task will be moved from one category to the next as the process evolves. The Kanban Board is a handy tool, as it makes it easy to track and evaluate the ongoing process continuously. It also visualizes any potential bottlenecks and enables optimization of the work.
- **Maximizes Productivity:** The Kanban Framework aims to maximize the overall productivity through optimization and scheduling of the workflow throughout the process. Different measures can be taken to improve productivity, but avoiding bottlenecks and hence reducing idle time is one of them.
- **Continuous Delivery:** Kanban releases new features in small iterations, similar to Continuous Delivery. Unlike releasing new features in bigger increments, this approach makes it easier to meet the customers' rapidly changing requirements.
- **Waste Minimization:** As mentioned earlier, Kanban aims to minimize the duration of each iteration. To reduce over-production and time spent on wasted work, tasks that are not required will not be considered for implementation at all.
- **Limits Work in Progress (WIP):** The main goal of the Kanban Methodology is to be able to limit the Work In Progress. As a measure to add limitations, the methodology motivates optimizing the workflow according to the constraining capacity, whether it involves the entire process or a sub-process. In other words: This will decrease the time it takes for a task to navigate its way through the Kanban Board.

2.3.3 DevOps

DevOps stands for Development and Operations and is, according to Mohammad [52] a software engineering culture and practice that aims to unify the two components, namely software development and software operations. The two components are integrated by implementing automated development, automated deployment, and infrastructure monitoring into the overall development processes. In other words: DevOps is emphasized around automation, end-to-end in software development and delivery [27].

As mentioned previously, the industry has shifted away from the more traditional approaches to software development and instead started implementing more agile practices over the years. Implementation of continuous integration has been a big part of this, aiming to improve the efficiency of the development teams. The focus has mainly evolved around optimizing the software development process, leaving the operations side of the process behind, thereby causing a bottleneck in the software delivery. This bottleneck will delay the entire delivery if one phase cannot keep up with the rest. That is also why involving optimization through continuous integration has shown not to be sufficient to make the entire lifecycle of development efficient [79].

DevOps is a cultural and organizational shift that aims to solve the issue mentioned above by unifying the traditionally opposing sides of development and operations. It efficiently integrates development, delivery, and operations, by connecting the three traditionally separated silos [27]. This involves dividing the existing silos into cross-functional teams to remove the distributed and disconnected groups that normally would operate separately. By implementing test-driven development and continuous integration, these teams will provide more rapid and continuous delivery. This means working full stack and necessitates additional knowledge beyond just general code knowledge for the developers. As they are now also responsible for testing and releasing, they must have the required skills to do so. When the boundaries between the silos vanish, this also creates a need for collaboration between the team members. DevOps encourage collaboration, along with teamwork and communication, which helps to reduce issues arising from miscommunication within the team [27] [33].

DevOps has a lot of similarities with the agile approach to software development, and they share a lot of the same principles and concepts. In both approaches, the software goes through development, testing, and deployment stages, but where the agile approaches stop after these three stages, DevOps continues. DevOps extends the usage of the principles by applying them to the entire software delivery pipeline, leading to continuous monitoring and development of software. Even though DevOps share many values and principles with agile approaches, there are also differences between them. Where agile methodologies provide the aids to improve efficiency for developers and make more frequent deliveries to the customer, DevOps provides culture, practices, and instruments that divide silos into teams and allow for continuous implementation and continuous delivery. Following is a list presenting additional differences between agile approaches to software development and DevOps [52]:

- Agile approaches provide a project management strategy, while DevOps focuses on optimizing the entire project pipeline.
- Agile approaches emphasize flexibility and advancement of functions, whereas DevOps focuses on implementing continuous integration and installing the already developed software.
- Agile approaches are often introduced by implementing an agile framework, like Scrum, while DevOps does not necessitate the implementation of any specific framework.
- Agile approaches are centered around operations, where DevOps is focused on automation and efficiency in the overall process.

Some of the main factors in DevOps are quality assurance, testing, and more effective delivery cycles. DevOps defines suggested principles, but "how to adapt to these" and "using what technology" are for the different organizations to decide. Also, each project is different, both in regards to the software product and varying lifecycle processes, which necessitates distinct adoptions of the principles. It is, therefore, up to each organization or team to find the best-suited approach to adopt the DevOps principles, both in regards to culture, architecture, and what tools to implement [27] [79].

2.3.4 Choice of Methodologies and Approaches

The IT industry today has a lot of different options to choose from when it comes to how to approach a software development process. There exists an ocean of software development methodologies providing frameworks that aids them in the process of planning, executing, and managing their process [78], and some of them have been introduced throughout this section. Whether they are part of the heavyweight or lightweight frameworks, each of the methodologies brings along its own strengths and weaknesses, making them suitable for different organizations and projects.

Deciding upon what approach to take on a software development project can be challenging, as the choice has a significant impact on the development process. The selection might be based upon many different factors, like team composition, project characteristics, or predetermined standards within the organization.

In 2015 Vijayasarathy and Butler [78] did research on what software development methodologies project managers and other team members use in their projects. The research was conducted through a survey based on the participants' most recent project, where they had been actively involved throughout the process. The participants were also asked to express their project role during the given project. The results show that 58.2% of the participants were leaders or managers of their teams, including the following titles: project manager, team leader, development manager, and quality assurance manager. 25.5% of the participants expressed they were involved in the project through more technical roles. These technical roles comprise developers, analysts, architects/designers, and testers. The majority of the participants showed to have a lot of experience in the industry, and more than 56% from both of the groups mentioned above stated that they had more than five years of experience in their respective roles [78].

The survey presented the participants with a list of methodologies to choose from (the same as the once listed in Table 2.2), containing everything from well known and general frameworks, like Scrum, to more specific and detailed processes and techniques, like XP for instance. The results show that none of the listed methodologies were used in more than a third of the projects. The most commonly implemented was one of the traditional approaches, namely the waterfall methodology. 32% of the participants answered that the last project they took part in used this methodology. Table 2.2 lists a complete summary of the results, including the distinct methodologies and their corresponding percentage of projects. Here they are sorted from most to least frequently implemented.

Software Development Methodologies	**Percentage of Projects**
Waterfall	32.0%
Agile Unified Process	28.1%
Scrum	20.3%
Test-Driven Development	19.6%
Rapid Application Development	18.3%
Joint Application Development	15.7%
Feature-Driven Development	14.4%
Adaptive Software Development	13.7%
Lean Software Development	13.1%
Structured Systems Analysis and Design Method	9.2%
Extreme Programming (XP)	8.5%

Rational Unified Process	5.9%
PRINCE or PRINCE2	3.3%
Crystal	3.3%
Dynamic Systems Development Method	1.3%

Table 2.2: The software development methodologies reported used in projects sorted by percentage [78]

Combining different methodologies to make a more hybrid and customized approach to software development is a rising trend in the IT industry, as discussed earlier. This was also one of the findings uncovered in Vijayasarathy and Butler's research. Due to this, they decided to divide the methodologies into categories to make it easier to classify the projects. The classification looked at the software development approaches rather than the individual methodologies. The result provides a simple overview, and a clear impression of the distribution [78]. Table 2.3 shows the distribution of approaches in terms of what percentage of the projects that implemented them. The middle column of the table declares what methodologies are included in the distinct approaches. The results show that most of the participants have been involved in projects using a more hybrid approach to software development, meaning they have chosen to implement methodologies from several of the other listed approaches.

Software Development Approach	**Included Methodologies**	**Percentage of Projects**
Agile	• Agile Unified Process • Scrum • Test-Driven Development • Feature-Driven Development • Adaptive Software Development • Lean Software Development • Extreme Programming • Crystal • Dynamic Systems Development	33.1%
Traditional	• Waterfall • Structured Systems Analysis and Design Method • PRINCE or PRINCE2	13.8%
Iterative	• Rational Unified Process • Joint Application Development • Rapid Application Development	7.7%
Hybrid	A mix of methodologies from the other approaches	45.3%

Table 2.3: The software development approaches used in projects sorted by percentage [78]

Kanban

Kanban has shown an increase in popularity in the software industry as an agile methodology over the last decade [50] [81]. This marked rise in popularity aligns with the findings of the "State of Agile" reports. These are reports formulated from surveys conducted annually by Digital.ai (a company formerly named VersionOne [24]). In 2015 the use increased from 31% to 39%, and it increased even further in 2016, from 39% to 50% [57] [58].

In 2021 the 15th number of the "State of Agile" report was published, involving 4182 participants. The report provides statistics showing that the use of Kanban Boards for visualizing and managing the workflow in projects has been more commonly adopted over the years. From the first version of the report, published in 2007, the number of participants who reported the use of Kanban Boards has increased from 6% to 61% in the 2021 report [25].

DevOps

According to the "State of Agile" reports, the adoption of DevOps has become more widespread throughout the last couple of years, according to the "State of Agile" reports. The surveys asked how important DevOps was within the participants' organizations, giving them options of 'Very Important', 'Important', 'Somewhat Important', and 'Not Important'. Results from the last four years show that the percentage of participants who consider DevOps 'Very Important' within their organization has increased from 33% to 42%. In the latest edition, published in 2021, 75% participants stated that the approach was either 'Important' or 'Very Important'. Only 9% of the participants answered that DevOps was 'Not Important' within their organizations [25].

Chapter 3

Literature Review

The following chapter will present a summary of a Systematic Mapping Study (SMS) written throughout the fall semester of 2021 [13]. The review was conducted to explore and uncover existing literature on the intersection between IT and gender equality. The findings identified several gaps in the current state of the art, explicitly concerning existing best practices to address gender equality in IT processes. The study identified several challenges to women's participation in the industry, such as gender bias, gender stereotypes, imposter syndrome, and retention problems. Further, any processes directed specifically at software development were not identified to diminish the discovered challenges. However, the review showed that implementing a process to address these challenges would also assist the SDG 5.5c in achieving gender equality and further inspired the theme for this thesis. This chapter will be dedicated to summarizing the literature review by presenting its motivation, method, and main findings. The chapter, in its entirety, is referenced from the literature review.

3.1 Motivation

While IT over the past years has gained more significant influence over people's everyday lives through the constant development of software artifacts, the lack of women in IT has become a matter of concern. When the United Nations (UN) presented 17 goals for sustainable development in 2015, one of the goals introduced was concerning gender equality.

3.2 Research Method

The research was conducted through a SMS and is influenced by Kitchenham's [45] procedures. The following section will introduce the research questions, search strategy, data extraction, and threats to validity. The SMS was conducted in the advanced search function of Scopus [8].

3.2.1 Research Questions

The following RQs were identified for the literature review [13] to get an overview of the state of the art while focusing on the affiliation between IT processes and gender equality. The RQs as presented in the literature study are listed below [13]:

RQ1 What kind of studies exist on IT processes and gender equality?

RQ2 What gender equality targets are addressed by IT processes?

RQ3 What are the main challenges to achieve gender equality in IT processes?

RQ4 What are the best practices established to address gender equality in IT processes?

3.2.2 Search Strategy

The fifth SDG of gender equality covers a broad field and has a set of nine sub targets. Therefore, it was decided to initiate one general search string and one for each of the belonging sub-targets of the goal, resulting in ten search strings and 4206 potential studies. The selection of studies was then reduced based on the selection criteria by reviewing the titles, abstracts, and keywords.

The potential studies were further reduced to 50 potential studies and, finally, to 15 primary studies.

3.2.3 Selection Criteria

The selection criteria were set to scope the papers from the preliminary search in a structured manner while still including all the relevant papers. The selection criteria for including a paper, as written in the report, was the following [13]:

Inclusion Criteria

i1 English articles published between 2016 and 2021 about gender equality in and by the IT sector

i2 Complete articles that are peer reviewed in journals or conferences

Exclusion Criteria

e1 Articles presenting opinions, or inadequate articles such as abstracts or presentations e2 Articles that do not revolve around IT processes and gender equality

e3 Replicated work, only the most recent will be taken into account

3.2.4 Data Extraction Strategy

The data extraction strategy utilized the RQs as a vantage point. For the first RQ concerned with what studies exist in the field, there were four potential answers: others, validation, proposal, and state of the art. For the second RQ, the studies assessed which targets they aided. For the third and fourth RQs, keyword extraction was used to avoid setting out answers that could limit the possible results.

3.2.5 Threats to Validity

Even though the literature search was limited to research that applied to Europe and North America, this does not directly imply that all research is valid and representative of the total area. The searches conducted in the project were only performed using Scopus. It would have been beneficial to use another search engine or a manual search to reduce the possibility of not finding all relevant papers.

3.3 Results

Table 3.1 shows the data extraction results for all studies and is fetched in its entirety from the literature review [13]:

Study ID	**Type**	**Targets**	**Challenges**	**Best practice**
S01	Others	5.1, 5.5	Implicit bias	
S02	Others	5.1, 5.5	Gender bias, Code acceptance, Disengagement, Imposter syndrome	
S03	Others	5.5	Gender bias	
S04	Validation	5.1, 5.4, 5.5, 5.5c	Few women, Gender bias, Retention problems, Imposter syndrome, Motherhood	Women workshops

S05	Proposal	5.1, 5.4, 5.5, 5.5c	Recruitment, Poor management, Retention problems, Gender bias	Nudging
S06	State of the art	5.1, 5.4, 5.5, 5.5c	Gender bias, Gender preferences, Pay gap	Habit breaking
S07	State of the art	5.5	Gender bias, Pay gap, Retention problems, Implicit bias	
S08	Validation	5.5, 5.5b, 5.5c	Gender bias, Imposter syndrome, Stereotype threat	Goal Congruity model
S09	Validation	5.5	Imposter syndrome	
S10	State of the art	5.4, 5.5, 5.5c	Gender preferences, Stereotype threat	Anti bias training, gender blind training
S11	Others	5.1, 5.5	Symbolic violence, Queen bee syndrome, Motherhood, Imposter syndrome	
S12	State of the art	5.1, 5.5	Pay gap, Gender based discrimination, Motherhood, Imposter syndrome	
S13	Validation	5.1, 5.4, 5.5	Implicit Bias, Self-efficacy, Imposter syndrome	
S14	Others	5.1, 5.5	Gender bias, Stereotype bias, Negative environment, Stereotype threat	
S15	State of the art	5.1, 5.5	Implicit bias	

Table 3.1: The data extraction results for RQ1 (Type), RQ2 (Targets), RQ3 (Challenges), and RQ4 (Best practices) [13]

3.3.1 RQ1: What kind of studies exist on IT processes and gender equality?

The literature review was based on 15 primary studies that created the basis for the findings in the report. The studies were categorized into four different categories based on the research method in the relevant studies; this is displayed in Figure 3.1. Five of the studies represented the State of the art, while four were labeled as validated studies. One study presented a proposal, while the remaining five did not fall under one category and were labeled as "Others" [13].

Looking at the year of publication for the primary studies, the theme of Gender equality in IT processes seems to be trending.

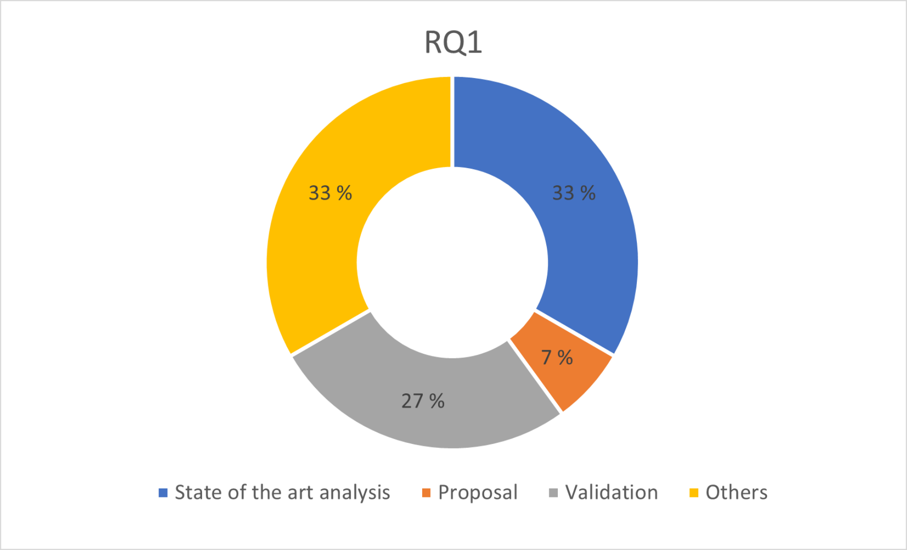


Figure 3.1: Results from data extraction of RQ1 fetched from [13]

3.3.2 RQ2: What gender equality targets are addressed by IT processes?

For the second research question, all the primary studies were evaluated on what targets of SDG goal 5, as mentioned in section 2.1 they aimed for. The results can be seen in Figure 3.2. Target 5, representing women's participation and equal opportunities, was a red line represented in every study. Target 1 concerning ending all discrimination against women was also represented in two-thirds of the studies. Targets 4 and 5c were both included in five of the studies and were concerned with recognizing unpaid care and domestic work, as well as strengthening the promotion of gender equality [13].

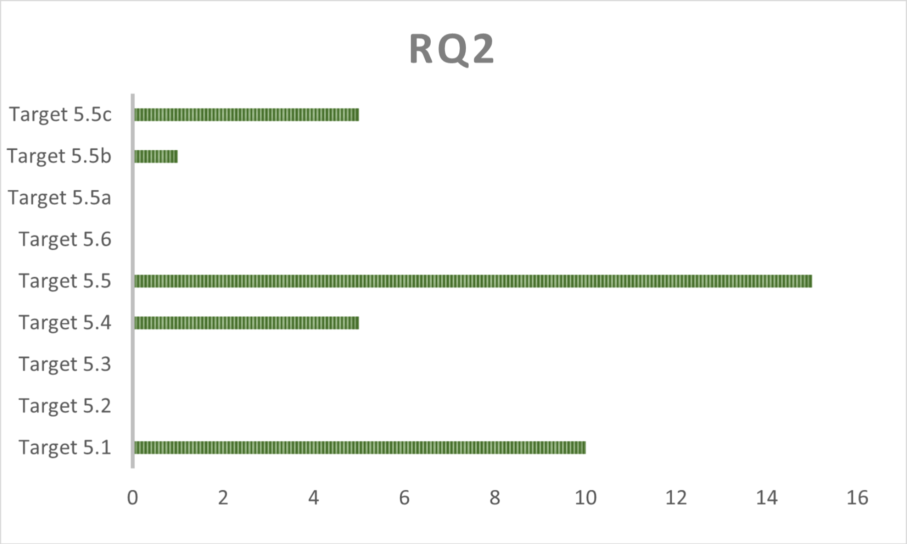


Figure 3.2: Results from data extraction of RQ2 fetched from [13]

3.3.3 RQ3: What are the main challenges to achieve gender equality in IT processes?

All the primary studies presented a set of challenges to achieve gender equality, where some challenges were more frequently mentioned than others [13]. Gender bias was the major challenge mentioned in eight studies and is strongly related to implicit bias mentioned in four of the studies. Imposter syndrome was mentioned in six of the studies as a challenge for women in IT as many addresses their success to luck instead of hard work and talent. One can also see imposter syndrome in relation to stereotype threat, which is the reluctance to do something out of fear of confirming a (gender) stereotype. Other intriguing results were discussions on to what degree gender preferences played a part or if women were a part of making the IT environment hostile for other women through the "Queen bee syndrome". One study also presented that women had more difficulties getting their code accepted in code reviews. There were also presented several other high-level challenges such as the pay gap, obstacles in connection to motherhood and the final challenge of retention problems [13].

Some of the challenges mentioned above are more relevant to software development processes. In contrast, others need to be addressed in a larger assembly as the society or companies as a whole.

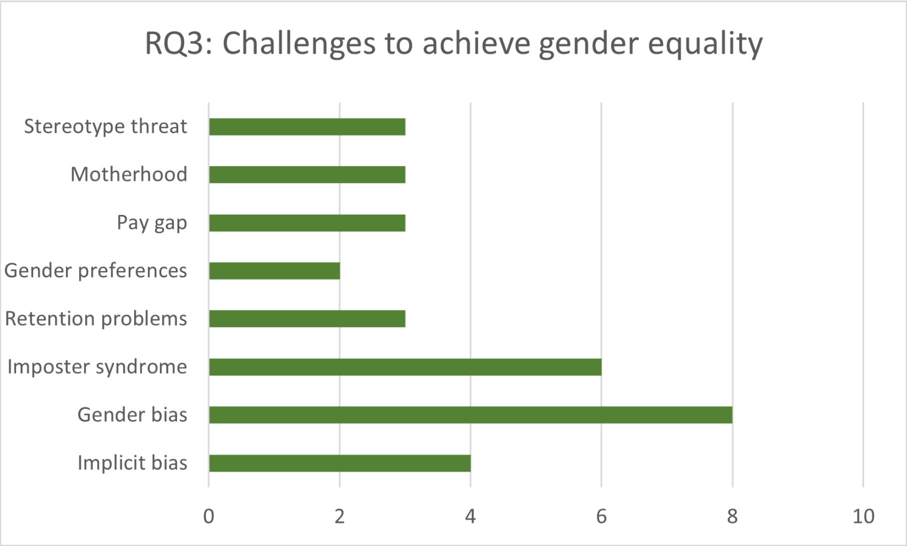


Figure 3.3: Results from data extraction of RQ3 fetched from [13]

3.3.4 RQ4: What are the best practices established to address gender equality in IT processes?

An exciting discovery in the established best practices to address gender equality in IT processes was that there were no established best practices specifically for IT processes. In total, five studies presented general best practices. These were women workshops [65], nudging [7], habit breaking [16], the Goal Congruity Model [23] and, finally, anti bias training and gender blind training [72] [13]. Further, the five studies that presented these best practices also hit target 5c of the SDG goal 5; this can imply that creating more best practices can further aid target 5c of strengthening policies for the promotion of gender equality [13].

3.4 Future work

The fact that no best practices were discovered that apply specifically for IT processes was surprising as there is a high demand for adding women to IT. Further, the IT processes constitute the structure for the way female developers work and interact with others. The retention problems for women in the field are also serious, and addressing the challenges women meet in the current IT processes might help in reforming both the processes and the industry itself to better hold on to and increase the number of women in IT [13].

Chapter 4

Research Method

The presentation of the research method comprises introducing and elaborating further on the different research methods and strategies chosen throughout the process and justifying these choices. This chapter aims to describe how the research presented in this Master's thesis has been conducted and will therefore go into further detail on the sequence of events leading up to the final results in chapter 5 and the discussion in chapter 6.

The chosen method of this Master's thesis is based on a strategy presented by Oates [56], as case studies. According to Oates, one of the characteristics of a case study is the natural setting around the research or case in question, implying that the given setting exists already prior to the beginning of this study and will continue to exist after the research for this Master's thesis is done [56]. Interviews were used as a data generation method to utilize the natural setting around the case. In further detail, the interviews were performed in a semi-structured manner, thus taking advantage of the natural setting and relaxed conversation. Interviewing people directly involved with software development processes provides the opportunity to get first-hand insight into and examine the case (RQs) directly from its natural setting. Oates points out that the use of multiple sources and methods is another typical characteristic of the case study [56]. The goal of the interviews was to gain insight into the industry based on the subjects' perceptions. The interviews were conducted with multiple relevant representatives from the industry to gain a broader and more diverse insight. The data collected from the interviews are considered to be qualitative.

4.1 Research Questions

The RQs set for this study should communicate what the research aims to discover [63]. This section further elaborates on the stated RQs and their related motivation. The SMS [13] revealed a gap in the literature for gender equality in IT processes. In particular, today, there does not exist any IT processes that address the identified challenges for achieving gender equality. Further, the study indicated that implementing a process to address the discovered challenges could directly aid SDG 5.5c with strengthening policies for the promotion of gender equality [61]. Inspired by the discoveries in the SMS the main prospect for this study is to address what actions can be taken to improve gender equality in software development processes.

Women are still a minority in the IT industry, especially in developer roles and within development teams [77] [26]. The implications the lack of gender diversity has on software products have occasionally been exposed in products that neglect important aspects that diverse users require. Increasing the presence of women in software development processes has been a focus for quite some time, but the fact is that women today remain underrepresented [77]. The consequences gender inequality has on software products and what can be done to improve software development processes while women remain a minority need to be addressed and constitutes the purpose for RQ 1.1.

It is currently unknown what impact the challenges identified in the literature review have on the Norwegian IT industry. However, research shows that the retention problems among women in the IT industry are consequential, with 50% of women having left the field within 12 years [35] even though 80% say they enjoy their work [36]. In order to reduce retention problems RQ 1.2 aims to tackle how the discovered challenges in IT can be diminished.

This motivation resulted in the formulation of the following RQs:

> **RQ1:** What actions can be taken to improve software development processes with unequal gender distribution?
>
> > **RQ1.1:** How can the artifacts resulting from software development processes with unequal gender distribution be optimized?
> >
> > **RQ1.2:** How can the challenges women meet in software development processes today be diminished?

4.2 Interviews

The interviews were conducted with multiple representatives from the IT industry to get a more comprehensive collection of data than one would get from including just a few. All the interviews are considered to be expert interviews. Sebele-Mpofu suggests aiming for a sample size of 12 semi-structured interviews in order to maintain appropriate sampling and achieve saturation [68].

The goal of the interviews was to gain insight into the industry and collect data based on the experiences and perceptions of our interviewees. In order to find participants considered relevant to this Master's thesis, a set of inclusion criteria was introduced. The inclusion criteria describe what requirements the potential participants must meet to be included as part of this study.

4.2.1 Selection Criteria

Inclusion Criteria

i1 The interviewee must have at least five years of experience in a software development team. Either as a project leader or developer

i2 The interviewee must have experience from using one or more development methodologies

i3 Minimum half of the interview subjects must be female

i4 The total group of interviewees must represent at least six different companies

Exclusion Criteria

e1 The interviewee can not have a close connection to one or both of the authors.

In order to gain a better understanding of the problem, it was natural to seek out people who have first-hand experience of unequal gender distribution. Therefore, the interviews will seek out two different groups with more than five years of experience in software development. This criterion further demands that the participant's experience originates from working in a technological role or being a project leader in a software development team. It was deemed essential that the project leaders are experienced with leading teams with and without women. The second inclusion criteria require participants to have experience with software development methodologies. Furthermore, the third inclusion criteria are in line with the object of the thesis, as it requires a minimum of 50% female participants. For the fourth and final inclusion criteria, it is required that the interviewees are from a minimum of six different companies.

As for the exclusion criteria, the interviewee cannot have a close connection to either of the authors, as this might influence their answers.

Due to the time constraints on the participant's schedule, we will time the interviews to 45 minutes.

4.2.2 Personal data

Throughout this project, we wanted to minimize the transfer of sensitive information involving the exchange of personal data as a measure to shield the participants as best as possible. In order to communicate with the candidates and set up the interviews, it was, however, necessary for us to collect their names and email address. The interviews were conducted through Microsoft Teams [51], and from there, digitally recorded by sound and video. Microsoft Teams also provides functionality that automatically produces transcripts throughout the interview session. These transcriptions were saved as text files.

In addition to their name and email, the participants provided some general information about themselves for the research.

- **Gender:** What gender the participant identifies with
- **Years of Experience:** How many years of experience the participant has from the IT industry
- **Working Title:** What current working title the participant has

The above-listed information is used throughout the research and this thesis but is completely anonymized. We have also taken the time to make sure that it is not possible to identify the participants based on the data presented.

All personal data will be deleted upon completion of this project and will not be accessible to the public through the publication of this thesis. It is also worth mentioning that participation in this project is voluntary, meaning the participants at any given time can withdraw their consent for participation without any further reason. All information gathered on the participants, including their data, will immediately be deleted.

We formulated an 'Information letter' to inform potential participants about the project and for what purpose this research was to be conducted. The letter comprises informing about the participant's privacy throughout the project and how we planned to handle their personal data. The full document can therefore be found in our 'Information Letter' in Appendix A. The letter is based on a template provided by the *Norwegian Agency for Shared Services in Education and Research (NSD)* [31].

Quality assurance for the collection of personal data

As a part of the quality assurance for master theses at the *Norwegian University of Science and Technology (NTNU)* we declared all the personal data collected for our thesis in an application to the *Norwegian Agency for Shared Services in Education and Research (NSD)*. As a part of this application, a 'Consent Form' that all participants received was also created. This form is a part of the 'Information Letter' that was sent out to all participants and can be further reviewed on the third page in Appendix A. The application was approved by NSD prior to the conduction of interviews.

4.2.3 Developing the Interview Guide

Kallio et al. [42] discovered five phases for developing an interview guide that has been adapted in the development of the interview guide for this thesis. The first phase is in place to establish whether the current project is suited for using semi-structured interviews as an approach to data collection. The second phase is concerned with collecting information and using previous knowledge to prepare for the process. Further, the third phase consists of formulating the first draft of the semi-structured interview guide. Finally, the two final phases consist of pilot testing the interview guide before presenting the final draft [42].

Identifying the Prerequisites for Using Semi-Structured Interviews

This first phase aims to justify our choice of the data collection method. That is, to assess to what extent using semi-structured interviews corresponds to our stated RQs. As stated by Kallio et al. [42], it is deemed essential to have some background knowledge on the research topic prior to the conduction of the interviews. This makes it easier to evaluate if semi-structured interviews are the correct approach for the stated RQs and their given context.

Semi-structured interviews are well-suited in situations where one wants to one wants to explore the participant's personal opinions and perceptions [9]. Gender equality in the IT industry is a disputed topic that people have conflicted opinions about, which makes semi-structured interviews applicable to our research as it necessitates for us to investigate people's thoughts and perceptions. The data collection method is also beneficial for addressing issues that are more complex and might be emotionally sensitive to some participants [6]. Considering our RQs, and especially RQ1.2 about challenges women meet in the industry, the questions formulated can feel personal and sensitive to some of our participants. Through semi-structured interviews, one can utilize the natural setting around the research context and establish a relaxed environment for conversation. Interviewing people directly involved with software development processes, therefore, provided us with first-hand insight into the research context and the opportunity to examine the RQs directly from its natural setting [56].

Retrieving and Using Previous Knowledge

The second of the five phases suggested by Kallio et al. includes retrieving information on the topics in question and making use of prior knowledge.

In the fall semester of 2021, a literature review was conducted to identify the literature existing on the intersection between IT and gender equality [13]. The results from the literature review identified several gaps in the current state of the art, explicitly concerning existing best practices to address gender equality in IT processes. A summary of the literature review is given in chapter 3.

Conducting an extensive literature review is also listed by Kallio et al. [42] as an approach to gaining preliminary knowledge of the research context. The literature review will further help create the conceptual basis for the current project [6]. That is also what happened in our case, as the literature review inspired us and made the foundation for our RQs.

To gain broader insight and supplement our prior knowledge, we have further studied the research context for this thesis by reading existing literature and empirical studies. This has resulted in chapter 2, which is a summary of our theoretical background. It is essential to avoid significant gaps in the knowledge and therefore seek to retrieve more knowledge on the research context [42].

Formulating the Preliminary Semi-Structured Interview Guide

Phase number three of developing an interview guide consists of formulating the interview guide by making a draft of the interview questions. This guide is supposed to aid the conversation and ensure that the research context is thoroughly discussed during the interviews. The European Institute for Gender Equality (EIGE) points out that to avoid gender bias in the creation of questions; "clarity and wording of questions so that no room is left for interpretation based on the respondents' (and in the case of one-to-one interviews) interviewers' personal beliefs and social norms" [34, p. 12]. Further, it is fundamental that the guide is well-thought-out, as it affects the interviews and further analysis of the retrieved data [42]. We, therefore, made sure we kept our RQs in mind while formulating the guide.

Preliminary, five main topics for the questions were identified. Firstly, enough data about the interview subject must be collected to justify their role as experts. Secondly, it was desirable to get some insight into the company that the subjects work for regarding the company's current motivation and achievement in gender balance. Further was considered attractive to get the interviewee's input on some of the challenges discovered for women in IT during the literature review Figure 3.3. The last two topics align directly with the RQs and seek input on exploring and improving the development process with diminishing challenges for women and improving the resulting software artifacts.

First, the subjects were eased into the interview by prompting them about general facts concerning their experience in the field. They were asked about their current role to establish their present responsibility and how many years they have been in the industry. Further, they were inquired as to what development methodologies they are familiar with. In conclusion, they were asked what gender they identify with without setting any constraints on potential answers. These answers can be seen concerning their experiences and outlook on the current situation of women in development processes.

The second part enlightens the gender data and actions of the company where the interviewee works. First, the participant is enquired on percentages of developers in the company and specifically female developers in the company. No specific numbers on the size or location of the company are requested to ensure the participants' anonymity. Further questions seek to attain what, if any, actions the company has implemented to increase or retain women in their company and if the interview subject has noted any differences due to the actions in question. These data are helpful to see both in relation to each other to get an overview of the company's frame of reference and the participant's inkling of the problems presented later in the interview.

The third section of the interviews takes on the development process. The participants are first asked about any differences they have noticed in how a development process advances when there are no women present versus when there are women on the development team. Subsequently, they are asked what changes they think can be implemented in development processes to fit women's way of development better.

The fourth section of the interview contains the challenges discovered for women in it from section 3.3.3. These challenges are bipartite and, as such, divided into two categories. Where the first consists of challenges that can occur when women interact with colleagues, such as gender bias and code acceptance in code reviews. The second part is concerned with women's self-efficacy and tackles the problems of imposter syndrome and gender stereotypes. EIGE [34] points out the value of using the correct terms and definitions that reflect gender issues, so for these questions, the context of the problem is stated before the interviewees are asked if they can think of a way to solve it.

The fifth and final section of the interview addresses women's impact on software artifacts. The first part takes on what views women bring to the development of software artifacts. As a follow-up, the participant is inquired about actions that can ensure women's input on software artifacts.

Pilot Testing of The Interview Guide

Once the preliminary interview guide is finished, following Kailo et al.'s five phases, it is then time for testing. The purpose of phase four is to work out if the drafted interview questions give answers that are relevant to the research and that they together provide a good enough coverage. This phase also provides the opportunity to make improvements based on the potential weaknesses or errors discovered during the testing, for example, in terms of adding, removing, or reformulating the questions [42].

During our testing phase, we performed both internal testing and expert assessment as measures to see if we could uncover any sections or questions that should be improved in the interview guide. Internal testing comprises evaluating the guide in collaboration with the research team [9], and that is what we did. One of us played the role of the interviewee, while the other performed the interview according to the guide. This way, we got to experience first-hand how it was to be asked the planned interview questions. According to Chenail's report, [17] on strategies in qualitative research, this is also a valuable alternative to pilot testing. This approach to testing also provided us with an impression of how the more personal or sensitive questions should be asked. As Chenail also expresses [17], this made us more prone to acting responsible and further helped us identify ethical concerns that should be kept in mind throughout the interviews. An example would be when asking for the participant's gender; one should not suggest or set limitations for the participants' answer.

Expert assessment is concerned with showing the preliminary interview guide to someone outside the research team that has knowledge in the research context, and its given field of expertise [42]. In our case, expert testing was conducted on our co-supervisor. As he was involved in the process of developing the literature review, he had a lot of insight and prior knowledge of the research context around our RQs and the interview questions. In alignment with what Barriball et al. [9] writes, this was beneficial to evaluate if the interview guide was appropriate for its purpose and if it would give the coverage we needed in regards to what we wanted. It also presented us with the opportunity to discuss the relevance, ordering, and formulations of the preliminary interview questions [9].

Presenting the Complete Semi-Structured Interview Guide

In the fifth and final phase of developing the interview guide, the guide should be complete and ready to be presented in the research publication [42]. We managed to produce a semi-structured interview guide we were delighted with throughout the four preliminary phases and the corresponding work conducted. We used the guide to aid in collecting data from

12 participants. The complete and finalized 'Interview Guide' can be studied further in Appendix B.

4.3 The Interview Process

Prior to the interviews, all the participants received information about the project and the interview through an 'Information Letter' (attached in Appendix A). This letter also contained more detailed information on how we would process and store their data. A 'Consent Form' was included at the end of the letter. This form had to be signed by the participants as an agreement to participate in our research. Once we received their written consent, we scheduled a time for the interview, followed by a meeting invite through Microsoft Teams.

According to Oates [56] it might be a good idea to send the interviewees a list of the topics or questions prepared for the interview to give them the time to think about their views. This was to allow the interviewees to prepare for the interview by reading the questions in advance and reflecting upon the topics addressed in the interview. In addition to the meeting invitation, the participants received our 'Interview Guide' (attached in Appendix B). The interview guide contains a summary of the purpose of this study and the questions planned for the upcoming interview.

The interviews were conducted digitally using Microsoft Teams. Throughout the interview process, both of us participated as interviewers, each responsible for our respective sections of questions as a measure to simulate a typical conversation between all parties better and make the interview feel less rigid and artificial for the interviewee. We also participated in all 12 interviews, either joining from the same computer or individual computers at different locations. As all of the participants are a part of the Norwegian IT industry, the interviews were mainly performed in Norwegian to put the interviewees at ease and use their natural vocabulary to elaborate on their experiences. There was one exception where English was the preferred language of the participant. Starting off the interview, we introduced ourselves to those who did not already know us. We engaged in some small talk to get the conversation going and hopefully make the interviewee more relaxed. All the measures mentioned above were made to make the interviewees feel as comfortable as possible and that they could trust us. According to Oates [56] this is essential, especially before asking more sensitive questions.

After some chattering, we introduced the background for this project as formulated in the 'Interview Guide' to set the context around the interview and remind the participant why their answers are valuable to us. We reminded them that they would be recorded through audio and video as soon as the central part of the interview started. According to Oates [56] informing the participants about the purpose of the research and that they will be recorded of great importance when it comes to online interviews. The interviewees were also encouraged to open the 'Interview Guide' themselves to better keep up with the line of questioning throughout the interview session. From this, the rest of the interviews went about, more or less, just according to our guide. In some cases, due to time limitations on specific interviews, some questions were skipped; this is also one of the characteristics of semi-structured interviews [56]. The questions left out were agreed upon beforehand and considered less necessary in maximizing the total coverage from our base of questions. In alignment with what is suggested by Oates [56], the interviews were closed off by inviting the interviewees to revisit previously discussed questions or address any additional topics. We also made sure to once again thank them for participating in our research project [56].

4.4 Data Analysis

Qualitative data analysis was conducted to process the data collected through semi-structured interviews. Qualitative data comprises all data that is not numeric, such as words, sounds and images [56]. This data was generated by a case study and collected through interview recordings in our case. As stated by Oates [56], once you have generated some qualitative data, the next step is to analyze it.

"Most qualitative data analysis involves abstracting from the research data the verbal, visual or aural themes and patterns that you think are important to your research topic" [56, p. 276]. Although Oates further states that there are no strict rules for qualitative data analysis, we have read his advice on approaching the process and tried to follow his guidelines. As a method for analyzing the qualitative data, the thematic analysis will be used as it is appropriate for comprehending experiences and opinions through a data set [44].

4.4.1 Transcription

The first step described by Oates [56] is data preparation which means making the qualitative data ready for analysis by getting all the material in the same format and stored in the same place. All our interviews were conducted and recorded using Microsoft Teams. This tool provides functionality that automatically transcribes the sound throughout the meeting stored as a text file. Therefore, resulting from each interview, we were both a video recording and a transcription of the session. These transcriptions are usually not accurate and error-free but were used as a solid foundation in preparing the data. Instead of transcribing from scratch, these auto-generated text files worked as an outline that could be corrected according to the video recordings. The motivation for getting all material in the same format is to ease the process further, both by making the collected data material more manageable and easier to skim through.

”The goal of transcribing is to be as true to the conversation as possible, yet pragmatic in dealing with the data” [10]. From this, Bazeley [10, p. 73] addresses issues that might arise when working with transcription and introduces some general guidelines as to what to keep in mind throughout the process. Below we have listed some guidelines inspired Bazeley [10] that we felt were the most important once; hence we decided to follow them when transcribing our interviews.

- Transcriptions will include all oral terms, repetitions, and general oral errors. We had to beware that these might indicate something more. For example, if the interviewee responded with ’uhm’, this might indicate hesitation or uncertainty.
- We did not complete or correct incomplete sentences or poor grammar in the transcriptions because the sentences and how they are composed are a part of how the given interviewee expresses himself/herself. It is essential to capture this.
- We included comments and questions from ourselves, the interviewers, just as they were phrased in the given interview. Even though we followed our pre-made ’Interview Guide’ to the letter, the questions might not be verbatim. As the wording of the questions might vary a bit from time to time, they are necessary to transcribe to provide an exact context around the data collected.
- We did not find it a necessity to include all digressions from the topic in our transcripts as it, in general, is a controversial issue. However, unless the material was of explicit significance to the project, it was skipped and not transcribed in detail—the theme and duration of the digressions were, however, clearly noted as part of the transcripts.
- We always made sure to keep a copy of the original transcriptions provided to us by Microsoft Teams and a copy of the finished transcriptions in their natural order. We always had the conversations available in their original form, how they happened sequentially, rather than rearranging and coded according to categories and topics.

The interviews were conducted in Norwegian; they were also transcribed and further analyzed in Norwegian. Snippets from different interviews considered relevant for the sake of the results and analysis in this thesis were, however, translated to English and are presented throughout chapter 5. As the interviews in their entirety were not translated, they are not attached to this thesis. Additionally, it proved challenging to anonymize the interviews as a whole, such that interviewees would not be recognized, which makes another reason for not publishing the transcribed files.

4.4.2 Thematic Analysis

Bazeley [10] emphasizes the importance of getting to know the entire set of data collected before deciding on themes for coding the interviews. The themes should take both the collected data and theories uncovered in the preliminary work with the thesis into account [10]. Further, Kiger and Varpio state that "Themes are actively constructed patterns (or meanings) derived from a data set that answer a research question, as opposed to mere summaries or categorizations of codes. Themes can be generated inductively or deductively." [44]. In the development of the thematic analysis, six steps inspired by Kiger and Varpio [44] were followed: get to know the data, create initial codes, search for themes, assess themes, name and define themes, and produce results.

In the first step of getting to know the data, the transcriptions were conducted before all interviews were read thoroughly by both authors, providing an exhaustive overview of the raw data. Consequently, the second step consisted of creating initial codes where the critical points of the interviews were discussed and potential typical demeanors and preliminary ideas presented. The inductive approach was utilized during the discovery, and the data was tried by putting it into the initial themes. By entering the third phase of searching for themes, they were derived from the transcribed interview data by identifying the main topic of each section in each interview and thus providing an extensive and comprehensive analysis of the data [44]. Further, the discovered themes were assessed for having a clear connection to the data and that a code covered all relevant data. Similar themes were merged while assessing that the relating data maintained appropriate coherence. After concluding the partitioning of themes, the following step is to group them before naming them consistently and descriptively and setting a description. The full table of names and descriptions can be seen in Table 4.1. Finally, the analyzed data is used by analyzing and extracting data from each theme until the saturation of each theme.

Codes/Themes

The themes and codes used for the thematic analysis of data.

Name/Theme	**Description**
General info	General info about interview subjects
Work title	The interview subjects work title
Years of experience	The interview subjects years of experience in IT
Development methodology	The development methodologies the interview subject has used
Gender	The interview subjects gender
Company	Group for themes related to the interview subjects company
Percentages	Percentages on female presence in company
Events for women	Events or actions that the company has introduced to combat gender inequality
Results for events for women	Results of the introduced events or actions
Challenges	Group for themes of challenges women can meet in IT
Gender bias	Interviewees perception of gender bias in IT
Imposter syndrome	Interviewees perception of imposter syndrome in IT
Stereotypes	Interviewees perception of gender stereotypes in IT
Code review	Interviews perception of code reviews affected by gender
Development process	Group of themes relating to the development process
Women in processes	What women bring to development processes
Improvements	How to improve gender equality in development processes
Software artifacts	Group of themes related to women affecting software products
Women affecting products	How women impact software products

Improvements	Changes to software development methodologies to secure women's input on software artifacts
Assessment	Interview subject's thoughts on topic
Perceived gender differences	Perceived differences between men and womens input on topic
General thoughts about the topic	Interview subjects general thought about subject
Other valuable input	All other input that does not fit in another theme, but is still deemed relevant for the thesis.

Table 4.1: The thematic topics used for coding interviews

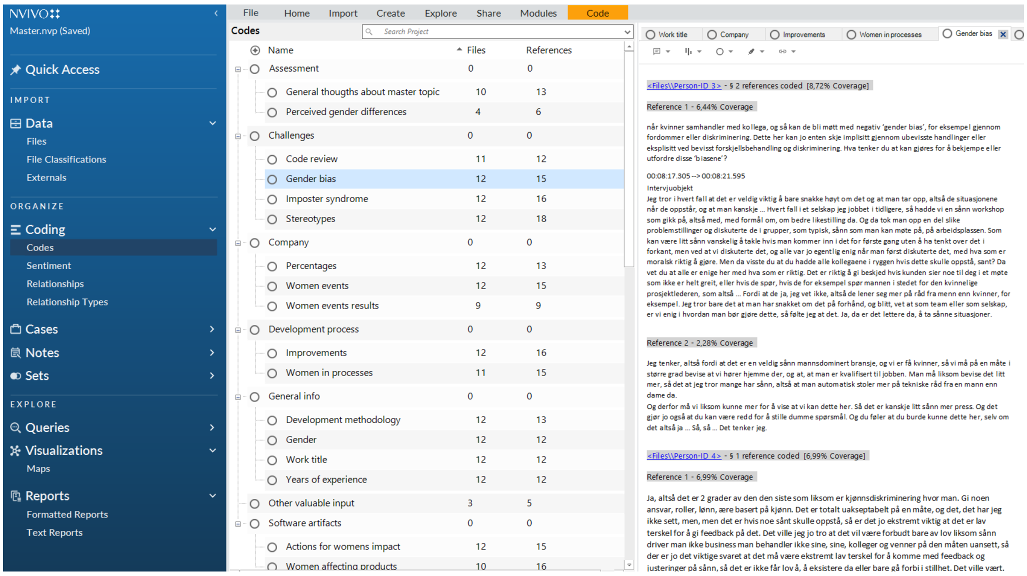
The interviews were coded into seven main themes with 18 sub-codes. The themes were created by uniting both a deductive and inductive approach, meaning that the themes are based on both preliminary theories and new themes revealed by the data itself.

4.4.3 NVivo

We started by researching tools for analyzing our data. There exist software programs that can be used and aid the process of analyzing qualitative data [56]. As this comprises both free and commercial software, Oates [56] suggests inspecting what possibilities are offered to us through our institution. We discovered that NTNU could provide us with licenses for the software tool NVivo [4]. NVivo is a tool that can be used to support both qualitative and mixed methods of research and is ”designed to help you organize, analyze and find insight in unstructured, or qualitative data like interviews” [67]. As this description fitted our chosen research method, we decided to use this software for our analysis. In addition to offering licenses, NTNU also provided us with user help on how to use NVivo, in terms of online resources and courses.

The thematic analysis was conducted using NVivo and was implemented from the second step introduced in section 4.4.2 in the first iteration of code creation. The transcribed interviews could easily be uploaded to NVivo as word files. NVivo gives the option to create a project that collects all files and codes.

Most of the software programs for analyzing qualitative data are mainly focused on textual qualitative data, hence providing facilities specialized for dealing with that, such as text



Name	Files	References
○ Assessment	0	0
○ General thougths about master topic	10	13
○ Perceived gender differences	4	6
○ Challenges	0	0
○ Code review	11	12
○ Gender bias	12	15
○ Imposter syndrome	12	16
○ Stereotypes	12	18
○ Company	0	0
○ Percentages	12	13
○ Women events	12	15
○ Women events results	9	9
○ Development process	0	0
○ Improvements	12	16
○ Women in processes	11	15
○ General info	0	0
○ Development methodology	12	13
○ Gender	12	12
○ Work title	12	12
○ Years of experience	12	12
○ Other valuable input	3	5
○ Software artifacts	0	0
○ Actions for womens impact	12	15
○ Women affecting products	10	16

<Files\\Person-ID 3> - § 2 references coded [8,72% Coverage]

Reference 1 - 6,44% Coverage

når kvinner samhandler med kollega, og så kan de bli møtt med negativ 'gender bias', for eksempel gjennom fordommer eller diskriminering. Dette her kan jo enten skje implisitt gjennom ubevisste handlinger eller eksplisitt ved bevisst forskjellsbehandling og diskriminering. Hva tenker du at kan gjøres for å bekjempe eller utfordre disse 'biasene'?

00:08:17.305 --> 00:08:21.595

Intervjuobjekt
Jeg tror i hvert fall at det er veldig viktig å bare snakke høyt om det og at man tar opp, altså de situasjonene når de oppstår, og at man kanskje … Hvert fall i et selskap jeg jobbet i tidligere, så hadde vi en sånn workshop som gikk på, altså med, med formål om, om bedre likestilling da. Og da tok man opp en del slike problemstillinger og diskuterte de i grupper, som typisk, sånn som man kan møte på, på arbeidsplassen. Som kan være litt sånn vanskelig å takle hvis man kommer inn i det for første gang uten å ha tenkt over det i forkant, men ved at vi diskuterte det, og alle var jo egentlig enig når man først diskuterte det, med hva som er moralsk riktig å gjøre. Men da visste du at du hadde alle kollegaene i ryggen hvis dette skulle oppstå, sant? Da vet du at alle er enige her med hva som er riktig. Det er riktig å gi beskjed hvis kunden sier noe til deg i et møte som ikke er helt greit, eller hvis de spør, hvis de for eksempel spør mannen i stedet for den kvinnelige prosjektlederen, som altså … Fordi at de ja, jeg vet ikke, altså de lener seg mer på råd fra menn enn kvinner, for eksempel. Jeg tror bare det at man har snakket om det på forhånd, og blitt, vet at som team eller som selskap, er vi enig i hvordan man bør gjøre dette, så følte jeg at det. Ja, da er det lettere da, å ta sånne situasjoner.

Reference 2 - 2,28% Coverage

Jeg tenker, altså fordi at det er en veldig sånn mannsdominert bransje, og vi er få kvinner, så vi må på en måte i større grad bevise at vi hører hjemme der, og at, at man er kvalifisert til jobben. Man må liksom bevise det litt mer, så det at jeg tror mange har sånn, altså at man automatisk stoler mer på tekniske råd fra en mann enn dame da.

Og derfor må vi liksom kunne mer for å vise at vi kan dette her. Så det er kanskje litt sånn mer press. Og det gjør jo også at du kan være redd for å stille dumme spørsmål. Og du føler at du burde kunne dette her, selv om det altså ja … Så, så … Det tenker jeg.

<Files\\Person-ID 4> - § 1 reference coded [6,99% Coverage]

Reference 1 - 6,99% Coverage

Ja, altså det er 2 grader av den den siste som liksom er kjønnsdiskriminering hvor man. Gi noen ansvar, roller, lønn, ære basert på kjønn. Det er totalt uakseptabelt på en måte, og det, det har jeg ikke sett, men, men det er hvis noe sånt skulle oppstå, så er det jo ekstremt viktig at det er lav terskel for å gi feedback på det. Det ville jeg jo tro at det vil være forbudt bare av lov liksom sånn driver man ikke business man behandler ikke sine, sine, kolleger og venner på den måten uansett, så der er jo det viktige svaret at det må være ekstremt lav terskel for å komme med feedback og justeringer på sånn, så det er ikke får lov å, å eksistere da eller bare gå forbi i stillhet. Det ville vært.

Figure 4.1: A screenshot from the NVivo workbench

search and coding [56]. NVivo is one of these tools and allows for coding the specific content of any memo (document) [10]. This is also something we experienced as a great advantage, as we could quickly adapt the use of NVivo to our specific needs. Figure 4.1 shows an outline of the NVivo workbench as it was used during the thematic data analysis of the interviews. It provided the appropriate flexibility as it was easy to add, delete or rename codes and group them. When coding an interview, one could mark the appropriate paragraphs and drag and drop them into a code. In order to access the coded materials, a code can be opened to show all marked paragraphs related to the code in the same document.

4.4.4 Quality Assurance

Translation comprises the transcribing of text in its original language into a target language [30] taking sentences or whole sections from the already transcribed Norwegian files and transcribing them into English. The translation is not straightforward; hence, considerations and measures had to be taken to provide quality assurance. One of the considerations taken was concerning local terms and phrases. Both the Norwegian language in general, but specifically within the different dialects represented by the different interviewees, there exist terms and words that do not necessarily have an exact equivalent in the English language [10]. To find a solution to this problem, making sure no data got 'lost by translation', it was therefore deemed essential to identify and beware these local (indigenous) terms and phrases to determine their meaning. This coincides with what Filep [30] points out in his paper on Interview and translation strategies: one needs to beware of the meaning of the given terms and phrases to be able to use them in the correct context and to translate them precisely.

Proofreading was another measure implemented during the translation process regarding quality assurance. In this study, proofreading consisted of cross-checking the translations for errors by reading the Norwegian and English versions of the same sentence or paragraph in parallel, making sure the meaning from the Norwegian version was expressed in its entirety through the English version. The cross-checking was done twice by first re-examining our work before verifying each other's work.

Chapter 5

Results

The following chapter will present the results found through the conduction of 12 semi-structured interviews, as described in chapter 4. The goal of the interviews was to gain insight into the IT industry and collect data based on the experiences and perceptions of the interviewees in order to investigate further and provide answers to the RQs stated in this Master's thesis. Six different sections are presented to introduce the results, each corresponding to its section from the 'Interview Guide', as shown in Appendix B. The sections will be presented in chronological order starting with the general information on the interview subjects.

5.1 Interview Subjects

The results presented in this Master's thesis are made up of expert opinions of representatives from the IT industry. Before introducing their opinions, it is essential to provide some background information as a general presentation of the interview subjects. When recruiting interview subjects for this thesis, it was assessed as necessary to choose subjects with different prerequisites, experience, and from different companies. This was to gain a diverse and representative view of the issue. The interviewees consisted of 12 representatives from nine different companies in the Norwegian IT industry, and they retained a minimum of five to 40 years of experience in the field. Out of the representatives, four identified as male and eight as female. More women were included because they seemed to be more engaged and interested and hence have more opinions. This was also suggested by two of the male interview subjects (P04 and P06). Table 5.1 gives an overview of the general information gathered on the distinct interview subjects. In the final column of Table 5.1 agile approaches as Scrum are added in parenthesis where the interview subject specified which approaches they had more experience with.

The subjects reported experience with different development methodologies, but all stated to have used an agile software development approach. Most of the interview subjects also noted that they did not follow any methodology 'by-the-book' but instead adapted some variation, i.e., Scrum, to fit their team's needs. Many of the interviewees had experience with Kanban, and some with more traditional approaches, such as Waterfall. DevOps was mentioned once, but this might not represent the field. One subject also mentioned using PRINCE2.

PID	CID	Years in IT	Role	Gender	Development methodologies
P01	C1	10	Project leader	Female	Agile (Scrum), DevOps
P02	C2	5	Developer	Male	Agile (Scrum)
P03	C2	12	Project leader	Female	Agile (Scrum, Kanban)
P04	C3	13	Developer	Male	Agile (Scrum, Kanban)
P05	C3	22	Project leader	Female	Agile (Scrum, Kanban)
P06	C4	9	Developer	Male	Agile, Waterfall
P07	C5	25	Project leader	Male	Agile, Waterfall
P08	C6	8	Developer	Female	Agile (Scrum, Kanban), Lean
P09	C7	20	Project leader	Female	Agile, PRINCE2
P10	C8	7	Developer	Female	Agile (Scrum, Kanban)
P11	C9	40	Project leader	Female	Agile, Waterfall
P12	C4	10	Developer	Female	Agile (Scrum)

Table 5.1: The general information on interview subjects. (PID=Person ID, CID=Company ID)

5.2 Companies

The interview subjects are currently working at nine different companies, and the following section will present what the interviewees expressed concerning their companies statistics regarding female employees, and their input on events and activities for female employees.

5.2.1 Female Presence

All interview subjects disclosed statistics about the female presence at their company. Table 5.2 shows the companies' total female percentage, in addition to the percentage of female developers within each company. The percentage of female developers is significantly lower than the total number of female employees for most companies. The exceptions are companies that consist purely of developers, like C5 and C8. These also have some of the lowest percentages of females overall.

Company ID	**% of females in company**	**% of female developers in company**
C1	20%	20%
C2	17%	12%
C3	42%	20%
C4	30-40%	20%
C5	6,25%	6,25%
C6	29.6%	28.8%
C7	29 %	21%
C8	9-10%	9-10%
C9	46%	39%

Table 5.2: The presence of females and female developers in interview subjects' companies

5.2.2 Events or Activities for Women

C1, C2, C3, C4, C6, C7, and C8 all have an exclusive network for the women within their company. This can comprise online networks or set meetings or events during the year. The same companies also participate in events or conferences where women in IT are the main topic. These are TENK, Oda network, Ada network, SHE awards, and "Kvinner i Tech" (Women in Tech). The main goal of these groups is to increase the female presence in the companies.

P07 states that C5 has not had any specific activities or groups for women.

P11 states that C9 has come to see employees as individuals without paying much attention to their gender and that it would go against their way of thinking to create something exclusively for women or men. P11 further states that she believes events exclusively for women might enhance the differences instead of erasing them. It would narrate the havoc if one today were to create a similar event exclusively for men. Further, representatives from C2 point out that differentiating between women and other employees might increase the gap and lead to women feeling alienated.

P05 highlights that C3 also consciously has a flat salary structure to reduce occurrences of the gender pay gap.

Results of events or activities for Women

P01 states that she and her colleagues take pride in the investment put in by her company (C1) to increase and empower women in IT. She further notes that the company has successfully recruited more female developers without being able to attest this solely to the environment they have built for women in the company. P03 notes that it is valuable to have a group in the company responsible for following women. P05 and P09 both share the views on increased recruitment for women in C3 and C7, with P05 highlighting that more female leaders in C3 have increased the focus on recruiting female developers.

P10 points out that the events for women in her organization led her to become acquainted with women in other roles that she might not have otherwise. This has further led her to widen her network of women within the organization, as there are few women in development roles in C8.

P02, P03, and P12 have not noticed any effects of the initiatives for women in their companies, C2 and C4. However, the actions and events were implemented recently.

5.3 Challenges for Women in Software Development

The following section will present the interview subject's views and experiences related to the different challenges discovered in the literature review, described in chapter 3.

5.3.1 Gender Bias

Most of the interview subjects highlighted that they had not perceived problems with gender bias in their companies. However, P12 presented a case where she felt she was not accepted as a developer by the team due to her gender. She described the situation as difficult to handle and was unsure whether to quit or address the issue. "When I started in the company that I work in now, I had not worked there for more than four months before I was unsure whether this was somewhere I would bother to stay. I felt like the team did not accept me, and this was to the highest degree related to my gender, that I am a woman. So I thought that either I give this up, or I have to confront the issue and try." She decided to fix the situation and raised the issue with her leader. They then had an open and honest conversation involving the whole team, addressing the issue. In the end, this turned out to resolve the problem.

The most frequently proposed action to tackle gender bias among the interviewees is raising awareness and recognizing the problem. Several interviewees mention going through workshops or taking courses. They further remark that it is easier to identify the issue and prepare a clear strategy after seeing explicit examples of bias. P03 also stated that it was helpful to get the chance to discuss real issues with colleagues to make sure everyone is on the same page and has a well-established strategy on how to handle issues that might arise. P09 identifies the two steps of first becoming aware of potential biases one might be affected by, and secondly realizing in which situations these biases might be present.

P04 points out that it might be hard to be aware of one's own implicit biases, so it is essential to establish an environment where one is encouraged to bring attention to bias or other issues. Therefore, it would be beneficial to have good customs for providing both negative and positive feedback. P06 also points out that calling out gender bias is a shared responsibility for the whole team and that it is not acceptable for other team members to remain silent if gender bias occurs. P08 and P12 remark that it is essential to have the courage to stand up for oneself and tackle the problem at its core.

P01 presents that combating gender bias will demand a long-term systematical approach where one acknowledges women's work by giving them responsibility, which is also pointed out by P04. Further, P01 thinks more female leaders will lead to a more diverse workgroup, providing different points of view and more acceptance for the differences. This might bring around different solutions that, in the end, are valuable for more people. Due to this, she has faith in focusing on diversity and diversity quotas as a solution to obtain this. Other interviewees mention the gender quotas, as introduced in section 2.2.1, as a possible reason for the gender biases that we face today. P02 mentions that this might create a partition between men and women and give the impression that it is easier for women to get admission to IT studies. The interview subject also presented that he thinks the work to diminish gender bias should start already at the university level.

P03 stated that: "Because it is such a male-dominated industry, we have to prove, to a higher degree, that we belong there, and that we are qualified for the job... I think many like, that one automatically have higher trust in technical advice from a man than from a woman."

5.3.2 Imposter Syndrome

P01 remarks that many women she has worked with seem to have very high standards for what they deliver and feel a greater responsibility than other male colleagues for delivering something beyond fault. In addition, she notes that many women are more prone to admit their insecurities, making them seem less in control. P03 points out that women might feel more pressured to show that they belong in IT because it is male-dominated. P01's proposed solution is to first and foremost trust the processes to check deliveries, i.e., 'QA' and testing. Further, she points out that it is crucial to take the time to acknowledge and reward oneself for a job well done because others rarely do. P2 also acknowledges this problem and states that one rarely gets feedback when doing an excellent job. He sees providing positive feedback as a point of improvement for everyone involved in the process.

Women might be underestimating themselves, according to P05. She observes that women rarely apply to new positions unless they meet almost every requirement in the job description, while many men confidently apply, having only 20% of the skills required. To combat this, she suggests that women should try to push outside their comfort zones more frequently and keep in mind that the worst case is that they get a rejection or have to ask for help to succeed. P09 also adds: "It has probably got something to do with the 'Good girl syndrome,' I think. Believing that one has to be 100% sure about something to have the right to give an opinion".

P10 points out that the industry might have to be better at valuing soft skills like communication and creating product descriptions and emphasizes that several roles are essential for an IT project to succeed. P12 also points out that it is important not to compare oneself with others because one often brings different values to the table. She encourages everyone to become more aware of the good attributes one contributes to the team. Among recent graduates, P09 has noticed that the men joining her company more often have more experience with coding from hobby projects or internships. This makes some women start to feel less competent, but P09 points out that this is just a slight difference at the starting line and usually evens out after a while.

Role models might play a more significant part in our subconscious minds regarding what we think we can achieve, P06 states. He thinks we should be better at valuing and broadcasting good female role models and, in general, encouraging a job well done in a way that makes it clear that one sees the hard work put into it. P08 also encourages women to seek out other women in the same situation or whom they feel safe around to get some support, encouragement, and motivation. P09 has been in IT for more than 20 years and promotes the responsibility leaders have to be role models. She believes that having made it to her current position makes her obligated to be honest about her doubts and lessons learned. Through hearing about difficult situations a leader might have gone through, maybe more employees in IT will be able to trust themselves in uncomfortable and new situations.

P02 points out that for some women, all the focus on getting more women into IT might feel like special treatment. P12 also mentions that it sometimes makes her feel more out of place when friends and family do not understand what she works with but give her credit for working in a male-dominated industry.

P11 thinks imposter syndrome must result from childhood and upbringing and that the current generation might have a greater need for positive feedback than earlier generations.

Many informants (P12, P04, ...) point out that imposter syndrome is also widespread among men in the industry. They seem to agree that to diminish it; one has to talk about it. Several of the informants also imply that the high occurrence of imposter syndrome in IT might be because the knowledge within the field is constantly changing with new technologies emerging and old ones evolving. IT is a specialized field but still very wide, which makes it impossible to master everything or anything completely. P04 says, "Imposter syndrome disappears when you talk about it, because you are not an imposter if it is no longer a secret. It is as if you go around waiting to be exposed at any point". Further, he establishes the importance of having a work culture where it is acceptable to express a lack of competence or confidence in one's ability to perform a task.

5.3.3 Code Review

When asked about women getting their code reviews rejected more frequently than men, almost all interview subjects stated that this was something they had never experienced. Only P01 did not give any statement on her experience with the question, and P12 said that this was indeed something she had experienced.

P01 stated that it is about trust and how thoroughly the review process is performed when reviewing something. For her, the statement indicates that female developers are less trusted than men.

P11 says: "I had a time on my team there all my 'pull requests' were not accepted and spammed with comments that did not have anything to do with the code structure [...] I got the impression that the comments were written to indicate that in the overview of 'pull requests,' mine had the most comments."

The most repeated solution suggested among the interview subjects (P12, P02) is defining what good code is. It is much easier to have an objective baseline for reviews when the whole team agrees upon clear guidelines for structure. How code reviews are conducted in a team can be discussed in a retrospective meeting if one follows the scrum methodology, as pointed out by P03. P07 notes that some people might have a bit 'sharper edges' and might not be the best person to review a new coworker's code.

Several of those who have not seen gender as an influential factor in code review rejections point out that there is an age bias in reviews. P04 says: "A cranky old man would easier get his code reviews accepted because he is old [...] There is clearly an age bias, but it also corresponds to experience and trust".

5.3.4 Gender Stereotypes

"One adopts the culture to fit in with the form of communication to be listened to, but it is a compensation for a problem, not a solution" is voiced by P01. She portrays how women in a company where she used to work acted like men to make sure their opinions were to be considered.

There are many limitations for women with small children, as stated by P01, as she feels like these limitations might be due to restrictions from stereotypes connected to motherhood. After noticing that she was not given specific opportunities because she had children, she stopped mentioning her children in her work life. She still feels like it has fewer implications for a man to say that he is staying home with sick children than for a woman.

Many women with engineering degrees within IT seem not to want to continue as developers, instead choosing to work as testers or functional architects, P10 states. She thinks many women visualize other roles as easier or less technical because it takes years to become comfortable in a role as a developer. However, there is a lot to learn no matter what role one chooses as a graduate. To solve this, she suggests following up with new employees through courses and allowing them to learn from other experienced women to build a good network.

Women can be considered better in less technical roles, such as designers or product leaders, due to stereotypical attributes for women, such as creativity and compassion, as expressed by P12 and P05. P02 further states that there are stereotypes concerning women's ability to be good backend developers and suggests this needs to be addressed through attitudinal adjustment.

P04 states that he thinks most stereotypes in IT are outdated and proved wrong. If some still exist, he proposes demonstrating that they are wrong by displaying examples that contradict the stereotype.

Women are often expected to take responsibility for the team's social activities, from what P08 and P06 have observed. P06 elaborates: "In software development or the IT field, women tend to get side-lined in many ways from the primary software development process with other activities that may take place. For example, there is a social event that has to be planned. Generally, people tend to assume that the females on the team would take responsibility for the social activities while the rest of the men are working towards the actual goal of the team." He further notices that the women in his department seem very eager to take on this responsibility. "It is important also to question why they want to do it, and double check with them to see if they want to, or if that is just their habit of saying that they can fix it."

It is essential to change the narrative of what a good programmer is, P07 states. In his company, they have tried to shift the focus from having a lone rider (programmer) working alone to more inclusive processes where more people participate in understanding the needs and requirements of the product. The change has appealed to everyone, but women seem to prosper under the new approach.

There might be more stereotypes towards women in IT from people outside the industry, P09 suggests. From the outside, women who go into IT can be seen as masculine or different for having an interest in a technical field. She sees stereotypes as a big issue to overcome and states that: "If we assess the stereotypes for women in IT as a limiting factor to get more women into IT, we have a big job ahead."

5.4 Adapting Software Development Processes

This section will introduce the expert opinions of the interview subjects on adapting software development processes. This includes presenting their thoughts and answers concerning women's approach to software development and listing their potential suggested improvements to make software development processes more inclusive.

5.4.1 Womens Approach to Software Development

Women might be more concerned with understanding the need a product is set to fill before tackling the task, as indicated by P01. P10 and P07 agree with this, stating that women might bring a more complete and detailed view to both the task and the process of solving it. P05 describes this as women providing a consensus-driven process. P12 also indicates that without women as a part of the process, it can become very tech-centered and compromise on the details of the project.

From P02's experience, adding women to the team might improve the team by following development methodologies and improving the work structure. P05 agrees with this, saying that the tone might be sharper in a team with only men. P04 further reflects that in teams with no women, fewer questions are asked, and men are less humble during the project's research phase and the process of understanding the task at hand,

Women might be better at voicing issues and asking for help, as stated by P03 and P04, which can positively affect team dynamics. The former (P03) further indicates that women can better consider and discuss different approaches before agreeing on a solution.

P11 thinks that an approach to software development is individual and not dependent on gender. P06 and P09 utter that women might have a different approach to development but have not noticed any specific differences.

P01's experiences reflect that many men seem to get more joy from coding than women. Many women need to understand the use of the product to feel content.

Women might be less inclined to take courses, get certified, and use all the technical terms, according to P09, but they often still have the corresponding knowledge. She thinks the industry must value the application of knowledge to a greater extent than certifications and vocabulary.

Several interview subjects point out that diversity in all forms is essential. "Because our experiences characterize us, we need people with different experiences to create software, and here gender is an important factor", P09 says.

5.4.2 Suggested Improvements

Some women might need prompting to provide their opinion, as stated by P10, who further expressed that it is helpful to accommodate so everyone gets the opportunity to say something without speaking up. P05 agrees with this statement but also says that women might also have to be better at speaking out without being prompted.

Including more people in all parts of the development process and diminishing individualistic thoughts about having one great coder in a team is deemed significant by P07.

P11, P02, P04, P05, P06, and P12 state that they do not think there is a need for improving software development processes specifically for women. P03 agrees, voicing that agile methodologies are created to be adjusted to the team and give a voice to the individuals.

Most of the interviewees also expressed that their suggested improvements, although targeting women, would benefit most people.

5.5 Womens Impact on Software Artifacts

This section will present the interview subjects' expert opinions on the topic of women's impact on software artifacts. Several interview subjects repeated statements they had voiced at previous stages during the interviews. They felt like the questions raised here were similar to the once asked regarding women's approach to software development in section 5.4.1.

5.5.1 Effect on Resulting Software Artifacts

Women are better at keeping the bigger picture in mind while simultaneously focusing on the details of the project, according to P10 and P12. Both P06 and P12 had experiences developing large products where women raised issues related to universal design that benefited the resulting product. P03 stated that women are better at raising essential discussions about a product and the process and are, in general, better at interacting within the team. When prioritizing the user's needs, P05 thinks women are better at listening without making assumptions about these needs.

P04 highlights that it is complicated to see whether achievement is dues to the gender of the developer or just individuality. He said, "I am not sure if it was because she was a woman who underlined the right part of the equation or because she was a sharp analyst [...]. There are many examples from other places where women have affected the product, but I am not sure if I want to attribute their ideas to their gender rather than being brilliant analysts." However, he has noticed that many women dare to ask questions that might be considered simple but that, in some cases, are what is needed to unveil undisclosed issues.

Many women seem to be perfectionists to a more significant degree than men, leading them to be thorough and reduce the possibilities for error in the processes they are involved in, according to P08.

"It is as elementary as ensuring equal diversion among the people supposed to make the products as those who will use them. Furthermore, that is why gender balance is important. A huge amount of new solutions is needed, and many of the challenges ahead are connected to occupations where the majority are women," was stated by P09. She further expressed that the interests and experiences play a part in what we focus on and prioritize and thereby affect the products we create. For example, she questions why there are no more good solutions in the health sector.

5.5.2 Ensuring Women's Impact on Software Artifacts

Many of the interview subjects pointed out good preparatory work, including user insight and frequent user testing with all user groups represented, as a necessity when there are no women present in the project's development. P11 highlights that involving users through user testing has become more critical in the past decades, and further that it is crucial to use tests if there are specific user groups with different needs involved. "If you are creating an end product for nurses at home for the elderly, you might consider using more women than men in the user testing and preparatory work because they are over-represented in the industry," was expressed by P09.

P01 emphasizes that women need a chair at the table to be able to influence the product. P10 agrees with this and voiced that women can be included in the development process through other roles than as developers.

Several interview subjects found these questions concerning women's impact on software artifacts challenging to answer. They felt it might be misleading to attribute good development choices to gender rather than individual qualities.

5.6 Additional Input

The interviews and their line of questioning brought up various feedback and additional input from the interview subjects. Their thoughts and opinions on the overall topic and the perceived gender differences will be presented throughout this section.

P01 was very enthusiastic about the topic of this Master's thesis, stating that women in IT development have been left off the agenda for way too long. P05 shared this perception and further stated that the results would be helpful in more industries than just IT.

The female developers with the least experience, P10, and P08, initially stated that they were unsure if they had much to contribute to the prepared questions. However, P10 concluded that her insight and experiences in the industry might not be as evident to others as she had assumed.

P03 points out that she thinks the issues addressed in the interview are important but that she has noticed more problems with, for example, the gender pay gap. She further points out that this can also affect software development processes when a female developer knows that an equally competent male developer earns significantly more.

P07 and P06 stated that they felt inspired to further work with addressing issues concerning gender equality within their organizations after participating in the interview.

When asked some of the questions, the interview subjects are inquired about characteristics or abilities that set female developers apart in software development. P04 points out that it is helpful to generalize women to a group, making it easier to process and potentially solve the problem. This action also enhances the gender differences, which is somewhat counterproductive. He further states that the better solution is to ensure we achieve a better gender balance in education.

P11 is the participant that comes from the company with the highest percentage of women in this research. She states that, in her opinion, it is more important to attract the right individuals to software development than to attract more women specifically.

"It all depends a lot on my insecurities, too, with being new in business in a male-dominated field. I also completely understand that women disappear from the industry. I am afraid that is hard to miss.", was voiced by P11.

P09 concludes with: "We need all kinds of people to create the products that all kinds of people will use. It is as elementary as to have an equally large variation among those who will create the products as those who will use them, and gender equality is so important because there are so many problems ahead that we need to solve."

At the end of the interview, P04 stated that: "I think that I would have interviewed both young and more experienced women because they have probably felt the consequences of this to a more considerable degree." He further questions men's input on the topic and voices: "It is easy for a man to say that women are not being discriminated against, so is it relevant to hear men's input here?".

Chapter 6

Discussion

This chapter contains discussions regarding the findings presented throughout this Master's thesis, intending to provide answers to the previously stated RQs. The following sections will discuss the results from our research, introduced in chapter 5, and reflect upon these in relation to the theory and previous findings, presented in chapter 2 and chapter 3.

The structure of the discussion is based on how the results as presented in chapter 5, starting with addressing the challenges listed in section 5.3 in the same order. Further, the discussion will revolve around the software development processes and the resulting artifacts in regards to women's participation and their impact. All the relevant challenges and topics will be discussed together with their corresponding results and previous findings before a suggested module and add-on will be introduced. After this, some additional proposals will be listed, followed by a section that describes potential threats to validity.

As previously mentioned in chapter 2, research reveals agile approaches to software development have shown an increase in popularity within the IT industry over the last decade. This also aligns with the results on what methodologies the interviewees in this study were familiar with, as they all reported to have previously worked using agile approaches. Even though the interviews were set to discover possible ways to improve software development to solve the retention problems and diminish barriers for women, the results and possible add-ons are set to involve the whole team and will be beneficial to all.

6.1 General Discussion

The results show diverse and disputed opinions on gender equality and how to reach it. Perhaps the most controversial point was to what degree the issues were present in the Norwegian IT industry. Some of the interview subjects viewed gender equality as a conquered issue, while others had recent encounters with challenges presented in the interview. The fact that one interview subject had experienced several of the challenges addressed through the interview and had considered quitting her job due to these, emphasizes the necessity of addressing them.

Seternes [69] indicated similar retention problems among women in the Norwegian IT industry to those discovered in other countries. Furthermore, it proved difficult to find female developers with a minimum of five years of experience as interview subjects for this thesis. It was easier to locate female project leaders. One of the female developers that were interviewed also addressed women choosing to leave roles as developers for less technical ones as a stereotype for the field, indicating this is a widespread issue.

Despite the disputed opinions on the general topic of gender equality and how to solve it, the study reached saturation on some suggestions to improve software development processes. In particular, many agreed and presented similar solutions to the challenges identified for gender equality and how to ensure women's impact on software.

Based on the interview subjects' reflections and suggestions on the different topics presented through the results in chapter 5, six distinct modules are proposed. Table 6.1 gives an overview of these modules, their goal, as well as the topic or challenge that inspired them. The following sections will further elaborate on these modules and what motivated them.

Challenge/Topic	**Module**	**Goal**
Gender Bias	Anti-Bias Training Module	Raising awareness and challenge biases within the development team
Imposter Syndrome	Feedback Module	Emphasize feedback more explicit to improve self-efficacy among team members
Code Review	Coding Standard Module	Establishing a common coding standard within the team to make sure there is no dispute on how to examine code reviews
Gender Stereotypes	Work Allocation Module	Allocating all types of tasks collectively within the team to provide transparency and more control on who is responsible for what

Women's Impact on Software Development	Clarification of Requirements Module	Ensure a common understanding of the project requirements within the development team and establishing a shared vision of the projects purpose and users needs
	Extended User Testing Module	Compensate lack of diversity within the development team through extended testing, to ensure more diverse software products

Table 6.1: The challenges and topics with suggested modules and their corresponding goals

6.2 Gender Bias

Gender bias was the challenge most frequently identified in the literature review performed this fall [13]. The interview subjects point out that raising awareness on the subject might be the best way to address it. Diversity training or anti-bias training can be used to raise awareness and can reduce implicit gender bias towards women [40]. Further, the results show that discussing possible biases or issues with coworkers can be beneficial and make it easier to raise issues that might occur and agree on a code of conduct. The team, and the organization as a whole, can accomplish this through workshops or discussions. Among the identified issues causing retention problems among women, hostile macho cultures [36] were introduced in section 2.2.1. By putting gender bias on the agenda at an early stage of a new team or when introducing a new team member to an existing team, both cultural issues and bias can be addressed and pointed out as a priority from the start.

Anti-Bias Training Module Raising awareness and putting gender bias on the agenda is a measure that could challenge and have a beneficial impact on the issue. In general, when discussing bias, one should be introduced to how it might occur and that there are different types, namely implicit and explicit bias. It can be easier to understand bias by being provided with examples of different kinds of realistic situations where it might manifest. This way, one can get a clearer picture of the problem, which provides a better foundation for further reflection and discussion on how one should address similar situations. Moreover, this leads to identifying different biases one might be affected by, in what scenarios they might occur, and how to deal with the biases strategically. This comprises both own biases and other team members' biases.

An Anti-Bias Training Module, consisting of realistic examples as described in the paragraph above, would be beneficial to implement in any development process. Especially in cases where the team is working together for the first time, or a new team member has arrived. The module could easily be adapted to any development process through the planning stage. The main goal of this stage is to prepare for the process moving forward, both in terms of deciding on technologies and tools and also deciding upon what software methodology to implement. Introducing the Anti-Bias Training Module through this stage would not only help with raising awareness and hopefully challenge any potential biases within the team but also affect the team environment, making every member feel comfortable and safe. Again, this would be beneficial for making it easier to raise issues that might occur as the process evolves.

The training module could be conducted using many different formats, either it is implemented like a crash course or using a more thorough approach, like through a workshop. Available time and resources are valuable aspects that should be considered when deciding what approach to take. However, the foundation for the module is fundamental, and it is crucial that it targets to reveal both implicit and explicit biases within the team. Further, the training module must be tailored to fit the given project and its needs.

6.3 Imposter Syndrome

Imposter Syndrome is another challenge that was identified through the literature review [13]. This challenge is faced by many women in IT, but also other industries, and concerns that the individual feels like their success comes as a result of pure luck, rather than talent and work [18].

Several of the interviewees point out that women seem to hold themselves to a higher standard in terms of their work, which can lead to a constant chase for perfection. The interview subjects emphasize the importance of acknowledging great work to alter this. That is, both in regards to rewarding oneself and recognizing each other. Taking the time to give positive feedback is rarely done and is addressed as a point of improvement in development processes in general. In addition, the results point out that imposter syndrome can be combated by voicing insecurities to coworkers in a supportive work environment. Chrousos and Mentis also point out acknowledging high performance and discussing the topic within the organization as essential [18] for combating imposter syndrome.

Feedback Module Feedback needs to be addressed more explicitly, and it needs to be put on the agenda to make sure it is not neglected and to benefit the team members' self-efficacy. That is why giving feedback should be introduced as a regular and standard procedure within the team and as part of the development process.

Scrum introduces the concept of retrospective meetings, which aims to improve the development process moving forward. This meeting gathers the entire team and provides an arena for collective reflections and evaluation of their work so far. A setting like this could be advantageous in the context of giving feedback, and the provided surroundings should be utilized to implement even more feedback throughout the development process. Whether the project implements Scrum or not, one should implement the Feedback Module to provide the means for a properer feedback session. Suppose Scrum is the software methodology implemented, and the development team already has scheduled retrospective meetings at the end of each sprint. In that case, this module can be integrated into these existing meetings directly. For other methodologies, this module can either be introduced as a small evaluation meeting scheduled at the end of a work period or added to another regular meeting or event that seems fitting. Anyhow the meeting is implemented, it should provide the means for a feedback session that nourishes a positive attitude. The main goal of the feedback sessions should be to give positive feedback on work well done. This comprises encouraging team members to acknowledge each other's work as well ass encouraging self-praise. It is also important to emphasize that everyone participates throughout this meeting by giving credit to both one self and other team members. The supportive environment provided through the feedback sessions is also suitable for team members to raise their voices about their fears or insecurities. An example is addressing a situation where one did not feel confident in completing a task.

The Feedback Modules should be planned ahead and adapted as a part of any given development process. The team should also customize the module according to their schedule and preferences to make it low effort and as convenient as possible for everyone involved in the project.

6.4 Code Review

In the SMS [13] code acceptance was one of the challenges identified that correlated the most to software development. As introduced in section 2.2.1, research showed that women have a higher chance of getting a pull request accepted when their code does not state their gender. This can be seen as a specialized case of gender bias. Through their interviews, almost all interview subjects expressed that they had not experienced this issue and further asked for the research behind the question.

In the SMS, Albusays et al. [3] was the source presenting the challenge cited from a study performed by Terrell et al. [75] on code acceptance in open source. As only one source on the topic has been uncovered, and it targets open source rather than standard software development processes, the foundation for this question could be more solid. However, the research indicates that gender bias in code reviews is significant as the reviewer is less inclined to accept code, knowing that it was a woman who wrote it. The research was conducted on open source code reviews, but this might also prove to be a good data set as the reviewer has very little personal info about the coder exceeding the chosen name that, in some cases, imply gender. This led to a slight change in question 6 as visualized in Table 6.2. The change is highlighted in red and was implemented from interview five and in the consecutive interviews.

Initial Question	**Changed Question**
”Women experience more problems with getting their code accepted than men. How can this be addressed or avoided in software development processes?”	”Women can experience more problems with getting their code accepted than men. How can this be addressed or avoided in software development processes?”

Table 6.2: The change for question 6 in the interview guide

However, the challenge was still relevant, and one of the interview subjects stated that she had indeed experienced problems getting her code accepted.

Coding Standard Module Several interview subjects proposed that to address problems surrounding gender bias in code reviews, the development team should have a predefined and standardized approach to code structure. By providing a common standard, there should be no dispute on how to examine the code during the reviews.

Extreme Programming introduces a set of simple and independent software development practices that can be implemented in any project. One of these is ’Coding Standards’, which is a well-known principle in the IT industry. It concerns the entire development team, and is in XP suggested as a measure to provide good communication and coordination within the team, as well as provide a solid end product of high quality. The practice involves for the team to define guidelines for what they acknowledge as good code structure and thereby determining these through establishing a common code standard. All team members should follow the coding standard throughout the entire development process. Also, the standard should work as guidelines when examining code, providing explicit instructions for what to look for when conducting code reviews. Implementing coding standards, in addition to commenting code, is an essential part of the development stage of any general software development process, intending to make it less complicated for other developers to understand each other's code.

The coding standard for a given project should be established upon the actual development such that all code is written using the same guidelines. The planning stage is the fundamental part of every software development project that aims to set everything in order before the project development can begin. Therefore, this stage makes for a suitable place to insert a module for discussing coding standards within the team. As the planning stage also consists of deciding upon what software development methodology to implement, it provides an excellent opportunity to integrate the Coding Standard Module into the chosen methodology. The module should be adapted as a part of the planning stage and can be fitted to any chosen approach to development. The main goal is to provide the team with an arena and the time to establish coding standards for their upcoming project.

6.5 Gender Stereotypes

Gender stereotypes are perceptions of a group of people that generalize characteristics of gender. One of the stereotypes that were discovered for women in IT is that they are often expected to take charge of social activities or other domestic-like duties that might arise within the team. This can consist of anything from being responsible for planning the summer party to organizing birthday presents for coworkers or cleaning up after a shared lunch. Taking on this responsibility shifts their focus from the team's primary goal and persists gender stereotypes about women as domestic providers. One of the interview subjects noted that women often volunteered for these tasks and questioned whether they felt expected to do so or genuinely do enjoy it. If the few women in IT take on these tasks, it will be at the expense of something else, either it is their development tasks or their free time. Either way, this will affect their overall work performance and workload.

Work Allocation Module Allocation of all tasks within the development team should be distributed fairly and allocated openly and collectively to provide more control on who is responsible for what. This applies to both development tasks that target the team's overall goal as well as more detached tasks concerning social activities or other domestic-like duties.

During the planning phase of a software development project, requirements for the project are broken down into smaller parts or tasks to make the work at hand more comprehensive. These tasks are further distributed within the development team to be completed, thereby working towards their shared goal of fulfilling the requirements. However, this distribution of tasks should not only take care of the development tasks but also incorporate mandatory supplemental tasks. Making sure all kinds of tasks are distributed collectively within the team would prevent individuals from volunteering to do additional work on top of their technical work. This would also stop them from taking on too many tasks, potentially affecting their spare time or delaying progress in the development process.

Work allocation is crucial for any well-functioning team, but how the teams carry out this distribution varies. This often depends on what software development methodology is implemented in the project. Introducing a Work Allocation Module that takes all types of tasks into consideration should be easily adaptable to any software development project, as it will only be utilizing and supplementing an already existing procedure. The team should conduct the regular work allocation process as usual within the development process and according to their own standards. However, incorporating the additional tasks will slightly reform the process.

Kanban introduces a well-known and commonly implemented tool to monitor and manage the workflow of a software development project. The tool, known as the Kanban Board, also provides a visual overview of all tasks and their current status of progress. Aiming to make work allocation even more open and transparent within a team, one could implement the Kanban Board, or a similar tool, as a part of the Work Allocation Module. As the board is an advantageous way of visualizing work and workload, this would provide more transparency into the process, making it even more evident to everyone who takes on which tasks. This will also make it easier for the team to keep track of each others tasks and hold all team members accountable for the tasks they have signed up for. Altogether, the introduction of the Work Allocation Module will hopefully contribute to individuals feeling less obliged to take on more tasks than others by creating a more significant threshold for them to do so. Even though the tasks are not development tasks and detached from the overall development process, they should be just as visual and clearly stated as a part of the overall workload.

One of the Kanban Board characteristics is visualizing potential bottlenecks in the development process, hence enabling optimization of tasks. Implementing this tool, or a similar tool, into the module would therefore also prove to be beneficial for work allocation in general.

6.6 Women's Impact on Software Development

One of the goals for this Master's thesis was to investigate how software development processes unfold with women present compared to how they do when women are not. Further, also looking at how software products are affected by the lack of women and what measures can be taken to ensure gender-inclusive products.

The first point to note is the controversy around whether women have a different approach to software development. It is a disputed topic, and can, as some of the interview subjects pointed out, feel marginalizing and stereotyping to reduce a diverse gender to a common set of characteristics. However, it is difficult to address the issue of a skewed gender distribution without discussing the differences between the genders. The IT industry's current acquisitiveness for women should be seen in relation to what women bring to the processes and what the potential pitfalls are if they remain a minority.

The results show that several of the interviewees agree that women often are more concerned with understanding the purpose of the product and the details of the product requirements before starting developing the technical solution. Further, women might be better at initiating structure to the development methodology events and generally improve the communication within the team. Some interviewees also expressed that it is challenging to attribute qualities or contributions to a project purely to a person's gender.

The interviewees also repeated the importance of diversity in the development team in order to ensure diverse solutions for the users. To ensure the diversity in the solution when the team is more uniform, the interviewees stressed the importance of good predatory work to understand the product's needs and perform user tests. Some also pointed out that user tests should aim to be diverse but still seek out the target group of the application.

Clarification of Requirements Module Understanding the product description and the requirements of a software product is crucial to deliver a solution that satisfies the product owner. Therefore, it is essential for the development teams to thoroughly comprehend the purpose of the software product while also gaining an adequate understanding of its user group. This should be more emphasized within software development projects, as it would benefit all involved parties—the development team, the product owner, and the project manager.

Within the research phase, at the beginning of a software development project, the product owner, the development team, and the project manager get together to exchange information about the product and discuss the outlines of the project. This sets the product owner to formulate and hand over the product requirements for the given project.

Once the development team receives the technical requirements, these are evaluated from a technical perspective. It is crucial that the team is aligned in their understanding of the requirements and has a clear sense of the project's overall purpose in order for them to proceed towards a common goal. Holding a meeting within the development team to clarify expectations before initializing further planning and development of the project would aid the team in creating a shared vision of the project at hand. This would ensure that all involved team members share the same understanding of the project's purpose and further what needs to be done to fulfill the requirements set for the resulting software product.

Having a common understanding of the work at hand is crucial within any development team to ensure progress and make sure all individual members contribute to working towards the common goal. Introducing a Clarification of Requirements Module that emphasizes the importance of this would therefore be beneficial to all software development projects. The module should be implemented as a short meeting to verify that the team's understanding of the product requirement is aligned and that they all share the same vision. This will serve as a cross-check before planning further development in the next phase of the project.

Implementing a meeting to clarify project understanding within the research phase should require minimal effort from the development team. The meeting should not have a strict structure and could be included in existing sessions or arranged independently. The research phase is fundamental in every software development project and is considered one of the seven general and distinct stages. This reasons for the Clarification of Requirements Module to be easily adaptable to any software development methodology. Integration of the module aims to ensure common understanding within the development team and a shared vision of the project at hand.

Extended User Testing Module Providing a diverse software product is highlighted as important and can be seen as an extension of thoroughly understanding product descriptions. Further, the results in chapter 5 repeatedly pointed out the importance of diversity within the development team to ensure a diverse product for the end users. This applies generally and to all types of diversity, and the lack of involvement from women throughout development processes is one of these scenarios. As a measure to compensate in such scenarios and make up for the lack of diversity within the development team, user testing should be even more emphasized and thoroughly planned as a part of the software development processes.

Throughout the initial stages of a software development project, the development team would be able to map out the diversity naturally provided both within their team and the target user group. This presents the opportunity for the team to plan for a more fitting and comprehensive testing regime already in the planning stage. The focus should be for the user testing to include multiple user groups, aiming to bring more diversity into the development process. The goal is to enable perspectives and visions not naturally represented within the development team to also be heard and potentially make an impact. Further, the aim is to avoid potential pitfalls that would result from not including them. Extended user testing, including user groups not naturally present within the development process, allows them to influence the resulting software product through feedback.

An Extended User Testing Module is suggested, as an additional component in the planning phase, to ensure more diverse software products. The planning conducted within this module should not affect an already existing testing plan but simply extend it by including more testing. Even though the Extended User Testing Module is suggested to be implemented within the planning stage of a software development project, the testing itself could be facilitated and conducted during later stages of the development. Integrating an Extended User Testing Module aims to ensure a more diverse software product by securing the involvement of user groups not represented naturally throughout the development process. This would also comprise making sure the products are gender-inclusive.

6.7 Additional Proposals

After presenting the suggested extensions to software development processes in the section above, there are still some additional takeaways on gender equality from this thesis that does not target the software development processes. These takeaways will be presented throughout this section as additional proposals that should be highlighted to create a better overview of how the challenges the IT industry is facing can be addressed.

Role models to combat imposter syndrome and gender stereotypes. Frequently mentioned by the interview subjects as a measure to address and battle imposter syndrome and gender stereotypes was the presence of role models. Further, lack of role models is also enhanced as one of Coqual's [36] four main barriers for women. There is a clear need for women in higher technical positions to make more women see the possibility of reaching the same heights. It is also important to have role models convey how they manage to go out of their comfort zone and for them to share their lessons learned along the way.

Events for women to give support and build their network. Within the companies, events for women were implemented and participated in different ways and to different extents. However, most of them had some kind of group for women, either through digital platforms or physical meetings. Coqual [36] points out that having a mentor or sponsor is paramount to get both the support and inside information needed to progress in the industry. Female events seem to have the most value for creating networks and finding mentors.

Encourage female developers in their technical roles. Gender stereotypes can for women include the viewpoint that attributes such as creativity and compassion are female attributes, making them better suited for less technical roles than developers. Seeing this in relation to the numbers showing the rate of women that are leaving IT as described in section 2.2.1 and the claim in section 5.3.4 of few women remaining in development roles, it is clear that an attitudinal adjustment is needed.

Value soft skills as an asset in technical roles. When describing what women are especially good at, many interview subjects pointed out understanding the user's needs, seeing the complete picture, or enhancing communication within the team. These can be considered soft skills, and as a measure to attract more women and reduce retention problems, it is important to value these soft skills in the same way as other skills required in software development teams. The value of soft skills were also highlighted by Seternes [69].

Mind the pay gap as it can affect team relations. The pay gap is not strictly a part of the development process but can affect the developers and teamwork if differences in salary seem unfair. This is also why several of our interview subjects brought up and emphasized the topic.

6.8 Threats to Validity

To ensure that the case study results are valid, it is necessary to assess possible threats that can have affected the research. To determine this as objectively as possible, Runeson and Hösts *Guidelines for Conducting and Reporting Case Study Research in Software Engineering* [63] have been implemented to review the construct validity, internal validity, external validity, and reliability of the research conducted through this Master's thesis.

6.8.1 Construct Validity

Construct validity is assigned to evaluate whether the research conducted assesses what it is set out to do [63]. It has been emphasized from the beginning of the project to eliminate the possibility of construct validity. For this thesis, the biggest pitfall is that the interview subjects interpret an interview question differently than the researches intention. To avoid this, the interview guide has been developed using an acknowledged set of guidelines proposed by Kallio et al. [42] that this far has been cited close to 2000 times. Further, the interview guide contains a descriptive introduction to the thesis and the interview, and clarity was emphasized when formulating questions. However, focusing on creating unique and comprehensive interview questions based on facts from the SMS might have led to questions that seem leading. Even though the interview guide was built to avoid misinterpretation, it is a common occurrence. Especially this applies to longer conversations that demand focus, such as in a semi-structured interview.

6.8.2 Internal Validity

Internal validity is a complex factor in place to evaluate if any results can be affected by unstudied causes [63]. As the semi-structured interviews are based on the interviewees' experiences and opinions, this is highly relevant in this case study. Their personal experiences and thoughts are strongly affected by their background, age, gender, and political views, to mention some factors. However, the differences represent the natural diversity and hence contribute to creating a representative data set. It is of importance to see the results in relation to some generalized data about the subjects, which is why Table 5.1 is implemented to provide an overview of the basic data on each interview subject.

The choice of interview subjects can also have affected the outcome. A study performed by Accenture [1] shows that leaders are more than twice as likely to say that it is easy for women to prosper in tech in comparison to the womens' own view on thriving in tech. This should therefore be further evaluated in regard to the group of project leaders participating in this thesis.

6.8.3 External Validity

External validity is concerned with how relatable and representative the findings are to a larger population, and the general interest of the subject and results of the project [63]. The scope of the project was set to the Norwegian IT industry. However, most of the current research used in the background of this thesis covers larger areas such as Europe or the US, and the basis of this Master's thesis relies on this data being relatable to the Norwegian IT industry. A comparison of statistics from Norway and the US is used as an indication of their comparable levels in section 2.2. Still, a threat to validity is the lack of statistically detailed data specifically for the Norwegian IT industry.

Concerning the general interest, the literature review performed in the fall of 2021 [13] showed that the topic is gaining momentum and attention, and most of the interview subjects also showed great interest in the matter.

6.8.4 Reliability

The reliability of the study reflects how the researchers might have affected the data and analysis. The interviews in this study were conducted by the authors, who are women about to enter the IT industry, and it is essential to reflect on how this might have affected the interview objects in their answers. Some of the interview subjects had the feeling that we were after specific answers and specified that they did not think they would provide the answers they presumed we wanted. It was elucidated in advance that the study is exploratory and that there was no presumption to their answers. This initial thought might have still affected their responses.

Parts of the thesis are grounded in the high demand for women in the IT industry, and some of the interview subjects showed interest in recruiting the authors. This might have led some to try to put both the industry and their company in a better light.

Runeson and Höst describe how case studies can be affected by bias from the researchers and additionally how it can be hard to generalize from [63].

For the interviews conducted in this case study, the interview subjects were recruited through the authors' networks. None of the interview subjects had any direct relations with the interviewees. Still, there is a possibility that the interviewers' previous knowledge of the interviewees could have affected the interviewees' position on the questions asked. Even though the interviewees are a diverse group, the fact that they were recruited from the authors' network might lead them to have unknown similarities.

6.8.5 On the Topic of Gender

All interview subjects were asked to state their gender at the beginning of the interview and were free to answer with any term they preferred. All the interview subjects identified with either the female and male gender. The fact that the interview subject had to state their gender at the start of the interview might have affected their answers.

Gender is a complex and challenging topic to address. Ideally, this thesis would have considered the obstacles of all minority genders, but the literature for this thesis was based on barriers specifically for females as it is hard to find both relevant literature and interview subjects for non-binary people in the IT industry in Norway.

6.9 Implications for Research and Practice

The literature review conducted on the intersection between IT and gender equality uncovered multiple gaps in the current state of the art. These gaps were specifically concerning existing practices to address gender equality in IT processes. The review also identified a series of challenges women encounter throughout these processes. With the aim of addressing these issues and starting to fill the uncovered gaps, the modules presented in the previous sections were introduced.

The modules presented are suggested add-ons to the software development methodologies used in the IT industry today. These add-ons are made to be straightforward and general, hence easily applicable for any kind of software development team and project. Depending on what they find the most suitable, the development teams can choose to adapt which add-ons they want, either it is only one or several of the suggested ones. Anyhow, the add-on(s) implemented will work as measures to improve software development in terms of solving the retention problems and diminishing barriers for women.

Our hope is for the research and the modules resulting from this project to be of inspiration and motivation for others and potentially serve as the starting point for further research on the topic. This comprises the suggested add-ons to be implemented and tested by companies in the industry. Through feedback from real-life evaluation of the add-ons, they can further be refined to become even more effective. Meaning each of them can be adapted to become even better suited to diminish its respective challenge. Further research on the topic, nevertheless, if it turns out to be confirming or contradicting, will bring more insight and provide a more solid foundation for improvement within the state of the art.

Chapter 7

Conclusion

In the pursuit of equalizing the currently unequal gender distribution in the IT industry and increasing the number of female employees, it is important to address the turnover issues among women within the field. The retention problems that influence the industry are complex. To further explore these problems, one must assess the software development processes to investigate how they might influence women and their participation.

Through the conduction of a Systematic Mapping Study [13] several challenges regarding women's participation in IT were identified. The review further revealed no measures or processes in place today that directly aim to address or diminish these challenges, which motivated the objective of our further research. Additionally, other literature displays how an uneven gender distribution in the development of software products can lead to products that neglect diverse needs, resulting in obstacles instead of aids.

This Master's thesis, therefore, aimed to investigate software development processes and methodologies used in relation to the challenges identified for women in the IT industry and what implications the lack of female developers potentially has on the software products resulting from the processes. In the same way, this thesis aimed to further map out what actions can be taken to improve software development processes with unequal gender distribution, both in regards to optimizing the resulting artifacts and diminishing the challenges women encounter during these processes.

A case study that consisted of 12 semi-structured interviews was conducted to investigate the topic and gather data from experienced representatives from the industry. The study included participants with a minimum of five years of experience in the field and from nine different companies, where 50% were developers and 50% were project leaders. The study revealed contradicting perceptions on to what degree gender equality is an issue and how to address it in the Norwegian IT industry. While some of the participants felt that the battle for gender equality belongs to the past and that one today focuses on each person's individuality, others had personal experiences with challenges like gender bias. Some also stated they had considered leaving their job due to the current issue. The thematic analysis considered all views and statements and extracted expert opinions on common themes and ideas.

The recurring themes and ideas in the thematic analysis created the grounding for a set of modules that can be adapted by the IT industry. The modules are created to be implemented as add-ons, supplementing already existing software development methodologies and improving the processes for all parties involved. Each of these modules is created to address and try to diminish its corresponding challenge. However, the overall goal for all of them is to make the software development processes more inclusive and help increase the diversity of the resulting software products.

The interview categories and questions prepared for this study were intentionally set to discover potential points of improvement within software development processes to solve retention problems and diminish the challenges women face during these processes. Nevertheless, the suggested modules and their corresponding add-ons are beneficial to all and set to involve the entire development team. The add-ons are also created to be easily adaptable to different software development methodologies. The add-ons are fashioned to be general and can be tailored to any approach to development, making it a low effort for the team to implement. The development teams can choose to adapt which add-ons they want, either it is only one or several of the suggested ones.

As the results from the study show, gender equality is a critical aspect of ensuring diversity in the software development team and the resulting products. As a part of working toward a more gender-inclusive software development environment, further research should also include non-binary genders. Future work should also have a more in-depth analysis of retention problems among women, specifically in the Norwegian IT industry, both in terms of what makes women exit and the extent of the situation compared to other countries.

The work with this thesis has shown what an intricate topic gender equality can be. Most interview subjects in this thesis agree that gender equality in software development is an important topic, but how to achieve it or if it, in fact, already has been achieved is disputed. We hope that this thesis shows that even though gender equality in the IT industry has improved in the past decades, the current retention problems show a need for addressing issues that our results show some women still experience.

The modules introduced in this Master's thesis aim to address gender inequality in software development and are, for the time being, only suggestions that have yet to be tested on real-life projects. Further investigation and evaluation of the modules are desirable to adjust the add-ons to fit various software development methodologies within the industry. Further testing would also be beneficial to improve their efficiency towards their corresponding goals.

Bibliography

[1] Accenture and Girls Who Code. *Resetting Tech Culture. 5 strategies to keep women in tech.* 2020. URL: `https://www.accenture.com/_acnmedia/PDF-134/Accenture-A4-GWC-Report-Final1.pdf` (visited on 05/04/2022).

[2] M. O. Ahmad, J. Markkula, and M. Oivo. "Kanban in software development: A systematic literature review". In: *2013 39th Euromicro conference on software engineering and advanced applications*. IEEE. 2013, pp. 9–16.

[3] K. Albusays et al. "The Diversity Crisis in Software Development". In: *IEEE Software* 38.2 (2021), pp. 19–25.

[4] Alfasoft. *NVivo.* 2022. URL: `https://www.alfasoft.com/en/products/statistics-and-analysis/nvivo.html` (visited on 05/16/2022).

[5] J. Almestad and H. Olssen. *How to Improve Development of Teamwork Skills in Software Engineering Education through Project-Based Learning with Agile Methodologies.* Tech. rep. Norwegian University of Science and Technology (NTNU), 2021.

[6] P. Åstedt-Kurki and R. Heikkinen. "Two approaches to the study of experiences of health and old age: the thematic interview and the narrative method". In: *Journal of Advanced Nursing* 20.3 (1994), pp. 418–421.

[7] N. Atal, G. Berenguer, and S. Borwankar. "Gender diversity issues in the IT industry: How can your sourcing group help?" In: *Business Horizons* 62.5 (2019), pp. 595–602.

[8] Elsevier B.V. *Scopus.* 2021. URL: `https://www.scopus.com/` (visited on 02/02/2021).

[9] K. L. Barriball and A. While. "Collecting data using a semi-structured interview: a discussion paper." In: *Journal of advanced nursing* 18.10 (1993), pp. 328–335.

[10] P. Bazeley. *Qualitative data analysis: Practical strategies.* Sage, 2013.

[11] K. Beck. "Embracing change with extreme programming". In: *Computer* 32.10 (1999), pp. 70–77.

[12] C. Bentley. *Prince2 revealed.* Routledge, 2005.

[13] S. Block. *Gender Equality in Information Technology Processes: A Syystematic Mapping Study.* 2021. URL: `https://sbs.idi.ntnu.no/master` (visited on 02/15/2022).

[14] K. Bradley. *Understanding Prince2.* SPOCE Group, 2002.

[15] T. Breda et al. "Gender stereotypes can explain the gender-equality paradox". In: *Proceedings of the National Academy of Sciences* 117.49 (2020), pp. 31063–31069.

[16] T. E. Charlesworth and M. R. Banaji. “Gender in Science, Technology, Engineering, and Mathematics: Issues, Causes, Solutions”. In: *Journal of Neuroscience* 39.37 (2019), pp. 7228–7243.

[17] R. J. Chenail. “Interviewing the investigator: Strategies for addressing instrumentation and researcher bias concerns in qualitative research.” In: *Qualitative Report* 16.1 (2011), pp. 255–262.

[18] G. P. Chrousos and A. A. Mentis. “Imposter syndrome threatens diversity”. In: *Science* 367.6479 (2020), pp. 749–750.

[19] E. Corona and F. E. Pani. “A review of lean-kanban approaches in the software development”. In: *WSEAS transactions on information science and applications* 10.1 (2013), pp. 1–13.

[20] J. L. Cundiff and T. K. Vescio. “Gender stereotypes influence how people explain gender disparities in the workplace”. In: *Sex Roles* 75.3 (2016), pp. 126–138.

[21] W. Cunningham. *Manifesto for Agile Software Development.* 2001. URL: `https://agilemanifesto.org/` (visited on 03/16/2022).

[22] M. L. Despa. “Comparative study on software development methodologies”. In: *Database Systems Journal* 5.3 (2014), pp. 37–56.

[23] A. B. Diekman et al. “A Goal Congruity Model of Role Entry, Engagement, and Exit: Understanding Communal Goal Processes in STEM Gender Gaps”. In: *Personality and Social Psychology Review* 21.2 (2017), pp. 142–175.

[24] Digital.ai. *Digital.ai Creating a digital world customers can trust.* `https://digital.ai/company`. Accessed: 2022-04-28.

[25] Digital.ai. *The 15th Annual State of Agile Report.* 2021.

[26] W. DuBow and A. S. Pruitt. *NCWIT Scorecard: The Status of Women in Technology. Boulder, CO: NCWIT.* 2019.

[27] C. Ebert et al. “DevOps”. In: *Ieee Software* 33.3 (2016), pp. 94–100.

[28] United Nations Department of Economic and Social Affairs. *The Worlds’s Women 2020: Trends and Statistics.* 2021. URL: `https://unstats.un.org/unsd/demographic-social/gender/index.cshtml#worldswomen` (visited on 08/02/2022).

[29] R. Eveleth. *How Self-Tracking Apps Exclude Women.* 2014. URL: `https://www.theatlantic.com/technology/archive/2014/12/how-self-tracking-apps-exclude-women/383673/` (visited on 05/19/2022).

[30] B. Filep. “Interview and translation strategies: coping with multilingual settings and data”. In: *Social Geography* 4.1 (2009), pp. 59–70.

[31] Norsk senter for forskningsdata. *NSD.* 2022. URL: `https://www.nsd.no/` (visited on 02/16/2022).

[32] N. A. Fouad and M. C. Santana. “SCCT and underrepresented populations in STEM fields: Moving the needle”. In: *Journal of Career Assessment* 25.1 (2017), pp. 24–39.

[33] E. Freeman. *DevOps for dummies.* John Wiley & Sons, 2019.

[34] EIGE - European Institute for Gender Equality. *Gender Statistics and indicators.* 2019.

[35] J. L. Glass et al. “What’s so special about STEM? A comparison of women’s retention in STEM and professional occupations”. In: *Social forces* 92.2 (2013), pp. 723–756.

[36] S. A. Hewlett et al. *Athena Factor 2.0: Accelerating Female Talent in Science, Engineering Technology.* Coqual, 2019.

[37] M. Hicks. *Programmed inequality: How Britain discarded women technologists and lost its edge in computing.* MIT Press, 2017.

[38] M. Hron and N. Obwegeser. “Scrum in practice: an overview of Scrum adaptations”. In: *Proceedings of the 51st Hawaii International Conference on System Sciences.* 2018.

[39] Z. Huang and L. Yuan. “Gender differences in tourism website usability: An empirical study”. In: *International Conference of Design, User Experience, and Usability.* Springer. 2017, pp. 453–461.

[40] S. M. Jackson, A. L. Hillard, and T. R. Schneider. “Using implicit bias training to improve attitudes toward women in STEM”. In: *Social Psychology of Education* 17.3 (2014), pp. 419–438.

[41] K. H. Judy. “Agile Values, Innovation and the Shortage of Women Software Developers”. In: *2012 45th Hawaii International Conference on System Sciences.* 2012, pp. 5279–5288. DOI: `10.1109/HICSS.2012.92`.

[42] H. Kallio et al. “Systematic methodological review: developing a framework for a qualitative semi-structured interview guide”. In: *Journal of advanced nursing* 72.12 (2016), pp. 2954–2965.

[43] A. I. Khan, M. Qureshi, and U. A. Khan. “A Comprehensive Study of Commonly Practiced Heavy & Light Weight Software Methodologies”. In: *arXiv preprint arXiv:1202.2514* (2012).

[44] M. E. Kiger and L. Varpio. “Thematic analysis of qualitative data: AMEE Guide No. 131”. In: *Medical teacher* 42.8 (2020), pp. 846–854.

[45] B. Kitchenham. *Procedures for Performing Systematic Reviews.* Keele, UK: Keele University, 2004, pp. 1–26.

[46] H. Kniberg. *Scrum and XP from the Trenches.* C4Media, 2015.

[47] H. Kniberg and M. Skarin. *Kanban and Scrum-making the most of both.* C4Media, 2010.

[48] M. Kramer. “Best practices in systems development lifecycle: An analyses based on the waterfall model”. In: *Review of Business & Finance Studies* 9.1 (2018), pp. 77–84.

[49] L. Lindstrom and R. Jeffries. “Extreme programming and agile software development methodologies”. In: *Information systems management* 21.3 (2004), pp. 41–52.

[50] G. S. Matharu et al. “Empirical study of agile software development methodologies: A comparative analysis”. In: *ACM SIGSOFT Software Engineering Notes* 40.1 (2015), pp. 1–6.

[51] Microsoft. *Microsoft Teams.* 2022. URL: `https://teams.microsoft.com/` (visited on 01/15/2022).

[52] S. M. Mohammad. “DevOps automation and Agile methodology”. In: *International Journal of Creative Research Thoughts (IJCRT), ISSN* (2017), pp. 2320–2882.

[53] C. A. Moss-Racusin et al. “Gender bias produces gender gaps in STEM engagement”. In: *Sex Roles* 79.11 (2018), pp. 651–670.

[54] United Nations. *Goal 5: Achieve gender equality and empower all women and girls.* 2021. URL: `https://www.un.org/sustainabledevelopment/gender-equality/` (visited on 11/11/2021).

[55] Statistics Norway. *Employment, register-based.* 2008-2021. URL: `https://www.ssb.no/en/statbank/table/08536/?rxid=446f3516-6812-4c6f-8430-73e2c33b56de`.

[56] B. J. Oates. *Researching information systems and computing.* Sage, 2005.

[57] Version One. *The 10th Annual State of Agile Report.* 2016.

[58] Version One. *The 11th Annual State of Agile Report.* 2017.

[59] Samordna opptak. *Universitet og høgskole - Kjønnspoeng.* 2022. URL: `https://www.samordnaopptak.no/info/opptak/opptak-uhg/poengberegning/legge-til-poeng/kjonnspoeng/` (visited on 02/05/2022).

[60] C. Pincemaille. “Prince 2: A methodology of project management”. In: *Cork, Ireland: Cork Institute of Technology* (2008).

[61] United Nations Development Programme. *Goal 5: Gender Equality.* 2021. URL: `https://www.undp.org/sustainable-development-goals#gender-equality` (visited on 02/11/2021).

[62] W. W Royce. “Managing the development of large software systems: concepts and techniques”. In: *Proceedings of the 9th international conference on Software Engineering.* 1987, pp. 328–338.

[63] P. Runeson and M. Höst. “Guidelines for conducting and reporting case study research in software engineering”. In: *Empirical software engineering* 14.2 (2009), pp. 131–164.

[64] S. Saeed et al. “Analysis of Software Development Methodologies”. In: *International Journal of Computing and Digital Systems* 8.5 (2019), pp. 446–460.

[65] F. Schmitt et al. “Spotlight on Women in Tech: Fostering an Inclusive Workforce when Exploring and Exploiting Digital Innovation Potentials”. In: *Proceedings of the 2020 International Conference on Information Systems (ICIS 2020).* 2020.

[66] K. Schwaber. “Scrum development process”. In: *Business object design and implementation.* Springer, 1997, pp. 117–134.

[67] Norwegian University of Science and Technology. *NVivo.* 2022. URL: `https://i.ntnu.no/wiki/-/wiki/English/NVivo` (visited on 05/05/2022).

[68] F. Y. Sebele-Mpofu. “Saturation controversy in qualitative research: Complexities and underlying assumptions. A literature review”. In: *Cogent Social Sciences* 6.1 (2020), p. 1838706.

[69] G. Seternes. *Tech Women, Where Did You Go?* 2021. URL: `https://betterprogramming.pub/tech-women-where-did-you-go-5d5c662fb47d` (visited on 01/28/2021).

[70] R. Singh et al. “Stemming the tide: Predicting women engineers’ intentions to leave”. In: *Journal of Vocational Behavior* 83.3 (2013), pp. 281–294.

[71] A. Smeding. “Women in science, technology, engineering, and mathematics (STEM): An investigation of their implicit gender stereotypes and stereotypes’ connectedness to math performance”. In: *Sex roles* 67.11 (2012), pp. 617–629.

[72] S. Stewart-Williams and L. G. Halsey. “Men, women and STEM: Why the differences and what should be done?” In: *European Journal of Personality* 35.1 (2021), pp. 3–39.

[73] T. Streule et al. “Implementation of scrum in the construction industry”. In: *Procedia engineering* 164 (2016), pp. 269–276.

[74] R. Tatman. “Gender and dialect bias in YouTube’s automatic captions”. In: *Proceedings of the first ACL workshop on ethics in natural language processing.* 2017, pp. 53–59.

[75] J. Terrell et al. “Gender differences and bias in open source: Pull request acceptance of women versus men”. In: *PeerJ Computer Science* 3 (2017), e111.

[76] C. Thompson. *The Secret History of Women in Coding.* 2019. URL: `https://www.nytimes.com/2019/02/13/magazine/women-coding-computer-programming.html` (visited on 04/18/2022).

[77] Kantar TNS. *Andelen kvinner i norsk IT-bransje.* 2019.

[78] L. R. Vijayasarathy and C. W. Butler. "Choice of software development methodologies: Do organizational, project, and team characteristics matter?" In: *IEEE software* 33.5 (2015), pp. 86–94.

[79] M. Virmani. "Understanding DevOps & bridging the gap from continuous integration to continuous delivery". In: *Fifth international conference on the innovative computing technology (intech 2015).* IEEE. 2015, pp. 78–82.

[80] A. Wroblewski. "Quotas and Gender Competence: Independent or Complementary Approaches to Gender Equality?" In: *Frontiers in Sociology* (2021), p. 167.

[81] W. Zayat and O. Senvar. "Framework study for agile software development via scrum and Kanban". In: *International journal of innovation and technology management* 17.04 (2020), p. 2030002.

[82] M. K. Zehnter and E. Kirchler. "Women quotas vs. men quotas in academia: students perceive favoring women as less fair than favoring men". In: *Frontiers in psychology* 11 (2020), p. 700.

Appendices

A Information Letter

Are you interested in taking part in the research project

"Gender equality in software development processes"?

This is an invitation for you to participate in a research project where the main purpose is to study gender equality in software development processes. This document will give you more detailed information about the purpose of the project and what your participation will involve.

Purpose of the project

The main purpose of this project is to study gender equality in software development processes. The goal of this study is to further map out what actions can be taken to improve software development processes with unequal gender distribution, in regards to optimising the resulting artefacts and diminishing the challenges women encounter during the processes.

This study is conducted by two computer science students, and is a part of a master thesis.

Who is responsible for the research project?

This project is a part of a master thesis written at Norwegian University of Science and Technology (NTNU). The students writing the master thesis will be Hanne Olssen and Sunniva Block, supervised by Professor Letizia Jaccheri.

Why are you being asked to participate?

You are being asked to participate in this interview because you are within one of the target groups for this research project. The scope of this project targets subjects who have been a part of a software development team for a minimum of five years. Additionally a requirement is for the target to have had a technical role within the team or have been the project leader of the team.

What does participation involve for you?

If you chose to take part in the project, this will involve participating in a digital interview that lasts approximately 30 minutes. The interview will mostly consist of questions defined prior to the interview, but some follow up questions might occur. The questions will be divided into different topics, targeting different aspects of software development processes and gender equality in the industry. The predefined questions will be available to you in advance of the interview, to provide you with the opportunity to prepare some thoughts on the given topics. Your answers will be recorded digitally, by sound and video recording.

Participation is voluntary

Participation in the project is voluntary. If you chose to participate, you can withdraw your consent at any time without giving a reason. All information about you will then be made anonymous. There will be no negative consequences for you if you choose not to participate or later decide to withdraw.

Your personal privacy – how we will store and use your personal data

We will only use your personal data for the purpose(s) specified in this information letter. We will process your personal data confidentially and in accordance with data protection legislation (the General Data Protection Regulation and Personal Data Act).

In regards to accessing the personal data, it will only be the listed students and supervisor who will gain access. To ensure no unauthorised access to the data, your name and contact info will be replaced with a code. This code and the corresponding name and contact info will be stored at a different location from the rest of the collected data. Name and other personal data will be anonymized in the publication, to make sure that you, as a participant, will not be recognizable.

What will happen to your personal data at the end of the research project?
The project is scheduled to end 31.12.2022, and all the data material collected for the purpose of this project will be anonymized by this date.

Your rights
So long as you can be identified in the collected data, you have the right to:

- access the personal data that is being processed about you
- request that your personal data is deleted
- request that incorrect personal data about you is corrected/rectified
- receive a copy of your personal data (data portability), and
- send a complaint to the Data Protection Officer or The Norwegian Data Protection Authority regarding the processing of your personal data

What gives us the right to process your personal data?
We will process your personal data based on your consent.

Based on an agreement with NTNU, Data Protection Services has assessed that the processing of personal data in this project is in accordance with data protection legislation.

Where can I find out more?
If you have questions about the project, or want to exercise your rights, contact:

- NTNU - Norwegian University of Science and Technology by Letizia Jaccheri, on email: (letizia.jaccheri@ntnu.no) or by telephone: +47 73 59 34 69.
- NTNU - Norwegian University of Science and Technology by Hanne Olssen, on email: (hannols@stud.ntnu.no).
- NTNU - Norwegian University of Science and Technology by Sunniva Block, on email: (sunnivbl@stud.ntnu.no).
- Our Data Protection Officer, Thomas Helgesen, on email: thomas.helgesen@ntnu.no or by telephone: +47 93 07 90 38.
- Data Protection Services, on email: personverntjenester@sikt.no or by telephone: +47 53 21 15 00.

Yours sincerely,

Letizia Jaccheri, Project Leader	Hanne Olssen, Sunniva Block
(Researcher/supervisor)	(Students)

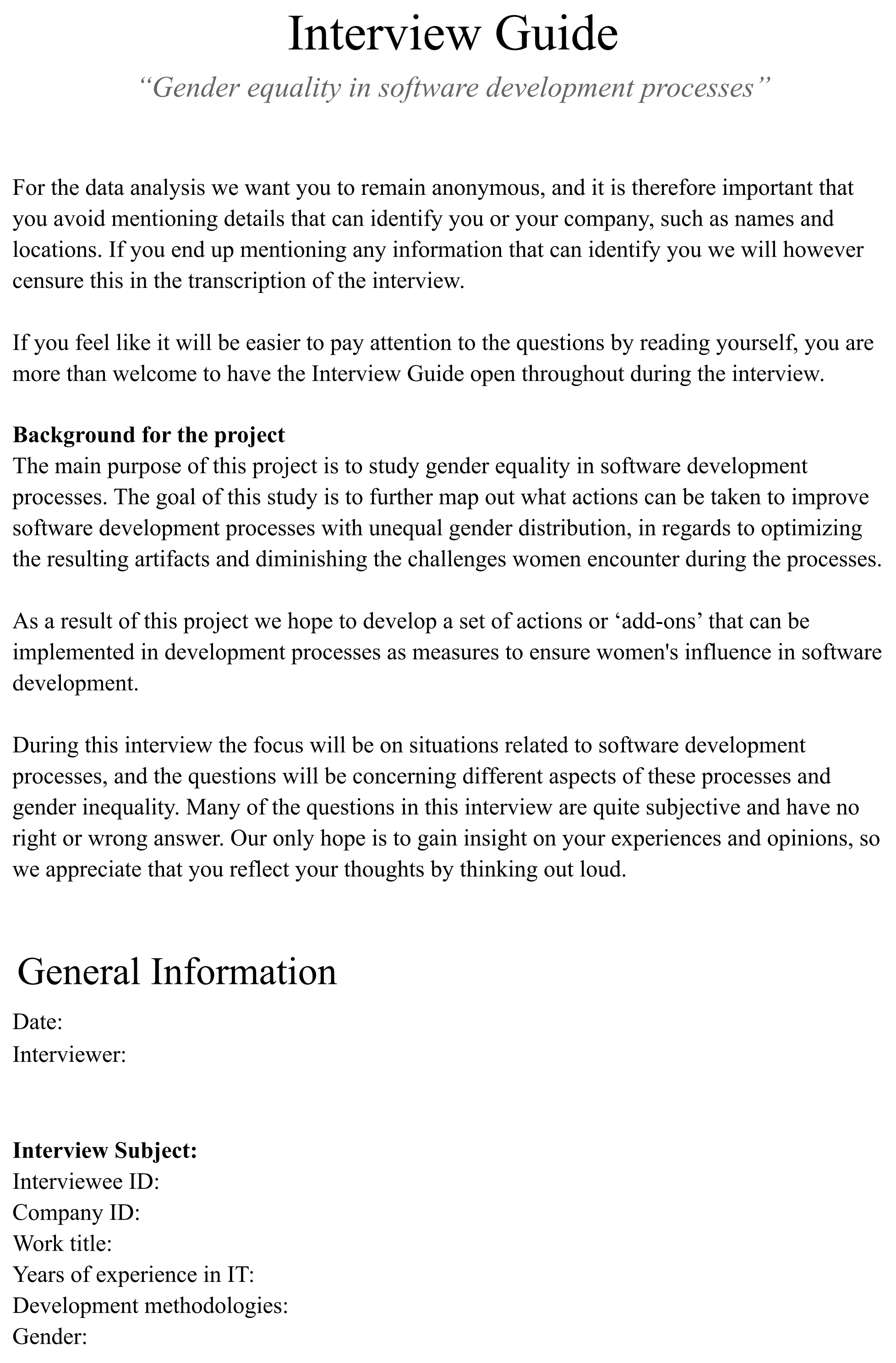
Declaration of consent to participate in research

I have received and understood information about the project "Gender equality in software development processes" and have been given the opportunity to ask questions. I give consent:

- To participate in an interview
- For my personal data to be processed until the end date of the project (31.12.2022).

(Signed by participant, date)

B Interview Guide



Interview Guide

“Gender equality in software development processes”

For the data analysis we want you to remain anonymous, and it is therefore important that you avoid mentioning details that can identify you or your company, such as names and locations. If you end up mentioning any information that can identify you we will however censure this in the transcription of the interview.

If you feel like it will be easier to pay attention to the questions by reading yourself, you are more than welcome to have the Interview Guide open throughout during the interview.

Background for the project
The main purpose of this project is to study gender equality in software development processes. The goal of this study is to further map out what actions can be taken to improve software development processes with unequal gender distribution, in regards to optimizing the resulting artifacts and diminishing the challenges women encounter during the processes.

As a result of this project we hope to develop a set of actions or ‘add-ons’ that can be implemented in development processes as measures to ensure women's influence in software development.

During this interview the focus will be on situations related to software development processes, and the questions will be concerning different aspects of these processes and gender inequality. Many of the questions in this interview are quite subjective and have no right or wrong answer. Our only hope is to gain insight on your experiences and opinions, so we appreciate that you reflect your thoughts by thinking out loud.

General Information

Date:
Interviewer:

Interview Subject:
Interviewee ID:
Company ID:
Work title:
Years of experience in IT:
Development methodologies:
Gender:

Interview

The company:

1. What percentage of your company are women?
   a. What percentage of the software developers currently working at your company are women?
2. Does your company offer any activities, programs, social events etc. for encouraging women in IT?
   a. If yes, have you noticed any effects of the activity in your company?

The development process

3. In your opinion, does the development process advance in a different way in your teams when there are no women present versus when there is one or more?
4. Have you at any point taken measures or adjusted the development process to ensure that the women on your team can fulfill their potential?
   a. Is there any part of the development process that you feel like could be adjusted to fit better with women's approach to development?

Colleagues and team members:

5. When interacting with colleagues women can face negative gender bias. This can happen both implicit through instinctive behavior, or explicit through discrimination.
   a. What do you think can be done to diminish these biases?
6. Women can experience more problems with getting their code accepted than men.
   a. How can this be addressed or avoided in software development processes?

Women's self-belief (self-efficacy):

7. Many women in the IT industry are affected by imposter syndrome, which can be described as crediting an accomplishment to luck rather than talent and work.
   a. Why do you think this affects women in IT more than other industries?
   b. What measures do you think can be taken to help women overcome their imposter syndrome in the software development process?
8. Many women are also affected by the fear of confirming gender stereotypes, where this fear can stop them from taking actions that they otherwise would have done.
   a. Are you aware of any gender stereotypes in software development?
   b. What do you think can be done to combat these stereotypes?

Software artifacts

9. Are there any particular views you have noticed that women bring to software development that can positively affect the resulting artifacts?
    a. Have you seen any specific cases where women have affected the software artifact in a software development process?
10. Are you aware of any cases where a development team has presented actions to secure womens input on the software artifact?
    a. Can you think of any actions or measures that can be taken or implemented in development processes to ensure diversity and womens input on the resulting software artifact?

Sunniva Block and Hanne Olssen

Addressing Gender Inequality in Software Development Processes

NTNU
Kunnskap for en bedre verden