Informatics

Impact of sudden increase in remote work on agile software engineering processes and productivity

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

BSc Vlad - Ioan Bălan
Registration Number 1126441

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Thomas Grechenig
Assistance: Univ.Lektor Dr. Martin Pazderka
Univ.Lektor Dipl.-Ing. Dr.techn. / Bakk.techn. Mario Bernhart

Vienna, 1st January, 2001

Signature Author	Signature Advisor

Erklärung zur Verfassung der Arbeit

BSc Vlad - Ioan Bălan

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. Jänner 2001

Vlad - Ioan Bălan

Acknowledgements

I would like to give my sincere thanks to Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Thomas Grechenig for supervising my diploma thesis. Furthermore, I would like to thank Univ.Lektor Dr. Martin Pazderka and Univ.Lektor Dipl.-Ing. Dr.techn. / Bakk.techn. Mario Bernhart for their feedback. Last but not least, I would also like to thank my family for their unconditional support.

Kurzfassung

Das Ziel dieser Arbeit ist der Einfluss des durch die COVID-19 Pandemie ausgelösten Zeitabschnittes von Remote-Arbeit auf der Softwareentwicklungsprozessen, der Arbeitsroutine der Teammitgliedern, sowie auf der Produktivität zu analysieren. Basierend auf eine Literaturrecherche werden die Herausforderungen der Remote-Arbeit im Fall der Arbeit nach der agilen Methodik identifiziert. Weiters werden die Auswirkungen dieser Herausforderungen auf den agilen Prinzipien sowie deren Einfluss auf den Kernpunkten der Forschungsfragen: agile Prozesse, Arbeitsroutine und Produktivität diskutiert. Um die Forschungsfragen umfassend abzudecken, werden halb-strukturierte Interviews mit agilen Fachleute aus kleinen und mittleren Unternehmen in Österreich geführt. Im Rahmen der Interviews sind Antworten auf den folgenden Fragen gesucht: wie haben sich die agilen Prozessen verändert, wie hat sich der Arbeitsalltag der Teammitglieder verändert und wie wird die Produktivität beeinflusst. Die Analyse der Veränderung der Agilität der Prozessen als auch die Analyse der kurz- und langfristigen Produktivität sind Teil der wichtigsten Ergebnissen dieser Arbeit. Zusätzlich werden die Auswirkungen der zunehmenden Remote-Arbeit auf der Kommunikation und Zusammenarbeit innerhalb des eigenen Team als auch mit anderen Teams untersucht. Schließlich wird analysiert, ob die Erwartungen einer gestiegenen Produktivität realistisch sind, unter der Annahme, dass die agilen Prozessen in Bezug auf der Arbeitsroutine der Teammitgliedern angepasst werden.

Abstract

The goal of this thesis is to analyse the influence of the increased period of remote work triggered by the COVID-19 pandemic on the agile software development processes, team members' work-routine as well as productivity. Literature research is conducted to identify the challenges caused by remote work when working according to the agile methodology. In addition, the impact of these challenges on the agile principles as well as their influence on the research questions' main points: agile processes, work routine and productivity is discussed. For a more comprehensive perspective, semi-structured interviews are conducted with agile practitioners from small to medium-sized companies in Austria. Furthermore, answers to: how have the agile processes changed, how has the work routine of team members changed as well as how is productivity affected are provided. Key results include the analysis of how the agility of processes has changed as well as the analysis of productivity both in the short and in the long term. Moreover, the impact of the increased remote work on communication and collaboration within the own team and with other teams is discussed. Last but not least, the feasibility of the expectation of increased productivity in case the agile processes are adjusted with respect to the team members' work routine is discussed.

Contents

CHAPTER 1

Introduction

1.1 Problem Statement

Business sectors all over the world have been significantly influenced by the COVID-19 pandemic effects. It has become a challenge to find a sustainable method to keep conducting their everyday business in a successful manner. Software development makes no exception, programmers around the world are being kept away from their everyday work routine and while some are seeing this as an opportunity to be closer to their family members, others are having a hard time to adapt sharing their work environment with their loved ones [13]. Due to the companies' desire to maintain their employees motivated and keep their productivity level as high as possible, it is in the companies' interest to invest time and money in enhancing their employees' remote work experience. As a consequence, companies are more willing to improve their work environment to support distributed software engineering teams. Given the current situation, the above-presented problem description is of significant interest to the Software Engineering/Development sector. As the Software Engineering department's effectiveness directly impacts the quality and the timeline of the product development, the research topic is also relevant from the Software Product and Project Management perspective.

1.2 Motivation

Being involved in the Software Engineering process during these hard times and having a positive experience, reshaped my view on the home office and remote work perspective. Over the last years, I have been part of multiple agile software engineering teams and realized that the key to migrating towards a distributed team highly depends on the processes and tools that support the work environment. Besides unimaginable chaos, I think that there are also positive outcomes that can be concluded from this COVID-19 pandemic situation. Namely, the opportunity to rely on the own established agile development processes and evaluate the efficiency and quality obtained with these processes during remote work. The effectiveness of the home office exceeded or fall short of the expectations of most employers and this is thanks to their employees' sense of responsibility and motivation. It must be noted that this performance is tightly coupled with the quality of their work environment.

My assumption is that in the upcoming years the software engineering work environment will experience considerable changes and will migrate more towards the home office. This is in the interest of both the employers and employees, as one party will lower their expenses per employee and the other will benefit from a more flexible time schedule. For a deeper understanding of the feasibility of this assumption, the impact on productivity during the pandemic needs to be observed, as well as how employers and employees have tailored their processes and work routines to cope with this unprecedented situation.

To sum up, I think this opens an up-to-date research area that focuses on the impact of COVID-19 on agile software engineering teams and reinterpreted agile software engineering processes tailored to adapt to the requirements of the new work conditions.

1.3 Expected Results

The aim of this thesis is to analyse multiple agile development workflows and determine how the adaptation of the existing processes can suit the new work environment. In this work, the current software landscape of agile development processes is reviewed. Moreover, the impact of the sudden increase in remote work on agile distributed software engineering teams and processes is explored.

To get a more practical overview of the current software landscape and its use during extended periods of remote work, this research is accompanied by a set of interviews conducted with practitioners of agile development methods. The results expected in this thesis will be divided into two parts: (i) provide answers to the research questions (ii) contradict or approve the hypothesis.

1.3.1 Research Questions

Based on the literature research and the conducted interviews, the first part of the expected results will try to provide answers to the following research questions:

RQ1 *How has the workflow of agile distributed software engineering teams changed during the period of sudden increase in remote work triggered by the COVID-19 pandemic?*

RQ2 *How has the work-routine of agile distributed software engineering team members changed during the period of sudden increase in remote work triggered by the COVID-19 pandemic?*

RQ3 *How is the productivity of agile distributed teams impacted during the period of sudden increase in remote work triggered by the COVID-19 pandemic?*

1.3.2 Hypothesis

As stated in Section 1.2, the assumption that motivates this research anticipates that in the upcoming years a significant amount of software engineers will conduct their work from home. This raises interest regarding the confirmation or contradiction of the hypothesis concluded from the above-mentioned research questions:

- *Adjusting the agile software development processes with respect to the distributed team members' needs and work routines improves overall efficiency resulting in increased productivity*

The second part of the expected results relies on the answers to the above-presented research questions and it aims to provide a reasonable explanation if it is realistic to expect the hypothesis to be confirmed or not.

1.4 Methodology

The methodological approach upon which the thesis is based consists of a literature research process, followed by an analysis of a set of interviews that are meant to provide a comprehensive overview of the topic from a practical perspective. The aim of these interviews is focused on exploring and understanding the topic rather than straightening some ideas about it. The structure of the interviews requires flexibility and fluidity, the number of interviewed people being less important. Consequently, the interviews are structured based on a qualitative approach and conducted in a semi-structured manner.

1.4.1 Literature Research

As already revealed, the theoretical part of this thesis consists of a literature research which serves as the research method. Its purpose is to identify and analyse the most commonly used agile development methods. These development methods are analysed concerning their suitability to be applied in agile distributed software engineering teams. In addition, several papers concerned with the influence of COVID-19 on agile distributed software engineering teams are reviewed. The goal is to identify how companies have adapted their processes to keep up with the pandemic's effects. Furthermore, it is part of this thesis to identify the impact of the pandemic on the agile distributed software engineering team members. Namely, how has their work routine changed and which benefits and challenges they are facing while working from home? Last but not least, a relation between the (i) productivity of agile distributed teams, (ii) followed agile software engineering processes and (iii) team members work-routine and well-being is sought.

1.4.2 Semi-structured Interviews

The second research method used in this work consists of a qualitative interviewing approach and is structured based on a semi-structured interview process. This interviewing method stands out of the lack of structure and high levels of flexibility [12]. Semi-structured interviews are very well suited for an agenda of open-ended questions that require follow-up queries [3]. This interview technique proves itself to be very useful if the independent thoughts of each individual in a group are of interest [3]. These interviews will target a minimum number of 10 professionals from at least 3 companies that are currently developing products based on agile software engineering methods. The goal of these interviews is to get in touch with the real world and gain a more practical perspective on the effects of the pandemic on the agile distributed processes and each agile distributed team member.

The interviews will be conducted by following the model described in [3] which (i) takes into consideration the disadvantages and advantages and (ii) provides a detailed description of how to design and conduct semi-structured interviews. In addition, (iii) polishing interview techniques as well as (iv) guidelines for analyzing and reporting are presented.

1.5 Structure

The structure of this thesis is constructed as follows. Section 2 comprises the state of the art. Relevant concepts are presented in Section 3. The interview structure and the interview analysis are contained in Section 4 and Section 5. Answers to the research questions are provided in Section 6. Last but not least, Section 7 contains the conclusion.

CHAPTER 2

State of the Art

Based on its relevance to this topic, the literature is analyzed by dividing the papers into 3 categories. Firstly, papers that describe the impact of COVID-19 on software development are taken into consideration. Secondly, software development is being analyzed from the agile point of view and therefore papers that focus on agile and distributed development practices are reviewed. Thirdly, studies that are concerned with qualitative interviewing methods are examined.

2.1 COVID-19 Impact on Software Engineering

The following papers' contributions relate to the faced challenges in software engineering from the standpoint of the COVID-19 pandemic. These studies were chosen because they consider the impact of the virus on software engineering.

Ford et al. [13] conduct two surveys (with 3,634 responses) on software developers during the COVID-19 pandemic with the goal to understand the benefits, challenges, and opportunities to improve this special circumstance of remote work. Based on their findings all benefits have a positive influence on productivity while all faced challenges have a negative impact on productivity. They identified that (i) *Improvements to communication tools* such as support for whiteboard drawing, support for multi-desktop sharing from multiple people and sharing on split windows or on local multiple screens and (ii) *Provide guidance for working from home* such as support for documentation, rank among the top 6 improvements that can impact productivity in a positive way. In the long run, they also consider the idea that the future of software development work is possible to develop towards fully remote work or some form of hybrid work.

A Deep Dive on the Impact of COVID-19 in Software Development is presented by da Mota Silveira Neto et al. in [9]. They conduct their research based on a mining software repository study that takes into consideration 100 GitHub projects evaluated by using 10 different metrics. In addition, 279 software development professionals are surveyed with the goal of identifying the impact of COVID-19 on their daily work routine and their well-being. As a result, 12 observations related to productivity, code quality and well-being are synthesized. To sum up, they revealed that does not affect code contribution and compared to the usual task completion time the overall productivity does not decrease. Furthermore, based on the conducted survey, during the pandemic does not impact code quality. In contrast to these findings, mentoring activities for newcomers have decreased during the pandemic.

Likewise, Rodeghero et al. reported in [30] that newly hired developers face communication and collaboration challenges as well as difficulties when finding documentation, asking for help, and bonding with teammates. In order to provide support during the remote onboarding process, they recommend a set of guidelines meant to help managers and teams when onboarding new developers. In this thesis, a survey of 267 new hires at Microsoft involved in their onboarding process to software development teams during the pandemic is presented. 61.9% of the participants reported that most of the face challenges are a consequence of the COVID-19 pandemic.

In contrast to [9], Ralph et al. [27] report based on a large-scale study that they have found evidence that the closely related attributes: productivity and well-being have declined during the COVID-19 pandemic. Hence, it is unrealistic for organisations to maintain their expectations regarding the normal productivity levels of their employees. Furthermore, they reveal recommendations regarding what companies can do to help their software developers maintain their productivity level as high as possible during these hard times. These recommendations vary from: (i) *organizations improving documentation of their processes*, (ii) *peer reviewing commits, change requests or pull requests*, (iii) *using a build system to automate compilation and testing* and (iv) *having a good work-from-home infrastructure (e.g. source control, VPN, remote desktop, file sharing)* to: (v) *companies paying for internet charges*, (vi) *organizations buying new equipment needed to* , (vii) *having virtual social events* or even (viii) *organizations sending food to staff* . However, they indicate that there is no general recommendation as different people appreciate different kinds of support from their organisations. The work is organized based on a questionnaire translated into 12 languages and the analysis of 225 responses from people from 53 countries.

The uncertainties caused by the COVID-19 pandemic as well as how to try to deal with them from the standpoint of an agile software startup are shared in [8]. The research is based upon the action research methodology and uses three cycles of action research. This method stands out by allowing both subjects and co-researchers to be involved as practitioners of agile development methods. The outcome of this study reveals how software startups can improve their processes to be able to cope with the unpredictability of the environment caused by the COVID-19 pandemic. The learned lessons as well as the main challenges such as: (i) *keeping the team's productivity levels*, (ii) *establish the required tools for remote work*, (iii) *trying to meet clients expectations*, (iv) *maintain the delivered quality*, (v) *preserve the employees well-being*, (vi) *support employees with* *the required infrastructure*, and (vii) *organize the development process* are part of the presented results.

During the 21st International Conference on Agile Software Development [4], discussions regarding the effects of the pandemic on agile practices and the impact of COVID-19 on digital workspace transformation took place. Mancl and Fraser [19] research the topic of *COVID-19's Influence on the Future of Agile* with respect to four key points synthesized from the panel discussions. To begin with, a set of communication tools are required to substitute face-to-face communication between developers working from home. Second, following agile practices is more challenging due to the absence of in-person communication and interaction. Likewise [13], support for brainstorming or whiteboard as well as informal discussions are hard to achieve as collaboration tools are not always suited for high-bandwidth or informal communication. Last but not least, similar to Ford et al. [13] and da Mota Silveira Neto et al. [9], the panel raised the topic of onboarding new employees in the virtual work environment.

2.2 Agile Distributed Development

Agile development practices and approaches for creating a distributed work environment are considered in the following papers.

To begin with, Ramesh et al. [28] conduct a research with the goal to provide an answer to a fundamental question. Namely, *Can Distributed Software Development be Agile?* Detailed case studies of agile distributed development are conducted in three companies. The objectives are to identify the faced challenges and how these can be managed by development practices and processes. The methodological approach used for data collection is based on open-ended (semi-structured) interviewing. Over a period of 9 months, a total number of 18 participants (e.g software developers, project managers and customer representatives) were interviewed one to three times. For analysing the data, the grounded theory methodology [35] was used. The outcome of the research reveals the fact that distributed software development can be agile if both environments (agile and distributed) are combined with respect to their unique characteristics.

Not only Ramesh et al. [28], but also Shrivastava and Date [34] are interested in how suited are agile practices for distributed development. Being in the same location and relying on a shot communication path are the core concepts of most agile methodologies. Whereas, cost reduction, access to skilled resources and having a high-quality employee pool while expanding to different markets is the whole reason why companies opt for remote software development. By taking into consideration the principles of agile practices, Shrivastava and Date [34] draw attention to potential compatibility issues of distributed development with agile development. However, if proper tailored agile techniques, prove very handy in overcoming distributed development challenges such as cultural issues, inadequate communication, technical issues and project and process management issues.

A distributed scrum approach as part of agile project management with outsourced development teams is presented in [37]. Many companies around the world such as Honda, Canon and Fuji-Xerox are developing products based on best business practices such as Scrum. But the company that really stands out is Toyota. This car manufacturer is believed to constantly be able to achieve higher productivity and quality levels than its competitors. More exactly, Toyota manages to reach four times the productivity of its competitors. From the quality point of view, the level is twelve times higher. The motivation behind this study is to find out if *Scrum can do the same for globally distributed teams*. To provide answers to this research question, two companies which achieved similar results based on agile development practices are taken into consideration. SirsiDynix and StarSoft Development Laboratories developed a Java application with the help of an agile team of 56 developers distributed in three countries: the USA, Canada and Russia. This paper recommends best practices which are radically different or counteractive to the ones promoted by the Scrum Alliance.

Poole [26] describes distributed product development based on Extreme Programming (XP). The author was acting as a senior manager at INOA Technologies and was responsible for delivering their Web Service Integration Platform (WSIP) product suite. Based on this experience, the development effort and the issues encountered during distributed development using Extreme Programming are revealed. The development effort is analysed from the perspective of its organization, practices and processes. Moreover, the qualitative and quantitative measures used in the evaluation method are considered. On top of that the four practices: (i) *distributed stand-ups*, (ii) *cultural exchanges*, (iii) *common source code* and (iv) *shared vision* on which the team agreed are detailed. The learned lessons from this experience provide meaningful observations of how Extreme Programming can complement the agile distributed development process.

An approach to how a work environment can support agile distributed software development is discussed in [18]. Even though the study is specialized on a single company and no generic way that can be applied to any company is presented, an in-depth example on how to organize a virtual work environment and how to run distributed projects efficiently are covered in this work. The research is based on a case study that shows how a company managed to gradually improve their processes over 1 year without neglecting the characteristics of the distributed development. The selected technologies are chosen by focusing on the team members' needs and the virtual work environment is designed to allow each team member independent of his location to fully benefit from it.

One of the main challenges encountered in distributed software development is considered to be the 'non-face-to-face' communication. In order to tackle the inefficiency of 'non-face-to-face' communication, Phalnikar et al. [25] propose a mix of technologies and best practices that can compensate for leadership struggle, cultural incompatibility and lack of trust. Agile distributed teams should be able to cope with the process adaptability during the entire life of the project and promote open collaboration throughout the development iterations. For this purpose, two team structures are proposed with respect to the benefits and challenges of using agile processes in distributed software development.

As face-to-face communication is infeasible in distributed development, Robinson [29] analyses the communication network of an agile distributed software development team by providing an experience report. The work is motivated by the research question: *"How well can a work environment support agile distributed software development?"*. The empirical research method consists of a case study and the source of empirical evidence is based on a questionnaire.

Lous et al. [17] present the journey that a medium-sized Danish company follows to tackle the challenges of distributed development in an agile team. To analyse how agile and distributed software development complement themselves, an exploratory case study was conducted. The data collection process is based on a one year period between November 2016 and October 2017. Several semi-structured interviews and full-day observations were carried out to find out how the company manages to respect the motivations behind agile software development and still dealing with the known problems of distributed development. Even though the study is based on a single nature reported case, key aspects identified during the research: (i) *agile servant-leader*, (ii) *agile team*, (iii) *trust*, (iv) *virtual work environment* and (v) *reduce waste* provide valuable insides on how the company managed to reach a maturity level that allows them to continuously evolve their software processes to keep up with the collocation restrictions of agile software development.

A strategy on how to adopt agile distributed development from the collective point of view rather than from the individual perspective is presented by Sureshchandra and Shrinivasavadhani [36]. A validation model based on Scrum, designed to support the transition to handling agile projects across different distributed teams is revealed. The validation model is structured based on a 4 stage process. Firstly, during the *Evaluation* stage, it is decided how suitable the project is to be developed in a distributed environment. The nature of work, infrastructure as well how much the development team is aware of the business concept are part of the evaluation. Secondly, the *Inception* phase requires core members from each satellite location to be familiarized with the workflow on a common location. Thirdly, the *Transition* is the phase where the core members mentioned during the second stage, transition their knowledge to their team on the satellite location. Last but not least, the *Steady* stage is reached where the location disparity barrier is broken and each team is able to work seamlessly. In addition to this validation model, measurement and analysis metrics for monitoring the distributed team's improvement as well as the learnt lessons and best practices are presented.

Based on a multiple case study in Germany, the influence of the COVID-19 pandemic on the agile methodologies in software development is analysed [22]. The research is organized based on a qualitative data collection approach and uses direct methods such as semi-structured interviews and observation of agile practices. This approach is complemented by an independent analysis of the project documentation and artefacts from the agile development teams. Neumann et al. [22] focus their research on: (i) *the impact on the performance of the agile development teams*, (ii) *the adjustments made on agile practices or roles* and (iii) *communication and collaboration aspects*. Besides the presented suggestions to the identified challenges of agile distributed practices, the results of the study reveal that the performance of agile development teams did not experience a negative impact.

Marek et al. [20] measures the impact of COVID-19 on the agile development teams by analysing the impact on the backlog and product vision. The study is based on a survey conducted on 120 agile software development practitioners involved in different industries. Their roles range from regular team members to higher management positions in organizations of different sizes. Due to the fact that most of the teams were involved in distributed and remote development prior the sudden increase in remote work caused by the COVID-19 pandemic, the results show that the work of agile development teams was not significantly impacted. Most of the teams managed to maintain the delivered quality by sticking to their existing definition of done and their release frequency. One of the key roles in the smooth transition to remote work are the used cooperation tools, therefore a collection of the most used cooperation tools is presented.

The research conducted by Griffin in [15] is based on a software development team that was used to operate in an agile manner prior the sudden increase in remote work triggered by COVID-19. Despite the fact that the team was already used to working according to the Scrum methodology and the fact that the team members conducted work both from the office and from home, a lot of challenges were encountered when applying the Scrum Guide [32] in a remote environment. Similar to the key aspects presented in [17], the concept of reducing the *waste* produced by the lack of agile process adjustments, is presented. Furthermore, suggestions on how to tailor the scrum methodology with respect to the challenges: (i) *scrum ceremonies*, (ii) *distractions* and (iii) *fatigue* are part of this study. Last but not least, even though scrum is based on strong collaboration and co-location principles, the importance of the chosen tooling set that supports the virtual environment is explained.

2.3 Qualitative Interviewing - Semi-Structured Interviews Methodology

Papers considered in this section describe the process of qualitative interviewing, more exactly semi-structured interviews. Fundamental, as well as specific aspects, are covered by this papers.

Edwards and Holland describe in [12] good practices for qualitative interviews. Fundamental aspects such as terminology and historical background are complemented by more specific considerations such as the domain of appliance, strengths and challenges. This work is also concerned with techniques meant to enhance the process of conducting such a type of interview. Guidelines on how talking, writing, seeing and creating should be done as well as hints on how to ask questions are presented. In addition, light is shed on particularities: (i) *how many interviewers should be considered?*, (ii) *how to start an interview?*, (iii) *listening, probing and following up* and (iv) *finishing the interview.*

The advantages and disadvantages as well as guidelines on how to design and conduct semi-structured interviews are presented in [3]. In addition, interview techniques and tips on how to draft questions and create an interview path are presented.

Extensive information on how to design, conduct and analyse semi-structured interviews is provided in the book [14]. The analysis consists of an ongoing and iterative process based on three steps. Firstly, after each interview, a post-interview analysis is conducted. Secondly, the goal is to execute an analysis meant to identify thematic patterns. Last but not least, connections are sought as part of an interpretative analysis.

CHAPTER 3

Relevant Concepts

This chapter is structured based on the relevant concepts of agile methodologies from two perspectives: co-located and distributed. Firstly, an overview of the agile methodologies from the co-located perspective is presented. The goal is to understand the fundamentals of agile development and its principles. Secondly, the perspective of applying the principles of agile methodologies in a distributed environment is researched. The goal is to take a look at the benefits of the agile principles in the distributed environment as well as understanding the challenges that may be encountered when conducting distributed development based on agile methods.

3.1 Agile Development

The method of continuous adjustment as a requirement to deliver work is a principle constructed upon evolutionary project management and adaptive software development that dates back to the 1950s. Therefore, even though the agile manifesto is a little bit over 20 years old, the roots of agile are significantly older [5]. Dave West, the Scrum.org CEO, suggests that the principles on which the foundations of agile are laid upon, are far from new. They are just used in a different manner [10].

> *"The tenets behind the #agilemanifesto weren't invented by us; they're the tenets of the scientific method Galileo used them. Archimedes used them."*
> — Dave West

In order to keep up with the demand for practices that value customer satisfaction, straightforward recommendations for creating future processes were required. As a consequence, based on pre-existing processes, a set of guidelines that are nowadays recognised by the software industry as the Manifesto for Agile Software Development were officially established [5].

3.1.1 Manifesto

In 2001, Beck et al. [6] gathered together with a common goal. They were a group of seventeen software development practitioners that had experience with Scrum, Extreme Programming, Adaptive Software Development, Feature-Driven Development and other software development methods. This group of people is nowadays called "The Agile Alliance". The main interest they all shared was to find a more lightweight, less documentation-oriented option to the existing heavyweight software development processes. What came out of that meeting is the Manifesto for Agile Software Development on which all participants agreed.

> ***Individuals and interactions*** *over processes and tools*
> ***Working software*** *over comprehensive documentation*
> ***Customer collaboration*** *over contract negotiation*
> *Responding to change over following a plan* — Manifesto for Agile Software Development

The Manifesto of Agile Software Development is built upon 4 fundamentals, which serve as guidelines for developing working software and deliver high quality.

First, the development process is driven by people and not tools, therefore individuals are more important than tools and processes. Otherwise, customer satisfaction is less likely to be achieved, because the team can not react as fast to changing requirements. Kate [16] best describes this idea by making an analogy to communication. In comparison to communication driven by processes, communication driven by people takes place whenever somebody feels the need, and not based on a predefined schedule.

The second fundamental tries to overcome the delays generated by extensive documentation before and during the development of the product. Comprehensive documents such as technical requirements, technical specifications, design documents and test plans require significant effort and time. In addition, most of these documents then need to be approved by others. Instead, developers can extract requirements from user stories. Still, agile is not about avoiding documentation, it is about understanding the limits of planning and against unmaintained and unnecessary documentation. Working software is a higher priority than documentation [6, 16].

The third fundamental encourages collaboration with the customer rather than contract negotiation. The reason behind this idea is the fact that negotiation limits the involvement of the customer in the development process as this takes place before and after the development is completed [16]. Customers are regularly provided with the opportunity of evaluating and giving feedback if working software is continuously delivered to them [34]. Involving a customer regularly in development meetings guarantees a higher probability of meeting the customer's expectation. Both parties benefit from this as it is easier for the development team to understand and meet the needs of the customer and the customer ensures the product fulfils the requirements.

Last but not least, agile embraces changes as changes are considered to add value to the product. Therefore, requirements changes need to be accepted at any point during the development stage. The idea is to be able to cope with and react fast to every change [34]. By following the agile tenets this is easier to achieve, as customer collaboration and continuously delivering working software help in reacting to changes. On the other hand, traditional software development methodologies provide a different perspective on changing requirements. Namely, changes are considered additional effort and more exactly, costs. This is a consequence of the detailed initial planning and no customer involvement during the development process [16].

To provide a more comprehensive understanding of the agile methodology, the manifesto is extended by twelve additional principles.

3.1.2 Principles

This section reveals the principle of the agile development philosophy. Similar to Séguin et al. [33], every principle is analyzed with respect to its corresponding agile fundamentals. Every principle provides an extension to at least one of the fundamentals, as can be seen in Table 3.1.

		Agile Principles											
		1	2	3	4	5	6	7	8	9	10	11	12
Agile Fundamentals	***Individuals and interactions*** *over processes and tools*				✓	✓	✓		✓			✓	✓
	Working software *over comprehensive documentation*	✓		✓				✓		✓	✓	✓	
	Customer collaboration *over contract negotiation*	✓	✓		✓				✓				
	Responding to change *over following a plan*	✓	✓	✓									✓

Table 3.1: Mapping of Agile fundamentals to principles

Principle 1

Our highest priority is to satisfy the customer
through early and continuous delivery
of valuable software.

This first principle is meant as an addition to the second, third and fourth agile fundamentals [33]. Although traditional software development promotes delivering the software product at the end of the development phase or after a longer development cycles, the agile philosophy is based on short iteration at the end of which working software development is delivered. Customer satisfaction is the most important thing, because of this the customer needs to be able to track the progress and assess the accomplishment of requirements as soon as possible [21, 23].

Principle 2

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

The second principle is suggested in order to be able to keep up with the fast-changing business environment. Due to the flexible requirements policy, the customer can start product development very early and specify requirements during the development process when further requirements are identified. The customer must not waste time on writing formal change requests, as changes are welcomed at any point during the development stage. This allows the customer to exploit every business opportunity and gain an advantage over its competitors [1, 21]. Therefore, this principle is bound to the third and fourth agile fundamentals [33].

Principle 3

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale

Williams [38] considers this principle to be to closely related to the first principle, making it redundant. On the other hand, others point out that this principle structures the development cycle by allowing to split the functionality into smaller features which are delivered in a shorter time frame. This strategy speeds up the development process, allows better response times to changing requirements and provides immediate value to the client [1, 23]. Therefore this principle incorporates values form the second and fourth agile fundamentals [33] .

Principle 4

Business people and developers must work together daily throughout the project.

The final product is the result of the collaboration between the people involved in planning and selling activities and the technical resources [1]. Due to the dynamic nature of the software development business environment, flexible requirements and lightweight documentation, collaboration between all parties that contribute to the final product is required on a regular basis [1, 21, 23]. This transparency in communication builds trust and ensures customer satisfaction [2]. Séguin et al. [33] correlates this principle with the first and third agile fundamentals, as it relies on the collaborations between individuals.

Principle 5

Build projects around motivated individuals.
Give them the environment and support they need,
and trust them to get the job done.

The fifth principle brings individuals to the fore by suggesting key points on which an agile team should be built and supported. Consequently, this principle is an addition to the first agile fundamental [33]. The goal when building a team is to select the right people for the right job and allow them to self-organize themselves. Avoiding micromanagement is a must. Allowing the team to work autonomously and taking the team members needs into consideration, makes people feel trusted and empowered. On this way, people will identify themselves with the purpose of the project and will deliver by reaching their full potential. To be able to get the most out of the teams potential, the team needs to be provided with the required environment and tools. Failing to do so will force team members to waste time with administrative issues and prohibit them to invest this time in adding value to the product [1, 2, 21, 23, 38].

Principle 6

The most efficient and effective method of
conveying information to and within a development
team is face-to-face conversation.

Séguin et al. [33] view the sixth principle as an addition to the first fundamental because individuals interaction is the main point of the principle. In contrast to documentation, which is the way of communication present in traditional software development, agile gives great importance to face-to-face communication. This reduces the overhead, a key factor that agile is trying to influence [21]. The same opinion is shared by Williams [38] who also draws attention over the importance of documenting important decisions. Moreover, face-to-face communication shortens the communication paths and allows stakeholders and software developers to address issues in a direct manner which leads to faster response times when taking a decision [1, 2, 23].

Principle 7

Working software is the primary measure of progress.

This principle is built on the second fundamental [33] and recommends a straightforward method of measuring progress. More exactly, working software. As already mentioned, in the agile philosophy the most important thing is to please the customer, therefore progress has to be evaluated from the customer's perspective. It is less significant how many hours are invested in the project and how many bugs are fixed or how many features are implemented if the software does not correspond with the customer expectations [1]. Similar to Misra et al. [21], Williams [38] consider that with each development iteration a step forward towards the end product must be achieved. Hence, every iteration has to deliver high-quality working software. Being the first to get a functional product to the market is a higher priority than delivering a complete product later on. Because of this, the customer has to be able to give feedback very early in the process [2].

Principle 8

Agile processes promote sustainable development.
The sponsors, developers, and users should be able
to maintain a constant pace indefinitely.

Agile is not about satisfying the customer at any cost, the work-life balance of all practitioners of agile methods is equally important to pleasing the customer. Getting done a constant amount of work in every development iteration is the goal and not working overtime to reach unrealistic targets. Panning without taking into consideration the developers' pace and work routine will lead to burnout which will decrease efficiency and lead to lower productivity in the long term [21]. To reach high-quality software, time for tasks such as exploring, refactoring and responding to customers' change requests needs to be planned [21, 38]. The eighth principle can be traced back to the first and the third fundamental of agile development [33].

Principle 9

Continuous attention to technical excellence
and good design enhances agility.

As the development process advances to later stages, just focusing on being the first to the market and just delivering working software is not enough to keep performing at the highest productivity levels. Being flexible enough to respond fast to changing requirements is also important. Over time technical debt accumulates [2], therefore investing time to cope with this as well as planning time for refactoring is required. Additionally, making use of best practice designs contributes to a good quality which provides the robustness needed to react to changing requirements and is less likely to introduce bugs when code changes are required [1, 21, 38]. This principle provides a different perspective over the second agile fundamental [33].

Principle 10

*Simplicity–the art of maximizing the amount
of work not done–is essential.*

Principle ten is best described by an analogy to the Pareto principle. Also known as the 80/20 rule, the Pareto principle states that about 80% of the consequences are achieved by 20% of causes [11]. In other words, with 20% of work, 80% of the desired target can be achieved. This principle promotes the concept of investing effort in the tasks that have the most impact [2]. For example, if being the first in the market is the goal, spent 20% of the development effort to try to deliver around 80% of the product [23]. Keeping things as simple as possible while trying to focus on the present rather than trying to anticipate the future, improves the flexibility of designs and processes [21]. Misra et al. [21] considers this principle to be related to the second agile fundamental.

Principle 11

*The best architectures, requirements, and designs
emerge from self-organizing teams.*

Misra et al. confirms in the study [21] that no unique methodology that can be applied in every team structure, exists. The freedom that self-organizing teams benefit from, allows space for creativity and flexibility. As a result, team members can find their balance and apply agile methodologies in the way that best suits the product, their needs and their work routine [38]. Another benefit of self-organized teams is that interaction and collaboration within the team improves. Thus, principle eleven promotes the values of the first and second agile fundamentals [21].

Principle 12

*At regular intervals, the team reflects on how
to become more effective, then tunes and adjusts
its behavior accordingly.*

Following rigid processes is discouraged by the agile philosophy and iterative improvement is sought [21]. On a regular basis, agile teams need to take a step back and do a retrospective about the achieved targets and address present issues. This is a direct method to evaluate the success rate of the applied agile methodologies and principles. If there is room for improvement, the team needs to adjust its plans and behaviour in order to become more effective. To ensure constant progress, this process needs to be iterative [38]. Due to the fact that this principle has an influence on how individuals respond to changes within the team, this principle extends the first and fourth agile fundamentals [21].

3.2 Remote Agile Development

In this thesis, the term remote agile development stands for the process of developing software products based on agile methodologies while working remotely. This development process is considered to be distributed due to the fact that team members are working remotely. More exactly, from home. Due to the agile and distributed nature of this development process, benefits, as well as challenges between the agile and distributed paradigms, are identified. The analysis of these benefits and challenges is conducted based on the agile fundamentals and principles with respect to the research questions' main points: (i) **RQ1** - *agile workflow*, (ii) **RQ2** - *team members work routine and needs* and (iii) **RQ3** - *productivity*.

3.2.1 Benefits

Alongside the challenges of remote agile development, there are also some benefits which can be identified. This section describes the identified benefits and their effects on the agile practitioners and agile practices as well as the impact on the research questions' main points.

Benefits	Impact on (→) / Influenced by (←)		
	Agile workflow	Team work-routine	Productivity
Fewer distractions	→	→	→
Involvement of stakeholders	→	→	→
Self organization	→	←	→
Transparency	←	←	→
Commute time		→	

Table 3.2: Relation between benefits and research questions main points

As can be seen in Table 3.2, all identified benefits can either have an impact on the team's work routine or can be influenced by this one. Commute time is the only benefit that only impacts the work routine of teams and has no influence on other research question main points. In contrast to the agile workflow and team's work routine which are impacted by the same amount of benefits, productivity is the most impacted main point. Self-organization and transparency are the only benefits which emerge as a result of at least one of the main points. For instance, self-organization is a result of the team's work routine while transparency is influenced by both the agile workflow and team's work routine.

Table 3.3 maps the benefits mentioned in Table 3.2 onto agile principles. The 4th principle which relies on daily cooperation between agile practitioners is in relation to all benefits except commute time. Commute time is the only benefit that has no relation to any of the agile principles. The second most impacted principles are the 6th and the 8th. These principles rely on how information is conveyed within the team and the importance of

		Agile Principles											
		Priority is customer satisfaction through valuable software	Welcome change requests for competitive advantage	Deliver working software frequently	Business people and developers work together daily	Motivated individuals, support their need, trust them	Face-to-face conversation is most efficient and effective	Working software is the primary measure of progress	Sustainable development, maintain constant pace indefinitely	Technical excellence and good design enhance agility	Simplicity – amount of work not done – is essential	Self-organized teams – best architectures and designs	Retrospective followed by behavior adjustment
		1	2	3	4	5	6	7	8	9	10	11	12
Benefits	Fewer Distractions				✓		✓		✓		✓		
	Involvement of stakeholder				✓	✓	✓		✓				
	Self organization				✓		✓		✓			✓	
	Transparency		✓		✓	✓							
	Commute time												

Table 3.3: Remote work benefits of agile principles

maintaining a constant pace throughout the development process. Last but not least, principles: 2, 10 and 11 are each influenced by only one benefit.

Fewer distractions

As a result of fewer distractions influencing the team members' work routine, in terms of quality, working time is considered to be used more efficient [22]. Just as Neumann et al. [22], Mancl and Fraser [19] noted that people consider that productivity has increased as a consequence of not being distracted that often. Not having to share the office with other colleagues and being reachable only over communication tools provides more control over the work environment and allows working without interruptions for longer periods of time [22]. Similarly, participants in the study conducted by Ford et al. [13] stated that they are able to concentrate better and work focused for longer periods. Besides this, a quiet work environment and having more control over potential distractions from other team members also increases productivity [9, 13]. Moreover, because of the reduced number of distractions, daily collaboration activities throughout the project can change. Also, information passthrough within the development team as well as development pace and self-organization can be expected to change. As can be seen in Table 3.3, the agile principles: 4 (daily collaboration), 6 (face-to-face conversation), 8 (constant development pace) and 10 (maximizing the amount of work not done) are impacted due to the influence on the agile workflow.

Involvement of stakeholders

As a consequence of virtual communication, product owners are considered to be more integrated into the teams in some cases. Neumann et al. [22] assumes that more objective and efficient, tool-supported communication are the reasons behind this findings. In more than half of the examined cases, stakeholder involvement remained constant while in 20.8% of the cases it was observed that stakeholders have been more involved [20]. From the perspective of the research questions' main points, the intensified involvement of the stakeholders affects the team members' work routine, the agile workflow and also the productivity (see Table 3.2). The motivation of stakeholder to step up and maintain a tight communication and collaboration relation between the business people and developers shape the way how the agile principles: 4 (daily collaboration), 5 (provide support), 6 (face-to-face conversation) and 8 (constant development pace) can be applied within agile workflow (see Table 3.3).

Self-organization

In contrast to the majority of the interviewed people, some interviewees stated that product owners are less involved in their teams. Better self-organized teams are considered to be one of the main reasons [22]. In spite of the fact that interaction between team members was negatively impacted by the period of increased remote work conducted during the pandemic, Neumann et al. [22] believe that communication achieved a more effective and fact-based character. In addition, how much team members communicate with each other is considered to be directly proportional to the quality of the communication [25]. Therefore, it is realistic to expect less stakeholders' involvement within teams that are better self-organized. Self-organization impacts how the agile workflow is applied within teams, by having an impact on the agile principles: 4 (daily collaboration), 6 (face-to-face conversation), 8 (constant development pace) and 11 (self-organized teams) (see Table 3.3). As can be seen in Table 3.2, self-organization is an effect of the team's work routine, while the agile workflow and productivity are impacted by it.

Transparency

Playing an important role in examining based on a goal-oriented manner and in adjusting how agile methods are applied [32], transparency is identified by Neumann et al. [22] as a determinant part of a teams performance. As a consequence of remote work, collaboration and communication are achieved through digitized means such as online tools. Not only Neumann et al. [22] but also Marek et al. [20] state that communication benefits from are more precise character. On top of that tools designed to track the project status also contribute to transparency [34]. Development processes that are transparent on how agile methodologies are applied, positively impact the agile approach [22]. Table 3.2 reveals the relation between the main points of the research questions and transparency. Lack of transparency within the agile process and lack of communication can negatively impact the response to changing requirements as well as how agile participants collaborate. In addition, in a remote environment transparency is also obtained by making use of online tools. Therefore, infrastructure plays an important role. Table 3.3 reveals on which agile principles transparency can have an impact.

Commute time

Being one of the most influential benefits of day-by-day work routine, reduced commute time is highly valued by software practitioners [19]. Not only Mancl and Fraser [19], but also Ford et al. [13] point out that 96% of the participants in the survey mentioned that lack of commute time has a positive influence on their well-being. While some engineers mentioned that they use this extra time to spend more time with their family, others mentioned that they use this time to work more hours. Yet, no significant impact on productivity was identified [13]. Furthermore, reduced commute time can not be associated with any influence of the agile principles.

3.2.2 Challenges

In this section, the faced challenges when conducting remote work are enumerated. By taking into consideration the main points of the research questions, the *agile workflow* (**RQ1**), *team members work-routine* (**RQ2**) and *productivity* (**RQ3**), the impact of every challenge on the agile fundamentals and principles is analysed. Due to the fact that remote work is a different form of distributed development, some challenges are also present in the distributed development process [22].

Table 3.4 reveals the relation between the encountered challenges and the research questions main points mentioned above. The direction of the arrow specifies if the challenge was influenced by the main point or if the challenge had a direct impact on the main point. For instance, communication challenges rise due to the team members work routine and have an impact on the agile workflow as well as on the productivity. Similarly, challenges related to delivering on time also impact productivity. In addition the bidirectional relation towards team work-routine symbolizes the reciprocal influence.

As can be seen in Table 3.4, all challenges except customer satisfaction have an impact on productivity. The team members work routine is impacted by more than half of the challenges. More exactly, 66,6%. On the other hand, 40% of the challenges rise as a consequence of the team members work routine. In comparison to the other two research questions main points, the agile work flow is not impacted by all challenges. 10 of the challenges have an impact on the agile work flow while only 2 are being influenced by this research questions main point.

Table 3.5 shows which challenges are faced when trying to apply the agile development principles while working from home. Since principle 4 relies on daily collaboration between team members, 12 challenges are encountered when trying to respect this principle. This makes principle 4 the most difficult to satisfy. The second most influenced principle is the 8th principle, followed by the 5th principle. These principles rely on developers' motivation, work environment infrastructure and trust. Their goal is to obtain a constant

Challenges	Impact on (→) / Influenced by (←)		
	Agile workflow	Team work-routine	Productivity
Customer Satisfaction	←	←	←
Collaboration and coordination	→	←	→
Communication	→	←	→
Conflicts		↔	→
Deliver on time	↔	←	→
Different working hours	→	←	→
Knowledge management	→	→	→
Management Issues	→	→	→
On-Boarding	→	→	→
Pair-Programming	→	→	→
Working overtime		→	→
Infrastructure	→	→	→
Shared work environment	→	→	→
Well-being		→	→
Ergonomics		→	→

Table 3.4: Relation between challenges and research questions main points

pace throughout the development process. In contrast to the other agile principles, principles: 9, 10 and 12 are not bound to any challenge.

Customer Satisfaction

Butt et al. [7] identified that during the increased period of working from home, 77% of the participants in the study think that the customer satisfaction experienced a significant decrease. This is the consequence of the agile practitioners' work routine which lacked work pressure and led to lower productivity. Therefore clients encountered delays and higher costs. In addition, due to fewer meetings with the developers, customer collaboration also decreased [7]. For customers used to the agile workflow, customer satisfaction was hard to reach and customer collaboration had a lot to suffer while conducting remote work. This makes principles 1 (customer satisfaction), 2 (welcome change requests), 4 (daily collaboration) and 8 (constant development pace) of the agile methodologies hard to satisfy (see Table 3.5). Moreover, based on Table 3.4, customer satisfaction is influenced by all research questions' main points.

Collaboration and coordination

Being one of the fundamentals of agile, team collaboration is a must when working together on a project, but this also makes team members dependent on each other [7]. Butt et al. [7] states that according to 80% of the developers that were part of the study, team coordination during the increased period of remote work triggered by the pandemic,

		Agile Principles											
		Priority is customer satisfaction through valuable software	Welcome change requests for competitive advantage	Deliver working software frequently	Business people and developers work together daily	Motivated individuals, support their need, trust them	Face-to-face conversation is most efficient and effective	Working software is the primary measure of progress	Sustainable development, maintain constant pace indefinitely	Technical excellence and good design enhance agility	Simplicity – amount of work not done – is essential	Self-organized teams – best architectures and designs	Retrospective followed by behavior adjustment
		1	2	3	4	5	6	7	8	9	10	11	12
Challenges	Customer Satisfaction	✓	✓		✓				✓				
	Collaboration and coordination		✓		✓		✓		✓			✓	
	Communication		✓		✓		✓		✓			✓	
	Conflicts				✓	✓						✓	
	Deliver on time	✓		✓									
	Different working hours	✓			✓		✓						
	Knowledge management		✓	✓	✓	✓							
	Management Issues				✓				✓				
	On-Boarding			✓	✓	✓							
	Pair-Programming		✓		✓	✓			✓				
	Working overtime	✓							✓				
	Infrastructure				✓	✓			✓			✓	
	Shared work environment				✓				✓				
	Well-being				✓	✓			✓				
	Ergonomics					✓							

Table 3.5: Remote work challenges of agile principles

encountered a decrease. Even though, Neumann et al. [22] considers communication to be more efficient and more objective, social interaction and togetherness suffered a significant drop. Coffee breaks and activities such as getting lunch with team members are not possible while working from home. Furthermore, new hired developers are making efforts in trying to stay socially connected to their team members [30]. Despite remote work being accepted, the preference of being able to socialize with colleagues in-person is still appreciated [20]. One of the key ideas embedded in the agile principles, more exactly having a fast response to changing requirements is considered to be promoted by cooperation and self-organization [20]. Marek et al. [20] also identified that in 16.8% of the cases in which they conducted the research, stakeholders' involvement decreased.

Especially challenging proved to be the collaboration with stakeholders in the early phase of conceptualization and design [19]. Stakeholders collaboration is also pointed out as a challenge by Neumann et al. [22]. Participants in the study conducted by Mancl and Fraser [19], encountered a hard time when trying to replace face-to-face whiteboard interactions with virtual brainstorming. Collaboration and coordination are directly influenced by the main point of the second research question: work routine. They have an impact on the first research question's main point: agile workflow. In addition, productivity is also impacted. (see Table 3.4). Furthermore, as can be seen in Table 3.5, this challenges affects principles: 2 (welcome change requests), 4 (daily collaboration), 6 (face-to-face conversation), 8 (constant development pace), 11 (self-organized teams) of the agile methodology.

Communication

Communication issues present in the distributed development teams [24, 34], are also faced when working from home. Breaking the communication barriers is a must when talking about agile practices applied in distributed environments [20], as smooth communication is one of the success factors of distributed teamwork [34]. Similar to distributed development, remote work requires vital communication channels. Failing to provide this communication means, negatively influences the communication frequency and quality. Phalnikar et al. [25] states that the quality of communication plays an important role in the volume of communication. Hence, a decrease in productivity can be expected [34]. Therefore, in order to adapt, agile teams started relying more on state-of-the-art communication practices [19]. Mancl and Fraser [19] raises awareness over challenges of mixed communication where meetings contain some attendees that are physically present and others that are taking part in the meeting remotely. Such situations require much more preparation and the used communication tools play an essential role. Communication without face-to-face interaction influences the interpersonal relationship and impacts motivation in a negative way because of the distance between development and business [25]. Responding to changes is considered by Marek et al. [20] to be a directly influenced by communication. Moreover, co-location is influential in supporting communication [20]. In general, in-person communication has the benefit of greater fidelity and therefore virtual communication should not be viewed as an alternative [19]. Mancl and Fraser [19] considers text communication less effective and more difficult to interpret, due to the absence of body language. Instead of just walking to a colleague's desk to discuss in person, many complained that communication through communication tools takes 2-3 times more than face-to-face communication [13]. As Ford et al. [13] identifies, work-routine changes thanks to the expectation of being responsive on multiple communication channels only adds extra communication overhead. Table 3.4 shows the relation between the research questions' main point and the communication challenges. Which agile principles are bound to this challenge are revealed in Table 3.5.

Conflicts

Conflict resolution is accelerated by non-verbal communication according to some interviewed persons in the research conducted by Neumann et al. [22]. In addition, conflicts are dealt with in a similar manner as before the period of increased remote work. The number of conflicts did not increase while working from home [22]. On the contrary, Mancl and Fraser [19] state the already present conflicts within team members have intensified. One of the reasons behind this could be text-based communication. Since text communication leaves a lot of space for personal interpretation it can be easily misunderstood or misinterpreted [19]. Conflicts have an influence on how people collaborate on a daily basis and also affect the internal relations between team members. Furthermore, motivation can also be affected by conflicts. As a consequence, conflicts affect principles 4 (daily collaboration), 5 (provide support) and 11 (self-organized teams) of agile methodology (see Table 3.5). The impact of conflicts on the research questions' main points is presented in Table 3.4.

Deliver on time

Tailoring the development cycles' length is considered more a usual adaptation as part of the agile philosophy than a consequence of the increased period of remote time triggered by the pandemic [22]. Even though Marek et al. [20] identified that the release frequency was in the majority of cases not affected by COVID-19, in 9% of the cases the release frequency changed due to the pandemic. Despite the fact that agile methodologies are built to raise professionalism among developers, 81% of the developers being part of the study conducted by Butt et al. [7] invoked non-professional reasons to explain late deliveries. The absence of work pressure negatively influenced productivity, as solving simple tasks took more than usual. This leads to late deliveries and higher costs [7]. As a consequence, the 1st and the 3rd agile principles are affected by this challenge (see Table 3.5). Table 3.4 shows the bidirectional relation between the delivery intervals and the agile workflow. Release intervals can be influenced as a consequence of the agile workflow and by not being able to deliver on time, the agile workflow is impacted. Furthermore, the work routine of developers influences the delivery intervals and impacts productivity.

Different Working Hours

Shrivastava and Date [34], raise the problem of time zone differences in distributed teams. Working from different time zones can be an impediment for communication [24, 34], which further leads to productivity decreases. The equivalence of this challenge while working from home are different working hours. Due to this fact, developers working schedules have less overlapping working time which also means that they have less time to communicate synchronously. Furthermore, Butt et al. [7] reveal that some developers being part of the study [7] did not join meetings because of family related reasons or absence of an official work environment. Working at different hours also affects customer satisfaction as planning meetings with clients is also dependent on overlapping work schedules [7]. Due to the lack of overlapping working hours, agile software development can not fully benefit from agile principles: 1 (customer satisfaction), 4 (daily collaboration), and 6 (face-to-face conversation) (see Table 3.5). Table 3.4 reveals that the main points of the first and third research questions are impacted while the main point of the second research questions influences and contributes to different working schedules.

Knowledge management

Proper knowledge-sharing mechanisms are required to get the most out of the used development methodologies, team members' experience and skills during the distributed development process [34]. As Marek et al. [20] identified, team dependency is a side-effect of the agile methodologies, therefore documentation can help reduce cost by avoiding redundant work [34]. Similar to Shrivastava and Date [34], Rodeghero et al. [30] considers that having up-to-date documentation regarding projects and how to set up the development environment allows new hires to avoid delays and work more efficiently. As text-based communication takes more time than face-to-face communication, proper documentation can save time by not being forced to wait until a colleague responds. In addition, storing documentation in as few locations as possible and making sure it is maintained, also contributes to working more efficiently [30]. Sharing knowledge by making use of a backlog to which every team member has access as well as using tools for keeping track of the issues and project management tasks reduces misunderstandings and improves team collaboration [34]. As can be seen in Table 3.4, knowledge management has an impact on how agile methodologies can be implemented in the development workflow as well as on the team members' work routine and productivity. Failing to support knowledge management can have an influence on the agile principles: 2 (welcome change requests, 3 (deliver frequently), 4 (daily collaboration) and 5 (provide support) (see Table 3.5).

Management Issues

Management challenges such as controlling and minimizing the impact of dependencies within distributed groups as well as identifying interrelated tasks are part of the development process and play an important role when talking about productivity. On top of that changing requirements in an environment that lacks of informal communication are more difficult to estimate and plan [34]. Phalnikar et al. [25] identifies the challenge of measuring the project progress and status with accuracy, which is even more difficult in a distributed environment. This challenge also applies while working from home or when conducting work based on a mixed routine that consists of both remote and onsite work. Therefore, management issues can impact all 3 research question main points (see Table 3.4). Due to management issues it is also harder benefit from the values mentioned in the 4th and 8th agile principles (see Table 3.5).

On-boarding

Just as in distributed development, training new team members remotely is more difficult than in a co-located environment because of the missing person to person communication possibility [22, 34]. Activities dedicated to integrate new employees have been reduced during the increased period of remote work triggered by the pandemic, as confirmed by 35.12% of the developers being part of the research conducted by da Mota Silveira Neto et al. [9]. Social integration is harder to maintain while conducting remote work, as social exchange activities such as taking a tea break in the kitchen or having lunch with the team members are not possible [22]. On top of that, remote training requires a high quality collaboration based on coaching and mentoring. This is significantly harder to achieve in a virtual environment as observed by Mancl and Fraser [19]. Making use of techniques such as virtual work shadowing, can make the process more effective due to the fact that an important part of integrating new comers relies on observing and following other team members work-routines [19]. To sum up, training new employees in a virtual environment is more difficult and requires more effort and time. As a result, the team members work routine is affected by the need of spending more time for training activities. Therefore, productivity can decreases due to less available time for solving actual tasks. In order to tackle the challenges encountered by new employees, the agile workflow can be adjusted. Techniques based on collaboration such as virtual work shadowing, pair programming or assigning a person as buddy for the new hires can ease up the process of integration. Providing the means for collaboration and communication is the most valuable aspect during the integration process of new developers. Motivation plays an important role both for the trainee as well as the trainer. Hence, principles 4 (daily collaboration), 5 (provide support) and 6 face-to-face conversation) of the agile methodology are impacted by issues encountered when trying integrate new members into the team.

Pair Programming

Pair programming is a common practice in agile methodologies. This was initially designed for co-located teams [20]. Like in distributed development, traditional in-person pair programming can not be achieved when conducting remote work [34]. Therefore, tools must be used in order to overcome this challenge. Being an important part of the agile teams' work routines, pair programming can be conducted either by using screen sharing or by using specific plugins which are part of the development environment [22]. Consequently, interviewees being part of the research conducted by Neumann et al. [22], stated that pair programming is conducted less frequent when working from home. Moreover, in some cases, the quality of this practice also decreased. Also Mancl and Fraser [19], reveal that in order to cope with the challenges of conducting pair programming several techniques that combine solo, pair and mob programming can be applied. Even though conducting pair-programming in a distributed environment is challenging, benefits such as improved product quality and exchange between team members as well as knowledge and skill transfers emerge from it [20]. Not being able to conduct pair programming as well as adjustments to this collaboration method leads to a different work routine which can change the agile process and affect productivity. Such changes can interfere with the ability to collaborate or self-organize and even with the ability to react to change requests. Hence, challenges encountered while trying to conduct pair programming leave their marks on how agile principles 2 (welcome change requests), 4 (daily collaboration), 5 (provide support) and 8 (constant development pace) can be applied.

Working overtime

The assumption that team members are working overtime during the increased period of remote work triggered by the pandemic is not confirmed by Neumann et al. [22]. Nevertheless, unfair management and pressure to work overtime can be harder to combat while working isolated from distributed locations than collectively [19]. In the long run, not having a clear delimitation between work time and free time, leads to working overtime and as work pressure intensifies developers became more proven to reach burnout [27]. Ford et al. [13] also mentions the issues of mixing the work routine with personal time and confirms that developers can lose track of time and work extra hours. Changing the work routine by working overtime can also affect productivity. Although productivity might increase due to working extra hours, in the long term, fatigue and even burnout can lead to a significant performance decrease. Working overtime does not influence how agile processes are set up. As productivity can oscillate from high in the short term to low in the long term, sustainable development and constant pace are hard to achieve. In addition, low productivity also decreases customer satisfaction. Therefore, making use of the advantages of agile principles 1 (customer satisfaction) and 8 (constant development pace) is more difficult while dealing with the disadvantages of working overtime.

Infrastructure

Adjustments to agile practices that enable the appliance of the principles in distributed environments and support communication, cooperation and transparency are reliant on infrastructure [20]. However, many companies are not properly prepared to support employees in conducting telework [27]. Marek et al. [20] reveals that applying the required agile methodologies in distributed development is realized by making use of online tools. Agile software development teams are highly dependent on communication tools [20, 22] as well as collaboration tools that provide digital whiteboard support [20]. Providing transparency as well as facilitating knowledge sharing is also obtained with the use of tools [20]. In addition, tracking work distribution as well as monitoring project progress and status are also enabled via tools [20]. In addition, internet speed can affect the quality of a meeting, leading to less collaboration and communication. Customer satisfaction is also impacted [7]. According to Ford et al. [13] connectivity problems such as internet disruption, slow connections and access problems to remote tools are the most popular challenges mentioned in the conducted survey. Productivity decreased with every encountered challenge, as a result of not being able to carry out the usual work-routine [13]. Without the required infrastructure the agile processes can not be maintained in a remote environment. Table 3.5 reveals which principles are difficult to apply without the right infrastructure.

Shared work environment

da Mota Silveira Neto et al. [9] agree that additional distractions are present while working from home, but do not confirm that these new distractions alongside the existing ones decreased productivity. This idea is supported by 41.60% of the developers being part of the survey. Nonetheless, maintaining high concentration levels required for the fulfilment of everyday work responsibilities, proved to be difficult while sharing the work environment with loved ones [13]. Although, the number of people with whom the work environment is shared while conducting remote work is not considered a source of distraction [9]. In particular, for parents work responsibilities blend with childcare responsibilities [27]. As a consequence, their ability to fulfil their work duties is limited [27] and directly impacts their work-routine [13]. Moreover, similar distractions as in open space offices are still present while conducting remote work from home [13]. For example, being interrupted by a colleague that walks to your desk is now replaced by distractions from family members. The shift towards the home office brought with it a lot of changes in developers' work routines. As some of these changes include distractions or temporarily limited capacity to work, the involvement in agile processes, as well as productivity, are subjected to change [8]. Table 3.4 shows how these changes interfere with the research questions' main points. Recommendation of agile principles: 4 (daily collaboration), 5 (provide support) and 8 (constant development pace) are also more difficult to follow while dealing with distractions and additional responsibilities which can reduce

Well-being

Through collaboration, transparency and fast reactions to changing requirements, agile methodologies are not only increasing productivity and customer satisfaction but also the work satisfaction of agile practitioners. Team members' well-being is highly impacted by the period of increased remote work triggered by the pandemic. Emotional factors such as stress and mental instability combined with home life without work pressure reduce productivity [7]. Ralph et al. [27] also raises attention over the negative effect of the pandemic on emotional stability and that this affects developers well-being. In addition, developers' productivity is highly bound to their satisfaction and emotional state [27]. da Mota Silveira Neto et al. [9] state that the reason behind high productivity and quality levels is high cognitive performance, which is often associated with developers being happy. Whereas, unhappiness is bound to emotional state, stress and disorder which lead to lower cognitive performance. Consequently, productivity and quality are affected. During the increased period of remote work triggered by the COVID-19 pandemic, 39.40% of the developers being part of the study [9] confirmed that they encountered sleep disorders and more than 50% mentioned that the stress level intensified. Thus, 34.76% of the developers consider that the happiness level did not increase [9]. Consequently, team members' work routine and productivity are impacted. Yet, no effect on the agile workflow was identified. For the relation between the agile principles and well-being please refer to Table 3.5. Agile principles: 4 (daily collaboration), 5 (provide support) and 8 (constant development pace) are based on collaboration and motivation, which are related to the state of mind and well-being of agile practitioners.

Ergonomics

In contrast to working on-site, the absence of a properly designed workspace can be an impediment when working remotely [27]. Ralph et al. [27] confirms that ergonomics directly influences that comfort experienced in the daily work-routine. Furthermore, many home environments are not designed to support working from home as people do not own a desk or do not have enough space for additional monitors [13]. Hence, productivity can be lower if the well-being though out the work routine is decreasing [8]. Likewise, despite the challenges of working overtime on well-being, agile processes are not influenced by this challenge of ergonomics. Furthermore, since ergonomics is related to the work environment and needs, satisfying the 5th agile principle is more difficult when facing ergonomics challenges.

To conclude, while all challenges have a negative impact on productivity, not all benefits improve productivity significantly [13]. As can be seen Table 3.3 and Table 3.5, the 4th agile principle is the most influenced principle while conducting remote work. An explanation for this could be the fact this principle relies on daily collaboration between agile practitioners and that most benefits and challenges impose changes on team members' work routines. In contrast to the 4th principle, principles: 7 (working software measures progress), 9 (agility enhancement) and 12 (retrospectives) are not influenced by any of the mentioned benefits or challenges. This finding may come as a consequence of the fact that these principles rely on measuring progress, technical aspects and adjustments of workflow. From the perspective of the research main point, as can be seen in Table 3.2 and Table 3.4, the most influenced is the team's work routine followed by productivity and the least influenced is the agile workflow.

CHAPTER 4

Interview Setup

4.1 Data Collection

Data collection is organized based on a qualitative approach and has an exploratory character. Therefore, data collection is obtained by using a direct data collection method. Namely, semi-structured interviews. By following the guidelines for qualitative interviewing presented by Edwards and Holland [12], 10 semi-structured interviews are conducted.

The selection of the participants in this interviewees, is based on several aspects. Software engineers that are developing products by following agile methods during remote work are targeted. To get a broader overview, agile practitioners that fulfill different roles and have different working experience are interviewed.

The Interview is organized based on several topics. The structure of the interview emerged as a result of the correlation between the Relevant Concepts and the research questions. Each topic contains a primary question and several follow-up questions. The purpose of these follow-up questions is to provide a more comprehensive understanding of the topic. Even though the interviews consist mostly of open-ended questions, periodically quantifying questions that make references to percentages, scales and ranges are asked. Initially, general questions are asked and as the interview advances more detailed questions are asked gradually. The question pool went through several refinement iterations based on the feedback of the advisors of this thesis.

4.2 Data Analysis

The analysis of the data collected as a result of the semi-structured interviews is inspired by the ongoing and iterative data analysis method described by Galletta and CROSS [14]. Even though the analysis is conducted with the goal of providing answers to the research questions, the construction of the responses is kept flexible during the early stages of the analysis. More exactly, in spite of the fact that some ideas may seem to directly answer the research question, several iterations might be required to refine the answer while others might prove to be misleading in the end. Moreover, data which does not seem to be directly related to the research questions can prove significant in the final analysis [14]. Similar to the method described by Galletta and CROSS [14], the data analysis process is structured based on the following stages: *post interview analysis*, *thematic patterns analysis* and *interpretative analysis*.

4.2.1 Post Interview Analysis

The process of data analysis starts immediately after finalizing an interview by following a set of analytical steps.

Interview Reflection

The purpose of this phase is to create an interview retrospective and reflect on the ideas that might have become obvious during the interview. Additional thoughts and questions that might have emerged during the interview are also documented during this phase. This retrospective serves not only as a prerequisite for further data analysis but also for ongoing data collection [14].

Data Organization

Independent of the source of data collection, recordings or notes, the data is organized and stored so that it is available for further analysis. In order to comply with the promise of anonymity, data should not be organized by referencing the interview participants' names [14].

Transcripts Accuracy

Galletta and CROSS [14] draws attention over the importance of maintaining authenticity and accuracy when creating a transcript from a recording or when taking notes during an interview.

4.2.2 Thematic Patterns Analysis

During the iterative process of data analysis, several topics become noticeable. The relevance of such thematic patterns is evaluated with respect to the research questions to which answers are sought. Although the interview questions are structured based on the research questions, during this phase, the goal is not to pursue a relation between theory and data but to document the thematic patterns that emerge [14]. The following guidelines are taken into consideration during the thematic pattern analysis.

Codes

Core ideas that stand out during the thematic pattern analysis are called codes. Codes are defined by one or more words. As a result of qualitative data collection, a wide range of codes is likely to seem relevant. Therefore, codes must be extracted with respect to the research questions [14].

Meaning

It is important to be able to trace codes back to data from which they materialized as well as establish a detailed meaning and how this codes are used [14].

Relation to other codes

Establishing a relation between other codes is not a goal during this analysis phase as relations often come to light as a result of the iterative analysis. Still, clear correspondence between codes must be documented [14].

Code status

Due to the high number of codes that emerge during the data analysis, some codes may overlap or refer to the same idea. The decision based on which these codes are considered to be still relevant or excluded from the analysis needs to be documented. This increases transparency and provides the possibility of reconsidering the decision in case this might prove to be inadequate for future outcomes of the analysis. The frequency of a code is important but not decisive, as "outliers" can provide a different perspective [14].

4.2.3 Interpretative Analysis

At this stage, as a result of the above-mentioned analysis phases, the list of codes finds itself in a more refined state. Popularity increases among codes with common significance. Nevertheless, variations of codes referring to the same thematic patterns can also be observed, resulting in multiple relations between these codes [14].

Refinement within the single interview

During the analysis of each interview in the process of identifying and interpreting the meaning of data from which the codes emerged, names and labels of current codes might be subject to change. This opens up the possibility of finding an additional instance of meaning with respect to the research questions as well as confirming or contradicting initial assumptions [14].

Connections across interviews

Galletta and CROSS [14] recommend to extend the analysis by trying to interpret how interviews relate to each other. In this way, particularities of single interviews can benefit from additional nuances. In addition, contrast and similarities are studied.

Thematic clustering

Once the codes are refined and analysed within and across the interviews, potential connections from one code to another code can be established. By making use of this association between codes, thematic clusters are delimited. Furthermore, through the meaning of the codes and the connections between them, thematic clusters provide an additional interpretation perspective that can offer viable answers to the research questions [14].

To sum up, the analysis is an ongoing process that blends with the data collection. The analysis starts gradually with identifying thematic patterns with respect to the research questions and as the analysis advances more detailed and precise answers to the research questions are sought. Due to the iterative nature, transitioning backwards and forwards between the three above-mentioned analysis phases reinforces the emerged concepts and patterns.

4.3 Structure

This section reveals the questions based on which the interview is designed. The interview path is structured with respect to the following key topics: (i) *personal presentation*, (ii) *social aspects*, (iii) *agile practices*, (iv) *work routine*, (v) *productivity*, (vi) *quality*, (vii) *on-boarding*, (viii) *motivation*, (ix) *trust*, (x) *overwork*, (xi) *conflicts* and (xii) *return to office vs. continue to work from home*. Therefore each question is part of one of the key topics mentioned earlier. The purpose of each topic alongside with the question assigned to it and its follow-up questions are revealed below.

4.3.1 Personal presentation

With the purpose of establishing a more personal relation with the interviewee and helping them fit into the context, some demographic questions are used. Edwards and Holland [12], propose to start by asking general, broad questions in the beginning of the interview. This phase marks the interview opening and is followed by a more specific part that goes more into the details of the interviewing topic. As the interview process continues, more specific questions are gradually asked.

1. **Describe your position and the role within the company, as well as your responsibilities?**

   1.1 What is your experience in this activity field?

1.2 Since how many years do you work in this company?
1.3 What is the activity domain of your company (e.g Automotive, Healthcare, Communication and Entertainment etc.)?
1.4 How many employees are working in the company?
1.5 Do you have any subordinates?
1.6 What is your educational background?
1.7 How old are you?
1.8 What is your country of origin?

4.3.2 Social aspects

Social aspects play an important role in the agile methodology as described by the agile manifesto [6]. Therefore, guidelines of agile methodologies [32] are constructed around interaction and collaboration principles. Collaboration and communication count themselves within the success factors of the agile methodology [22]. Furthermore, according to the participants in the study, the second biggest positive introduced thanks to the increased period of working from home, is the increase in communication [22]. Marek et al. [20] confirms that the communication within the teams has improved due to the fact that all communication was done using online tools.

According to Neumann et al. [22] and Marek et al. [20], meetings were reduced and are conducted in a more efficient and goal-oriented way since the beginning of the increased period of remote work. Techniques that can be applied during virtual meetings are essential in trying to compensate for not being in the same location. In order to provide added benefits over being physical together during meetings, virtual meetings have the advantage that they can be kept shorter when scheduled more often. In addition, practices such as using "round robin" to provide the opportunity for each participant to speak, can help in engaging each participant in the meeting [19]. In order to tackle the challenges faced when trying to bond with the teammates [30], virtual coffee breaks and virtual social events could be organized [22, 27].

2. **What are the differences in the way you interacted/communicated with your colleagues before and during the period of increased remote work?**

   2.1 Is there a difference in how meetings were conducted before and during working from home?
   2.2 How is the number and duration of meetings been influenced by this period?
   2.3 On a scale from 1 to 5, how much less do you socialize with your colleagues during a normal work day?
   2.4 What could be done to compensate for the physical absence when socializing with your colleagues? Do you think virtual breaks (e.g virtual game/coffee breaks) could help?

2.5 What are the key factors that influence this change in how much you communicate with your colleagues?

2.6 How much time a day is lost/gained because of communication aspects (e.g communication with colleagues using chat)?

2.7 How have your responsibilities changed to cope with the change? Did you experience any adjustments in your role?

4.3.3 Agile Practices

In this section, the focus lies on the software development processes. More exactly on the agile ones. Alongside the agile practices, roles have also encountered various adjustments [22]. The migration towards a distributed agile development strategy gives rise to a set of challenges for both employees and employers. From the employer's point of view, organizing the development process [8], providing guidance for working from home [13] and having a well-designed development pipeline that allows employees to work remotely [27] should be the priority. Similar to their employers, some software developers encounter problems while trying to adapt to the new working regime, with productivity, delivered quality and their well-being being most affected [13, 9, 30, 27]. To provide a different perspective, the effect of the sudden increase in remote work on the release frequency and intervals is analysed in [20]. Besides this analysis, also the involvement of the stakeholders was examined. In contrast to Marek et al. [20] where 57% of the subjects stated that the involvement of the stakeholders did not change and only 17% of the interviewers experienced an increase, the majority of the participants involved in the study conducted by Neumann et al. [22] felt that the collaboration with the stakeholders increased.

The goal is to find out more about how the sudden increase in remote work triggered by the COVID-19 pandemic forced companies to tailor their agile development processes. With the following questions, answers for the research question **RQ1** are sought.

3. **Describe the software development process that you were using before the sudden increase in remote work.**

3.1 What are the differences compared to the workflow used during the period of increased remote work?

3.2 On a scale from 1 to 5 how difficult did you find it to adapt to the changes?

3.3 What were the challenges you faced?

3.4 What are the benefits and drawbacks of the two software development processes?

3.5 Which development process do you prefer and why? A mix of both?

3.6 Which tools do you use during your development process?

3.7 How did remote work influence the way you do pair programming?

3.8 How would you rate the involvement and collaboration with the stakeholders? Did it increase/decrease? Why do you think this happened (e.g better self-organized, better collaboration)?

4.3.4 Work routine

As a result of the period of increased remote work triggered by the COVID-19 pandemic, not only the agile software development processes but also the work routine of the software engineers went through changes. Marek et al. [20] states that previous experience in conducting remote work or working in distributed teams has a significant influence on how smooth the transition from on-site work to fully remote work is. In the pursuance of identifying the impact of these changes on agile distributed software engineering team members and their work routines, answers to the second research questions **RQ2** are expected.

4. **Describe how your work routine looked like during a normal work day before the period of increased remote work.**

   4.1 How much of your working time was conducted from home before the COVID-19 pandemic?

   4.2 What changed in your everyday work routine during the fully remote work?

   4.3 On a scale from 1 to 5 how difficult did you find it to adapt to the changes?

   4.4 What were the challenges you faced?

   4.5 What are the advantages and disadvantages of the two working routines?

   4.6 Which work routine do you prefer and why? A mix of both?

4.3.5 Productivity

The third research question **RQ3** is based on the impact of the sudden increase in remote work on productivity. As McConnell observed, high-performing teams succeed in maintaining their high productivity level when working from home. Moreover, some people consider themselves to be more productive during remote work due to the fact that they have fewer distractions [19]. Similarly, Neumann et al. [22] state that the performance of agile software development teams is impacted by the distributed working nature and confirm that the efficiency and productivity of agile teams has not decreased. Also, almost half of the interviewed people would appreciate themselves as being even more productive and none of them considers the productivity level has dropped [22]. Based on the responses of their subjects to an open-ended question, Marek et al. [20] identified that the third most common positive thanks to this shift towards remote work, is considered to be increased productivity. However, even though others are pleased with the fact that they are able to spend more time with their loved ones, they encounter difficulties while sharing the work environment with their family [13]. Moreover, it is unrealistic for companies to expect their employees to perform at the same level as before the COVID-19 pandemic, as productivity and well-being suffered a decline during this period [27]. Previous experience in working from a distributed or remote environment as well as good applied agile principles combined with the suited tools are the key factor in maintaining the performance of an agile distributed team [20].

The following interview questions are constructed with the goal of finding out which factors are more determinant when analysing productivity during the period of increased remote work. Furthermore, detailed information about the influence of software development processes and team members' work routine on productivity, bring the hypothesis of the thesis in the foreground.

5. **How have the software development processes and your work routine influenced your efficiency during the sudden increase in remote work triggered by the COVID-19 pandemic?**
   5.1 How would you rate your increase/decrease in productivity during the period of increased remote work (e.g percentage-wise)? Did it increase/decrease over time?
   5.2 What influences your productivity (e.g processes, work routine, infrastructure, distractions, tools, previous remote work experience etc.)?
   5.3 Do you think your efficiency would have increased/decreased with/without the company tailoring the software development processes? Why?
   5.4 Should your employer tailor the development processes with respect to your work routine? Why? What would be the benefit?
   5.5 On a scale from 1 to 5, how well do you think the used software development processes were suited for distributed development? Why?

4.3.6 Quality

Even though the productivity of agile development teams is a key factor by which the team is analysed, the delivered quality should play an important role when examining the performance of the team. da Camara et al. [8] identifies the ability to maintain the delivered quality as one of the main challenges when trying to adapt the software development processes in order to be able to react to the unpredictable environment caused by the COVID-19 pandemic. On the other hand according to the survey conducted by da Mota Silveira Neto et al. [9], code quality did not have to suffer while working from home. Furthermore, due to fewer distractions compared to everyday office life and other potential disruptions, the time when working from home is considered to be more effective from the perspective of the quality aspect [22]. Similar to [22], in most of the cases, the delivered quality of agile development teams did not decrease [20]. Marek et al. [20] points out that the success factors for maintaining the quality level are the facts: (i) that the teams did not change their definition of done with respect to their tasks and (ii) the fact that the release frequency was not changed.

6. **How do the used software development processes influence the delivered quality?**
    6.1 How would you describe the relation between productivity and quality?
    6.2 Has the delivered quality of your team experienced any changes during the period of increased remote work? What influenced this change (processes, fewer distractions etc.)?
    6.3 How often did you release before the pandemic? What about now? Why (e.g processes)?
    6.4 Did the definition of done change with respect to the tasks? How?

4.3.7 On-boarding

New hires are experiencing a steep productivity curve and require additional help from their colleagues. As the time invested in mentoring newcomers decreased during the period of sudden increase in remote work triggered by the pandemic [9], onboarding employees has become more difficult. Furthermore, newly hired software engineers revealed that most of the difficulties they face are due to the increased virtual work influenced by the COIVD-19 pandemic. Most of them struggle to overcome the communication and collaboration challenges encountered while asking for help or bonding with teammates [30]. Despite the extra motivation or increased working hours, training new employees still proves to be more difficult in the distributed environment [22]. The following set of questions explores the onboarding process challenges. The goal is to understand how this process could be optimized so that new employees are being able to deliver higher productivity levels earlier.

7. **How does the onboarding process of new employees look like?**
    7.1 What are the challenges faced while onboarding new employees during the period of increased remote work?
    7.2 Has the time spent for onboarding new colleagues increased/decreased during the remote work? By how much (e.g percentage)?

4.3.8 Motivation

While each faced challenge encountered while working from home impacts productivity in a negative manner, each benefit offered by the employer motivates the employee and increases productivity [13]. Neumann et al. [22] opens up a new research topic by expressing interest in conducting further work to analyse if switching to distributed development impacts the motivation of agile team members.

8. **How do you keep yourself motivated and make sure you perform at your best every day?**

8.1 Did you experience any changes during the increased remote work period? Did you experience any ups and downs?
8.2 What are the challenges that influence your motivation?
8.3 Do you think your employer could help in keeping you motivated (e.g offer benefits, take over some cost when working from home, pay for lunch, buy equipment etc.)?
8.4 In which order do you think these benefits contribute to your productivity?

4.3.9 Trust

Mancl and Fraser [19], bring up one of the key points in the relationship between employer and employee while conducting remote work, namely trust. The CEO and Chief Software Engineer at a worldwide software consulting and training company embrace remote collaboration but also remains concerned over how people are working from home. The absence of face-to-face interaction with the team members could have a significant effect on the trust within teams. While this might not seem a problem in the short term, it may cause friction between the team members in the long term [19]. Not only Mancl and Fraser [19], but also Phalnikar et al. [25] identify trust as a potential issue within the distributed environment.

Not being able to have trust in your colleagues' work ethic and constantly having doubts about their delivered work, leads to a lack of motivation and therefore has a direct impact on productivity. The following question set researches the consequences of having trust issues within an agile software development team.

9. **In the context of allowing people to conduct remote work, how important do you consider trust? What about other attributes?**

    9.1 Which are the personal values that you appreciate most at your colleagues (e.g trust, respect etc.)?
    9.2 Why do you consider these personal values important?
    9.3 What are the consequences of not being able to trust a team member? Motivation drop?
    9.4 On a scale from 1 to 5, how much can lack of trust increase/decrease your motivation?

4.3.10 Overwork

Mancl and Fraser [19] raise attention over the effect of working from home for long hours without taking breaks. In contrast to virtual work, being physically in the office allows people to engage in playing different group games while taking regular breaks. In addition, due to the fact that people that conduct remote work are very often isolated, they are more predisposed to work unpaid overtime. This makes it hard to oppose unfair management. On the other hand, the assumption that team members have worked quantitatively more hours during the increased period of remote work is not confirmed by the interviewees in the research conducted in [22].

10. **Did you work overtime during the increased period of remote work compared to before this period? Why?**

    10.1 How much more did you work overtime during the period of increased remote?

    10.2 How many breaks do you take while working from home compared to being physically in the office?

    10.3 Can working from home be more prone to overworking and therefore leading more quickly to burnout?

4.3.11 Conflicts

On the interpersonal level, if friction existed within the team before migrating to the virtual environment, this was intensified while working from home [19]. Nevertheless, the majority of the interviewed test subjects in [22] stated that the number of conflicts did not increase or decrease while conducting remote work. Neumann et al. [22] also add that around 10% of the interviewed people think that written communication contributes to solving conflicts faster. On top of that, by moving all communication to online tools, co-located colleagues did not have the possibility to discuss issues in person and therefore distributed colleagues could not be excluded [20].

9. **How have you dealt with misunderstandings or conflicts before the sudden increase in remote work and how did you do it while working from home?**

    9.1 Has the number of conflicts increased/decreased?

    9.2 How did existing conflicts evolve?

    9.3 What are the consequences of having conflicts at your work?

4.3.12 Return to office vs. continue to work from home

The future of software development working environment is likely to develop itself as fully remote or at least became a mix of both remote and on-site working environment [13]. The new working virtual regime, namely working from home, requires organizational changes as noted by Reinitz [19]. The challenge remains to find a way to be able to conduct the everyday face-to-face activity in the virtual environment. From the employee standpoint, introducing remote work was rated as the best change, by the majority of the participants in the study [20]. In addition, Mancl and Fraser [19] reveal that there is a high demand in continuing to work from home. Employees are already being provided with the alternative of working entirely remotely by companies [20]. Ralph et al. [27]

also states that many companies are considering the option of allowing at least 5% of their previously co-located resources to work from home after the COVID-19 pandemic.

10. **What are the advantages/disadvantages of returning to the office instead of continuing to work from home?**

    10.1 How much of your working time was conducted from home how during the period of increased remote work?

    10.2 Do you see the possibility of working from home as a benefit? What about before this period?

    10.3 Which work routine do you prefer and why? A mix of both? How much 50%/50%

    10.4 What should companies do in order to motivate their employees to return to work?

    10.5 How do you see the future of software development: on-site, remotely or a mix?

4.3.13 Finish up

1. What is the best/worst thing introduced during the period of increased remote work?

2. Provide the opportunity to express ideas that were not covered by the questions.

4.4 Summary

In conclusion, the interviews are conducted by following a structure which intends to provide answers to the research questions with respect to the Relevant Concepts. The structured is organized based on several main questions and follow-up questions which are assigned to several topics. Data is collected by keeping the interview participants antonymous. The analysis of the collected data is done in three phases. This process is an iterative approach that gradually refines the details of the analysed data.

CHAPTER 5

Analysis of Interviews

5.1 Interview Group Overview

During the interviewing process, 10 people from 8 different small to medium-sized companies in Austria were interviewed. The size of the companies varies from 10 to 60 employees. All but one of the participants (P03) have software development responsibilities, with 5 of them having also other responsibilities such as leading a team (P02, P03, P08), designing software architectures (P07) or even being responsible for technical guidance (P06). While all interviewees have at least a bachelor's degree, not all of them have an educational background in computer science (P01, P08). The activity domain of most companies is consulting or communications while only 2 companies are involved in test automation or healthcare. Table 5.1 reveals more details about each interview participant.

5.2 Social Aspects

Table 5.2 shows how several communication aspects were influenced during the increased period of remote work triggered by the pandemic.

In order to try to compensate for the lack of face-to-face communication, the number of meetings increased. This is confirmed by 70% of the respondents (see Figure 5.1). Before the increased period of remote work, people could also be involved in meetings on demand, as they were physically in the office. During remote work, finding a time slot is more difficult as people have different work schedules. As a result, more people get invited to meetings just to be able to provide input in case they are required. In fact, in a lot of cases, people take part in meetings in which they are not necessarily required to participate.

Interviewee	Roles	Education Background	Experience	Industry
P01	Software Developer	Bachelor in Economics	4 years	Healthcare
P02	Team Lead, Software Developer	Master in Business Administration	15 years	Consulting
P03	Team Lead	Bachelor in Business Informatics	10 years	Consulting
P04	Software Developer	Master in Software Engineering	10 years	Banking
P05	Software Developer	Master in Software Engineering	8 years	Consulting
P06	Technical Lead, Software Developer	Master in Computer Science	15 years	Video Communication
P07	Software Architect, Software Developer	Master in Computer Science	17 years	Video Communication
P08	Team Lead, Software Developer	Bachelor in Physics	25 years	Real-Time Communication
P09	Software Developer	Master in Software Engineering	16 years	Test Automation
P10	Software Developer	Bachelor in Software Engineering	4 years	Banking

Table 5.1: Details of interview participants

Software Developer P04:
"At work, I could interact with colleagues at any point, now I need to schedule a meeting."

Team Lead, Software Developer P02:
"There were fewer meetings before because you could see if somebody is in the office."

Software Developer P03:
"To avoid creating additional meetings, you get invited to meetings even if they are not sure you can help."

Further, P04 reveals that his employer had a similar initiative to Rodeghero et al. [30]. Namely, developers were encouraged to turn their cameras on during meetings. P04 rates this initiative as a success as more than 50% of his colleagues continue to attempt meetings with cameras turned on. Similar to Mancl and Fraser [19], P02 also states that calls without being able to see the other participants are less effective in transmitting the desired message.

Interviewee	Number of meetings	Duration of meetings	Socialization with colleagues	Communication with stakeholders
P01	↑	↓	↑	↓
P02	↑	↑	↓	↓
P03	↑	↑	↓	=
P04	↓	=	↓	=
P05	=	=	↓	=
P06	↑	↓	↓	↑
P07	↑	↓	=	↑
P08	↑	↑	↓	↑
P09	=	=	=	=
P10	↑	↓	=	=

Table 5.2: Communication aspects

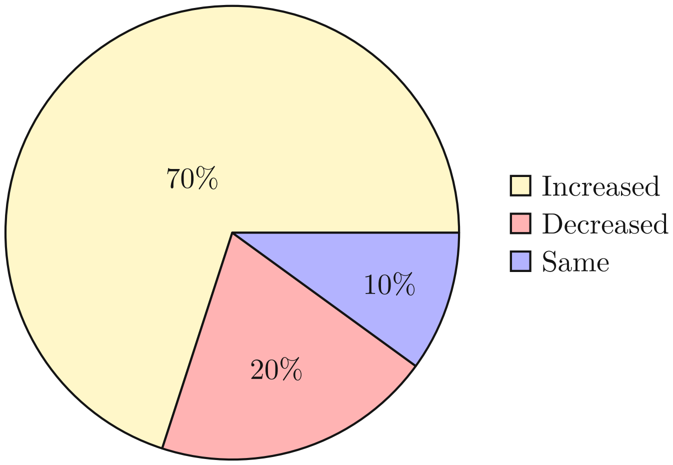


Figure 5.1: Number of meetings

🗨 Software Developer P04:
"I feel responsible for turning my camera on when so many people have their cameras on during meetings."

🗨 Team Lead, Software Developer P02:
"Calls without camera are a problem. Seeing other people's reactions is important to avoid misunderstandings."

However, when asked if virtual coffee breaks can compensate for the physical absence when socializing with colleagues, all but one participant agreed that this is not possible.

Software Developer P05:
"Instead of taking scheduled breaks, I prefer to get the job done and leave earlier."

Team Lead, Software Developer P02:
"Virtual coffee brakes are still meetings."

Software Developer P01:
"Better decide on a day to meet in the office."

P06 thinks that virtual coffee brakes are useful in large companies and that they provide the opportunity to get to know colleagues with whom you are not in direct contact on a daily basis. Yet, he is reluctant if this practice can be applied successfully in his company as this depends on the company culture and finding common topics could be challenging.

Technical Lead, Software Developer P06:
"In my current company most likely people will skip virtual breaks."

Technical Lead, Software Developer P06:
"More than 5 people could struggle to find common topics. Still can't compare them with face-to-face conversations."

On the other hand, P03 and P04 revealed that their companies tried to organize virtual coffee breaks and virtual events but they gave up due to lack of success. In addition, P10 states that the company in which he is working continues to have virtual coffee breaks 2 times a week, but he rarely joins these breaks.

Software Developer P04:
"Discussions in the kitchen can not be scheduled in online."

Team Lead P03:
"Our Christmas party took place online and it was a failure, only 4 people showed up."

Compared to Neumann et al. [22] and Ralph et al. [27] who suggest organizing virtual social events to cope with the faced challenges when trying to bond with the colleagues, this thesis can not confirm the success rate of this techniques. Clearly, as Marek et al. [20] identifies, in-person communication is highly valued.

Just like Neumann et al. [22] and Marek et al. [20] who reported that meetings are more efficient and conducted in a more goal-oriented way, 70% of the interviewees confirmed that the duration of the meetings went down or at least stayed the same (see Figure 5.2).

Software Developer P01:
"Meetings have become shorter, more structured and more precise."

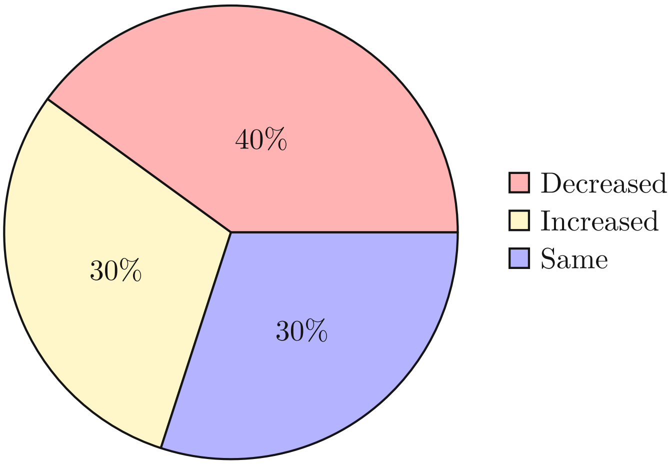


Figure 5.2: Duration of meetings

Software Developer P04:
"We have fewer meetings, but they are more focused."

Due to the fact that the number and duration of meetings are indirect proportional, Mancl and Fraser's [19] statement that online meetings have the benefit of being kept shorter but scheduled more often, is supported. Moreover, according to the conducted interviews staying focused during virtual meetings is more difficult than in face-to-face meetings. Not only Mancl and Fraser [19], but also P02 draws attention to the importance of being able to maintain the participants in a meeting, involved. Particularly, people that seem shy need to be given the opportunity to express their opinions.

Software Architect, Software Developer P07:
"It is hard to pay attention in long meetings online."

Software Developer P09:
"Sometimes people seem to be just physically in meetings."

Team Lead, Software Developer P02:
"People can hide easily. Introverts also have good opinions, so it's a pity that they do not say anything."

An additional factor that plays an important role when conducting meetings is considered to be the group size (P02, P07, P09).

Team Lead, Software Developer P02:
"Meetings with a lot of people are troubling."

Software Architect, Software Developer P07:
"Interaction in meetings depends on group size."

Software Developer P09:
"Some people are shutting down."

In contrast to Neumann et al. [22], who reported increased communication as one of the positives of remote work, 6 out of 10 participants stated that their socialization with colleagues decreased. Furthermore, only 1 interviewee stated that communication increased while 3 other participants did not experience any changes in the communication volume. Compared to Marek et al. [20] who claim that communication with teams improved, only 30% of the participants reported an increase in communication within the team.

Team Lead, Software Developer P02:
"Direct communication within the team, might even be better."

Software Developer P01:
"Communication increased both privately and work-related."

Figure 5.3 reveals on a scale from 1 to -5, how communication changed during the increased period of remote work triggered by the pandemic. 1 means that communication increased, while values from 0 to -5 define if and by which factor it decreased. Communication with the team members was significantly less impacted than communication with other teams.

Team Lead, Software Developer P02:
"No idea what is going on in other teams, communication went down by a lot."

Technical Lead, Software Developer P06:
"Before, I was interacting with colleagues from other teams in the office."

Figure 5.4 shows that the majority considers that the required time dedicated to communication has increased. In addition, none of the interviewees considers that they spent less time for transmitting the intended message during conversations. Like Mancl and Fraser [19] and Neumann et al. [22], especially in early stages of the project, participants reported the inconvenience of not being able to use a physical whiteboard.

Team Lead, Software Developer P08:
"Particularly when trying to sketch something, I realise how much I miss the whiteboard."

Team Lead P03:
"Do you have any idea how difficult it is to draw something using the mouse?"

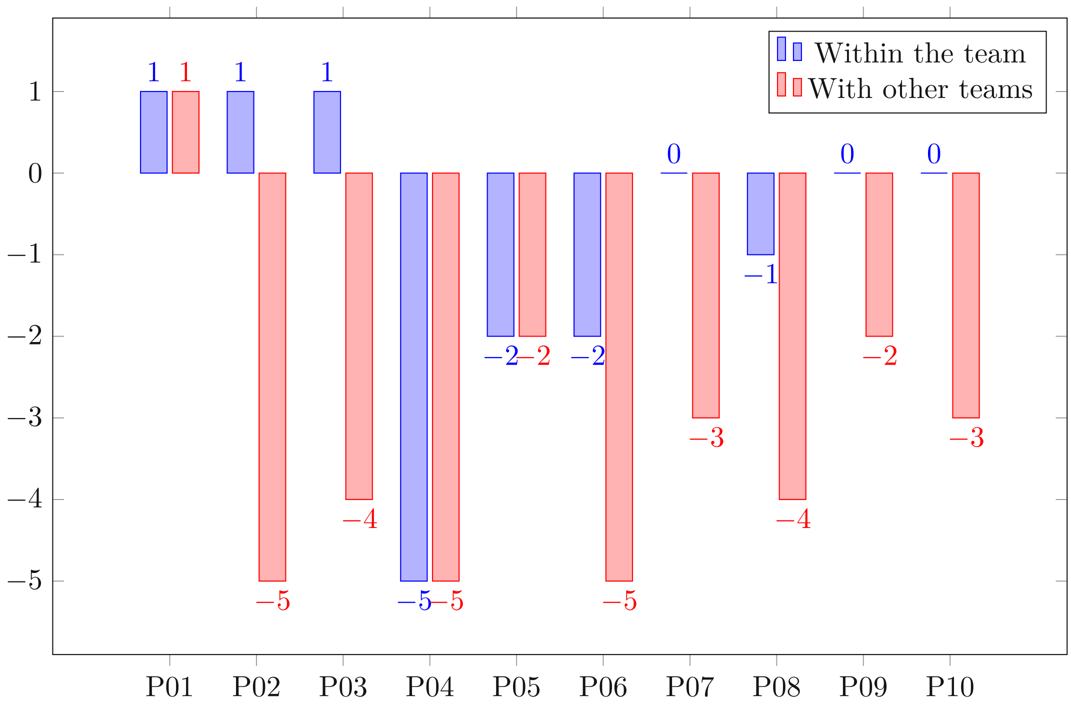


Figure 5.3: Communication with colleagues

Similar to Ford et al. [13], 60% of the interviewees confirm the fact that being active on multiple communication channels adds up extra communication overhead. Furthermore, in order to support focused work, some companies have introduced time intervals when employees should not be disturbed. 70% of the participants are either using or would appreciate these time intervals when other colleagues are not allowed to distract them.

Team Lead, Software Developer P02:
"It is important to know when to chat and when to call."

Software Architect, Software Developer P07:
"I would really appreciate it if people would respect the period of focused time."

Team Lead P03:
"Lack of distractions, makes me more productive."

None of the interviewees stated that their responsibilities have changed as a consequence of remote work. Some did experience some adjustments in their roles, but this happened due to the company being downsized, taking on new responsibilities or being promoted.

Even though 70% of the participants reported that the number of meetings increased and 60% stated that the duration of the meetings increased or stayed the same, socialization with colleagues experienced a decrease according to 60% of the interviewees. In addition,

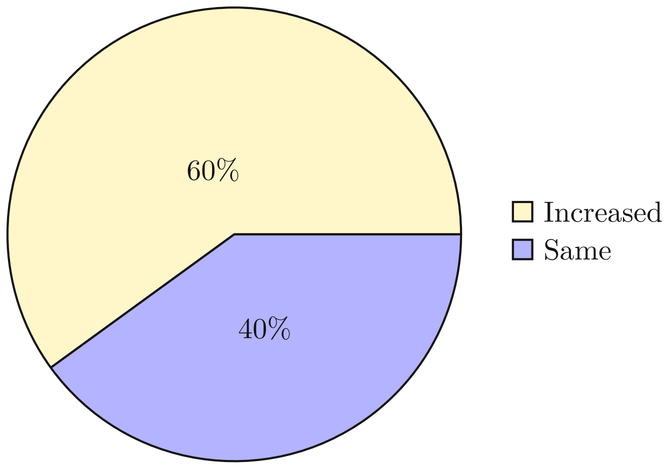


Figure 5.4: Addition time required for transmitting the intended message

30% stated that communication with colleagues did not suffer a decrease while only 1 participant stated that communication with colleagues increased. The majority reported that stakeholders were at least as involved as before the period of remote work (50%) or were more involved while conducting remote work (30%).

5.3 Agile practices

This section examines the software development process before and during the period of increased remote work triggered by the pandemic. As part of this analysis, we take a look based on which agile methods the development process is conducted and which agile practices are part of the development process. In addition, the influence of this period on pair programming and the development cycles is examined.

Based on the conducted interviews, it was identified that the participants are developing products based on 3 agile development methods (see Table 5.3). The majority are using a software development process that is based on Scrum. In contrast to them, two interviewees (P02, P05) are using Kanban, while only one (P04) is using a mixed approach of Scrum and Kanban.

All but one (P02) of the participants, kept using the same agile development method during the increased period of remote work. P02 revealed that the reason behind this change is a result of the downsizing that happened in the company due to the reduction of business opportunities.

Team Lead, Software Developer P02:
"During the pandemic we stopped using sprints because of the reduction of business. We made only fixes and did not develop any new features."

Agile method	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10
Scrum	✓	✓	✓			✓	✓	✓	✓	✓
Kanban		✓			✓					
Mixed approach of Scrum and Kanban				✓						

Table 5.3: Used agile software development methods

Similar to Neumann et al. [22], a set of agile development practices with the goal of analysing how the software development workflow changed while working from home, is established. For this purpose, we take a look at which agile practices were embedded in the software development process before the pandemic and which were continued or not to be used during the increased period of remote work.

Table 5.4 reveals how the development process changed with respect to the analyzed agile practices. The symbol (✓) denotes the fact that the agile practice is being used while (×) marks the fact that it is not part of the development workflow. In addition, transitions that indicate that an agile practice started or stopped being used during the period of increased time are represented with the help of the symbol (→). For readability purposes, each transition that reveals the fact that an agile practice was introduced in the development process is marked with the color blue, while the color red reveals the fact that the specific agile practice is not being used anymore. For instance, P01 stated that they started using *code reviews* when they started working from home while in the case of P06 they stopped doing *pair programming* when conducting remote work. However, in the case of P04 the development process did not experience any changes during the increased period of remote work.

As can be seen in Table 5.4, only three of the interviewees (P03, P04, P09) did not change the software development process during the increased period of remote work. All the others have experienced changes in their development process either by removing or adding new agile practices. *Continuous integration* and maintaining a *product backlog* are the only practices that were used before and during the pandemic by all interviewees. In contrast to *continuous integration*, only 30% used *continuous deployment* while working from home or while working on-site. The most frequently adopted agile practices are the *daily stand up* meeting and the introduction of a *definition of done*. In contrast, *pair programming* is the agile practice that stopped being used the most(30%). Alongside, *continuous deployment*, the least used practices before working from home are conducting a *retrospective meeting* and *test driven development*. The reason behind this is that good self-organized, mature teams do not see the need for retrospectives as well as companies not imposing the usage of this agile practice.

🗨 Software Developer P05:

n	Agile practices AP(n)	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10
1	Code review	× → ✓	✓	✓	×	✓	✓	✓	✓	✓	✓
2	Continuous deployment	×	✓ → ×	✓	×	✓	×	×	× → ✓	×	×
3	Continuous integration	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	Coding guidelines	× → ✓	✓	✓	×	✓	✓	×	✓	✓	×
5	Daily stand-up	✓	✓ → ×	✓	×	✓	× → ✓	✓	× → ✓	✓	× → ✓
6	Definition of done	× → ✓	✓	✓	✓	✓	× → ✓	× → ✓	✓	✓	×
7	Estimation	✓	✓ → ×	✓	✓	× → ✓	✓	×	×	✓	✓
8	Kanban board	×	× → ✓	×	✓	✓	×	×	×	×	×
9	Pair programming	✓	✓	✓	✓	✓	✓ → ×	✓	✓ → ×	×	✓ → ×
10	Planning meeting	✓	✓ → ×	✓	✓	✓	✓	✓	✓	×	✓
11	Product backlog	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
12	Retrospective meeting	×	×	✓	×	×	× → ✓	× → ✓	×	✓	×
13	Review meeting	✓	✓ → ×	×	✓	✓	✓	✓	✓	×	✓
14	Sprint backlog	✓	✓ → ×	✓	✓	×	✓	✓	✓	✓	✓
15	Test driven development	×	×	✓	✓	✓	×	× → ✓	× → ✓	✓	×

Table 5.4: Used agile practices before and during the pandemic

"Test driven development is not enforced in our company, but people do it because they understand the benefits."

Team Lead, Software Developer P02:
"Our team was so mature that there was no for regularly retrospective. There was no need to reflect over what could be improved or what went wrong. Everybody understood this and tried to do better."

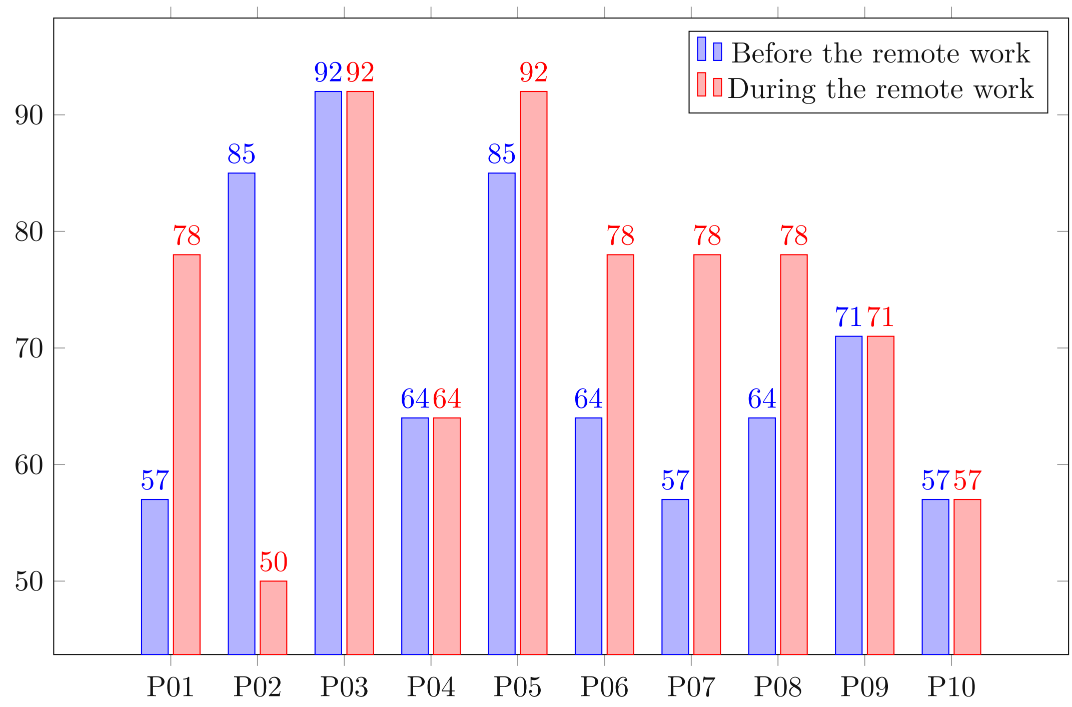


Figure 5.5: Agility index before and during remote work

In order to be able to assess how the agile software development process changed during the period of remote work, an *agility index* is computed. Based on Table 5.4 and on an own defined equation (see Equation 5.1) an agility index for each interviewee, is calculated, as can be seen in Figure 5.5. This index varies from 0 to 100, meaning that a higher value defines a more agile development process.

Based on the value of the agility index in Figure 5.5, the changes in the software development process can be observed for each interviewed person. In comparison to P01, P07 and P10 who had the least agile software development process before conducting remote work, P03, P02 and P05 were using the most agile software development process. While P01 encounters the highest improvement with respect to the analysed agile practices, P02 encounters the highest decreases thanks to the downsizing process suffered by the company he is working in. As also reveled by Table 5.4, P03, P04 and P09 did not experience any adjustments in the software development process. Even though in case of P10 some adjustments were made, the agility index did not change. As a consequence of the adjustments that took place during the increased period of remote work, 90% of the participants in the interviews either maintained (40%) or improved (50%) the agility of the used software development process.

Equation 5.1 reveals the formula used for calculating the agility index. This is a sum that iterates over the number (15) of agile practice defined in Table 5.4 and adds up the result of the function $AP(n)$. The denominator of the fraction is 14 because according to the definition of $AP(n)$ (see Function 5.2), the result of the sum ranges between 0 and 14. In case, all the all the agile practices defined in Table 5.4 are used, the result of the fraction is 1 and the agility index is 100.

$$AgilityIndex = \frac{\sum_{n=1}^{15} AP(n)}{14} \times 100 \tag{5.1}$$

For the evaluation of the function $AP(n)$, the following cases can be distinguished (see Function 5.2). Firstly, $AP(n)$ evaluates to 1 if the agile practice n is part of the development process. Secondly, if the agile practice: *Kanban board* is being evaluated ($n = 8$), then $AP(8)$ is 1 if the agile practice *sprint backlog* is not being used. Otherwise, $AP(8)$ is 0. Thirdly, the agile practice *sprint backlog* is also evaluated with respect to the practice *Kanban board*. In this way, software development processes that combine Kanban and Scrum do not achieve a higher agility index, than software development processes that are based only on Kanban or Scrum. The reasoning behind this logic is to avoid obtaining a higher agility index for processes that contain of a mix of Kanban and Scrum than software development process than just using Scrum or Kanban. Last but not least, $AP(n)$ evaluates to 0 in case the agile practice defined by n is not being used in the development process.

$$\begin{gathered} AP(n) = f : X \rightarrow Y \\ X = \{1, \ldots 15\}, Y = \{0, 1\} \\ AP(n) = \begin{cases} 0 & if\ n = 8\ and\ AP(14) = 1 \\ 0 & if\ n = 14\ and\ AP(8) = 1 \\ 1 & if\ agile\ practice\ n\ is\ used \\ 0 & if agile\ practice\ is\ not\ used \end{cases} \end{gathered} \tag{5.2}$$

Being one of the challenges encountered during remote work, pair programming was conducted less often when working from home. As shown by Figure 5.6, 60% of the participants confirmed Neumann et al. [22] findings. Neumann et al. [22] state that pair programming is conducted via screen sharing. Therefore the time spent for pair programming decreased. Still, 30% think that they spent the same amount of time for pair programming.

🗨 Team Lead, Software Developer P08:
"Pair programming was highly affected. Before we were doing a lot of pair programming. Basically, we stopped doing it."

🗨 Software Developer P05:
"Even before the pandemic we did not do that much pair programming. I think it stayed the same."

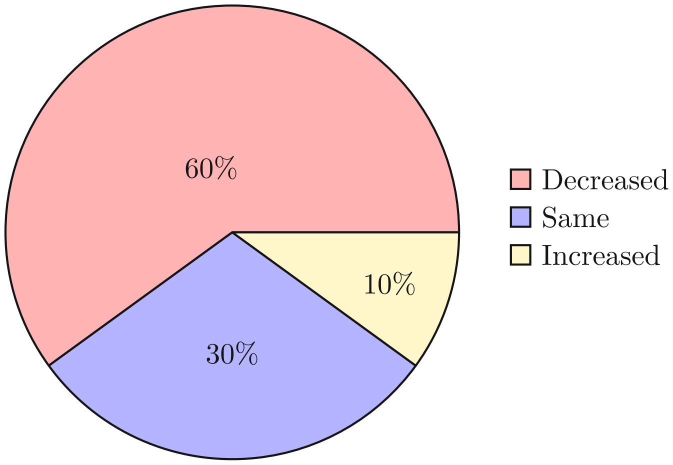


Figure 5.6: Time spent for pair programming

In addition, all interviewed developers that continued practicing pair programming during the pandemic stated that the experienced quality decreased. Moreover, in contrast to Mancl and Fraser [19] who suggested techniques to coupe with the faced challenges, none of the participants are using any technique to improve the experience of remote pair programming.

Software Developer P01:
"Pair programming was better before. Now we do it via screen sharing."

Software Developer P04:
"It was casual before, we could do it spontaneously. Now it is more official. We need to schedule a meeting."

However, P02 reported a positive experience with remote pair programming and stated that the time spent increased particularly because this method is used to mentor developers during the increased period of remote work triggered by the COVID-19 pandemic.

Team Lead, Software Developer P02:
"Pair programming improved during the pandemic. We use it for complex problem solving and also to educate people."

Just like Marek et al. [20], as can be seen in Figure 5.7, the majority (60%) of the interviewees maintained their development cycles, while 30% (P01, P06, P08) have shortened the development cycle. This decision was taken because they assumed that remote work will affect their response to changes and shorter cycles will help them react at least in the same amount of time to changes. In the case of P02, they stopped using Scrum, and therefore analysis of how the development cycles changed does not apply for 10% of the interviewees.

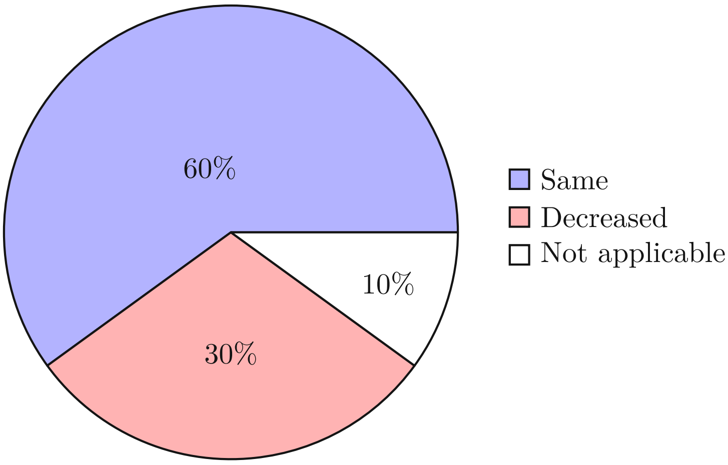


Figure 5.7: Changes of the development cycles

> Technical Lead, Software Developer P06:
> *"We presumed that remote work will slow us down and for a better response to changes we decided to change the sprints from 2 weeks to 1 week."*

> Team Lead, Software Developer P08:
> *"I think we wanted to asses potential problems faster."*

In contrast to Neumann et al. [22], the majority (50%) of the interviewees (P03, P04, P05, P09, P10) reported that during the period of increased remote work, the stakeholders continued to be as involved as before working from home. The involvement of the stakeholders increased for 30% (P06, P07, P08) of the participants. Marek et al. [20] also states that lowest percentage (16%) was encountered for a decreased in the stakeholders collaboration, while the highest percentage (57%) is assigned to constant stakeholder involvement.

> Team Lead P03:
> *"Our stakeholders were involved a lot and they continued to be as involved."*

> Technical Lead, Software Developer P06:
> *"Stakeholders involvement has gone up. Maybe also because we go a new project manager."*

> Software Architect, Software Developer P07:
> *"Because of online meetings, stakeholders could join from other locations. Stakeholders joined meetings instead of doing business trips."*

Table 5.4 confirms the fact that 60% of the participants did not change the definition of done during the remote work period. According to the study conducted by Marek et al.

[20], the majority did also not change the definition of done. In the case of this thesis, 30% of the participants (P01, P06, P07) decided to introduce a formal definition of done but stated that this did not change their perspective on the resolution of a task.

Overall, all participants except P02 and P10 stated that they prefer the software development used during the pandemic. P02 expressed its disappointment over the fact that the company had to downsize, while P10 considers that nothing has really changed. This is understandable, as according to Figure 5.5, P02 has the lowest ability index during the period of remote work, and in the case of P10 it did not change. In addition, when asked how difficult it was to adapt to the new software development process, none of the participants mentioned such difficulties. This can be a consequence of the fact that all companies improved their software development process by introducing additional agile practices and the fact that all participants were already accommodated to an agile software development process. The challenges that were mentioned are related to infrastructure, as in the early stages after switching to remote work some interviewees had issues with VPN and remote access to the office infrastructure. Furthermore, almost all interviewees experienced issues with their internet connection.

5.4 Work routine

This section provides an analysis of how the work routine of agile software engineers was influenced during the increased period of remote work by the factors presented in Table 5.5. Positive changes are marked with the symbol (↑) while (↓) describes a negative impact on the work routine. If the analysed factor does not have any impact on the work routine, the symbol (=) is used. For example, 60% (P03, P04, P05, P07, P09, P10) of the participants revealed that not having a dedicated workspace at home has a negative impact on their work routine. However, 40% (P01, P02, P06, P08) stated that their work routine was not influenced by this factor as they had a dedicated workspace at home. Flexibility is the only factor that was recognized by all participants as a positive influence.

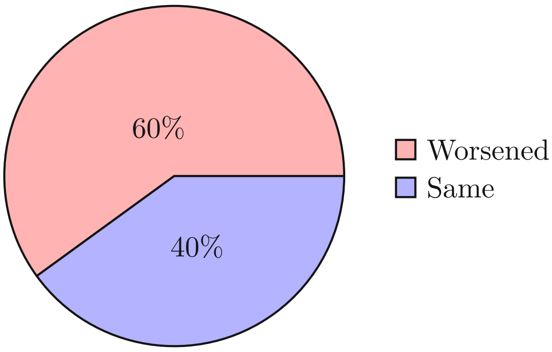
As can be seen in Table 5.5, 70% of the interviewees are very pleased by the lack of commute time, as identified by Mancl and Fraser [19] and Ford et al. [13]. Initially, it was assumed that every participant finds lack of commute time as a positive factor that influences their work routine. This assumption alongside the findings of Mancl and Fraser [19] and Ford et al. [13] are contradicted by 30% of the participants. In this case, P03, P04, and P05 consider lack of time as a missing part of their day-by-day work routine.

- Team Lead P03:
  "Commute time was good for me, as I could listen to audiobooks. This was time that I spent every day to improve myself."

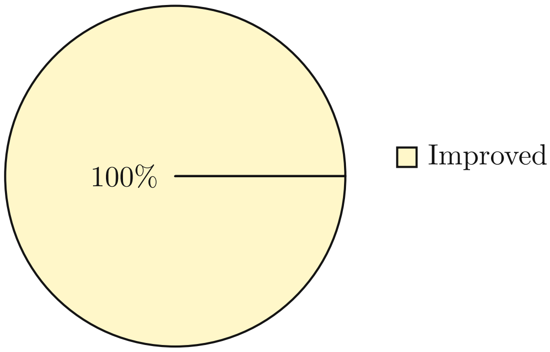
- Software Developer P04:
  "Routine is very important when trying to achieve something. The first part of my work routine was going to the office."

Work routine factors	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10
Commute time	↑	↑	↓	↓	↓	↑	↑	↑	↑	↑
Different working hours	=	↓	↓	=	=	↓	↓	↓	=	↓
Flexibility	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Distractions	↓	↑	↓	↓	↓	↑	↓	=	=	↓
Work overtime	↓	=	↓	↑	↓	↓	↓	=	=	↓
Ergonomics	=	=	↓	↓	↓	=	↓	=	=	↓
Well-being	↑	↓	↓	↓	↓	↓	↓	=	↓	=
Dedicated workspace	=	=	↓	↓	↓	=	↓	=	↓	↓

Table 5.5: Factors that influence the work routine



(a) Working hours - Impact on work routine



(b) Flexibility - Impact on work routine

Figure 5.8: Different working hours and flexibility - Impact on work routine

Software Developer P05:
"Commute time is important for me as it makes the transition from home to work. It helps me prepare for work and disconnect after work."

The contrast between different working schedules and flexibility is very surprising (see Figure 5.8), as allowing employees to work at different hours goes hand in hand with flexibility. Yet, the result is understandable. When asking about working at different hours, participants expressed their opinion about how their work routine is affected by their colleagues working at different hours. In contrast, when talking about flexibility, participants relate to themselves.

According to Figure 5.8b, all participants appreciate the possibility of having a flexible working regime. Most of the participants think that flexibility is a must when talking about software development and that flexibility improves their productivity. During the period of increased remote working, flexibility helped people deal with their everyday responsibilities during work time. In addition, all participants stated that they would not consider working for companies that do not offer a flexible working schedule.

Team Lead, Software Developer P02:
"Flexible working hours are a must. Core hours are accepted."

Software Developer P04:
"I can not imagine how we could have taken care of our baby without flexible working hours."

On the contrary, Figure 5.8a reveals the fact that the work routine is impacted by colleagues working based on different working schedules. Paasivaara et al. [24] and Shrivastava and Date [34] also warn about communication being affected by different schedules in distributed development.

Software Architect, Software Developer P07:
"Trying to reach somebody while doing home office is significantly slower than walking to his/her desk."

Team Lead P03:
"Sometimes due to synchronized working schedules, solving issues is delayed for tomorrow."

Furthermore, as mentioned by Butt et al. [7], the frequency of people not joining meetings due to private matters has increased.

Team Lead, Software Developer P08:
"I can't even remember a whole week when all participants in the daily were present or did not join later."

Technical Lead, Software Developer P06:
"Being in the office did not make it possible to receive deliveries or schedule some appointments at the time when meetings take place."

Software Developer P10:
"People find it more normal to be able to skip a meeting now."

Figure 5.9 shows that 20% of the participants were less distracted (P02, P06) when conducting remote work. These interviewees supported the findings of Neumann et al. [22] and Ford et al. [13] who reported that remote work allows people to have better control over their work environment and work more focused for longer periods of time. P08 and P09 felt the same level of distraction while working from home, as office distractions were replaced by other distractions. Nonetheless, 60% of the participants in the interviewing

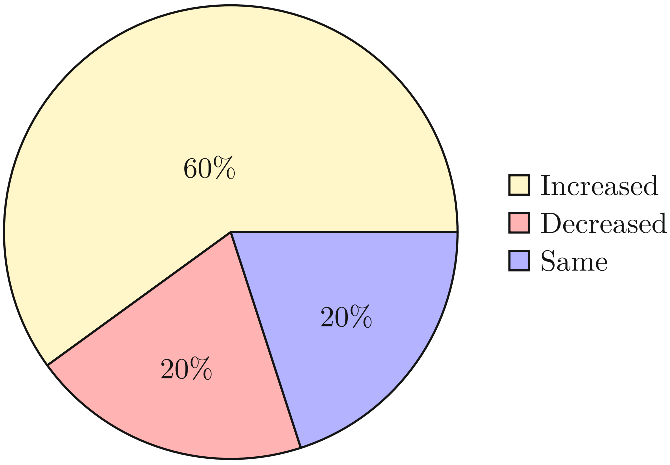


Figure 5.9: Distractions - Impact on work routine

process support da Mota Silveira Neto et al. [9] findings by revealing that they are more distracted while working from home. The participants involved in the conducted interviews, invoked similar distractions to the ones observed by Ralph et al. [27] and Ford et al. [13].

🗨 Software Developer P01:
"If you have personal problems you can not detach by going to work."

🗨 Software Architect, Software Developer P7:
"Trying to work with 3 children at home is pretty challenging."

🗨 Team Lead P03:
"Sometimes it happens that my wife and I distract each other by having meetings at the same time. If we had a dedicated space for working this would not be a problem anymore."

Same as Ford et al. [13], who mentioned that blending personal time with working time can lead to working extra hours, a majority of 60% (P01, P03, P05, P06, P07, P10) of the interviewees worked overtime (see Figure 5.10). Especially lack of making progress during regular work hours, too much time spent in meetings, as well as no strict separation between personal time and work time, are the main reasons mentioned by the interviewees.

🗨 Software Developer P05:
"There were days when I just wasn't as efficient as I normally am and I worked extra hours to compensate for it. In the office, I would have not done it as everybody could have seen I was in the office and I did work."

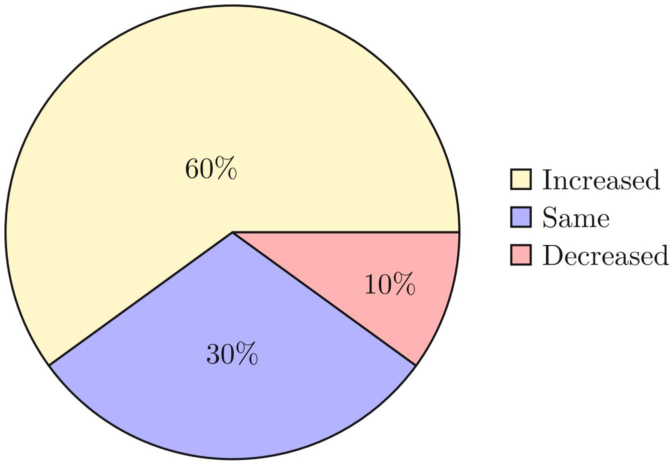


Figure 5.10: Work overtime

🗨 Team Lead P03:
"Because I spent so much time in meetings, I often need to work overtime to finish my tasks."

🗨 Software Architect, Software Developer P07:
"Sometimes after I finished a workday, something comes up and I log in just solve something quickly. Unfortunately, in most of the cases, this takes more than expected."

Nevertheless, 30% (P02, P08, P09) did not experience any changes in how much time they worked overtime. Meaning, that they either worked the same amount of extra hours or did not work more than the regular working schedule. Only, P03 clearly revealed that he decreased the amount of extra hours to the point where he is not working overtime anymore. This happened as a result of having a clear separation of work and personal time, having previous experience with remote work as well as being able to carry on with their work routine.

🗨 Team Lead, Software Developer P02:
"There needs to be a clear difference between personal life and work."

🗨 Software Developer P09:
"I did work extra hours in the past while working from home, so I knew what to expect and not work overtime."

🗨 Team Lead, Software Developer P08:
"Even though I was the only one going to the office, I kept going to the office during the whole time."

Similar to Mancl and Fraser [19] who raised attention over the potential issues of unfair management, P01 and P04 expressed disappointment over micro-management.

Software Developer P01:
"It is a little bit more difficult to delimit work time when the project manager writes you at 9 PM."

Software Developer P04:
"You can not expect that whenever you decide to write you get an instant response. Even in the office I am not available the whole time."

As revealed in Table 5.5, ergonomics and dedicated workspace had a negative impact on most of the interviewees. This findings are also supported by and suggested by Ralph et al. [27] and Ford et al. [13] who mentioned this challenges in their research. In addition, when referring to ergonomics and dedicated workspace, no one experienced an improvement while working from home office. Meaning that nobody has better work conditions at home than in the office.

In conclusion, all the interviewees have experienced changes in their day-by-day work routines. Even though several factors impacted their work routines, nobody had a hard time adapting to working from home. Furthermore, all participants expressed their preference for a mixed work routine that consists of both the home office and going to the office.

5.5 Productivity

This section analyses the perceived impact on productivity with respect to each interviewee. Table 5.6 reveals the productivity changes experienced while working from home. An increase in productivity is marked with the symbol (↑), while the symbol (↓) represents a decreases. If the participants did not encounter any productivity changes, the symbol (=) is used.

Productivity	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10
Short term impact	↓	=	↓	↓	↓	↓	=	↓	=	↓
Long term impact	↑	=	=	=	=	=	=	=	=	↓

Table 5.6: Impact on productivity

During the interviews, it was observed that in most of the cases productivity was not constant during the increased period of remote work. Therefore, the decision to analyse the impact on productivity in the short term and in the long term, was taken. Short term refers to the initial phase of the remote work period, while long term refers to the rest of the period. As can be seen in Table 5.6 only 30% (P02, P07, P09) of the

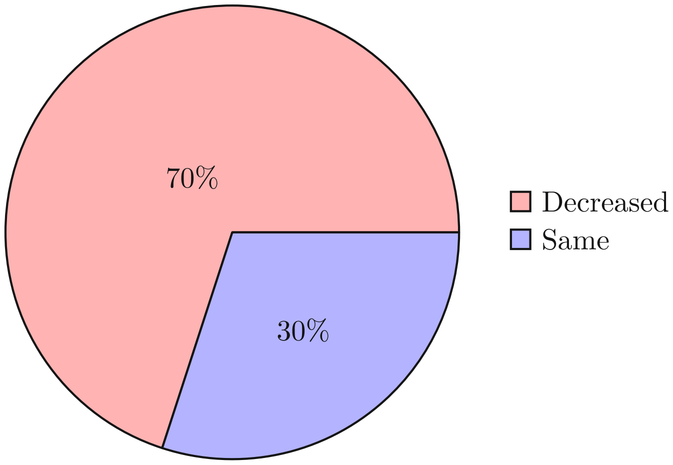


Figure 5.11: Short term impact on productivity

interviewees reported that productivity remained constant both in the short and long term. Furthermore, P01 is the only participant that feels more productive while working from home. In contrast to all other interviewees, P10 did not manage to achieve the same productivity level as before the increased period of remote work.

Figure 5.11 illustrates that 70% (P01, P03, P04, P05, P06, P08, P10) of the participants encountered a decrease in productivity during the initial phase of the remote work period. Infrastructural issues, unsuitable development processes as well as the mental state of developers are the main factors that contribute to this decrease.

Software Developer P05:
"It took some time until we managed to get access to our resources and be able to properly work from home."

Software Developer P04:
"Besides concerns brought by the pandemic, I also proved to be very responsive to distractions."

Technical Lead, Software Developer P06:
"Our processes and infrastructure were not that suited for remote work. We did not have a proper way to access and work with our systems."

In contrast, 30% (P02, P07, P09) managed to keep being as productive as before the switch to the home office due to working extra hours, trying to follow a routine, and due to previous experience with remote work.

Software Architect, Software Developer P07:
"While working from home, I tried to follow my usual routine. Wake up as I normally do and work based on the same time schedule as in the office."

🗨 Software Developer P09:
"I would not say that I was less productive, but I definitively worked more hours."

Compared to the initial impact on productivity, the majority (90%) managed to restore or even exceed the productivity levels in the long term as can be seen in Figure 5.12. It must be noted that out of the 90%, 60% (P01, P03, P04, P05, P06, P08) managed to reestablish the achieved productivity level as before the increased period of remote work. In addition, 4 (P01, P03, P05, P06) out of these 6 participants stated that they are only able to sustain their productivity level by working extra hours and by following the adjusted development processes. As this represents, 40% of all the participants and 66% of the participants that managed to recover from the productivity drop experienced in the short term, there is doubt over the sustainability of productivity under the circumstances of the increased period of remote work triggered by the pandemic. This also mentioned by Ralph et al. [27].

🗨 Software Developer P04:
"From home, I am regularly working more hours to try to be as productive as in the office. This impacts my well-being."

🗨 Software Developer P05:
"Productivity recovered after the initial phase, but with more invested time. This and the lack of separation between private space and work are an issue for me in long term."

🗨 Technical Lead, Software Developer P06:
"After we tailored to development processes to be suited for remote work, my productivity increased to the level as before the pandemic. I would not say I am more productive during home office."

To get a better understanding on what influences the perceived productivity, the participants in the interview were asked to rate the factors in Table 5.7 on a scale form 1 to 5. Meaning that a low impact is marked with 1 and 5 represents a high impact. To begin, processes stand for the relevance of used agile practices when referring to productivity. Work routine includes the influence of working based on different time schedules, flexibility as well as working overtime. Accessing the required resources from remote work as well as the influence of the used tools on productivity are denoted by infrastructure. Last but not least, the influence of distraction and previous experience while working from home are also taken into consideration.

By taking into consideration the evaluation provided by each participant in Table 5.7, Figure 5.13 reveals the overall impact of each influence on productivity as the sum of each individual evaluation. Similar to Marek et al. [20], processes designed based on agile principles, as well as the infrastructure, have the most impact on agile practitioners performance (see Figure 5.13).

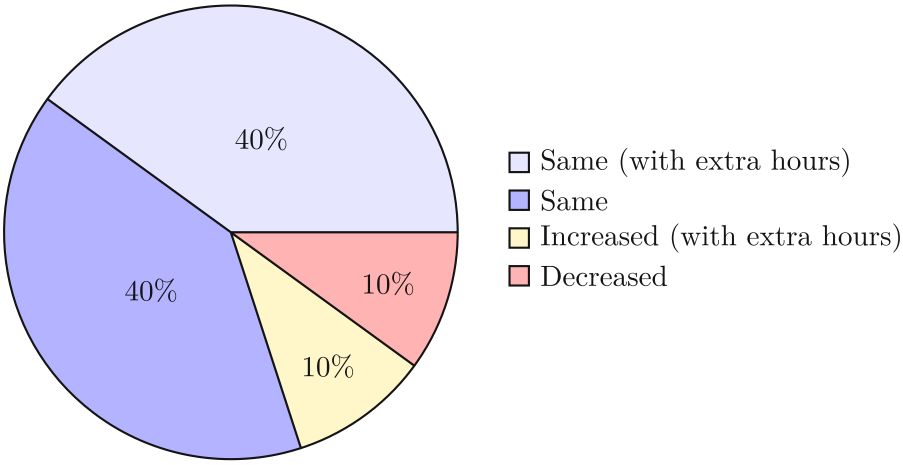


Figure 5.12: Long term impact on productivity

Productivity Influences	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10
Processes	3	4	5	4	3	5	4	4	3	3
Work routine	3	4	4	5	3	4	3	2	5	3
Infrastructure	4	5	4	4	5	5	4	4	4	5
Distractions	3	3	3	4	2	4	3	4	4	5
Previous experience with remote work	1	2	1	1	2	1	2	2	3	2

Table 5.7: Productivity influences

Team Lead P03:
"The agile development method should be chosen based on the project. Personally, I consider Scrum better for developing new features and Kanban better for maintenance."

Team Lead, Software Developer P02:
"I consider infrastructure the most important factor when talking about productivity. Processes improve productivity, but without infrastructure, processes can not be properly followed from the home office. "

Team Lead, Software Developer P08:
"Things started to move forward again after we solved our issues with the infrastructure and adjusted the development workflow."

As observed in Figure 5.13, work routine and distractions are very close to each other when analysing the impact on productivity. The high estimation for how distrusting

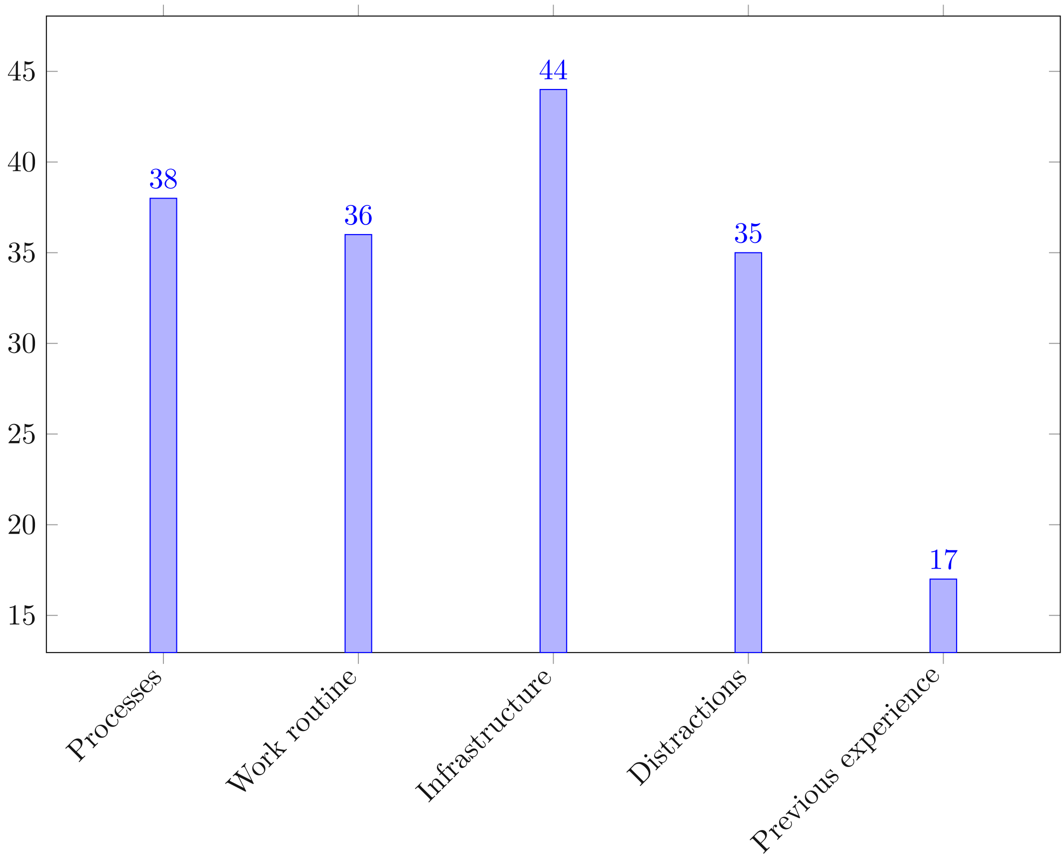


Figure 5.13: Productivity influences

distractions are from the participants' perspective, confirms Neumann et al. [22], Mancl and Fraser [19] findings that people are able to concentrate better and appreciate them to be more productive in an environment with fewer distractions.

- Software Developer P04:
  "Having a flexible work routine is one of the most important factors that help in being productive. Especially when you share your workspace with your family."

- Technical Lead, Software Developer P06:
  "Due to our open space office I feel more distracted at work than at home."

Last but not least, the agile practitioners involved in the interview shared their opinion on how suited they think the used software development process is for conducting remote work. Figure 5.14 reveals a comparison between the software development processes before the adjustment that took place during the increased period of remote work and after the adjustment. The evaluation is made based on a scale form 1 to 5. More exactly,

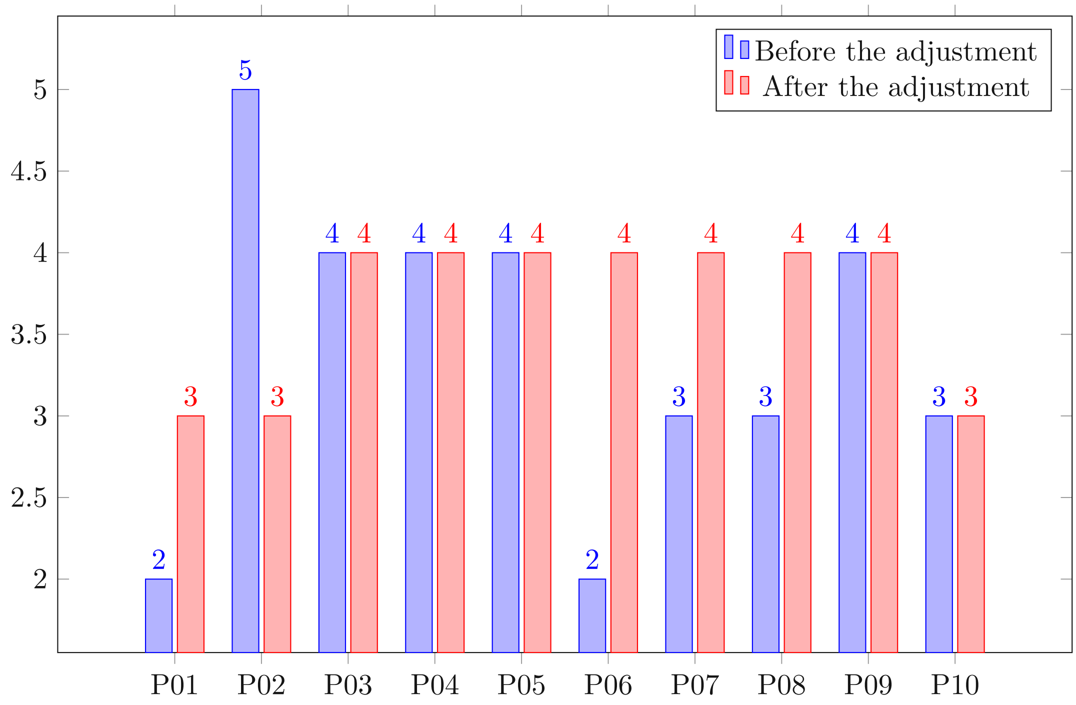


Figure 5.14: Suitability for remote work

1 means that the processes based on which the interviewees develop software are not a good fit for remote work. In contrast, 5 denotes that the software processes are very suitable for working from home.

As can be seen in Figure 5.14, 20% (P01, P06) of the participants consider the used processes before the increased period of remote work as the least suited within the whole interviewee group.

Software Developer P01:
"We were very dependent on each other, we are giving more importance to the documentation now."

Technical Lead, Software Developer P06:
"Issues are maintained in a more detailed way now. The goal is that everybody should be able to understand what is going on in the issue."

On the other hand, P02 is the only participant that considers that the used processes before the pandemic did not require any improvement. The majority (P03, P04, P05, P09), consider the used processes prior to the adjustment, as suited for remote work as after the adjustment but with room for improvement. Furthermore, 70% (P03, P04, P05, P06, P07, P08, P09) of the participants who adjusted the development processes as a response to remote work, consider the adjusted processes suitable (4) for remote work.

Moreover, all these interviewees managed to cope with the initial productivity decrease and regain their productivity level as before the increased period of remote work (see Table 5.6).

- Software Developer P05:
  "If we could reduce the meetings, I would consider rating the suitability to remote work with 5."

- Software Architect, Software Developer P07:
  "There is still room for improvement, but I consider our processes well designed for working from home."

According to Figure 5.14, P02 and P06 reported the highest difference in how suitable the processes are before and after the adjustment. P02 considers the processes less suited, while P06 considers the processes have improved. This fact, is also supported by the agility index (see Figure 5.5). In addition, all but one participant who improved their agility index (see Figure 5.5) consider the adjusted development processes more suited for remote work. The only exception is P05, who considers the processes before the adjustment to be as suited for remote work as after the adjustment.

To conclude, similar to Mancl and Fraser [19], who observed that high-performing teams continue to perform while working from home, the participants confirmed that tailored agile development processes alongside work routine, play an important role in maintaining and regaining the productivity levels when conducting remote work. In addition, a correspondence between the agility index and suitability for remote work was identified. Furthermore, by analysing Table 5.6 and Figure 5.5, 5 out of the 6 interviewees who managed to regain their productivity levels (P01, P04, P05, P06, P08) to the ones before the period of remote work, have improved their agile processes and obtained a higher agility index. The other interviewee (P03) who was able to cope with the initial productivity decrease, maintained his agility index before and during the remote work period. More exactly, the used development process did not become less agile when switching to remote work. Hence, all participants that managed to achieve being as productive as before the period of remote work, experienced an improvement or at least maintained the agility of the used development process.

5.6 Quality

The impact on quality is assessed with respect to Table 5.8 by examining if the development cycle decreased (↓), increased (↑), or stayed the same (=) and by analysing if the definition of done changed (✓) or not (×).

By analysing the impact of the increased period of remote work on the delivered quality, da Mota Silveira Neto et al. [9] statement that quality did not decrease is backed up. According to Table 5.8 all interviewees consider that the quality level before and during

	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10
Quaility	=	=	=	=	=	=	=	=	=	=
Development cycle	↓	=	=	=	=	↓	=	↓	=	=
Definition of done changed	✓	×	×	×	×	✓	✓	×	×	×

Table 5.8: Impact on quality

the period of remote work is the same. Particularly, agile practices such as code review, continuous integration, coding guidelines, pair programming, and test-driven development are considered to be a key factor in maintaining the quality level.

Software Developer P05:
"Quality is assured by having code reviews, automated deployment and tests. We also do pair programming for complex tasks."

Team Lead, Software Developer P08:
"Our quality did not drop and this is because of the processes that we use."

In fact, not only agile practices and quality processes, but also developers' desire to continuously improve themselves is very appreciated.

Software Developer P05:
"Most of my colleagues understand the importance of self-development."

Software Developer P09:
"Tools and process help improve the quality, but I also think people should not rely only on tools and processes and seek to improve themselves."

Software Developer P09:
"Improvement is an iterative process."

Furthermore by analysing Figure 5.15, even though Neumann et al. [22] considers developers to be more efficient due to lack of distractions, an impact on quality can be observed. 40% (P01, P03, P07, P08) consider they are only able to sustain the quality level only by investing more time.

Team Lead P03:
"As a team, we maintained the quality, but everybody is investing more time."

Team Lead, Software Developer P08:
"We started to invest more time in testing so that we can sustain the delivered quality."

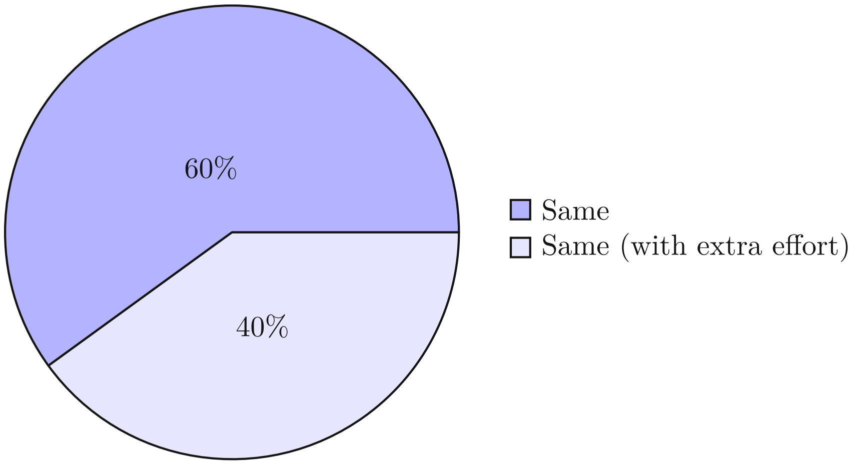


Figure 5.15: Impact on quality

In spite of the fact that P01 considers the delivered quality level remained constant during the increased period of remote work, customer satisfaction increased due to the fact that a dedicated person was assigned to support the customer when reporting problems.

🗨 Software Developer P01:
"Customer satisfaction increased in our case. We now have a dedicated person that supports the customer. This allows us to better use our time for development as only in rare cases, we still need to provide customer support."

On the contrary, P03 mentioned that some stakeholders expressed dissatisfaction due to communication issues with other stakeholders.

🗨 Team Lead P03:
"Sometimes stakeholders are not aware of other stakeholders' previous decisions and are not satisfied with some features."

Figure 5.16 reflects the percentage of interviewees that changed their definition of done during the increased period of remote work. Just like da Mota Silveira Neto et al. [9], the majority (70%) of interviewees did not change the definition of done, while 30% consider it was improved.

🗨 Technical Lead, Software Developer P06:
"We did not have a written DoD, but I think it is better that we have an official DoD now."

🗨 Software Architect, Software Developer P07:
"The DoD is more comprehensive now, but I think people should be aware of when a task is done or not."

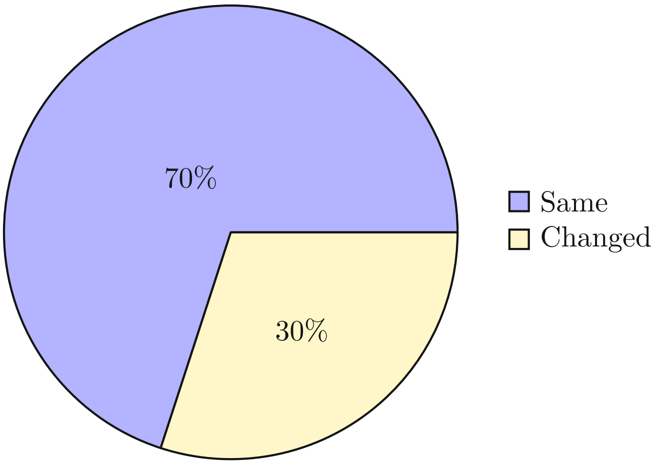


Figure 5.16: Impact on definition of done

In a similar manner to Marek et al. [20], who stated that not increasing the development cycle as well as having a clear definition of done maintains a high level of quality, none of the participants experienced a decrease in quality. In addition, none of the participants increased their development cycles. Further, only 30% of the interviewees changed their definition of done by making it more comprehensive.

5.7 Onboarding

This section reveals the influence of the increased period of remote work on new hires. Table 5.9 illustrates if the time dedicated by the interviewees to mentoring activities increased (↑), decreased (↓), or stayed the same (=) during the period of remote work triggered by the pandemic. In addition, the impact on the time required to accommodate newcomers (onboarding time) is also analysed.

	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10
Mentoring time	↓	↑	↑	=	↓	↓	↓	↓	=	=
Onboarding time	↑	↑	↑	↓	↑	↑	↑	↑	↑	↑

Table 5.9: Impact on onboarding

As can be seen in Figure 5.17, only 20% (P02, P03) of the participants consider that they spent more time conducting mentoring activities, while 30% (P04, P09, P1) consider that they spent the same amount of time as before the increased period of remote work.

Team Lead, Software Developer P02:
"My role has shifted a little bit more towards mentoring."

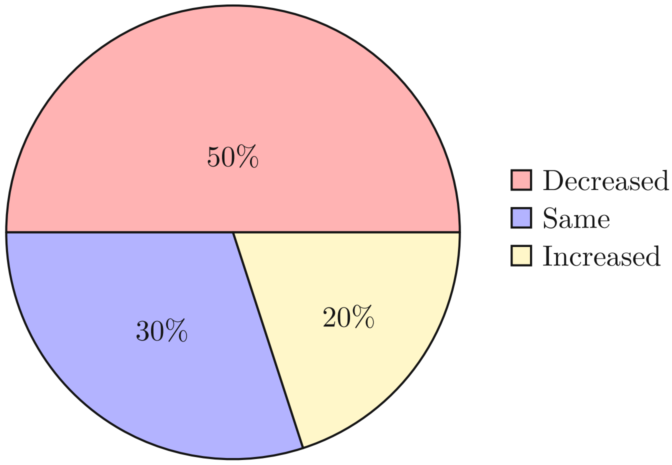


Figure 5.17: Time spent for onboarding new hires

Team Lead, Software Developer P08:
"Even the initial setup takes more time. It is slower to tell somebody to go access this and that resource instead of sitting next to them and showing them."

Similar to da Mota Silveira Neto et al. [9], 50% (P01, P05, P06, P07, P08, P10) of the participants dedicated less time to supporting other colleagues.

Team Lead, Software Developer P08:
"In the office, I am more involved in helping other colleagues as they can walk over to my desk and ask me for assistance."

Software Architect, Software Developer P07:
"The amount of questions I get asked has decreased significantly during remote work."

Further, 90% of the interviewees consider that onboarding time for newcomers has increased (see Table 5.9). Firstly, setting up the environment requires more time over remote work. Secondly, communication issues are less frequent in the office. For example, delegating a task and offering to provide help is different during remote work, as you do not have visual contact and it is harder to understand when somebody is struggling. Last but not least, establishing trust and integration within the team takes more time without face-to-face interaction.

Team Lead, Software Developer P02:
"I used to delegate tasks and frequently provide the opportunity to ask questions. During remote work, newcomers need to take the initiative and ask questions."

🗨 Team Lead P03:
"Working with an existing team during remote work is fine, but building a team over remote work is a challenge. It is a lot harder to read people and understand their needs. Trust is also more difficult to build."

Likewise Rodeghero et al. [30], the majority of the interviewees reported communication and collaboration challenges that contribute to a slower integration process for new developers, as also mentioned by Neumann et al. [22].

In summary, even though the onboarding time required by new hires to integrate into software development teams has increased according to 90% of the participants, the time dedicated by the participants to mentoring activities has decreased.

5.8 Return to office vs. continue to work from home

In this section, the debate about returning to the offices or continuing to work from home is analysed. The goal is to understand the preferences of the interviewees and discuss the topic of the remote and on-site working environment.

As identified by Marek et al. [20], Mancl and Fraser [19], the possibility of working from home is very appreciated. As can be seen in Table 5.10, all participants expressed their desire to be able to work from home.

Preferred work time	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10
In office	20%	20%	40%	20%	40%	60%	60%	80%	20%	40%
Home office	80%	80%	60%	80%	60%	40%	40%	20%	80%	60%

Table 5.10: Preferred work time distribution

Based on Table 5.10 and on the fact that a work week consists of 5 days, it was observed that 70% (P1, P2, P3, P4, P5, P9, P10) of the interviewees would prefer to conduct remote work between 3 to 4 days per week and 30% (P6, P7, P8) between 1 and 2 days per week, as can be seen in Figure 5.18. Furthermore, all participants consider the possibility of concluding home office as a benefit.

🗨 Technical Lead, Software Developer P06:
"Working on-site has it benefits but it should not be mandatory."

🗨 Team Lead P03:
"Home office is a benefit, but only home office is not ok for me."

Moreover, some interviewees demand to have this option.

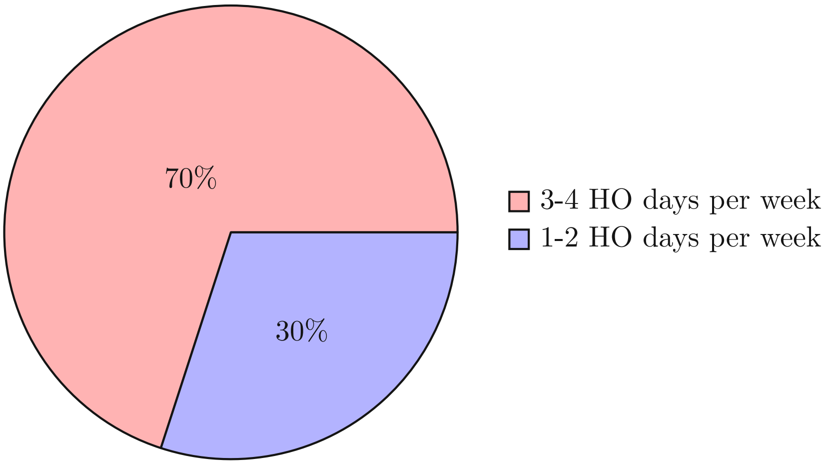


Figure 5.18: Preferred home office days per week

- Software Developer P05:
  "Software development is about flexibility. I do not understand why they focus on bringing people back to work. Not having the option of home office, would make me think about changing my employer."

- Team Lead, Software Developer P02:
  "Convincing people to return to fully work in the office is hard as productivity did no go to the floor while working from home."

Even though, as noted by Mancl and Fraser [19], the virtual working regime requires organizational changes, I can support Ford et al. [13] statement that the software development working environment is developing towards a mix of both remote and on-site working.

Last but not least, the participants were asked what was the best and what was the worst thing experience during the increased period of remote work. Likewise, Marek et al. [20] most of the participants consider remote work as one of the best things that were introduced. Companies were forced to find solutions and therefore the process of migrating towards a mixed working environment accelerated. On the other hand, lack of interaction with colleagues is considered one of the worst things encountered during the period of remote work.

CHAPTER 6

Discussion of results

The analysis and therefore the results mentioned in this thesis are exposed to the threats mentioned below. Even though several steps were taken to reduce these concerns, some threats to validity must be taken into consideration. Therefore, the threats are analysed from three perspectives.

Firstly, construct validity, which according to Runeson and Höst [31] reflects *"to what extent the operational measures that are studied really represent what the researcher have in mind"*. The interview is structured based on many questions and several topics. During the interviewing process, this can lead to participants giving less extensive answers as the interview advances. To avoid misinterpretation, additional follow-up questions were always addressed in such cases. In addition, the interview is structured as a result of the conducted literature research and the questions are designed with respect to the research questions of the thesis.

Secondly, internal validity takes into consideration key aspects such as study design and the collection and analysis of data. As mentioned by Neumann et al. [22], establishing the chain of evidence is a challenge in qualitative research. Therefore, based on the conducted literature research, the questions of the semi-structured interview were created by taking into consideration the research questions and the hypothesis. The questions are organized in several topics. Guidelines suggested by Edwards and Holland [12] were followed when conducting the interviews. Each topic consists of a primary question and several follow-up questions meant to provide a more comprehensive answer for the topic. In addition, several refinement iterations shaped the question set according to the feedback provided by the advisors of this thesis. For data analysis, the method suggested by Galletta and CROSS [14] is used. The analysis is organized based on three stages: executing a post interview analysis after each interview, finding thematic patterns and conducting an interpretative analysis (see Section 4.2).

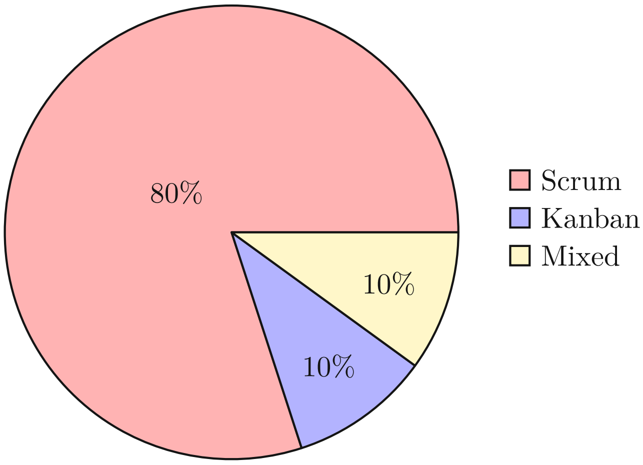


Figure 6.1: Used agile software development methods

Thirdly, external validity, to which degree the results can be generalized. 10 interviews with people from 8 different small to medium-sided companies from Austria were conducted. Consideration must be given to the fact that the interviewees are active in 5 different activity domains and fulfill different roles. In addition, the companies are not compared in other aspects than their size. Moreover, productivity stands for perceived productivity as no metric was used to measure it. For providing answers to the research questions, a literature research and a systematic analysis of the interviews were conducted (see Section 5).

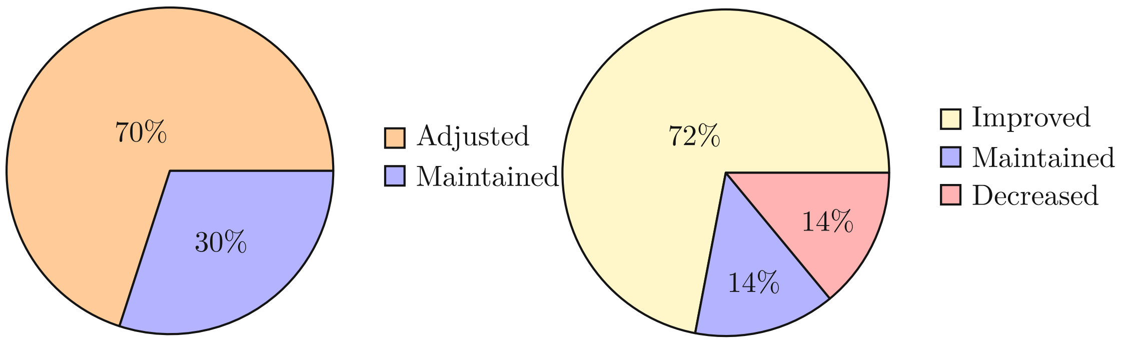
As part of the results, this chapter outlines the conclusions identified during the research carried out in the thesis with respect to the Expected Results. Answers to the research questions mentioned in section 1.3.1 are provided in conjunction with the conducted literature research and the analysis of the semi-structured interviews that were held (see Section 5). Last but not least, the hypothesis mentioned in section 1.3.2 is discussed.

6.1 Agile processes

This section is organized based on the central idea of the first research question: agile processes. The outcomes of this thesis regarding the adjustments of the agile software development processes that took place during the increased period of remote work triggered by the COVID-19 pandemic are revealed.

RQ1 *How has the workflow of agile distributed software engineering teams changed during the period of sudden increase in remote work triggered by the COVID-19 pandemic?*

To begin with, as can be seen in Figure 6.1, 80% of the participants in the semi-structured interviews used scrum as their agile development method. In contrast, 10% used Kanban



(a) Adjustment of agile process

(b) Agility changes

Figure 6.2: Agility changes of adjusted agile development process

and another 10% used a mixed approach between scrum and Kanban. In addition, only 10% of the interviewees switched the agile development method from scrum to Kanban during the remote work period. This decision was purely strategic and not directly related to the switch towards remote work.

In order to overcome the challenges encountered while working from home mentioned in section 3.2.2, 70% of the participants experienced adjustments in their development process (see Figure 6.2a). The adjustment is denoted by the introduction or exclusion of one of the agile practices inspired form Neumann et al. [22] in Table 5.4. Out of these mentioned interviewees, only 1 person representing 14% out of the participants who encountered an adjustment, experienced a decrease in agility according to the agility index (see Figure 6.2b). In opposition, 72% of the previously mentioned 70% improved the agility of their processes.

	Agile practices														
	Code Review 1	Continuous development 2	Continuous integration 3	Coding guidelines 4	Daily stand-up 5	Definition of done 6	Estimation 7	Kanban board 8	Pair programming 9	Planning meeting 10	Product backlog 11	Retrospective meeting 12	Review meeting 13	Sprint backlog 14	Test driven development 15
Introduced by	10%	10%	-	10%	30%	30%	10%	10%	-	-	-	20%	-	10%	20%
Excluded by	-	10%	-	-	10%	-	10%	-	30%	10%	-	-	10%	-	-

Table 6.1: Adjustment of agile development process

Moreover, Table 6.1 presents by which percentage of the interviewees each agile practice is adopted or excluded as part of the adjusted software development process. Table 5.4 reveals how the development process changed for each interviewee. It can be observed that *continuous integration* and *product backlog* are the only agile practices that continued to be used by all the interviews before and also after the increased period of remote work (see Table 5.4). Being the most adopted agile practices, the *daily stand-up* is introduced in order to overcome the communication and collaboration challenges, while the adjustment of the *definition of done* provides to transparency. In addition, 20% introduced *test driven development* to obtain more self organization and raise the delivered quality. In contrast, *pair programming* which is also mentioned as a challenge is the agile practice that stopped being used by the most interviewees. Different working hours as well as lack of communication and collaboration are the main reasons. The time dedicated to pair programming decreased according to 60% of the interviewees (see Figure 5.6).

The majority of the interviewees changed their development processes by making them more agile during period of remote work. The most used agile method continued to be scrum also while working from home and only 1 participant switched to Kanban. In a proportion of 70%, the development cycle stayed the same (see Figure 5.7).

6.2 Work-routine

By taking into consideration the main point of the second research question: the work routine of agile team members, in this section details on the changes that were observed during the period of remote work are provided.

RQ2 *How has the work routine of agile distributed software engineering team members changed during the period of sudden increase in remote work triggered by the COVID-19 pandemic?*

The work routine of agile development team members has changed during the increased period of remote work. Table 6.2 provides an overview on the benefits from the work routine perspective with respect to the benefits discussed in section 3.2.1 and the analysis conducted in section 5.4. Similar, Table 6.3 reveals the perception of the interviewees on the challenges presented in section 3.2.2 by taking into consideration the interview analysis in sections 5.4 and 5.7.

	Commute time	Flexibility	Involvement of stakeholders
Benefit	70%	100%	30%

Table 6.2: Influence of benefits on work routine

To start with, 70% of the interviewees confirmed the benefit of lack of commute time as previously identified by Mancl and Fraser [19] and Ford et al. [13]. Particularly, 30%

	Communication	Distractions	Work overtime	Different working hours	On-boarding
Challenge	60%	60%	60%	60%	90%

Table 6.3: Influence of challenges on work routine

consider commute time a part of starting their day-by-day work routine and do not consider it a benefit.

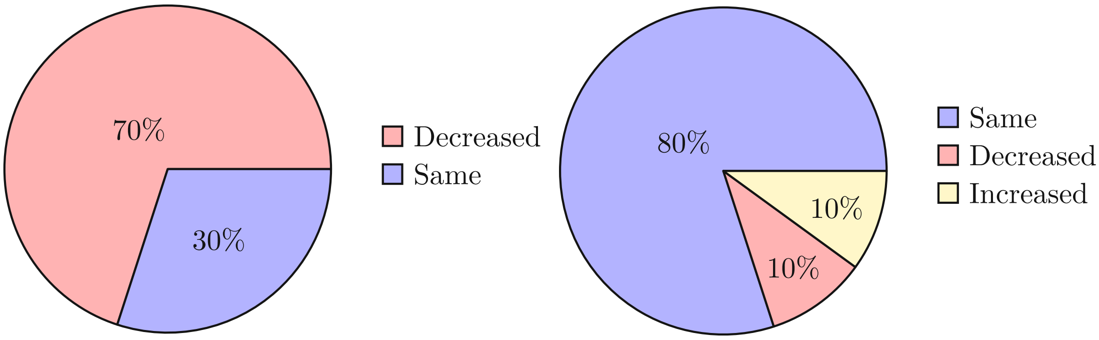
In contrast, flexibility is very appreciated by all interviewees. Even though flexibility is achieved by allowing people to work based on different working schedules, 60% of the interviewees encountered communication and collaboration challenges when working based on a different work schedule than their colleagues. Similar to Paasivaara et al. [24] and Shrivastava and Date [34], this thesis confirms that remote work is prone to communication and collaboration issues.

In addition, as identified by da Mota Silveira Neto et al. [9], distractions play an important role while working from home. According to 60% of the participants, more distractions are present while working from home. As a consequence, interviewees have a hard time when trying to concentrate for longer periods while conducting remote work.

During the remote work period triggered by the pandemic, 60% of the interviewees stated they worked overtime. As identified by Ford et al. [13] and confirmed by the interviewees, this happens due to the fact that private time blends into working time. Moreover, lack of progress during regular work hours is also a reason. Consequently, as suggested by Mancl and Fraser [19], employees can have a hard time when trying to oppose potential unfair management.

Due to the fact that the number of meetings is higher during the period of increased remote work, extra hours are required for fulfilling the usual responsibilities. Even though 70% of the participants stated that they are involved in more meetings, additional time is required for transmitting the intended message while doing remote work (see Table 6.3 and Figure 5.4) Further, as revealed by Figure 5.3, communication with colleagues has decreased for the majority of the interviewees. It was also observed that communication with other teams was more impacted than communication within the own team.

Last but not least, 90% of the interviewees consider that the time required to integrate new hires has increased during the increased period of remote work triggered by the pandemic. In addition, as can be seen in Figure 5.17, 50% decreased their mentoring activities while 20% are putting in the same effort in trying to provide support for onboarding new colleagues.



(a) Short term impact on productivity (b) Long term impact on productivity

Figure 6.3: Impact on productivity

6.3 Productivity

In this section, answers about the impact of the increased period of remote work triggered by the pandemic on the productivity of agile distributed teams are revealed.

RQ3 *How is the productivity of agile distributed teams impacted during the period of sudden increase in remote work triggered by the COVID-19 pandemic?*

In order to get a more accurate perspective on the development of productivity, the impact is analysed in the short term as well as the long term. The short term impact is analysed on the initial period during which all participants got accustomed to remote work. The influence on productivity during the remaining period is considered as long term. Figure 6.3 depicts based on the data reveal in Table 5.6 how productivity was impacted during the increased period of remote work.

As revealed in Figure 6.3a, 70% of the participants appreciate themselves to be less productive then usual in the short term. In addition, none of the remaining 30% consider themselves more productive. On the other hand, on long term all but one participant managed to regain the productivity level as before the increased period of remote work. Furthermore, only one interviewee reported a higher productivity level than the one prior remote work (see Figure 6.3b).

For a better overview on how much each analysed factor influences productivity, the values revealed in Figure 5.13 are presented as percentages in Figure 6.4.

As can be seen in Figure 5.13, the most determinant factor that influences productivity is considered to be the infrastructure, followed by the main points for the first and second research questions: agile processes and team members' work routine. Moreover, all interviewees that enhanced their development process by integrating more agile practices and consequently encountered an increase in their agility index (see Figure 5.5), consider the adjusted development process more suited for remote work (see Figure 5.14).

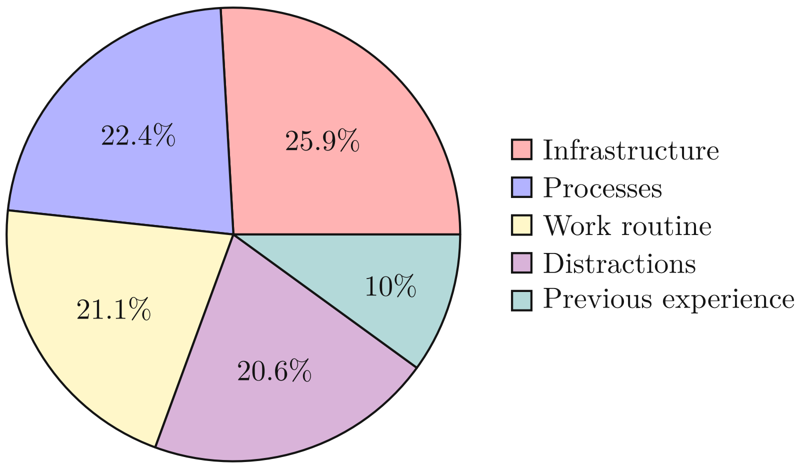


Figure 6.4: Distribution of productivity influences

As identified by da Mota Silveira Neto et al. [9] and confirmed by the participants in the interview, the delivered quality stayed the same during the increased period of remote work (see Table 5.8). Further, all participants that recovered their productivity level in long term did not increase their development cycles.

6.4 Hypothesis

In order to discuss the hypothesis, the previously mentioned research questions are taken into consideration.

Hypothesis *Adjusting the agile software development processes with respect to the distributed team members' needs and work routines improves overall efficiency resulting in increased productivity*

Initially, there was assumed that productivity will increase as a result of the adopted agile practices as part of the adjusted development process and flexible team members' work routine. This assumption was confirmed only by one interviewee (P1) and disproved by all other participants (P2-P10). This majority only managed to recover the productivity level in long term (see Table 5.6). Therefore, this thesis can only support the fact that adjusting the agile software development processes by introducing additional agile practices and with respect to the team members' work routine helps team members to maintain their productivity in long term.

Furthermore, the interviewees that improved the agility of their development processes by introducing additional agile practices (see Figure 5.5), all regained their productivity levels to the ones prior to remote work. As presented in Figure 5.13, the used agile development processes alongside the work-routine of the team members are determinant for productivity.

In conclusion, 90% of the participants recovered from the encountered productivity decreased (see Table 5.6) and 90% of the interviewees that improved the agility of their development processes consider the adjusted processes more suited for remote work (see Figure 5.14). However, due to the fact that half of the interviewees managed to recover their usual productivity level by working overtime (see Figure 5.12), potential sustainability and burnout issues must be taken into consideration.

CHAPTER 7

Conclusion

The agile methodology is one of the most implemented software development methods nowadays [7]. Thanks to its definition and to the principles that provide a more into detail understanding, the software development processes as well as the team members work-routine are tailored with the goal of achieving a higher level of productivity in order to satisfy the customer. Collaboration, communication as well as face-to-face interaction are highly valued in the agile methodology. Therefore the increased period of remote work triggered by the COVID-19 pandemic, faced agile practitioners with multiple challenges while working from home.

In this thesis, several papers that examine the COVID-19 impact on software engineering as well as papers concerned with agile distributed development are analysed as part of the literature research. Relevant concepts of agile development are discussed with respect to the agile principles. Benefits, as well as challenges raised by remote work, are identified and their impact on the research questions' main points: agile processes, work routine and productivity is discussed. In addition, the impact of the identified benefits and challenges on the agile principles is examined.

To provide a more comprehensive perspective and a better understanding of the effects of the pandemic on the agile processes and the work routine of agile team members, a qualitative interviewing approach is chosen. Due to their lack of structure and high level of flexibility, semi-structured interviews have an exploratory character. Therefore, the interviews are based on several open-ended questions that require follow-up questions divided into multiple topics. As part of the data collection process, 10 professionals that are developing products based on agile software engineering methods are interviewed. The goal is to identify the challenges raised by remote work and how these challenges affect the agile processes, the team members' work routine and their productivity. For the analysis of the semi-structured interviews, a three-stage data analysis process is used.

Based on the literature research and the conducted semi-structured interviews, answers to the research questions are provided. By using an own defined agility metric, this thesis confirms the fact that the majority of the interviewed people improved their development processes by integrating more agile practices. While agile practitioners' work routine is improved by lack of commute time and higher flexibility, challenges such as collaboration issues, working based on different working schedules and lack of separation between personal time and work time must also be considered. During the increased period of remote work triggered by the pandemic, the majority of the interviewees experienced a productivity decrease in the short term, but they managed to regain their productivity level in the long term. In addition, all participants that improved the agility of their workflow, regained their productivity levels to the ones prior to remote work. Last but not least, the feasibility of the expectation of increased productivity as a consequence of the adjusted agile processes with respect to the team members' needs is discussed.

The findings of this thesis reveal the fact that during the sudden increase in remote work triggered by the COVID-19 pandemic, the majority of the interviewees integrated additional agile practices as part of their development process. The adjusted development workflow became more agile and helped agile practitioners to cope with the challenges faced during remote work. Moreover, after a decrease in productivity, they regained their usual productivity levels. Future studies should aim to replicate results in a larger scale by making use of a quantitative interviewing method. It is a question of future studies to investigate the achieved productivity during remote work on a larger time frame.

List of Figures

List of Tables

Bibliography

[1] The 12 agile project management principles explained. URL https://kanbanize.com/agile/project-management/principles. Date accessed: 29/01/2022.

[2] What are the 12 agile principles? | definition and overview. URL https://www.productplan.com/glossary/agile-principles/. Date accessed: 29/01/2022.

[3] W. Adams. *Conducting Semi-Structured Interviews*. 08 2015. doi: 10.1002/9781119171386.ch19.

[4] A. Alliance. Xp 2020 conference, 2020. URL https://www.agilealliance.org/xp2020/#q=~(infinite~false~filters~(postType~(~'aa_event_session)~sources~(~'XP*202020))~searchTerm~'~sort~'createdDate~sortDirection~'desc~page~1).

[5] F. ANNA. How the world of agile has changed since the agile manifesto. URL https://bigpicture.one/world-since-agile-manifesto/?utm_source=google&utm_medium=cpc&utm_campaign=bp_search_austria&gclid=Cj0KCQjw8p2MBhCiARIsADDUFVH_9IsV6pXwsnhHDw5WyI9EGSV8wBjh4KOL36-SGmohUtBZSG369LEaAuBrEALw_wcB. Date accessed: 21/01/2022.

[6] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. Manifesto for agile software development, 2001. URL http://www.agilemanifesto.org/.

[7] S. A. Butt, S. Misra, M. W. Anjum, and S. A. Hassan. Agile project development issues during covid-19. In A. Przybyłek, J. Miler, A. Poth, and A. Riel, editors, *Lean and Agile Software Development*, pages 59–70, Cham, 2021. Springer International Publishing. ISBN 978-3-030-67084-9.

[8] R. da Camara, M. Marinho, S. Sampaio, and S. Cadete. How do agile software startups deal with uncertainties by covid-19 pandemic? *International Journal of Software Engineering Applications*, 11(4):15–34, Jul 2020. ISSN 0976-2221. doi:

10.5121/ijsea.2020.11402. URL http://dx.doi.org/10.5121/ijsea.2020.11402.

[9] P. A. da Mota Silveira Neto, U. A. Mannan, E. S. de Almeida, N. Nagappan, D. Lo, P. S. Kochhar, C. Gao, and I. Ahmed. A deep dive on the impact of covid-19 in software development, 2020.

[10] C. DRUMOND. Is the agile manifesto still a thing? | atlassian. URL https://www.atlassian.com/agile/manifesto. Date accessed: 21/01/2022.

[11] R. Dunford, Q. Su, and E. Tamang. The pareto principle. 2014.

[12] R. Edwards and J. Holland. *What is qualitative interviewing?* 10 2013.

[13] D. Ford, M.-A. Storey, T. Zimmermann, C. Bird, S. Jaffe, C. Maddila, J. L. Butler, B. Houck, and N. Nagappan. A tale of two cities: Software developers working from home during the covid-19 pandemic, 2020.

[14] A. Galletta and W. E. CROSS. *Mastering the Semi-Structured Interview and Beyond: From Research Design to Analysis and Publication*. NYU Press, 2013. ISBN 9780814732939. URL http://www.jstor.org/stable/j.ctt9qgh5x.

[15] L. Griffin. Implementing lean principles in scrum to adapt to remote work in a covid-19 impacted software team. In A. Przybyłek, J. Miler, A. Poth, and A. Riel, editors, *Lean and Agile Software Development*, pages 177–184, Cham, 2021. Springer International Publishing. ISBN 978-3-030-67084-9.

[16] E. Kate. The 4 values and 12 principles of the agile manifesto. URL https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto. Date accessed: 22/01/2022.

[17] P. Lous, P. Tell, C. B. Michelsen, Y. Dittrich, and A. Ebdrup. From scrum to agile: A journey to tackle the challenges of distributed development in an agile team. In *Proceedings of the 2018 International Conference on Software and System Process*, ICSSP '18, page 11–20, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450364591. doi: 10.1145/3202710.3203149. URL https://doi.org/10.1145/3202710.3203149.

[18] P. Lous, P. Tell, C. B. Michelsen, Y. Dittrich, M. Kuhrmann, and A. Ebdrup. Virtual by design: How a work environment can support agile distributed software development. In *2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE)*, pages 97–106, 2018.

[19] D. Mancl and S. D. Fraser. Covid-19's influence on the future of agile. In M. Paasivaara and P. Kruchten, editors, *Agile Processes in Software Engineering and Extreme Programming – Workshops*, pages 309–316, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58858-8.

[20] K. Marek, E. Wińska, and W. Dąbrowski. The state of agile software development teams during the covid-19 pandemic. In A. Przybyłek, J. Miler, A. Poth, and A. Riel, editors, *Lean and Agile Software Development*, pages 24–39, Cham, 2021. Springer International Publishing. ISBN 978-3-030-67084-9.

[21] S. Misra, V. Kumar, U. Kumar, K. Fantazy, and M. Akhter. Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality & Reliability Management*, 2012.

[22] M. Neumann, Y. Bogdanov, M. Lier, and L. Baumann. The sars-cov-2 pandemic and agile methodologies in software development: A multiple case study in germany. In A. Przybyłek, J. Miler, A. Poth, and A. Riel, editors, *Lean and Agile Software Development*, pages 40–58, Cham, 2021. Springer International Publishing. ISBN 978-3-030-67084-9.

[23] J. A. Ondiek. The 12 agile manifesto principles simply explained, 6 2015. URL `https://www.linkedin.com/pulse/12-agile-manifesto-principles-simply-explained-jacob-aliet-ondiek`. Date accessed: 29/01/2022.

[24] M. Paasivaara, S. Durasiewicz, and C. Lassenius. Using scrum in distributed agile development: A multiple case study. In *2009 Fourth IEEE International Conference on Global Software Engineering*, pages 195–204, 2009. doi: 10.1109/ICGSE.2009.27.

[25] R. Phalnikar, V. S. Deshpande, and S. D. Joshi. Applying agile principles for distributed software development. In *2009 International Conference on Advanced Computer Control*, pages 535–539, 2009.

[26] C. J. Poole. Distributed product development using extreme programming. In J. Eckstein and H. Baumeister, editors, *Extreme Programming and Agile Processes in Software Engineering*, pages 60–67, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-24853-8.

[27] P. Ralph, S. Baltes, G. Adisaputri, R. Torkar, V. Kovalenko, M. Kalinowski, N. Novielli, S. Yoo, X. Devroey, X. Tan, M. Zhou, B. Turhan, R. Hoda, H. Hata, G. Robles, A. M. Fard, and R. Alkadhi. Pandemic programming: How covid-19 affects software developers and how their organizations can help, 2020.

[28] B. Ramesh, L. Cao, K. Mohan, and P. Xu. Can distributed software development be agile? *Communications of the ACM*, 49(10):41–46, 2006.

[29] P. T. Robinson. Communication network in an agile distributed software development team. In *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*, pages 100–104, 2019.

[30] P. Rodeghero, T. Zimmermann, B. Houck, and D. Ford. Please turn your cameras on: Remote onboarding of software developers during a pandemic, 2020.

[31] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131–164, 2009.

[32] K. Schwaber and J. Sutherland. The scrum guide, 2020. URL https://scrumguides.org/scrum-guide.html.

[33] N. Séguin, G. Tremblay, and H. Bagane. Agile principles as software engineering principles: An analysis. In C. Wohlin, editor, *Agile Processes in Software Engineering and Extreme Programming*, pages 1–15, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-30350-0.

[34] S. V. Shrivastava and H. Date. Distributed agile software development: A review, 2010.

[35] A. Strauss and J. Corbin. Grounded theory methodology: An overview. 1994.

[36] K. Sureshchandra and J. Shrinivasavadhani. Adopting agile in distributed development. In *2008 IEEE International Conference on Global Software Engineering*, pages 217–221, 2008. doi: 10.1109/ICGSE.2008.25.

[37] J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov. Distributed scrum: Agile project management with outsourced development teams. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 274a–274a, 2007.

[38] L. Williams. What agile teams think of agile principles. *Commun. ACM*, 55(4): 71–76, apr 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133823. URL https://doi.org/10.1145/2133806.2133823.