TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Kirke Krämann 192445IABM

COMMUNICATION CHANGE IMPACT ON TEAM PERFORMANCE BY EXAMPLE OF AGILE DISTRIBUTED TEAM X

Master's Thesis

Supervisor: Karin Rava
MSc

Tallinn 2022

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kirke Krämann 192445IABM

KOMMUNIKATSIOONI MUUTMISE MÕJU MEESKONNA TULEMUSLIKKUSELE HAJUTATUD AGIILSE MEESKONNA X NÄITEL

Magistritöö

Juhendaja: Karin Rava
MSc

Tallinn 2022

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Kirke Krämann

10.05.2022

Abstract

The agile Team X who used to work co-located together before the pandemics were forced to work in distributed team setup overnight, as well as the rest of the industries. The information system development practices in team X remained the same, the communication showed a lack of natural conversation, discussions and team coordination. The team experienced problems in information system development outcome delivery and speed.

The research main question is to find out does change in communication in agile distributed team impact team performance and how.

The research gives an overview of information system development process characteristics in distributed teams, what are the differences in communication compared to co-located teams, what is a team performance and its' corresponding metrics, what is communication and its' quality and the ways to improve communication, and presents how changes in it have impacted team performance in an agile distributed team.

The theory is applied to a distributed agile information system development Team X, changes were made in its' communication, and the quality of communication and team performance was measured for 4 months before and after the changes in communication.

The result of the paper indicates communication changes in the team impacted the team performance. Changes made in Team X communication mostly indicate better team performance metrics results.

This thesis is written in English and is 55 pages long, including 6 chapters, 13 figures and 10 tables.

Annotatsioon

Kommunikatsiooni muutmise mõju meeskonna tulemuslikkusele hajutatud agiilse meeskonna X näitel

Agiilne meeskond X, kus meeskonnaliikmed töötasid ühises kontoris, olid sunnitud pandeemia mõjul üleöö kohanduma hajutatud meeskonnatööga sarnaselt teistele majandusharudele. Infosüsteemi arendusmeetodika jäi samaks, kuid suhtluses vähenes loomulik suhtlus, arutelud ja meeskonna koordineeritus. Meeskonnas esinesid probleemid arendusväljundite edastamise ja nende loomise kiirusega.

Uurimise põhiküsimus kas ja kuidas mõjutavad muudatused kommunikatsioonis hajutatud agiilse meeskonna tulemuslikkust.

Uurimistöö annab teoreetilise ülevaate infosüsteemi arendusprotsessi omadustest hajutatud meeskonnas, millised on erinevused suhtluses hajutatud meeskonnas võrreldes ühises kontoris töötavaga, mis on meeskonna tulemuslikkus ja vastavad meetrikad, mis on kommunikatsioon ja selle kvaliteet ning kuidas parendada meeskonna kommunikatsiooni. Töös esitletakse kuidas muudatused mõjutavad meeskonna tulemuslikkust agiilses hajutatud meeskonnas.

Teooriat rakendatakse agiilses hajutatud meeskonnas X, kus suhtluses tehakse muudatusi, ning mõõdetakse meeskonna kommunikatsiooni kvaliteeti ja tulemuslikkust 4 kuu jooksul enne ja pärast vastavate muudatuste rakendamist.

Uurimise tulemusena selgus, et muudatused meeskonna kommunikatsioonis mõjutasid meeskonna tulemuslikkuse mõõdikuid. Rakendatud muudatused toovad enamasti endaga kaasa paremaid meeskonna tulemuslikkuse näitajaid.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 55 leheküljel, 6 peatükki, 13 joonist, 10 tabelit.

List of Abbreviations and Terms

Output	Action, item, and a result which contribute to the outcome
Outcome	Business intention or need, the intended result
UX Design	User Experience (design), a user-centric design to provide a meaningful experience to the user
UI	User Interface, end-user view
SDLC	System Development Lifecycle, steps in the development process
QA	Quality Assurance, software engineer specialised in testing
R&D	Research & Development, innovation department
CMMI	Capability Maturity Model Integration, process improvement model

Table of Contents

1 Introduction 11
1.1 Problem 11
1.2 Research Questions 13
1.3 Methodology 13
1.4 Structure 14
2 Theoretical Overview 15
2.1 Distributed Teams 15
2.1.1 Characterization of The Development Methods in the Context of Distributed Teams 15
2.1.2 Communication in Distributed Teams Compared to Co-located Teams 17
2.2 Team Performance Measurement 18
2.2.1 Level of Ownership 20
2.2.2 Responsibility Division 21
2.2.3 Time Metrics 21
2.2.4 Software Quality Metric 22
2.2.5 Testing Efficiency 23
2.2.6 Development Performance 23
2.2.7 Team Members Satisfaction 24
2.3 Communication and Measurement of Communication Quality 25
2.3.1 The Quality of Communication 25
2.3.2 Communication Channels 26
2.3.3 Conway's Law 26
2.3.4 Measurement of Communication Quality 27
2.4 Communication Quality Improvement Methods 29
3 Implementation Changes of Communication for Improved Team Performance 32
3.1 Introduction to Team X 32
3.2 Communication Changes to Improve Team Performance 33
4 Results of Changes 37

4.1 Communication Quality 37
4.2 Team Performance Measurement Results 41
4.2.1 Results of Level of Ownership 41
4.2.2 Results of Responsibility Division 42
4.2.3 Results of Time Metrics 44
4.2.4 Results of Software Quality Metrics 46
4.2.5 Results of Testing Efficiency 48
4.2.6 Results of Development Performance 50
4.2.7 Team Satisfaction Results 52
5 Analysis and Conclusions.. 61
6 Summary.. 65
References 66
Appendix 1 – Lõputöö lihtlitsents 72
Appendix 2 – Team Survey Questionnaire.. 73
Appendix 3 – Results of Team Performance Metrics Details 74

List of Figures

Figure 1. Lead time and product cycle time. Author’s illustration................................ 22
Figure 2. Graphs. Author's illustration. 29
Figure 3. Communication paths in the team before changes... 40
Figure 4. Communication paths in the team after changes. 40
Figure 5. A Number of commits.. 41
Figure 6. Domain expertise results.. 43
Figure 7. Testing review efficiency.. 48
Figure 8. Defect removal efficiency... 49
Figure 9. A Number of releases.. 51
Figure 10. The ratio of tickets created to solved. 52
Figure 11. A Number of repliers per score.. 55
Figure 12. Results of Team Health Metrics (1st time). 59
Figure 13. Results of Team Health Metrics (2nd time). 59

List of Tables

Table 1. Communication frequency. 38
Table 2. A Number of commits to other repository. 42
Table 3. Lead time. 44
Table 4. Product cycle time. 45
Table 5. Automation test pass percentage. 46
Table 6. Pipeline pass rate. 46
Table 7. A Number of unit tests. 47
Table 8. A Number of code smells. 48
Table 9. Sprint velocity. 50
Table 10. A Number of tickets in progress. 50

1 Introduction

Information system development is human-centric socio-technical activity [1]. It is more than just writing code - it is a process, a combination of requirements, communication, collaboration, technical dependencies, and then, implementation and deployments, where one cannot exist without another. The collaboration and cooperation between different parties, also team performance, are achieved with effective communication [2].

Information system development in distributed teams has become a common way of working. The distributed teams have team members in multiple locations, the everyday information exchange is held over online mediums, and it has affected the communication and information system development processes. [3] Due to the global COVID-19 pandemic, most of the teams were forced to remote work in the distributed team setup, which highlights the need for the research [4].

1.1 Problem

The agile Team X who used to work co-located together before the pandemics were forced to work in distributed team setup overnight, as well as the rest of the industries. The software engineering practices remained the same, even though the team members needed to adapt to the distributed way of working. Team X realised the communication did not work the same way as sitting in the same room. The Team X members complained about a lack of natural conversations, idea bouncing discussions and structure of team coordination. The team experienced problems in information system development outcome delivery and speed, and the alignment of different team members' tasks.

Previous studies have found connections between working in the distributed teams and communication, less is known about how communication in distributed teams affects team performance [5], [6], [8].

A study by Kennedy, McComb, and Vozdolska “An Investigation of Project Complexity's Influence on Team Communication Using Monte Carlo Simulation" analyses team communication and performance dependent on goal achievement and efficiency [5].

An article by Hummel, Rosenkranz, and Holten, reviewed different communication factors in the information system development process based on literature reviews and brought out the previous studies’ findings, reporting the importance of communication in software development and practices used for distributed team process [6].

The book “Agile Metrics in Action” says The Agile Manifesto sets the metrics to the background and focuses on the process, collaboration and working software, the measurement is more based on reflections of the team [8], [9]. Therefore the team performance has not really been measured for the Team X, until the beginning of the research, it is hard to pinpoint where the team performance was lacking due to communication.

The gap in the state of art is communication changes impact on the distributed development team’s performance. The research is looking into the connection between team performance and its metrics measurement and how these are impacted by the changes in communication.

The purpose of the thesis is to identify if the changes in agile distributed team communication impact team performance and how. It is divided into subsections:

1. overview of the theory
    a. what is a team performance and its corresponding metrics,
    b. what is communication and its quality,
    c. identify ways of improvements for team communication and team performance,
2. apply the theory in experiments and measure the results of changes.

To verify the results, a case study is conducted and measured based on agile distributed Team X. The team performance is measured by a selection of metrics for 4 months before and after changes in the team communication.

1.2 Research Questions

The main research question is divided into sub-questions as follows:

RQ: Does changes in communication in agile distributed team communication impact team performance and how?

1. What are the characteristics of practices of the information system development process and communication in the distributed team compared to co-located team?
2. What is a team performance and what are the corresponding metrics?
3. What is communication and how to measure its quality?
4. What are the ways to improve team communication and team performance?
5. What are the results of the case study?
   5.1. What is the team quality of communication before and after changes?
   5.2. What is team performance before and after communication changes of communication?

1.3 Methodology

The methodology chosen for the research is a mixed methods research [10]. For the data collection, a mixture of approaches is used, a case study in a distributed agile team, a collection of team performance metrics supported by an interview survey and team experiments.

The methodology of the research is done in multiple steps:

- Theoretical overview for the case study
- Experiments to improve the team communication and collaboration
- Collection of feedback from the team members
- Measurements of team performance metrics for 4 months before and after changes in communication
- Analysis of the results

In the term of the research, a case study based on the team X will be conducted. The research will be applied to a distributed agile team. The qualitative data will be collected using 2 methods – an online survey of the team members and team performance measurement by involving team members in experiments. The online survey will be conducted to understand the team member's perception of the team process and well-being, as well as to see changes in the team mentality towards the information system development. The analysis of the team performance metrics, and the quantitative results, will be based on project management software Jira data, code analysis, team members' commits, Azure pipelines, app store reviews, and author's calculations.

1.4 Structure

The paper is written in English as the official communication language in the company is English and the survey questionnaire and workshops are conducted accordingly.

The structure of the paper is following:

The first chapter is an introduction to the problem and the research questions. The second chapter is a theoretical overview, divided into sections:

- The first section is the characterization of the information system development process in distributed teams and a comparison of communication in distributed teams to co-located teams.
- The second section is team performance measurement and its' metrics.
- The third section is an overview of communication and measurement of communication quality.
- The fourth section is introducing the improvements of team communication improvements.

The third chapter is an introduction to the agile distributed Team X used for the case study and a presentation of the implementation of the changes of team communication for improved team performance.

The fourth chapter is the presentation of the results of the changes in the team, what were the communication quality and team performance metrics before and after the changes in communication.

The fifth chapter is an analysis and conclusions of the research, including the answers to the research questions. The sixth chapter is the summary of the paper.

2 Theoretical Overview

The following paragraph gives a theoretical overview of distributed teams, characterization of information system development methods in distributed teams and communication in distributed teams compared to co-located teams.

2.1 Distributed Teams

A team is two or more people working together on different coordinated tasks to achieve common, specified goals, where each individual has a function [11].

Distributed information system development teams are geographically allocated yet working on the same software product or project. The primary factors of success are cooperation and collaboration, which depend heavily on communication [2].

2.1.1 Characterization of The Development Methods in the Context of Distributed Teams

The quality of the process affects the quality of the product. Selection of the most appropriate information system development model is a factor to define the process, the requirement may vary depending on the project or a company [20], [12].

To compare methods in information system development processes in distributed teams two of the most common practices were chosen – the traditional waterfall methodology and widely used agile development practices. The paper focuses on the agile methodology for the information system development process, where communication between team members is more highlighted [6], and it is used by the company and team in the case study.

The contrary approach to agile is a classical linear process, often called waterfall, which follows the steps of the system development lifecycle (SDLC), consisting of stages one after another, which cannot overlap [13]. Each of the phases need requirement documentation to be met, before entering the next stage of the development. The waterfall model is widely used for critical, well pre-defined projects where requirements are set early in the process [15].

Waterfall requires documentation before finishing each the stages of the development, in the distributed information system development there is an extensive need documentation [21], the ad-hoc informal communication in the co-located team cannot always be relied on, especially in the case of added time difference between the members. As the product evolves, the processes of the development need to be adjusted [22].

The constant improvement is the centre of agile software development methodologies, which were aimed at small projects for co-located team members who were to work together in the same premises on the same product [16]. The Agile Manifesto declares individuals, interactions and collaboration are prioritized over plans, documentation, and contacts. The measurement of the process is working software [9]. The agile way of working is based on continuous feedback loop – data collection, analysis, changes, and repetition of the steps [8]. The most common agile methodologies are Scrum, Kanban and Scrumban [19].

When the market and requirements are constantly changing, the team needs to adjust the processes to offer the best solutions: the agile methodologies are built for it, whereas the waterfall has fixed and more dependant processes [23]. It does also reflect on the human resources in the development process, who may not be available in the waterfall to step back into the development cycle, whereas in agile it is a team effort [19].

The communication in the agile methodologies is highly encouraged between all the team members. In contrast, the sequential waterfall relies mostly on the designated team leader to divide the information [21]. Knowledge management is continuous question in software development, especially in the distributed teams [23].

Both development approaches have two things in common – creating quality product, and people who are involved in the process. By the book, Scrum does not work for global software engineering, but modifying agile process for the team gives better results [2].

Quite often, the waterfall methodologies are used by critical projects, developed for corporations or governments, in contrast the agile methodologies are preferred by private sector companies and start-ups, where requirements change rapidly over brief period. [25]

2.1.2 Communication in Distributed Teams Compared to Co-located Teams

The challenges for the distributed teams have three dimensions - communication, coordination, control -, and three distances – time, geography, socio-culture [27]. Moving from co-located teams to distributed teams introduces challenges in team control and coordination, loss of communication richness, and management of the team spirit [28].

The article by Sablis, Smite and Moe “Exploring Cross-Site Networking in Large-Scale Distributed Projects” associates successful software development with the knowledge within the team and information gathered from the social network from the other teams. Good collaboration with other teams may give insights to the task in hand and raise the efficiency of the development [7]. On the other hand, Alex “Sandy” Pentland has said that one of the predictors of team performance is the communication within and outside the team [29]. And the success of information system development depends on communication participants and the information system development processes [33].

A study by Gutwin and Greenberg “The Importance of Awareness for Team Cognition in Distributed Collaboration” has confirmed, collaboration in the project is working on the same artefacts is possible only with verbal communication [30]. The team members working physically closely together have better opportunities for communication and collaboration [31]. Otherwise, everyone would focus on their actions, priorities, and their view of the workspace [30].

The distributed teams may have lower performance caused by information asymmetry and ambiguous authority. The first is caused by initiated changes where team needs better coordination and communication. The second, the ambiguous authority, is lack of control over the processes on different sites [32]. Distributed development teams have reported to have difficulties understanding the business domains and requirements, therefore making inadequate decisions and incompatible developments. [21]

The duration and frequency of the formal communication is similar between co-located and distributed team, the informal communication is more frequent in co-located teams, but the duration is comparable [22].

In case of working in the multinational teams brings in cultural differences, understanding these increases the teamwork and may benefit the development process. differences enriches as everyone has different approach to problem solving, creativity and communication [1]. Tackling the cultural differences in distributed team setup can bring challenges, the lack of direct communication increases the gap of understanding the mimics, body language, intonation etc which is easier being co-located [1].

In the light of understanding everyone's cultural heritage, it should not be forgotten what is most important in the team – the skills. How each individual tackles requirement gathering, design architecture, software implementation, testing, and collaboration with the other counterparts [1]. The distributed teams experience more disagreements on decision making as well as understanding the product or business domain [21].

One additional aspect in distributed team is a difference in time zone, in case the team is located in different countries. It is recommended to keep the time zones close to synchronize work hours and keep the working time close to normal work hours [25]. Managing hand-overs or synchronizing calls should be set in the beginning or end of the day to avoid disturbing usual life, for example, early calls or disturbing dinners [21].

When working in distributed teams, the collaboration needs to be highlighted to avoid gaps in information system development and unify the process for increased team performance, therefor effective communication in the team is important [21].

2.2 Team Performance Measurement

Team performance is defined as a combination of process and output measurement results, dependent on input (type of tasks, training), throughput, and output or the product. Team performance measurement is rarely a binary set of data, it is about learning team performance, communication and adjusting to achieve better results. [11], [24]

Traditionally, in a context of a project, the team performance has been measured with quality, on-time completion, on-budget completion. Research by Alzoubi and Gill added functionality as an extra metric. [35]

The complexity in measuring team performance lies in combining all the different metrics [8]. The distinction of process and outcome is rarely differentiated, and measured what is relevant in the given context [11]. However, the previous studies have found correlations between the processes and the quality of the software products [12].

A previous study by Hinds and McGrath “Structures that Work: Social Structure, Work Structure and Coordination Ease in Geographically Distributed Teams” confirmed, the relationship between team members in different locations affect the structure and performance of the teams in research and development (R&D). It also noted the difficulties in team performance may be caused from coordination of the tasks between the distributed sites. [36]

In a wider organisational view, one way to look at the improvement of performance in product and service development is with Capability Maturity Model Integration (CMMI) model. It is used to analyse the changes in the lifecycle of the product in the organisation, by creating levels of maturity. [37], [38]

The gap in the previous research is the combination of different, wide range team performance assessment metrics and the impact communication has in distributed team performance. However, it is confirmed by previous researchers, communication and collaboration have significant impact to the team performance [11]. Whether is it caused by the coordination of tasks, uneven distribution of information [36], or in contrast, the questionability of the conclusions of the previous measurements, such as time, unit of analysis or reliability of the results [11].

The basis of the team performance measurement is a selection of metrics, connected to the characteristics of software quality, team processes and output measures [37], [11]. Applying changes to the team can enhance the processes, improve the work, and help to remove unnecessary activities [39].

The measurement of team performance should take into consideration the purpose, the attributes and behaviours measured, verification of the results, the time and expense of the measurement [11]. When looking at the team performance metrics, in agile teams, it can be looked at trends, not only the numerical value of the metric. If the given metric trend is increasing or decreasing over period in the hoped way [24].

The list of metrics chosen for team performance measurement, based on research about team performance and communication in relevance to the Team X [40], [41], [42] :

- Level of ownership
    - Number of commits

    - Number of commits in other responsibilities

- Responsibility division
    - Clear responsibilities in domains
    - Incident severity index
- Time metrics
    - Lead time
    - Product cycle time
- Software quality metrics
    - Automation test pass percentage
    - Pipeline pass rate
    - Number of code smells
- Testing efficiency
    - Testing review efficiency
    - Defect removal efficiency
- Development performance
    - Sprint velocity
    - Number of releases
    - Number of unit tests
    - Number of code smells
    - Number of work-in progress items
    - Ratio of tickets created to tickets solved
- Team members satisfaction
    - Team members feedback survey
    - Team health monitor

The descriptions of the metrics are following in the upcoming sections.

2.2.1 Level of Ownership

The depth of expert ownership is the level of control depending on the number of edits done in code repository. The expert in ownership is given to the person whose reporting engineers perform the most edits. [41]

The chosen metric for measurement of ownership [42]:

- Number of commits in repository

- Number of commits in other repositories by developer

Higher ownership indicates focused activities, communication, and responsibility, as well as higher control of the properties. Without a clear ownership of the product, decisions taken may create new bugs to the system and affect the quality of the information system. [41]

2.2.2 Responsibility Division

The clear split of responsibilities affects the development team performance.

- Clear responsibility domains in the team

If the domains of the product are defined, the well-connected team members can get valuable information from the experts [7]. Team members with clear responsibilities achieve better results in the team' shared purpose [43]. The individuals' actions are based on the people around us, and to succeed as a team, it is beneficial to determine everyone's' focuses and align with the 'why', the goal, of the team [44].

- Incident severity index [45]

$$Incident\ Severity\ Index\ (DSI) = \frac{\sum(Incident * Severity\ Level)}{Total\ number\ of\ incidents}$$

Lack of purpose and responsibility creates incidents. Defect severity Index gives input about the quality of the product. The severity levels can be divided to three: Critical=3, Major=2, Minor=1. [45]

2.2.3 Time Metrics

Time behaviour is capability to provide processing and throughput under stated conditions [42], in the context of information system development , the following metrics were chosen to present team performance in the matter of feature delivery time: lead time and product cycle time as presented in the following figure.

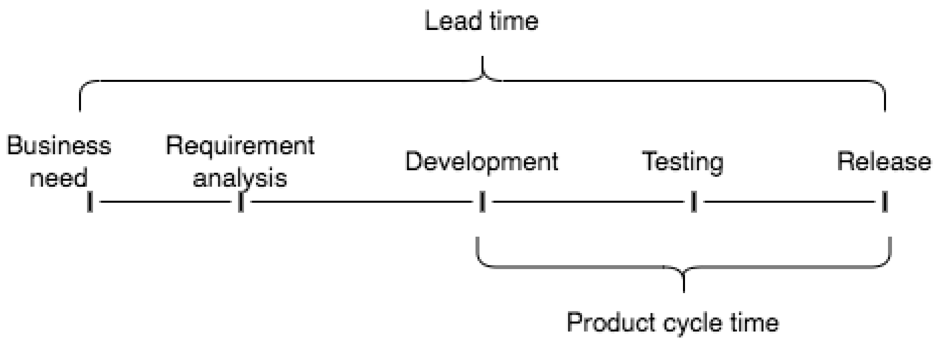


Figure 1. Lead time and product cycle time. Author's illustration.

- Lead time [46]

The lead time, or time to production, it is a period from business need to analysis and writing the requirement, until it gets to checked-in code and available to end user, with all the wait time, for example until included into the sprint [46], [47]. It is mostly used by business perspective to realise the length of the development cycle [48], the lead time includes cycle time [47].

- Product cycle time [46]

The cycle time of a ticket is in simple terminology the development time, from the business written requirement to coding and releasing to the end user, considering the wait time between each step – development, testing and release [46], [47]. The shorter the cycle time, the faster is the return of investment for the product [49].

2.2.4 Software Quality Metric

Percentage of quality towards a quality benchmark, aimed to collect information to improve team processes and product [45]. There are few possibilities for quality measurements [50], [51]:

- Automation test coverage rate [45]

Automation test coverage is metric for software product's functionality coverage by automatic test run in the system. [40]

- Pipeline pass rate [46]

Software builds, tests and deployments are automated with pipelines, supporting continuous integration and continuous delivery [46]. The Azure DevOps pipeline pass rate shows a percentage of successful pipeline runs to total pipeline runs, in other words, the rate includes the failure of the deployments [52].

- Number of code smells [53]

Code smell is an indicator for underlying problem, introduced to the system. It could be an inconsistency in the naming, a method being too long, or doing way too many functions. Code smells indicate technical debt and need for refactoring [53]. In terms of team performance, it is beneficial to prevent problems when these are small [54].

2.2.5 Testing Efficiency

Clear process metrics associated with fault prevention and removal. Estimation changes since last examination. [45]

- Testing review efficiency [45]

$$Testing\ review\ efficiency = \left(\frac{\text{defects found in review}}{\text{total number of testing defects}} * 100\%\right)$$

Testing review efficiency gives the defects found during the testing. Used to Reduce number of defects before release. It is used for effort utilization, increase of product quality and reduce number of bugs. [51]

- Defect removal efficiency [51]

$$Defect\ removal\ efficiency = \left(\frac{\text{defects fixed}}{\text{total number of defects}} * 100\%\right)$$

Defect removal efficiency provides measurement for bugs fixed towards total number of defects in the system. [50], [51]

2.2.6 Development Performance

The team development performance is a combination of technical and business goal setting, as well as efficiency, the ability to achieve the results [5].

- Sprint velocity [24]

Throughput is units of work per unit of time, alias how many items in sprint/release. It can be measured as sprint velocity, where it is average number of tickets or story points completed in each sprint. [24]

- Number of releases [49]

In the agile development, every sprint is aimed to create an increment, ready to be delivered to the end user [17]. Nevertheless, the industry is working towards multiple releases during the sprint, to release the tasks immediately after finishing. The number of releases indicates how often new improvements were deployed to end-customer.

- Number of Work-in-Progress items [57]

Number of work-in-progress (WIP) tickets is connected to Little's Law, indicating fewer number of tickets in sprint increases the throughput of the tickets. The team members are encouraged to collaborate with fewer tickets for better results. [57]

- Ratio of tickets created to tickets solved [57]

Ratio of tickets created over tickets solved shows the backlog status. Growing trend of tickets created to solved, indicates the business is bringing in tickets more than the team solves, creates confusion and reduces the accomplishment feeling of the team. [58]

2.2.7 Team Members Satisfaction

One of the indicators of the team performance is the satisfaction of the team. The indicator might be the user feedback, the business feedback, or the team members themselves. The development process satisfaction and the result itself may differ from one another. [59]

- Team members feedback survey [11]

Team members feedback is crucial to understand the team members mindset, the team performance is impacted by the satisfaction of its' members. Often the employees know where the problem is, and it is important to listen and make improvements for team's success. [11]

- Team health monitor [60]

One part of improvement of team performance is feedback [11]. As a part of measuring the team satisfaction of the team and collaboration, team health metrics were developed in a workshop by the team, and measured the change of satisfaction with the team metrics. The team health monitor would be one way to track the team collaboration in repeated workshop to see the trends, take actions, and aim to better satisfaction of the team. [60]

The member satisfaction metrics were chosen to get the qualitative input, but also something measurable, to see a trend over time.

2.3 Communication and Measurement of Communication Quality

The definition of communication states it is interaction with someone to exchange information by using words, sounds, signs, behaviour, or technology [62]. Communication involves efficient change of information between project participants and controlling the optimal flow of information for all of the receivers [63]. The goal is to create mutual understanding of the topic [64].

2.3.1 The Quality of Communication

The quality of communication, by definition, is the message is delivered and it satisfies the needs and expectations of the recipient. It needs to be aimed at the target and in a suitable format [64]. The quality of communication has been less researched in the field of information technology [59], the topic is also valid in the field of oil, gas, and construction industries [39]. An article by Lynda Bourne does confirm the quality of communication affects the quality of the product and cost effectiveness of the project management [65].

Another factor in communication quality is the richness of the communication channel – the effectiveness of delivering the information, it considers social presence, such as body language, and the need of feedback and level of details delivered during the exchange. The richest channel is face-to-face communication with a whiteboard, whereas the least effective method is simple documentation. [31]

2.3.2 Communication Channels

The communication in development team does not usually remain in the meetings, quite often questions arise in the middle of the process, therefor it is important to be open for ad-hoc communication which may affect software development and its process [66], [67].

The previous research lists the most common channels for distributed teams [31], [66]:

- Emails
- Meetings in person
- Instant messaging – Slack, Microsoft Teams, Glitter, Spectrum [21] , [66]
- Video calls – Slack, Microsoft Teams, Zoom [21] , [66]
- Project management tools – Jira, Trello
- Development collaboration tools - GIT, Microsoft TFS, CVS, deployment tools [68]
- Online whiteboards – Miro, Mural [31]
- Documentation management – Confluence, shared documents like shared Word documents or Google Docs [31]

When choosing communication channel, the richness of each communication method should be considered. It would consider the need for feedback, details to be delivered, and social presence, such as verbal cues and body language [31].

It is important to realise different roles in the development cycle have unique way of working, not all the same communication formats can be used for all the teams members. The message and delivery channel used should be taking into account the receiver role in the team for the most effective result of communication. [14]

2.3.3 Conway's Law

The communication when transforming business processes and requirements to information system features is reflected as Conway's Law [69]. Conway's Law declares organisations produce designs which are reflected by company's internal structures. Closer units working together would have more interaction and better collaboration in the development of software architecture. [70] Conway's Law declares organisations produce designs which are reflected by company's internal structures. Closer units working together would have more interaction and better collaboration in the development of software architecture. [70]

Conway's Law is connected to organization communication silos, where different teams are working in isolation, not sharing the information between each other, the risk is higher in the international distributed teams due to time zones and cultural differences, as well as in the previous way of working. [71]

To avoid applying Conway's Law to the information system, the design methodology should consider organisational and technical requirements as well as communication between the developers [69].

2.3.4 Measurement of Communication Quality

The communication quality metrics are chosen to grasp the definition of communication, as well as the quality of communication.

The communication metrics chosen for the research based on previous research in connection to team performance [1], [72]:

- The number of participants in communication
- Communication frequency
- Number of communication channels
- The number of communication paths
- Communication paths on a graph

The previous research indicates information exchange may affect team performance [5]. The communication frequency and quality play a role in the result on the information system. Too much communication may decrease output, especially in terms of development, yet in design team it may increase the result [5]. While delivering a message, multiple channels could be used – sharing the news verbally, writing on PowerPoint slides, reminding in a message, or email [21]. Nevertheless, it is almost impossible to overcommunicate the message needed to be delivered in the context of distributed team [5], [44].

The metrics chosen in research for the quality of the communication are [1], [72]:

- The number of participants in communication

The number of engineers who have touched the code affect the need to coordinate the communication [41]. The challenge of administrative and expert coordination increases with the size of the team [1], [32]. Communication is easier within smaller teams which builds tactic shared knowledge, the optimal size of a team is said to be 10 people [6], [20].

- Communication frequency

$Communication\ Frequency = \ Number\ of\ chats * Number\ of\ messages$ [73]

Communication frequency is number of messages in a chat over period, in a day or a week [22]. The duration and frequency affect the communication which impacts the information system development, as some of the requirements or details might get lost in the distributed teams [5], [74]. In order to ensure the information is delivered, one cannot assume the message is understood, and might need to repeat it [74]. The media channels which do not require active communication, such as emails, documentation, project management systems and shared codebase in development collaboration might reduce the need for frequent repetition [6].

- The number of communication channels

It is important to manage the communication channels in the team, depending on the number of team members and complexity of the project. Integration with meetings, communication tools, information system development tools and documentation are a way of communication [66]. The team should have rules on the communications channels. Too many messages in different channels might shift focus and create confusion [21].

- The number of communication paths for engineers in the project [41]

$$Communication\ Paths\ (CP) = \frac{N(N-1)}{2}$$

N – number of engineers

The number of engineers N in the project indicates x=(N*(N-1))/2 communication paths in the project. Communication path refers to potential communications between humans, it associates with the chances of defective code, chances of miscommunication and need for possible development work [41]. The communication paths are calculated if all the nodes (people) are connected to each other.

- Communication paths on a graph [72]

Communication paths can be described by using graphs [72]. A graph consists of nodes, connected with edges. In communication, a node is a person, and an edge is a connection between them. In a complete graph, all the nodes are related toes. [75]. Organisation communication is seen as bidirectional, but the information system message exchange can be one-sided [72].

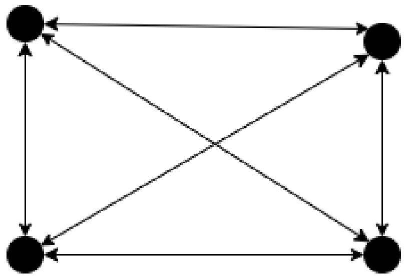


Figure 2. Graphs. Author's illustration.

The communication shown on a graph presentation gives an overview, how the communication between different people is structured [72]. Closely connected teams, such as illustrated on a complete graph, require less coordination, and improved performance in the team [36].

2.4 Communication Quality Improvement Methods

Effective communication improves the task, processes, and reduces the risks of conflicts in the mentioned areas, therefore positively affect the project success [76].

Suggested changes in the team to improve communication in distributed teams based on previous research to improve the team communication [1], [6]:

- Common meetings

Software systems that depend on one another should be developed by co-located teams to avoid system tweaks. If not possible, the teams should at least be in constant communication [77]. One of the methods to align different team whose work affect one another is to set up common meetings – all teams sprint planning meetings, “scum-of-scrum”, common demos, common retrospectives, it is aimed to all the teams give overview of their plans, impediments, or insights for the upcoming or past period, as well as ensure the flow of information between team members [16], [36].

- Identify team members skills and knowledge

In order to create understanding of the team communication, it is important to identify each individual skills, knowledge, behaviour, and interests. One of the possible ways to improve the communication is to explain expectations, and assess personal knowledge, motivations, interests, and goals. [1]

- Add coworking time

The importance of collaboration in information system development has proved itself by faster development time and improved quality of the product [68]. One-on-one collaboration creates connection, builds mutual understanding, trust, and provides motivation which increases the chances for team members to interact more to resolve difficulties in the project in the remote setup [5], [74]. One way for collaboration improvement in information system development projects is adding pair-programming sessions, mob-programming [78] or workshops.

- Improve documentation

Even though documentation is a formal way of communication with poor interaction level, it is a way of tracking of information system development [31]. The processes in the organisation, dependencies between tasks and management are in relation to one-another. The lack of documentation in distributed teams, may result in teams' understanding, or the inconsistencies in the development process [1], [36].

- Schedule retrospectives

The agile practices are built on continuous improvement, looking over the past period, and adjusting where needed, either in the product or in the process [8]. The agile principles are aimed at co-located teams [8]. Retrospectives are aimed at giving the team time to reflect on the last sprint, mostly focusing at 3 questions – what went well, what were the problems, and actions for solutions. [17] In the distributed teams, retrospectives could be the time to bring the team together to discuss processes in the team, whether it is over video calls and/or virtual whiteboards [17], [36].

- Introduce the role of Scrum Master

Scrum Master is responsible for the effectiveness of the team [17]. The role's main goal is to help everyone understand the scrum theory in practice, as well as improve the development progress and process [17]. There is a cost of communication in the team to get the work done, and scum expert, can be one lead to the architecture of communication to avoid contacting everybody, who is irrelevant to the topic and shorten the communication lead [44].

The list of the responsibilities of the Scrum Master according to the Scrum Guide 2020 [17]:

- Coaching team members in cross-functionality and management
- Helping the team to focus on the tasks
- Removal of impediments in the team’s progress
- Ensuring the scrum events take place and are productive within the time
- Facilitating collaboration with stakeholders
- Help with the product backlog
- Tacking the product and team progress

The role of the Scrum Master in distributed team enhances the team priorities, and manages the process, but also brings different roles in the team together [17], [36].

3 Implementation Changes of Communication for Improved Team Performance

To avoid implementation of Conway's Law and improve team communication and performance, changes which were made to are presented in the following sub-chapters, beginning with an overview of the Team X.

3.1 Introduction to Team X

The team in the case study is working in payment-after-delivery solution named XYZ by company A. It is aimed and omni-channel businesses to separate customer checkout process from payment, taking over the credit check and fraud risk from the merchant. An increasing number of customers make purchases in online stores on mobile devices. The goal of XYZ is to make payments quick and easy.

To finalize the payment process after delivery of the goods, customers can access their invoices through emails or website, which is managed by the team, or newly developed mobile application.

The core development team of 6 engineers in the case study has operated since 2017. The current setup with 10 engineers, 3 of them being front-end engineers, 3 product managers and 2 UX designers have been working all together from 2019. The team consists of different nationalities – Estonians, Swedes, Germans, Ukrainian and Russian.

Due to the Covid-19 pandemics in 2020, all the team members were separated, and team adapted to remote, all distributed team way of working [4].

In the beginning of 2020, a mobile application development initiative was started with the given team size of 10 engineers, where the team was all working remotely in distributed setup. As most of the members were familiar with one another, the transition of a new team went well.

The mobile application front-end is built by 3 front-end developers, based on the design team designs and requirements, the data management and behind the scenes data management is developed by 7 back-end software engineers. The current setup refers to potential implementation of Conway's Law, where neither of the sides have good overview what happens on the 'other side' of the application.

The team follows agile principles with some modifications. Due to the distributed setup implementation, the retrospectives and informal communication had decreased in the team.

3.2 Communication Changes to Improve Team Performance

Changes in team communication applied to the team X in order to improve performance are presented in the following sections.

Set of changes were applied to the team with the aim to improve the team performance. The changes applied were indicated from theory and the team need.

- Encouragement of reaching out directly

To reduce the impact of Conway's Law in the application's front-end and back-end components, the team members from back-end and front-end expertise were encouraged to reach out to one-another and plan pair-programming sessions. In case of open questions, reach out ad-hock and have a call, if needed, bring in other team members. All to improve communication and collaboration.

- Setting up common meetings

As part of communication improvements, all the team members were gathered for common meetings. It included daily stand-ups, common planning across functions, demo sessions and retrospectives.

During the morning stand-up call, all participants gave overview of their previous day tasks and upcoming day priorities, as well as open questions or impediments were highlighted. Common plantings were aimed to share with all the team members about upcoming period development priorities, the demo sessions to present to the wider audience the new functionalities and retrospectives of the process improvements needed to be done.

- Coworking time

Considering the need for extensive collaboration, the team were given tasks shared between developers, the division could have been done by two developers in the same expert areas or divided by front-end and back-end developers, in both ways the engineers needed to work together on the same ticket also called as pair-programming. It was either done at the same time over a video call or by sharing the same resource, nevertheless, the changes made individually were affecting the other developer. Coworking time was also encouraged with business representatives.

The different specialities were pushed to work together – the feature prioritisation was done among product managers along with analysis, the designers put the ideas in visual format and developers needed to bring them to life. The whole team were included in an analysis of the user needs, discussions on the design approaches and technical restriction on the development.

It also brought together the front-end and back-end team members to align with one-another which did increase the collaboration and understanding, as well as reduced the flaws of the information system

- Documentation improvements

Improvements for documentation included:

- Review of current documentation
- Creating definition of done for tasks required documentation,
- Scheduled technical knowledge sharing session where responsible presenter created documentation
- Specifications and documentation of upcoming features

The purpose of alignment of the team members and performance of the team, the documentation created another layer of communication channel, where to control the technical or product related questions.

- Retrospectives

For the sprint's reflections, the Scrum Master scheduled and prepared retrospective meetings in the end of every sprint. In the distributed team setup, the meeting took place in Teams videocall, accompanied with virtual Miro whiteboard or easyretro.io template. Each of the retrospectives were tried to keep unique, with varying the angle of the question.

The meeting usually started with a simple check-in question to engage everyone. The questions varied form ‘how was your last weeks’ to ‘what the title of a movie about your life would be’. Then the team members were asked to fill in each other with the biggest events, changes, or directions.

As first step, everyone filling in their highlights of the sprints. As a second part, the participants were highlighting their problems. Both sections were ending with discussion, in large group, in breakout rooms from 2 to 4 participants. In case of breaking the discussion, a summary of discussions was added. The third part of the call was creating actions, the Scrum Master driving the discussion to conduct concrete actions with responsible team member.

In the last calls of the measuring scope, to create trust in the team, the team members were asked to show their appreciation for the team members, whether it was the whole group or someone specific for a certain thing. The impact of the call created discussions outside of the meeting.

The recurring topics of retrospectives:

- Good collaboration and team spirit
- Missing alignment of focus areas for different people in team
- Over time the complaints about lack of documentation decreased, as the team took actions
- The need for certain meetings – product related specification, team fun time, pair programming
- Incident handling and prevention discussion in case of incidents

The repeating topics were minimised by creating action, also the trend of recurring topics importance were decreasing, when the actions were taken and kept consistent by the team to improve.

- Introduce the role of Scrum Master to the Team

To improve the information system development process, one of the developers was signed to a Scrum Master role to improve the team flow, create structure, and measure the team performance.

The first step was taking over the Scrum ceremony meetings, scheduling and facilitating the Teams calls, preparation of meetings and retrospectives. While being at it, the person tried to understand the flow, where do the team members get stuck, how to draw attention to the matters, who to contact to ease the process. The Scrum Master was also following along the action items created in retrospectives.

One of the aspects team members were complaining in the survey was a lack of documentation. The Scrum Master stressed with ticket creation, part of solving a task is adding necessary documentation for future references. It took about 2 months, until the team members proactively created documentation.

To measure the number of releases, the release processes needed to be set in order. The Scrum Master took the ownership of creating releases in Jira, assigning the tickets which were released and created documentation which features were released when. Soon after the process was set, it was re-introduced to the team members, and after 2 months the developers were adjusted to the steps and contributed to the documentation.

4 Results of Changes

The chapter gives overview of the changes results: the quality of communication, as well as the team performance before and after changes in communication.

4.1 Communication Quality

The communication quality metrics in the Team X, based on the theoretical overview:

- Number of Engineers

In the team size no changes were done- The team under observation continued to have 10 engineers, 3 product managers and 2 designers.

- Number of communication paths

The number of communication paths each of the team members has is 45, where number of engineers in the team being N=10.

$$Communication\ Paths\ (CP) = \frac{10(10-1)}{2} = 45$$

Given the team has 10 engineers and the number of communication paths is 45 in a complete graph, it would mean each of the developers should be aware there is 45 ways to communicate inside the team.

- Communication frequency

Table 1. Communication frequency.

	Communication Frequency			
	Before		**After**	
Team member	**Number of chats * number of messages in chat**	**Communication frequency**	**Number of chats * number of messages in chat**	**Communication frequency**
K	5*6	30	8*7	56
OS	6*6	36	12*6	72
J	8*4	32	14*4	64
AK	4*6	24	7*6	42
SZ	4*5	20	8*5	40
FO	4*6	24	6*8	48
FJ	4*6	24	5*6	30
SN	3*6	18	11*4	44
QA	3*8	24	7*8	56
AE	2*6	12	4*6	24
Average		24,4		47,6

Communication frequency was considering all the chats a team member had during a day given an example day of the period. The final communication frequency number also included conversations outside of the team. The communication frequency has almost doubled compared to the result before communication changes, going from 24.4 messages to 47.6 messages.

The communication frequency includes also other topics and chats the team members encounter during the day. Not always the communication stays in the given team, the product development might include collaboration with other teams or stakeholders outside the team.

- Number of communication channels

In total, the team is using 8 different tools for communication.

The list of communication channels the team is using:

1) Microsoft Teams instant messaging and video calls
2) Emails
3) Document sharing software Confluence
4) Project tracking software Jira
5) Online whiteboard software Miro
6) Development collaboration tools
    a. Git
    b. Microsoft Azure for cloud computing
7) Interface design tool Figma

Each of the communication channels have a purpose. The team has agreed on communication principles. The prioritised channels were Microsoft Teams for more personal communication and holding calls in distributed teams, the emails are reserved for formalities. The rest of the communication channels have no written agreements, the team is using common sense and following previous documents examples. The product development related communication channels are divided into three: current task related updates in Jira, supported by document sharing software Confluence and for collaboration in the team calls Miro, as whiteboard. The development collaboration tools are related to development.

- Communication paths on a graph

Each of the node represents a team member, and edges the communication between them. The graphs are based on the communication patterns the team members admitted having regularly.

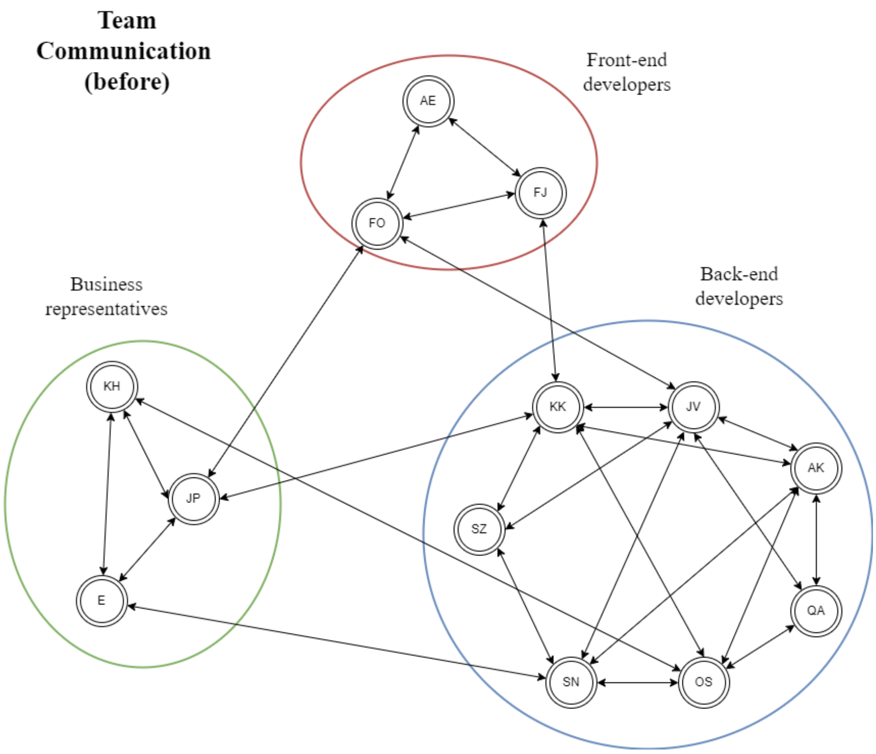


Figure 3. Communication paths in the team before changes.

The communication in the product before the communication, concentrated mostly in the expertise areas. The lack of alignment between different roles could reflect on the Conway's Law, and creates technical misalignments.

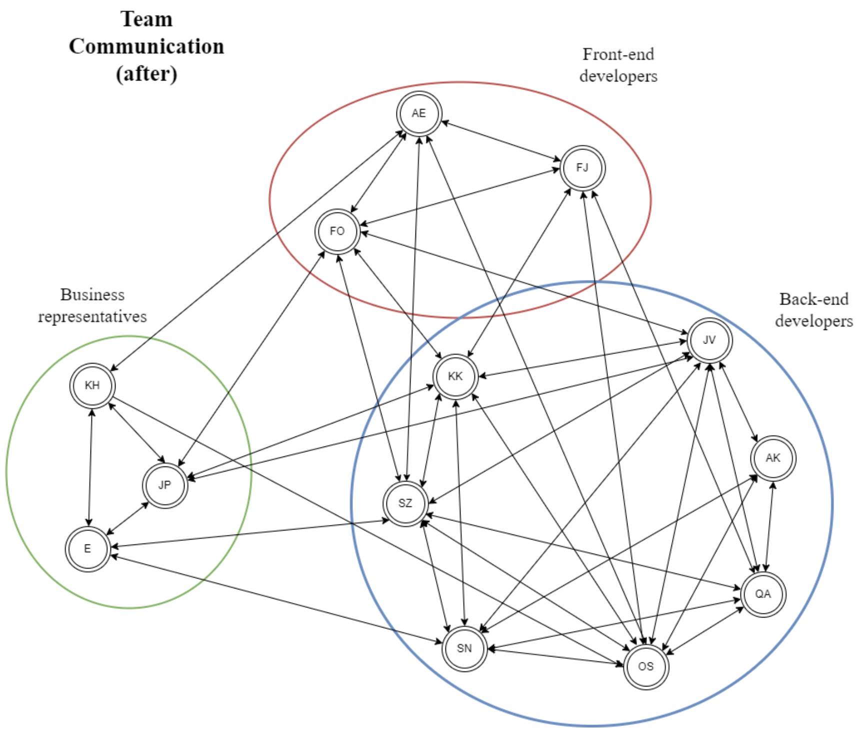


Figure 4. Communication paths in the team after changes.

The communication improvements can be reflected on the graph. Inside each role, the communication has been improved, but also the communication between different roles has been increased. The change can be seen the same way when looking at the communication frequency graph.

4.2 Team Performance Measurement Results

The changes in the team communication were applied within 4 months period, the team performance was measured for 4 months before making any changes, there was a month period when modifications were done, and the team performance after results were collected for another 4 months.

4.2.1 Results of Level of Ownership

The results of the team's level of ownership measured by the number of commits and number of commits in other repository, evaluated 4 months before and after changes in team's communication.

- Number of commits in repository

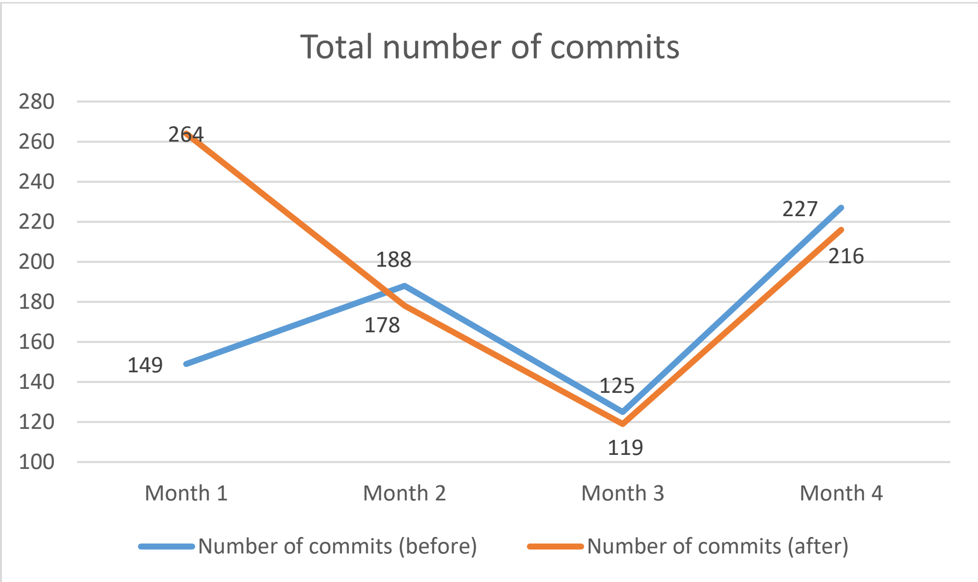


Figure 5. A Number of commits.

The number of commits in the repository does not correlate with the communication changes, the impact is not correlated to the data. Each developer contribution is in appendix 3. The measurement month 3 did fall in the vacation period, which explains the sudden drop in the number of commits. Both results do indicate a slow trend of growth in the number of commits, with some throwbacks. If to look at the table and the total number of commits, the trend of growth per team is more visible.

As a result of the communication changes, the effect is not connected to the number of commits in the team.

- Number of commits in other repository

Table 2. A Number of commits to other repository.

Number of commits to other repository				
	Month 1	**Month 2**	**Month 3**	**Month 4**
Before		1		
After	2	1	6	1

In order to bring the team members closer, and increase collaboration, several on commits in other repositories were counted. Compared to the first counting period, only one commit was done, whereas after the improvements in total 10 commits were done. The increase of commits to another repository is the result of communication and pair-programming where the whole feature was developed by front-end and back-end developer.

4.2.2 Results of Responsibility Division

The responsibility division in the terms of sharing domains and responsibilities for areas, as well incident severity index results:

- Mapping the expertise in the team

Tracking responsibilities and knowledge of the team members, an experiment workshop was created. The aim was to identify the specific areas of the information system and address a responsible team member or members to the area. The approach was targeted to create and share the knowledge about the domain and enhance collaboration by mapping the knowledge areas and being open about who knows the best about which area. [23]

Some of the domains were covered by multiple professionals, or assigned a new person to get more familiar with the certain field. The experiment was held in online Teams video call and actions were taken using Miro collaborative whiteboard.

The first part of the experiment was to distinguish different key areas in the information system. All the participants in the experiment were aware of the product and were given a chance to add their input about the different domains.

The second section of the experiment was to map the team understanding of the key domains and the experts of the domains, both could have been multiplied.

The third sector was time for self-reflection. The key areas were added to a matrix with the names of the team members and were asked to evaluate their knowledge of each domain. The possible answers for their understanding of the areas were good (+), fair (+−), poor (−), will learn (↑).


What is my understanding of the area?

Legend
+ Good
+/- Fair
-- Poor
↑ will learn

Domain	J	AK	OS	KK	SZ	AE	FJ	QA
Queues	+	+	+ −	−	−	−	−	+ −
Payments	+	+	+	+ −	+	+ −	−	+
app releases	↑	↑	↑	↑	+	+	+	−
Invoice	+	+	+	+ −	+	+	+	+ −
Monitoring	+	+	+	+	+ −	−	−	+
The front-end	−	−	−	+ −	+	+	+	+ −
MyAfterPay	+	+	+	+	+	+ −	−	+
Push notifications	+	+	+ −	+ −	+ −	−	−	+ −
Returns	+	+	+	+ −	+ −	+	+ −	+
Insights about users	−	−	−	+ −	−	+ −	−	−
Flex Accounts	+	+	+ −	−	+	−	−	+

Figure 6. Domain expertise results.

As a result of the experiment workshop, the team got a clearer understanding of the key domains of the information system, and better understanding of the expertise. On one hand, the participants reflected on their knowledge, and on another hand, it was clarified who have better overview of the topic. All the aspects are in favour of communication improvement, to define the key areas and assign the experts of the domain. The result of the experiment would benefit the communication to ask the area specific question to knowledge team member and reduce the loop of forwarding people to the expert. It was also an opportunity to learn about the product and create better team dynamics.

- Incident severity index

$$Incident\ Severity\ Index\ (DSI) = \frac{\sum(Severity\ Level * Incidents)}{Total\ number\ of\ incidents}$$

$$= \frac{(3*1) + (3*1) + (2*1) + (1*1)}{4} = 2.25$$

The severity Index of the team over a year, which considered system malfunctions that were listed as incidents. The overall score for the Incident Severity Index was 2.25 over the measurement period (9 months).

After the communication changes, the team occurred with one incident, with the Incident Severity being on the lowest level. It is an indicator team was able to decrease the incident level.

4.2.3 Results of Time Metrics

The team’s results of time metrics before and after changes in communication:

- Lead time

Table 3. Lead time.

Measurement week	**Lead Time (before)**	**Lead Time (after)**
Month 1	5w 2d 7h	2w 1d 6h
Month 2	5w 6d 23h	3w 4h
Month 3	5w 5d 2h	2w 3d 16h
Month 4	4w 4d 14h	3w 4d 23h

The lead time of the team has decreased from an average of 5 weeks to over 2 weeks. The reasoning has been collaboration with the business representatives for requirement specifications, as well as updating the ticket statuses on project management tool Jira, where the information was collected.

The communication changes had positive outcome, as the lead time decreased.

- Cycle Time

Table 4. Product cycle time.

Measurement week	**Cycle Time (before)**	**Cycle Time (after)**
Month 1	3w 1d 22h	2w 3d 15h
Month 2	3w 2d 12h	1w 2d 12h
Month 3	3w 4d 5h	1w 4d 21h
Month 4	3w 4d 12h	2w 4d 1h

The cycle time of the team started from 3 weeks, 1 day and 22 hours, and increased to 3 weeks 4 days 12 hours in the first measurement period. After the changes applied to the team, the team cycle time decreased to either 2 weeks 4 days or even 1 week 3-4 days.

The change was created by taking less tickets, setting pair-programming sessions, and increasing collaboration between the team members.

The communication changes had positive outcome, as the cycle time is shorter.

4.2.4 Results of Software Quality Metrics

The results of team's measured software quality metrics were following:

- Automation Test Pass Percentage

Table 5. Automation test pass percentage.

		Automation Test Pass Percentage			
		Total	**Passed**	**Failed**	**Pass percentage (%)**
Before	**Month 1**	483	111	372	22,98
	Month 2	490	118	372	24,08
	Month 3	490	118	372	24,08
	Month 4	498	118	380	23,69
After	**Month 1**	511	126	385	24,66
	Month 2	520	135	385	25,96
	Month 3	529	144	385	27,22
	Month 4	529	227	302	42,91

The automation test pass percentage shows the ratio of test passed compared to all the automation tests. The result were stable throughout the measurement period, just below or around 25%, expect the last month where it peaked to 42.91%. Reason being, the quality assurance specialist, changed the test according to the product and it had an impact for 80 automation tests which now passed, and increased the pass rate.

The changes done in the team, had minor impact on the automation test pass rate, the trend being slightly increasing every month.

- Pipeline pass rate

Table 6. Pipeline pass rate.

	Pipeline pass rate				
	Month 1	**Month 2**	**Month 3**	**Month 4**	**Average Rate**
Before	79%	95%	100%	95%	92%
After	100%	82%	95%	100%	94%

The pipeline pass rate shows how is the quality of releases. Before the changes in the team, the trend was increasing, from 79% to 95%. On month 3 (before changes), it was 100%. After the changes, on two months the pipeline pass rate was 100%. The average rate for before shows an increase from 92% to 94% of the pipeline runs to succeed.

The increase could be mostly connected to improvement of the release frequency, when smaller changes are being released, as well as collaborations in the development, pair programming and documentation enhancements.

The communication changes had a positive impact on the pipeline pass rate.

- Number of unit tests

Table 7. A Number of unit tests.

	Number of unit tests			
	Month 1	**Month 2**	**Month 3**	**Month 4**
Before	95	92	96	101
After	112	119	124	140

The number of unit tests is one of the quality metrics, ideally all the functionality of the application should be covered with unit tests. The team has not prioritised unit testing, therefor the overall low score. The before period in the month 2 tests were deleted, and compared to month 1 and month 3, only one test was added. Overall in the before period 6 test were added. The trend after the communication changes is increasing every month from 112 test to 140. With the communication changes, information sharing, the team put more effort into writing unit tests.

The communication changes improved the team effort into writing unit tests.

- Number of code smells

Table 8. A Number of code smells.

Number of code smells				
	Month 1	**Month 2**	**Month 3**	**Month 4**
Before	253	269	275	288
After	268	320	380	372

The number of code smells indicates the quality of the product, depending on the code quality, and code written. The results show increase of code smells over both measurement periods. The decline of the trend can be seen between the month 4 (before) and month 1 (after), and between month 3-4 (after).

The communication changes impact on the code smells cannot be confirmed. The lower the number of code smells, the better the quality of the code, therefor on the aspect of code smells the team showed better results before the communication changes.

4.2.5 Results of Testing Efficiency

The results of testing efficiency, to ensure best quality of the product.

- Testing review efficiency

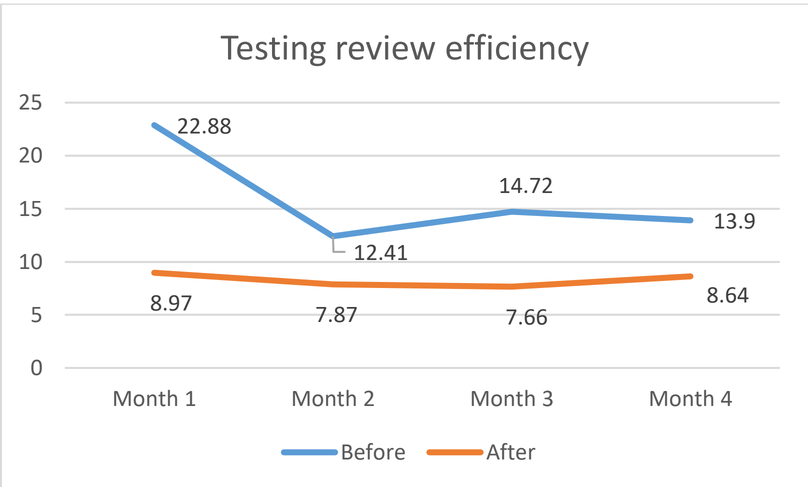


Figure 7. Testing review efficiency.

The team testing review efficiency had a decrease over the measurement time.

The first month of the measurement, the testing review efficiency was at 22.88%, in the last month of the whole period, it had decreased to 8.64%. Before the communication changes, the efficiency got from 22.88% to 12%-15%, whereas after the changes, the efficiency fluctuated between 7.66% to 8.97%, being quite stable. The testing review efficiency in the number of tickets can be seen in appendix 3.

One of the reasons could be, as the team has only one QA specialist, and the team performance increased, the QA role became a bottleneck, and the team could not fulfil the support. The team has realised by themselves as well the need for testing support. As the development team performance increased, the testing efficiency decreased.

The communication changes have had on the testing review efficiency a negative result.

- Defect removal efficiency

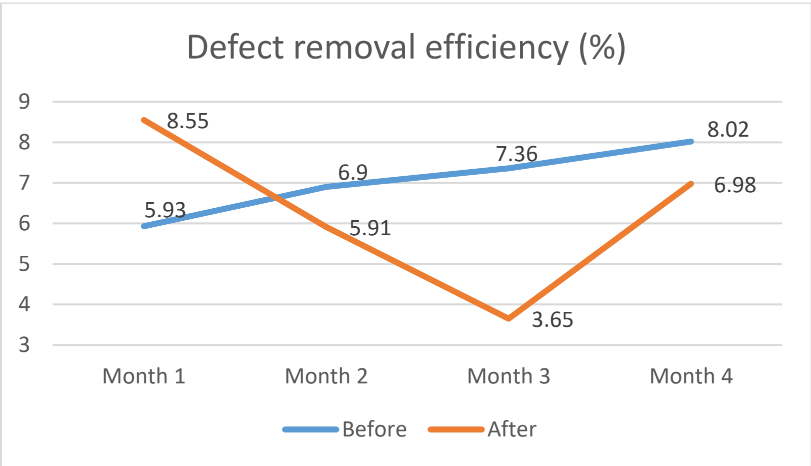


Figure 8. Defect removal efficiency.

The defected removal efficiency showed a positive trend in the beginning of the measurement, having a steady increase from 5.93% to 8.02%, whereas after the changes, the trendline was negative. Starting from 8.55% month 1(after changes), going to 3.65% in month 3(after) and finishing at 6.98%. Which was bit higher than the month 2 in the before result (6.90%). The defect removal ratio in the number of tickets can be seen in appendix 3.

The explanation of the behaviour is the growing backlog for the team, as well as focus on the development of the features rather than bug fixing.

The communication changes do not show positive result on the testing removal efficiency.

4.2.6 Results of Development Performance

The results of team X development performance metrics were following:

- Sprint velocity

Table 9. Sprint velocity.

Number of tickets delivered		
Measurement period	**Tickets delivered (before)**	**Tickets delivered (after)**
Month 1	44	64
Month 2	23	68
Month 3	39	66
Month 4	25	95

The team followed one-month sprints, the releases happening as required. In the beginning, the number of tickets delivered was fluctuating every month, being either 44 or 23 tickets in the sprint, which impacts the team predictability. After the changes in the team communication, the number of tickets delivered kept an increasing, stable trend.

- Number of work-in-progress items

Table 10. A Number of tickets in progress.

Number of tickets in progress		
Measurement period	**Before**	**Final**
Week 1	22	13
Week 2	25	14
Week 3	21	16
Week 4	23	15

As a result of changing the information system development process, adding the Scrum Master role, and stressing the importance of communication and collaboration in the team, the team reduced the number of tickets assigned as a whole. In the beginning, each developer had 2-3 tickets on them marked as ‘in progress’, in the team it being 22 until

25 tickets in progress, whereas in the end, it was average of 1.5 tickets on a developer, being 13 until 16 work-in-progress items . It could mean the developers were taking less tickets on, and sharing the load between each other.

- Number of releases

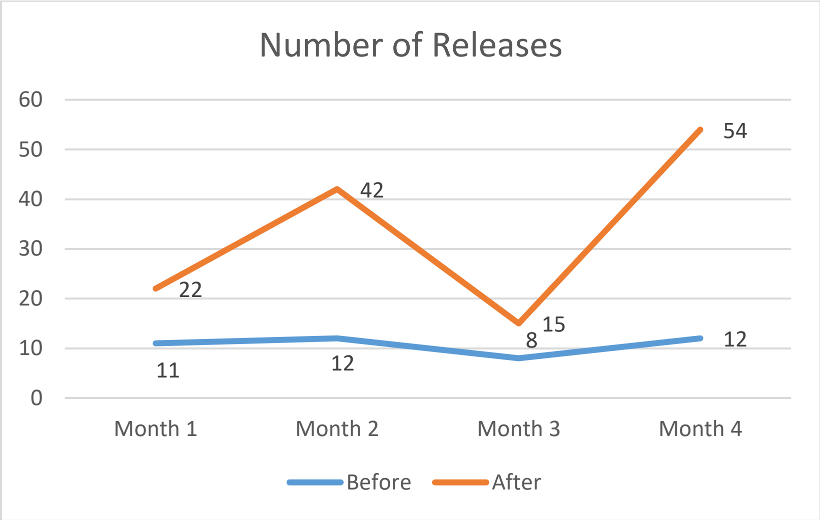


Figure 9. A Number of releases.

The number of releases indicates how often the team delivers improvements to end-user. The releases after the improvements have at least doubled for each month, from 8-12 releases to 15 or 54 releases. The third month setback, is also in correlation of decreased number of commits due to vacation period. After the communication changes, the results indicate a strong trend of incline.

- Tickets created to solved ratio

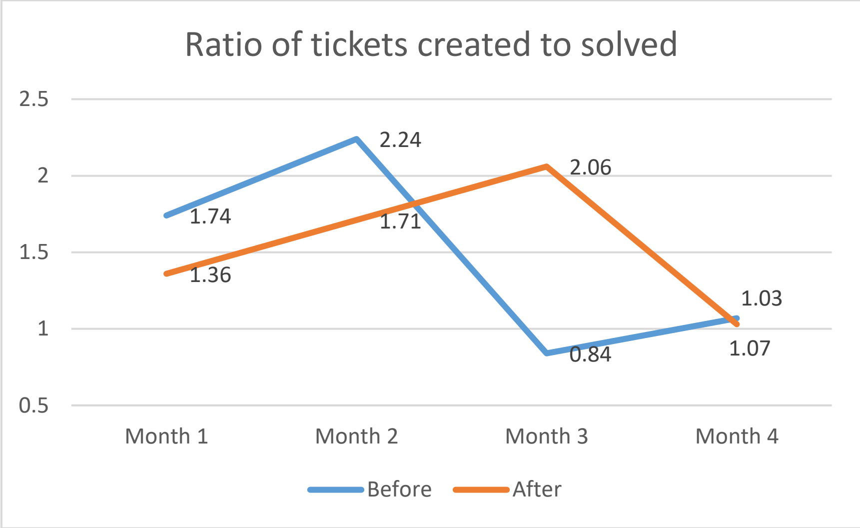


Figure 10. The ratio of tickets created to solved.

The ratio of tickets created to shows the relation for ticket creation to solved tickets in the month, it is also an indicator to development team reaction to new tickets.

Before the team communication changes, the first two months the ticket creation was more than solved, whereas on the third month the trend changed, and the final month as many tickets were solved as created. The detailed number of tickets created to solved overview is in appendix 3.

After the changes, more tickets were added, compared to what were solved, on month 3 there were double of tickets created than solved, on month 4 it was almost equal.

The communication changes give negative impact to the ratio of tickets created to solved. The trend of the ratio after changes indicates a growth of the backlog.

4.2.7 Team Satisfaction Results

The analysis of team satisfaction is divided into two: the results of team survey and the results of team health metrics.

- **Results of Team Survey**

To measure team members' satisfaction with the team process, a survey was conducted. The questions were the same and was sent to the participants three times to see the trend in answers. The first round of questions was sent at the beginning of the measurement, before the communication changes. The second time after working together for 5 months, and the third time in the end of the measurement period.

The survey questions were answered team members online, all answers being anonymous. The first survey had 11 participants and the second survey received 13 answers, and the third round 8 repliers answered.

1. When you think about working in the team, what is the first thing that comes to your mind?

The first question was to associate first thing that comes to mind when thinking about the team. The first time, 5 answer were associated with teamwork. Two answers were mentioning teamwork, others brought out aspects such as agility to tackle challenges together, everyone's involvement in progress and team alignment. Three answers brought out process – being productive, receiving fast and direct feedback and involvement and progress in the cycle.

In the second round of the survey the teamwork was mentioned 7 times, all the answers were straightforward, using either word teamwork, collaboration (4 answers) or helping each other (2 repliers). Two of the answers mentioned adaptability with words varied and change, another one brought out flexibility, flat organisation. Another participant mentioned customer experience as their first thing about the team.

The third trial for the survey gave answers on the same theme, mentioning collaboration, support and teamwork. Two participants brought out fun, and one said 'good atmosphere'. Through the observations, the main theme has remained the same.

2. If you think about the team setup, would you recommend it to others?

The second question was 'would you recommend the team setup to others?' with the answer options of 'yes', 'yes, with adjustments' and 'no'. None of the participants answered 'no'. In the first try 6 out of 11 said yes, others would adjust. The second time 7 out of 13 said yes. The third time, 2 participants said yes, and the rest of the 6 would adjust the teamwork.

3. What adjustments would you recommend to the team?

The participants were asked to clarify the adjustments they would suggest.

In the round one, the most common change would have been structure in the task planning, either by setting goals, contributing to the product backlog or processes, or adding a Scrum Master or project manager. The answers was mentioned the team has multiple missions and focus teams working in parallel, which creates confusion inside and outside of the team.

In the round two, the theme of the changes was same – structure. To create and follow goals, allow everyone to speak up and received needed support, as well as throughout documentation. In the processes the design flow should be better communicated.

The insights from third round of questionnaire, focused on 3 topics – two answers mentioned the role of product owner, another two replies on high level coordination of the separate roles and third being roadmap for the teams focuses. One participant requested more focus on the testing, another structuring of the refinements, and last brought out physical meeting with the team members.

The common theme within the team throughout the survey seems to be structure and clarity of the roles.

4. What is the biggest advantage of the team in your opinion?

When asked about the biggest advantages of the common theme of the answers were communication and flexibility. Three of the replies mentioned flexibility in the processes and team members. Twice was mentioned cross-functionality and information sharing, to improve the process, more people should be part of the information flow. When asked the same question again, the first common team was agility, in speed as well as competences. Three of the answers stressed collaboration in the context of being self-organised and cross-functional, the team can change focus and collectively make changes.

5. What is the biggest disadvantage of the team in your opinion?

The main disadvantage of the team the first time was lack of focus, it was caused by multiple priorities and lack of setting goals, missing documentation, or trying to collaborate on everything. The answers also mention the size of the team, which complicates getting the team together, makes the meetings longer and one clearly states 'remote teams require a lot of communication'.

On the retry questionnaire, the multiple focuses were even more strongly mentioned, which were caused by new priorities, unclear roles, responsibilities, and cross-functionality between the team members. The lack of structure and discipline were mentioned as well.

The last round of survey brought out prioritisation from different angles as the biggest disadvantage – the bigger goal of the team, the balance between work load, or structure and freedom, or business improvements to technical updates, or bug handling. One person brought out the team is bit too person-dependent, another one mentioned lack of QA created bottleneck.

6. How would you rate the collaboration between the different functions (backend, frontend, UX, UI, business)? (10 is excellent, 1 is really bad)

The qualitative question was 'how you would rate the collaboration between the different functions' with the rating from 1 to 10, where being excellent was marked as 10 and one being bad was 1.

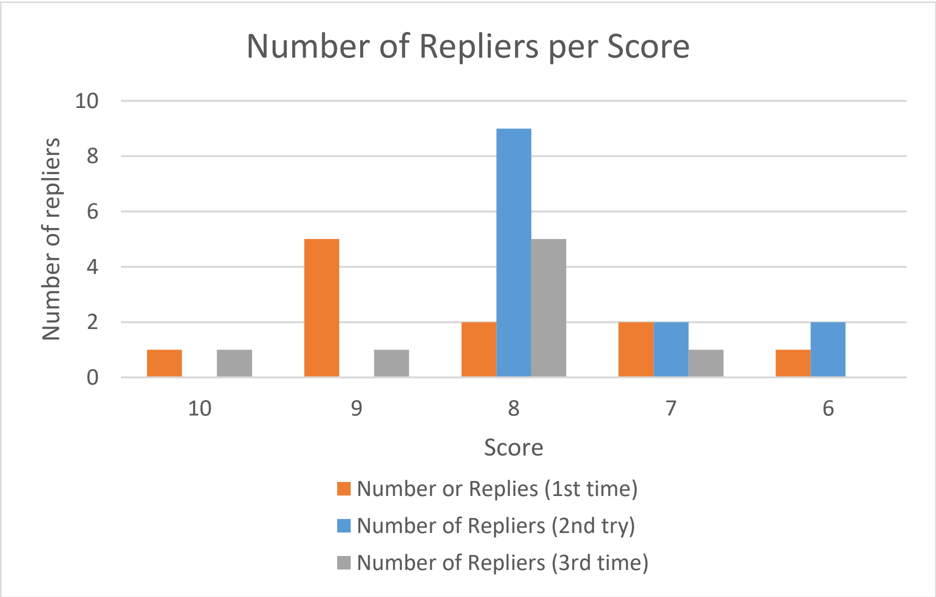


Figure 11. A Number of repliers per score.

The first time and third time one participant rated teamwork to be 10, second not time no one said this. The lowest grade given was 6, by one participants on the first time, and by two in the third round. The most common rank in the first round was 9 (by 5 participants), the second time, the score had decreased to be 8 (by 9 team members), which stayed the same in the third round (by 5 repliers).

7. If you would be the single decision maker of the team, and could decide on one thing, what would it be?

The team members were asked what changes they made, when being the decision maker. Both times one answer was to remove them as main decision maker, as well as one answer saying more time should be invested in team building in a fun way. Among other things were mentioning setting achievable goals, by reducing the features in the most-viable-product and releasing when the feature is ready.

The second time answering the same question, three answers were about celebrating and having some dedicated fun time. Two participants suggested to increase the size of the team by hiring more.

In the final round of answers, two participants suggested expanding the team size, for faster delivery and continuity during the vacation period. Two would commit using planning and roadmaps, plus two added physical meetings, team building and idea developing sessions. The last person said ‘decisions should be the result of a discussion, one decision-maker would be the worst-case scenario’.

8. How do you see the team in one year?

The last question was where the team would be in a year. Team members hope to see the maturity of the team and application development. The team setup should be smaller and more focused but have a better understanding of the product and ever-evolve team collaboration.

The second answer round gave positive answers, according to the team members the team should be more mature, have better structure, tackle more challenges, be bigger in size of members, but overall have the same mentality and achieve remarkable results.

The third time, the team would like to see growth in the team, improving services, handling more topics, yet keeping focus and improving structure. The stress of the team should be on improvement, growth, innovation, initiatives and maintaining the positive team spirit.

The repetition of the survey created insights to seeing trends in the team, what has changed and what need attention to create a change.

The list of most recurring topics needing change in the team:

- The need for improved documentation
- The usage of frequent releases
- Unclear understanding of the roles and domains of expertise
- Collaboration between different team members and roles
- Clear focus on team goals and priorities

The team survey gave insights into team collaboration, performance, and another platform for communication.

- **Results of Team Health Monitoring**

The team members satisfaction was measured using Team Health monitor. The team health monitor would be one way to track the team collaboration, the idea would be to repeat the experiment workshop over time, see the trends, take actions, and aim to better satisfaction of the team. [60]

The experiment workshop consist of three parts: creation of the team health metrics, evaluation of the metrics by each team member, unification of the answers, discussion of the results and creation of action points.

The first part of the team health monitor workshop the team members created the measurements for satisfaction monitoring. The question asked for it was ‘what aspects or words describe the ideal team?’ The result was a combination of mindset, values and process actions which the team formed to the metrics.

The team health metrics definitions:

- Fun – it is about being fun and friendly, mixing good humour into the team, but also organise get togethers, team activities and meetups.

- Empower and evolve – it is about empowering, and motivating each other, acknowledging each team member strengths and weaknesses. The actions taken to empower the team is to have knowledge sharing sessions, take time for pair programming, include different roles into testing, and through all understand dependencies in the team.

- The collaboration – it is about helping team members, being supportive and helpful, and praising cooperation.

- Common goal – it is to work together towards a common goal. It reflects in the ‘get things done’ attitude, setting ‘definition of done’ to tasks and having the confidence in objectives.

- Transparency – it is being open and transparent in the way of working, having free flow of communication. The actions the team needs to follow includes creating meeting notes, updating documentation and ticket statuses, and preferring chat over emails as the communication channel.

The second part of the workshop, each of the team member were asked to evaluate their understanding of the current level of each of the aspect in the metric. The possible answers were ‘good’, ‘could be better’, ‘bad’.

After the individual assessment, the third part of the experiment, was to unify the answers. The lowest answers were asked to be explained to understand, and to make adjustment accordingly. The team overall result created was based on the most common answer. If the results were equal, lower was chosen.

The final part of the workshop was to discuss the results, and create action point to improve the areas needing improvements.

For the Team X, the team health monitoring workshop was held twice: the first time in the beginning of the team performance measurement, the second time 3 months later. The team decided not to repeat the workshop more, as for them, the retrospectives and performance metrics created more value to evaluate and reflect their satisfaction of teamwork.

Sample results of the first round:

Figure 12. Results of Team Health Metrics (1st time).

The results of the second time:

Figure 13. Results of Team Health Metrics (2nd time).

As a result of the workshops, each time, a certain level of team health metrics was developed. Over time, while repeating the workshop a trend can be seen. In the first time of experiment, the team was more unified and more moderate with the answers given. After the overall going through of the results, action points for improvements were discussed.

The term ‘fun’ got a tie, therefore the lower was chosen. The action points were to have an online game time, a free space to bond and create connections. For the improvement of ‘empowerment and evolvement,’ a series of knowledge sharing sessions were created, on the topics about product technical setups, improvements, as well as modern technologies around, and booked pair programming sessions for at least 2 pairs. For the ‘transparency’ a promise to improve documentation was given, 3 team members were committing to add documentation to their certain domain areas. In order to progress with ‘common goal achievement’ testing sessions with business, developers and testers were created, ticket statuses needed improvement, and emphasise ticket solving steps (developing, testing, documentation) in the team process. The team was happy with their collaboration.

In the second time of the workshop, the steps promised to be committed were evaluated, and new round of self-assessment was done. Compared to the first time, the commitment to ‘common goal’ had gotten a worse result. The ticket solving steps were dragging the results down, as well as the team commented the general goal of the team had gotten hazy. Even though the team did not seem to improve the end result of the empowerment, the actions were success.

5 Analysis and Conclusions

The chapter begins with answers to the research questions followed by conclusions.

The answers to research questions are as follows:

RQ: Does changes in communication in agile distributed team's communication impact team performance and how?

The team changes in communication has effect on team performance, most of the measured team performance metrics had improved.

Some of the most significant improvements: the product lead time and cycle time got almost two times faster, the number of incidents reduced to one minor incident, the sprint velocity stabilised and increased, the number of releases per month increased almost twice.

The detailed overview of each team performance metric is answered in question 5.2 in the current chapter.

1. **What are the characteristics of practices of information system development process and communication in distributed team compared to co-located team?**

The information system development process in distributed teams is following the same practices, yet the importance of communication should be prioritized, whereas in the co-located team the communication comes naturally.

Communication in distributed teams is dependent on participants involved, coordination and control, as well as time, geographical location and socio-cultural background. Communication in the distributed teams gives a good insight into the success of the team, the duration and frequency of are reported to be the same as in co-located teams, but depending heavily on different communication channels to insure the message has been delivered to the recipient.

2. **What is team performance and what are the corresponding metrics?**

Team performance metrics are combination of process and output measurement results, depending on the input and product.

The variety for team performance metrics chosen for the research were based on agile development, considering the relevance for the given Team X. The measurement were covering various aspects of the team performance – time, ownership, responsibilities, development performance, testing efficiency, team members satisfaction.

3. What is communication and how to measure its quality?

Communication is exchange of information with another party. Communication quality is effectiveness of delivering the information in effective way to both sides. Communication measurements chosen were to measure the frequency and patterns of communication in the team.

The chosen metrics were:

- The number of participants in communication
- Communication frequency
- Number of communication channels
- The number of communication paths
- Communication paths on a graph

4. What are the ways to improve team communication and team performance?

The main improvements for team performance are taken from the theoretical overview and team feedback, what needs to be done, either is it creating documentation, adding alignment meetings, sharing responsibilities of the team, encouraging pair-programming

The actions chosen for team communication improvements:

- Common meetings
- Increased coworking time
- Improved documentation
- Scheduled retrospectives for the team
- Introduction of the Scrum Master role

- Identification of team members skills and knowledge for focused communication

The previously listed changes were applied to the Team X.

5. What are the results of the case study?

The results of the case study is divided into two, the results of team quality of communication and the team performance before and after changes in communication.

5.1. What is the team quality of communication before and after changes?

The team communication quality was improved by setting team expertise about the domains, which brought clarity about the knowledge base of different topics.

The communication frequency has increased almost double due to the communication changes, showing average result of 47.6. The number of team members remained the same as well the communication channels, and the communication patterns illustrated on a graph show an increase of communication between different members of the team.

5.2. What is the team performance before and after changes in communication?

Most of the team performance metrics were improved due to the communication changes. There were metrics which also showed either a negative trend or not much correlation for the communication.

The team performance which have improved as result of the communication changes:

- Number of commits to other repositories increased to 1 commit over the period to at least 1 or 2 commits per month
- Number of incidents reduced to 2.25 to 1 minor incident
- Lead time reduced from over 5 weeks to 2 weeks 3days
- Product cycle time reduced from 3 weeks 4days to 2 weeks 4 days, or even 1 week 2 days
- Automation test pass rate increased from 23% to 26% or even 43%
- Pipeline pass rate increased from average of 92% to average of 94%
- Number of unit tests increased from 95 to 140 test
- Sprint velocity increased from fluctuating between 23 to 39 to increase around 65 tickets delivered in a month long sprint

- Number of tickets in progress reduced from 22 to 15
- Number of releases increased from 11 in a month to 54 in a month

The team performance metrics having decrease after communication changes:

- Number of commits were fluctuating between the measurement period
- Number of code smells increased from 253 to 372 code smells reported, indicating decrease in code quality
- Testing review efficiency started from 23%, decreased to 8.64%
- Defect removal efficiency showed a steady increase from 5.94% to 8%, but ended with a fluctuation with the rate of 6.98%

The change in the communication is one-way for team performance enhancements, other way to improve development team performance is enhancing technical or administrative sides, such as updating frameworks, highlighting importance of code clean up, but as well as improving deployment pipelines, integration to cloud platforms, and applying modern technologies to the applications. It is important to understand why the team performance is changed, what are the metrics chosen and how they affect the team.

The side note of measuring team performance is understanding the bigger picture - the context of the product, what is happening in the organisation, team members joining or leaving, etc - which all affect the metrics and communication in the team.

6 Summary

The research paper looked into connection between the team communication and its impact to team performance in a distributed agile team, based on Team X example.

The aim of the paper was to find if changes in communication in agile distributed team has impact on team performance and how.

The given paper gave a literature review about communication and team performance, and what are the ways for measurement. The findings were used on the Team X in order to apply changes and measure the team performance, and see if the communication changes had an impact on the metrics results.

The result of the research is indicating the changes made in communication impacts team performance, the result is based on analysis on the communication metrics and the team performance before and after changes in the team communication. Most of the Team X performance metrics indicate positive results of performance metrics and communication quality. The negative trend of performance metrics were connected to testing, which got less attention in the team’s communication and collaboration.

The established questions for the case study were answered, therefore the purpose of the paper was fulfilled.

The future improvements for the research can be done by increasing the size of the teams under observations, either by the number of members in the team or comparison of different teams. Another approach for future development is adding the number of information system development process metrics. Third approach for the future investigations would be looking into team performance metrics over longer period of time.

References

[1] M. Marinho, A. Luna and S. Beecham, “Global Software Development: Practices for Cultural Differences,” 03. 11. 2018. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-03673-7_22. [Accessed 16. 11. 2020].

[2] P. T. Robinson, “Communication Network in an Agile Distributed Software Development Team,” 22. 08. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8807485/. [Accessed 12. 10. 2020].

[3] J. Eckstein, “Agile Software Development with Distributed Teams: Staying Agile in a Global World,” 07. 2013. [Online]. Available: https://www.oreilly.com/library/view/agile-software-development/9780133492385/. [Accessed 29. 10. 2021.].

[4] World Health Organization (WHO), “A timeline of WHO’s response to COVID-19 in the WHO European Region: a living document,” 31. 07. 2021. [Online]. Available: https://www.euro.who.int/en/health-topics/health-emergencies/coronavirus-covid-19/publications-and-technical-guidance/2021/a-timeline-of-whos-response-to-covid-19-in-the-who-european-region-a-living-document-update-to-version-2.0-from-31-december-2019-to-. [Accessed 12. 11. 2021.].

[5] D. M. Kennedy, S. A. McComb and R. R. Vozdolska, “An Investigation Of Project Complexity's Influence On Team Communication Using Monte Carlo Simulation,” 09. 04. 2011. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0923474811000105. [Accessed 18. 05. 2021].

[6] M. Hummel, C. Rosenkranz and R. Holten, “The Role of Communication in Agile Systems Development. An Analysis of the State of the Art,” 04. 05. 2013. [Online]. Available: https://www.researchgate.net/publication/259221905_The_Role_of_Communication_in_Agile_Systems_Development_An_Analysis_of_the_State-of-the-Art. [Accessed 16. 05. 2020.].

[7] C. W. H. Davis, Agile Metrics in Action, New York: Manning Publications Co, 2015.

[8] K. Beck, A. Cockburn, M. Fowler, J. Sutherland, R. C. Martin and e. al, “Manifesto for Agile Software Development,” 2001. [Online]. Available: https://agilemanifesto.org/. [Accessed 28. 02. 2022].

[9] J. Schoonenboom and R. B. Johnson, “How to Construct a Mixed Methods Research Design,” 05. 06. 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5602001/.

[10] M. T. Brannick, E. Salas and C. W. Prince, Team Performance Assessment and Measurement: Theory, Methods, and Applications, New York: Taylor&Francis, 2009.

[11] B. Singh and S. Gautam, “The Impact of Software Development Process on Software Quality: A Review,” 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8082729. [Accessed 10. 05. 2021].

[12] W. W. Royce, “Managing the Development of Large Software Systems,” 08. 1970. [Online]. Available: http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf. [Accessed 10. 10. 2021].

[13] H. F. Cervone, “Effective communication for project success,” 06. 05. 2015. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/OCLC-02-2014-0014/full/html. [Accessed 2021].

[14] W. Van Casteren, “The Waterfall Model and the Agile Methodologies: A comparison by project characteristics - short,” 02. 2017. [Online]. Available: https://www.researchgate.net/publication/313768860_The_Waterfall_Model_and_the_Agile_Methodologies_A_comparison_by_project_characteristics_-_short. [Accessed 28. 12. 2021].

[15] M. Paasivaara and C. Lassenius, “Scaling Scrum in a Large Globally Distributed Organization: A Case Study,” 05. 08. 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7577422. [Accessed 11. 05. 2021].

[16] J. Sutherland and K. Schwaber, “The 2020 Scrum Guide,” 2020. [Online]. Available: https://scrumguides.org/scrum-guide.html.

[17] M. O. Ahmad, J. Markkula and M. Oivo, “Kanban in software development: A systematic literature review,” 2013. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6619482. [Accessed 10 10 2021].

[18] M. Stoica, B. Chilic-Micu, M. Mircea and C. Uscatu, “Analyzing Agile Development – from Waterfall Style to Scrumban,” 04. 2016. [Online]. Available: https://pdfs.semanticscholar.org/84b3/46b298ff706bc55c9dae57bc34798e4ae416.pdf. [Accessed 19. 09. 2020].

[19] P. B. Nirpal and K. V. Kale, “A Brief Overview Of Software Testing Metrics,” [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.5632&rep=rep1&type=pdf. [Accessed 15. 11. 2021].

[20] M. Cohn, “Succeeding with Agile. Software Development Using Scrum,” [Online]. Available: https://learning.oreilly.com/library/view/succeeding-with-agile/9780321660534/ .

[21] B. Al-Ani and K. H. Edwards, “A Comparative Empirical Study of Communication in Distributed and Collocated Development Teams,” 03. 10. 2008. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4638651. [Accessed 22. 02. 2021].

[22] S. Ashmore, “The Impact of Process on Virtual Teams: A Comparative Analysis of Waterfall and Agile Software Development Teams,” 2012. [Online]. Available: https://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=3267&context=etd.

[23] C. Swetel, “The Development Metrics You Should Use (but Don’t),” in *GOTO Conference*, 2020.

[24] A. Scandaroli, R. Leite, A. H. Kiosa and S. A. Coelho, “Behavior-driven development as an approach to improve software quality and communication across remote business stakeholders, developers and QA: two case studies,” 22. 08. 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8807783. [Accessed 16. 05. 2020].

[25] A. Shakeri and M. Khalilzadeh, “Analysis of factors affecting project communications with a hybrid DEMATEL-ISM approach (A case study in Iran),” [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7408334/.

[26] I. Marin, “Data Science and Development Team Remote Communication: the use of the Machine Learning Canvas,” 19. 08. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8807620. [Accessed 04. 03. 2020].

[27] Y. Yue, I. Ahmed, Y. Wang and D. Redmiles, “Collaboration in Global Software Development: An Investigation on Research Trends and Evolution,” 26. 05. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8807779. [Accessed 20. 04. 2022.].

[28] A. Sablis, D. Smite and N. B. Moe, “Exploring Cross-Site Networking in Large-Scale Distributed Projects,” 2018. [Online]. Available: https://sintef.brage.unit.no/sintef-xmlui/handle/11250/2590009. [Accessed 18. 05. 2021].

[29] A. Pentland, “The New Science of Building Great Teams,” 05 2012. [Online]. Available: https://hbr.org/2012/04/the-new-science-of-building-great-teams. [Accessed 02 05 2021].

[30] G. Arias, D. Vilches, C. Banchoff, I. Harari, V. Harari and P. Iuliano, “The 7 key factors to get successful results in the IT Development projects,” *ScienceDirect,* pp. 199-207, Procedia Technology 5 2012.

[31] C. Gutwin and S. Greenberg, “The Importance of Awareness for Team Cognition in Distributed Collaboration,” [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.17.3674&rep=rep1&type=pdf.

[32] Amysoft Inc., “Communication on Agile Software Teams,” [Online]. Available: http://agilemodeling.com/essays/communication.htm. [Accessed 30. 11. 2021].

[33] N. Ramasubbu and R. K. Balan, “Globally Distributed Software Development Project Performance: An Empirical Analysis,” 09. 2017. [Online]. Available: https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=1818&context=sis_research. [Accessed 11. 11. 2021].

[34] Hofstede Insights, “WHAT IS THE RELATIONSHIP BETWEEN STRATEGY AND COMPANY CULTURE?,” 05. 05. 2021. [Online]. Available: https://news.hofstede-insights.com/news/what-is-the-relationship-between-business-strategy-and-company-culture.

[35] Y. I. Alzoubi and A. Q. Gill, “An Empirical Investigation of Geographically Distributed Agile Development: The Agile Enterprise Architecture Is a Communication Enabler,” 19. 03. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9078794. [Accessed 25. 02. 2022].

[36] P. Hinds and C. McGrath, “Structures that Work: Social Structure, Work Structure and Coordination Ease in Geographically Distributed Teams,” 04. 11. 2006. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/1180875.1180928. [Accessed 16. 05. 2021].

[37] A. Kobyliński, “The Relationships between Software Development Processes and Software Product Quality,” 09. 2013.. [Online]. Available: https://www.researchgate.net/publication/300710746_The_Relationships_between_Software_Development_Processes_and_Software_Product_Quality . [Accessed 20. 10. 2021].

[38] S. K. White, “What is CMMI? A model for optimizing development processes,” 01. 06. 2021. [Online]. Available: https://www.cio.com/article/2437864/process-improvement-capability-maturity-model-integration-cmmi-definition-and-solutions.html. [Accessed 15. 10. 2021].

[39] A. A. Bubshait, M. K. Siddiqui and A. M. A. Al-Buali, “Role of Communication and Coordination in Project Success: Case Study,” 11. 03. 2014. [Online]. Available: https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29CF.1943-5509.0000610. [Accessed 16. 05. 2021.].

[40] N. Bettenburg, “Studying the Impact of Developer Communication on the Quality and Evolution of a Software System,” 05. 2014. [Online]. Available: https://qspace.library.queensu.ca/bitstream/handle/1974/12191/Bettenburg_Nicolas_JSA_201405_PhD.pdf?sequence=3&isAllowed=y. [Accessed 22. 05. 2021].

[41] N. Nagappan, B. Murphy and V. R. Basili, “The Influence of Organizational Structure on Software Quality: an Empirical Case Study,” 2008. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2008-11.pdf.

[42] R. Heinrich and B. Paech, “Defining the Quality of Business Processes,” [Online]. Available: https://subs.emis.de/LNI/Proceedings/Proceedings161/133.pdf. [Accessed 2021].

[43] A. Kambil, “Drivers of team performance: Do you have a team? Do you want a team?,” 2015. [Online]. Available: https://www2.deloitte.com/content/dam/insights/us/articles/team-performance/DUP_1389_DoYouHaveATeam.pdf . [Accessed 28. 04. 2021].

[44] A. Harvey, “Your Team as a Distributed System,” 10-13. 06. 2019. [Online]. Available: https://learning.oreilly.com/videos/your-team-as/0636920338635/0636920338635-video327242/. [Accessed 19. 09. 2021].

[45] R. H. Yacobellis, “Software and Development Process Quality Metrics,” 27. 04. 1984. [Online]. Available: https://ieeexplore.ieee.org/document/7271280. [Accessed 11. 05. 2021].

[46] R. Osherove, Writer, *Elastic Leadership.* [Performance]. 05.11.2021.

[47] PremierAgile, “What Is Lead & Cycle Time? Why Is It Important?,” [Online]. Available: https://premieragile.com/lead-time-and-cycle-time-in-scrum/. [Accessed 16. 02. 2022].

[48] Agile Alliance, “Lead Time,” [Online]. Available: https://www.agilealliance.org/glossary/lead-time/. [Accessed 15. 10. 2021].

[49] Center of Business Practices , “Measures of Project Management Performance and Value,” [Online]. Available: https://www.pmsolutions.com/audio/PM_Performance_and_Value_List_of_Measures.pdf. [Accessed 19. 10. 2021].

[50] Simplilearn, “Project and Process Metrics Classifying the Process Metric Measurement,” 23. 03. 2021. [Online]. Available: https://www.simplilearn.com/project-and-process-metrics-article. [Accessed 24. 10. 2021].

[51] Thinksys, “Software Testing Metrics & KPIs,” 05. 11. 2018. [Online]. Available: https://www.thinksys.com/qa-testing/software-testing-metrics-kpis/. [Accessed 25. 11. 2021].

[52] Microsoft Azure, “Pipeline reports,” 11. 02. 2022. [Online]. Available: https://docs.microsoft.com/en-us/azure/devops/pipelines/reports/pipelinereport?view=azure-devops. [Accessed 18. 03. 2022].

[53] M. Fowler, “CodeSmell,” 09. 02. 2006. [Online]. Available: https://martinfowler.com/bliki/CodeSmell.html. [Accessed 16. 03. 2022].

[54] T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel and K. Petersen, Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation, Rome: Springer, 2014.

[55] Merriam-Webster, “Adabt,” [Online]. Available: https://www.merriam-webster.com/dictionary/adapt. [Accessed 30. 10. 2021].

[56] S. Park and S. Park, “Employee Adaptive Performance and Its Antecedents: Review and Synthesis,” 03. 04. 2019. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/1534484319836315. [Accessed 28. 10. 2021].

[57] A. Chowdhury, “Agile Metrics: The 15 That Actually Matter for Success,” 22. 02. 2021. [Online]. Available: https://www.plutora.com/blog/agile-metrics. [Accessed 02. 03. 2022].

[58] Atlassian, “"Created vs. Resolved" Report,” 09. 12. 2020. [Online]. Available: https://community.atlassian.com/t5/Jira-articles/quot-Created-vs-Resolved-quot-report/ba-p/1553129https://community.atlassian.com/t5/Jira-articles/quot-Created-vs-Resolved-quot-report/ba-p/1553129. [Accessed 15. 03. 2022].

[59] B. Aubert, V. Hooper and A. Schnepel, “Revisiting the role of communication quality in ERP project success,” 12. 04. 2013. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/19355181311314770/full/html. [Accessed 28. 09. 2021].

[60] H. Kniberg, “Squad Health Check model – Visualizing What to Improve,” 16. 09. 2014. [Online]. Available: https://engineering.atspotify.com/2014/09/squad-health-check-model/. [Accessed 01. 06. 2021].

[61] O. Blais, “Product Analytics — How to Measure Your Software Development Success,” 02. 02. 2018. [Online]. Available: https://towardsdatascience.com/product-analytics-how-to-measure-your-software-development-success-7a6bc765dbab. [Accessed 30. 10. 2021].

[62] Merriam-Webster, “Communication definition,” [Online]. Available: https://www.merriam-webster.com/dictionary/communication. [Accessed 25. 10. 2021].

[63] MSI - Management and Strategy Institute, “Project Quality and Communication Management,” 08. 09. 2017. [Online]. Available: https://www.msicertified.com/blog/project-quality-and-communication-management. [Accessed 20. 10. 2021.].

[64] J. Anttila, “Effective Quality Communication,” 06. 2001. [Online]. Available: http://www.qualityintegration.biz/Communication.html. [Accessed 24. 10. 2021].

[65] L. Bourne, “The True Cost of Quality in Project Communication,” 17. 10. 2012. [Online]. Available: https://projectmanager.com.au/cost-quality-project-communication/. [Accessed 21. 10. 2021].

[66] C. C. Silva, F. Gilson and M. Galster, “Comparison Framework for Team-Based Communication Channels,” 18. 11. 2019. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-35333-9_22.

[67] V. Stray, N. B. Moe and M. Noroozi, “Slack Me If You Can! Using Enterprise Social Networking Tools in Virtual Agile Teams,” 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8807500.

[68] G. Kanakis, S. Fischer, E. Khelladi and A. Egyed, “Supporting A Flexible Grouping Mechanism for Collaborating Engineering Teams,” 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8807504.

[69] J. McManus, “Conway’s Law: A Focus on Information Systems Development,” 18. 11. 2019. [Online]. Available: https://academic.oup.com/itnow/article-abstract/61/4/50/5628364 . [Accessed 29. 09. 2020].

[70] M. Conway, “How Do Commitees Invent?,” 1968. [Online]. Available: https://www.melconway.com/Home/pdf/committees.pdf.

[71] LighterCapital, “Silos and Turf Wars: What are Organizational Silos and Why Do They Exist,” 2020. [Online]. Available: https://www.lightercapital.com/blog/what-are-organizational-silos/. [Accessed 30. 01. 2021].

[72] K. L. Blatter, T. J. Gedhill, J. L. Krein and C. D. Knutson, “Impact of Communication Structure on System Design: Towards a Controlled Test of Conway’s Law,” 14. 11. 2013. [Online]. Available: https://ieeexplore.ieee.org/document/6664728. [Accessed 25. 09. 2020].

[73] V. Luu, “How to Improve Communication Frequency With Your Remote Team,” 25. 07. 2019. [Online]. Available: https://betterhumans.pub/how-to-improve-communication-frequency-with-your-remote-team-a446e15e5bb5. [Accessed 22. 01. 2022].

[74] C. Sethuraman and K. S. Srivatsa, “Effective communication for software professionals,” 2009. [Online]. Available: https://ieeexplore.ieee.org/document/4909188.

[75] I. Petuhhov, “Põhimõisted graafi kohta,” 2020. [Online]. Available: http://www.cs.tlu.ee/~inga/AAS/graph_2020.pdf. [Accessed 30. 11. 2021].

[76] G. Wu, C. Liu, X. Zhao and J. Zuo, “Investigating the relationship between communication-conflict interaction and project success among construction project teams,” 05. 09. 2017. [Online]. [Accessed 18. 04. 2021].

[77] R. Kazman, “Managing Social Debt in Large Software Projects,” 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8882853/.

[78] S. Harrer, J. Christ and M. Huber, “Remote Mob Programming,” Innoq, 2022. [Online]. Available: https://www.remotemobprogramming.org/. [Accessed 02. 01. 2022].

Appendix 1 – Lõputöö lihtlitsents

Mina, Kirke Krämann

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Communication Change Impact on Team Performance by Example of Agile Distributed Team X” (“Kommunikatsiooni muutmise mõju meeskonna tulemuslikkusele hajutatud agiilse meeskonna X näitel”),
   mille juhendaja on Karin Rava, MSc
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

10.05.2022

Appendix 2 – Team Survey Questionnaire

1. When you think about working in the team, what is the first thing that comes to your mind?

2. If you think about the team setup. Would you recommend it to others?

3. What adjustments would you recommend to the team?

4. What is the biggest advantage of the team in your opinion?

5. What is the biggest disadvantage of the team in your opinion?

6. How would you rate the collaboration between the different functions (backend, frontend, UX, UI, business)? (10 is excellent, 1 is really bad)

7. If you would be the single decision maker of the team, and could decide on one thing, what would it be?

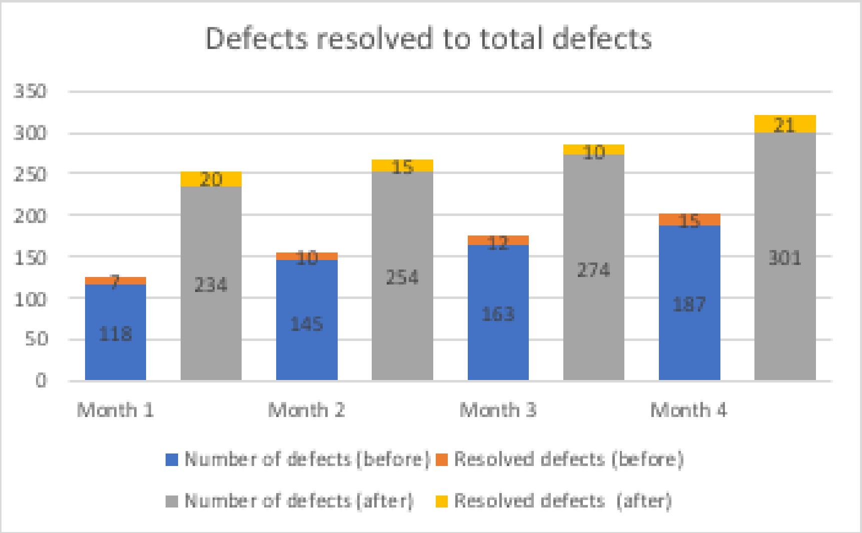
8. How do you see the team in one year?

Appendix 3 – Results of Team Performance Metrics Details

- Testing review efficiency

	Testing review efficiency			
	Measurement period	**Defects created**	**Total number of bugs**	**Testing review efficiency (%)**
Before	**Month 1**	27	118	22,88
	Month 2	18	145	12,41
	Month 3	24	163	14,72
	Month 4	26	187	13,90
After	**Month 1**	21	234	8,97
	Month 2	20	254	7,87
	Month 3	21	274	7,66
	Month 4	26	301	8,64

- Defect removal efficiency



	Defect removal efficiency percentage			
	Measurement period	**Defects resolved**	**Total number of bugs**	**Defects removed percentage(%)**
Before	**Month 1**	7	118	5,93
	Month 2	10	145	6,90
	Month 3	12	163	7,36
	Month 4	15	187	8,02
After	**Month 1**	20	234	8,55
	Month 2	15	254	5,91
	Month 3	10	274	3,65
	Month 4	21	301	6,98

- Number of Releases

	Number of Releases			
	Month 1	**Month 2**	**Month 3**	**Month 4**
Before	11	12	8	12
After	22	42	15	54

- Ratio of tickets created to solved

	Ratio of tickets created to solved			
	Measurement period	**Tickets created**	**Tickets solved**	**Ratio**
Before	**Month 1**	40	23	1,74
	Month 2	56	25	2,24
	Month 3	54	64	0,84
	Month 4	74	69	1,07
After	**Month 1**	90	66	1,36
	Month 2	84	49	1,71
	Month 3	68	33	2,06
	Month 4	98	95	1,03

- Number of commits by developer

	Number of commits by developer							
	Before				**After**			
	Month 1	Month 2	Month 3	Month 4	Month 1	Month 2	Month 3	Month 4
Developer	59 AK 25 J 12 SN 15 FO 12 FJ 9 KK 6 SZ 5 OS 5 SZ 1 AE	86 J 24 SN 22 AK 22 KK 16 AE 8 OS 7 SZ 6 FJ	48 J 38 AK 17 SN 6 OS 6 FO 4 KK 3 SZ 2 FJ 1 SZ	95 J 71 AK 22 SZ 16 KK 6 FJ 5 FO 4 AE 4 OS 2 SN 2 QA	131 J 66 AK 14 SZ 14 SN 9 QA 8 FJ 7 OS 6 AE 5 KK 4 FO	61 J 56 AK 14 OS 12 SZ 9 SN 9 FJ 7 FO 7 SZ 2 KK 1 AE	32 AK 30 J 19 SZ 8 SN 7 QA 6 AE 8 KK 4 OS 4 FJ 1 FO	100 J 41 AK 25 SN 17 SZ 15 KK 8 OS 4 AE 3 QA 3 FJ
Total	149	188	125	227	264	178	119	216