Mahmood Reza Alizadeh Ashrafi

Improving the Hardware Modelling Framework in the Case Company

Metropolia University of Applied Sciences

Master of Business Administration

Business Informatics

Thesis

30.11.2021

Abstract

Author:	Mahmood Reza Alizadeh Ashrafi
Title:	Improving the Hardware Modelling Framework in the Case Company
Number of Pages:	51 pages
Date:	30 November 2021
Degree:	Master of Business Administration
Degree Programme:	Business Informatics
Instructor:	Antti Hovi, Senior Lecturer

The objective of this thesis was to propose an improved framework for the hardware modelling team in the case company. The outcome of this thesis consists of a set of proposals under two different categories, the software development practices in an Agile environment, and proposals relevant to the capability and performance management at the case organization.

The thesis was carried out in seven stages. In the first stage, the thesis established the business challenge and the objective of this study. In the second stage, the research approach, research design, data collection plan and its schedule, and research quality criteria for this qualitative study were presented.

At the third stage, the current state analysis was conducted to examine the current way of working and software development practices. This stage included the first the data collection plan, in which several interviews with modelling personnel were conducted. In the fourth stage, existing knowledge and available best practices were explored, which were investigated based on the main areas discovered in the current state analysis. It covered domains such as Agile methodology, project management, DevOps, and capability and performance management in an Agile environment.

In the fifth stage, the initial proposal was co-created through a mixture of the information from the current state analysis, existing knowledge and best practices, and interview and discussions in the second data collection phase. The proposals included guidelines for two distinct domains. The first one proposed improved practices for the software development in an Agile team, and the second category proposed key actions in the Agile performance management. In the sixth stage, the initial proposal, which was partly implemented by the team, was reviewed by the key stakeholder and the validity of the initial proposal was acknowledged and the final proposal was created. Finally, the seventh stage concluded the thesis by presenting its summary, evaluation, and the future direction.

Keywords: Hardware Modelling, Agile Methodology, DevOps Practices, Performance Management

Contents

1 Introduction

Fifth generation (5G) technology is the most recent and advanced mobile telecommunication standard, which is aimed to replace the existing 4G long term evolution technology. Standardization and technology definitions of 5G are still evolving, but commonly, 5G is characterized by 3 major improvements over the previous generation, i.e., enhanced mobile broadband, ultra/high reliability and low latency, and massive machine-type communication. The first one declares that with deployment of 5G solutions, gigabytes in seconds will be achieved. The second one emphasizes decreased network latency and increased reliability compared to its predecessors, and the last one promises facilitation of internet of things in a larger scale. (Shafi et al., 2017.)

To illustrate the massive changes that 5G will bring, it is worth mentioning that its market size is estimated to increase from approximately $6 billion in 2020 to $670 billion in 2026. (PR Newswire, 2020.)

Telecommunication industry is known to be an oligopoly market, which is due to the fact that a significant infrastructure and resources are required to establish and support cellular networks and related services. Also, note that major players in the telecommunication infrastructure industry have been in this business for several decades, and therefore have constantly rectify their technological assets and infrastructure, which make it harder for new entities to compete. However, in an ongoing race for the 5G technology development and deployment, it results in a fierce competition. Currently, three major 5G vendors dominate the market with close to 80% of the market share (Telecoms, 2020). Therefore, optimization of their resources and frameworks in all phases of product development and delivery to achieve a better customer share and higher profitability is of utmost importance for these companies.

However, designing and building elements of such networks is a very difficult task. A widely used methodology to evaluate the early behaviour of such complex systems or any subsystem under developments is modelling the hardware, i.e., integrated circuit chip, via software-based models, which aims to demonstrate the algorithmic and functional correctness of the design. Modelling the hardware component through computer simulations are used early in the development process to gain confidence on the alignment of algorithms and functionality to the target application. Therefore, modelling is a crucial approach to emulate the efficiency and feasibility of the system.

1.1 Business Context

The case company within the scope of this thesis is a telecommunication equipment, service, and consumer electronics provider company.

1.2 Business Challenge, Objective and Outcome

This thesis focuses on studying and improving the current hardware (integrated circuit) modelling practices in the case company. Before moving to manufacturing and deploying phases of the hardware intellectual property (IP) and chipsets, the design validity, accuracy, and performance should be modelled and verified against several design criteria making it an essential part of the overall process. It is clear that faulty modelling practices will consequently lead to underperforming outputs, where cumbersome and expensive fault identification and debugging processes are required to revert to an acceptable performance, which eventually affect the product delivery pipeline.

As mentioned, an important issue to be tackled is the improvement of the hardware modelling framework. Behavioural and performance modelling of products to be fabricated is an initial and crucial step in the development of hardware IPs. A better design of the modelling process will reduce the potential fault occurrences and eliminate lengthy investigations required to identify the root

causes, and therefore facilitate the overall verification process of the designed hardware, which consequently accelerates the timely and successful deliveries of integrated circuit modules.

The objective of this thesis is *to improve the current hardware modelling practices (framework)* in the case under the study, which is achieved via analysing a pool of qualitative data. The goal is to extract potential trends, strengths and weakness of the existing way of working, and further ameliorate the resource allocation and possibly the workflow of the hardware modelling team.

The outcome of this thesis is *a proposal for improved hardware modelling practices (framework)*. A successful implementation of such a framework can potentially enhance the project delivery setting and avoid fault incidents and expensive delays.

1.3 Thesis Outline

This thesis is organized in seven sections. Section 1 provides a general overview of the scope of the thesis and defines the business context, research problem, and the outline of the thesis. Section 2 renders a detailed description of the utilized methodology in the thesis including the design and data collection plan for the research. In Section 3, the current state of the research problem is analysed and subsequently in Section 4, the corresponding conceptual framework is built based on the available knowledge and best practice. In Section 5, the proposed solutions for the research problem are defined and the modality of implementation of the proposed framework is discussed. Section 6 focuses on the validation of the proposed framework, and finally, Section 7 concludes the thesis with a summary of the study.

2 Method and Materials

In this section of the thesis, research methods, study design, and data collection strategies are discussed, and finally research quality criteria are reviewed and evaluated. First, the research approach and its core features are described. Next, research design and data collection process are presented. Furthermore, validity and reliability of the study are also discussed. The research method is designed in a particular manner to address the specific business problem and ultimately achieve the study objective, which aims to improve the hardware modelling framework and practices in the case company.

2.1 Research Approach

Research approach generally include several stages, i.e., data collection, data analysis, interpretations of findings through research methods and techniques in the corresponding approach. There are several options for the research approach, each of which poses various strengths and weaknesses. However, the research problem itself may direct an appropriate research methodology as the research approach. (Kananen, 2017)

For example, the exploratory qualitative research explores the meanings of people's experiences or cases. Data collected in the qualitative research is often in the form people's words, e.g., interviews of involved people. The target of qualitative research can either be finding the truth about a specific research topic or the interpretation of the topic. In other words, the aim of the qualitative research is to improve the understanding through collection and analyzing the data to reflect the findings to the known knowledge to move the researcher's comprehension closer to the answers to the research questions. (Aspers & Corte, 2019.)

Regarding the research strategy, action research can be defined as "*an approach in which the action researcher and a client collaborate in the diagnosis of the problem and in the development of a solution based on the diagnosis*" (Bryman

& Bell, 2011). Thus, one of the main characteristics of the action research is the collaboration between the researcher and members of the organization to solve the organizational problem specified in the research. In this thesis, therefore, the research approach is action research, which focuses on tackling a specific business problem in the case company. The study is expected to approximately continue over the span of a year (January 2021 – November 2021). In this thesis, qualitative research methods were selected for collecting the data.

2.2 Research Design

Figure 1 illustrates the research design in this study.

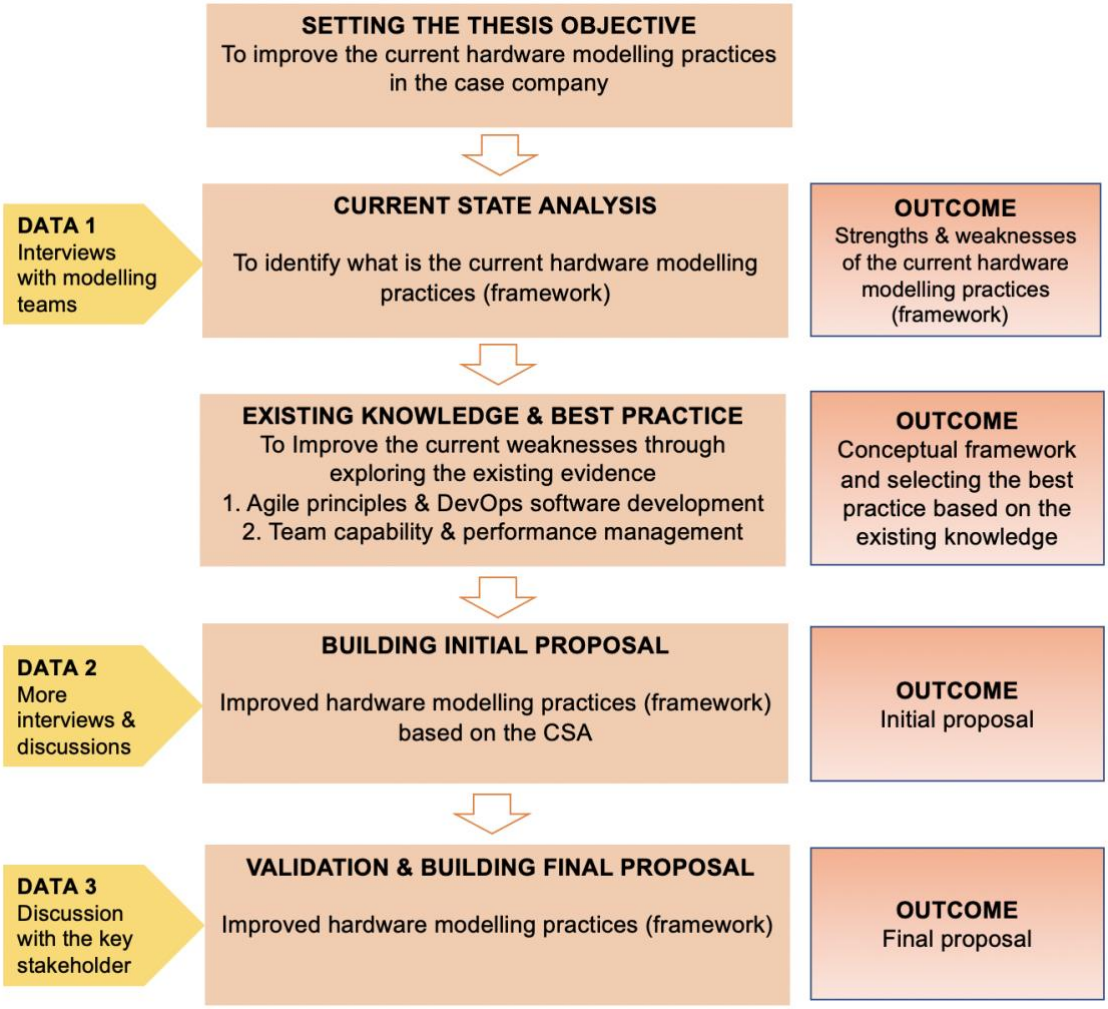

Figure 1. Research design.

As seen from Figure 1, the thesis starts by setting the objective. After that, the next step is the current state analysis. This step concentrates on studying the current practices and identifying the strengths and weaknesses in the scope of the existing business problem, i.e., the current workflow of the hardware modelling. It is followed by gaining knowledge through available literature and best practice, upon which a conceptual framework is created, which later yields to an initial proposal. Finally, validation of this initial proposal takes place through discussions with key stakeholders to ensure that the proposal renders an improved modelling framework in terms of the product delivery indicators.

2.3 Data Collection and Analysis

The data for this study is collected in three stages as shown in Figure 2.

	Content	Type	Timeline	Outcome
Data 1: **Current state analysis**	Interviews with: • Modelling engineers	Online meetings	Feb. 2021 - Apr. 2021	Identification of strengths and weakness in the current framework
Data 2: **Proposal building**	Interviews with: • Modelling leads • Project managers	Online meetings	May 2021 - Aug. 2021	Initial proposal
Data 3: **Validation**	Discussion with • Quality Manager	Online meetings	Sep. 2021 - Oct. 2021	Validation & final proposal

Figure 2. Sequence of three data collection stages in this thesis.

As seen from Figure 2, in the first round (Data 1), multiple stakeholders involved in the modelling process are interviewed to obtain a better understanding of the current practices. Permission to conduct the interview was requested from interviewees and managers, and if granted, an online meeting session was set up. Each interview mainly consisted of a free-format discussion to discuss the general-level matters, e.g., (but not limited to) the person's perception of the

current framework to follow in the hardware modelling tasks, highlights of such practices, and possible weaknesses of the current format.

Answers were recorded in meeting notes to be later modified into a more comprehensive summary, which covered main issues brought up and discussed. To maintain a transparent and integrous process, interviewees could review the meeting summary for possible edits and updates to ensure notes indeed reflect their true beliefs and ideas.

During the second stage of data collection, the aim was to co-create the initial proposal, and to validate the effectiveness of the proposal, more interviews with stakeholders were conducted to receive their feedbacks on the feasibility of the proposal, and based on these feedbacks, the proposal was further modified to build a consensual model for the next phase.

The aim of the third stage of data collection was to validate the proposal previously suggested. Note that due to the iterative nature of the proposal building, data collection in stages 2 and 3 could overlap and simultaneously coexist. Through the early validation, corresponding stakeholders were involved with the up-to-date progress to provide further feedbacks. These final stage feedbacks were collected to fine tune the proposal model and the process iteratively continued toward a validated final proposal model implementation.

2.4 Thesis Evaluation and Research Quality

In this subsection of the thesis, validity and reliability criteria of the research are discussed. The evaluation of research quality can be done based on four criteria for trustworthy qualitative research, which are credibility, dependability transferability, and confirmability (Shenton, 2004). The elements important to the evaluation of these research quality criteria for this work are summarized in Table 1.

According to Shenton (2004), creditability describes how well the research findings represent the reality, in other words, it shows the consistency of the research output with the reality of the case under the investigation. Dependability shows the research process and explicit presentation of the utilized process and methods in the research. Transferability emphasizes the replicability of the study, and confirmability ensures that the research findings are profoundly shaped by study participants than the researcher's own biases.

Table 1. Evaluation of credibility, dependability, transferability, and authenticity (Shenton, 2004).

Research quality criteria for action researchers	How to ensure high quality of research
Credibility	• **Deep involvement with the topic/organization/problem**: The researcher works in the case company and was involved in the problem of interest. • **Sufficient data collection**: Relevant data was collected in 3 phases through several multiple free-format interviews and discussions with different stakeholder involved in the team. • **Careful documentation of data & research process**: Meeting notes were documented, based on which CSA was constructed. • **Deep & reliable analysis**: The qualitative analysis was perfomed in several stages in an iterative manner, i.e., based on the discussion and received feedback, propsals were refined and validated. • **Involvement of others to reflect & discuss ideas**: Data was gathered via several rounds of interviews and proposals were co-created by the researcher and stakeholders.

Dependability	• **Producing a reliable (dependable) explanation (account) of the research focus & results**: Research focus and research design and implementation steps are explicitly described in Section 1 and Section 2 of this thesis, respectively. • **Demonstrating clarity & consistency of interpretation**: All research design steps are clearly described throught the thesis text. • **Other means to prove the quality of findings**: The project and effectiveness of the utulized methods are discussed in the last section of the thesis.
Transferability	• **Building & explicit reporting of research design, objectives:** Research focus and research design and implementation steps are explicitly described in Section 1 and Section 2 of this thesis, respectively. • **Giving enough picture of the context:** The research context are thoroughly described throught the thesis. • **Explicitly reporting on the findings and interpretations**: The findings and corresponding interpretations and analyses are reported in Section 3 and Section 5 of this thesis, respectively.
Confirmability	• **Avoiding researcher's bias**: Although the researcher works in the case company, his opinions were not reflected in gathered data and proposals were co-creaceted based on the feedback from different stakeholders, where they were allowed to freely express their opinions related to the research focus.

3 Current State Analysis

This section reports on the results of the current state analysis (CSA) of the hardware modelling responsibility within the project team. The section includes the description and analysis of the current software development framework and practices around it, and it ends with identifying the strengths and weaknesses of the current practices.

3.1 Overview of the Current State Analysis

In this part of the thesis, the current state analysis (CSA) of the hardware modelling framework within the project team is carried out. The focus is on the current software development framework and practices, which leads to identifying the potential strengths and weaknesses of the current state.

The goal of the CSA is to understand and report the way of working and practices in the hardware modelling team. The analysis is based on interviews (first data collection phase) with several modelling engineers, which form the basis of the Data 1. In the remainder of this section, job responsibilities are shortly introduced and findings and conclusions from the interview process are shown. Lastly, key findings of the current state analysis are summarized.

3.2 Hardware Modelling in the Case Company

A development team consists of several smaller groups, each of which are responsible for different aspects of the hardware design, model, and verification. The basic responsibility of the hardware (integrated circuit) modelling team is to produce a software code that models (emulates) the behavior of the actual hardware in scope. This is crucial due to the fact that with such a software piece, extensive tests and system simulations can be applied to ensure that the final integrated product meets the required target performance, and all performance degradation risks are eliminated. The modelling source code is developed and shared with other teams, whose responsibilities are to further test and match the

performance with the design criteria. Naturally as a software development team, the organization and the working framework have been constructed to follow the Agile methodology and DevOps practices, however as it was evident in findings of the next subsection, the framework needs improvement to align with the core components of the best practices in an Agile software development environment.

3.3 Analysis and Key Findings from the Current State Analysis

Table 2 summarizes the key findings of CSA, which are gathered from feedbacks of conducted interviews.

On the strength side, involved members feel generally positive and empowered about the recognition of their efforts and contributions as an important part of the whole process in creating a very complex system, which has potential to revolutionize the industry. Furthermore, abundant recourses of learning materials along with support of knowledgeable professionals make the task easier to tackle, if there are some issues and topics, where the responsible person has not enough knowledge on. Furthermore, due to the regular daily and weekly technical and status update meetings, involved team members are well informed about the potential problems and challenges ahead along with possible remedies.

On the weakness side, the alignment of the team structure with core Agile settings, and lack of full continuous integration and continuous delivery (CI/CD) practices and tools, as one of the best features of the Agile software development (Sacolick, 2020), is major concerns of professionals, who took part in interviews. A weak communication link toward other teams, e.g., modelling team clients or other teams which modelling activities depends on their input, was also highlighted as another area to be improved.

Furthermore, it is identified that some team members are not initially well-equipped with the required tools and services, e.g., programming language proficiency or the theoretical background of the problem. Lack of technical

capabilities, competence, and performance lead to a wider learning curve, which may result in expanding the workload of other members, or even project delays.

Overall, weaknesses can be categorized into two groups, the first one mainly concerns the Agile methodology and best software development practices in such environment. (Association for Project Management [APM], n.d.; Hüttermann, 2012; Chowdhury, 2019.) The Second one concerns the capability and performance management of the team (Stange, 2020; Darino et al., 2019; Mosley, 2021; Williscroft & Uddeen, 2021).

Table 2. Key findings in the current hardware modelling framework.

Strengths	• Hardware modelling has become a vital part of the process, so does the modelling team. • Abundant learning materials and helpful experts • Good communication and learning practices within the team
Weaknesses	• The team structure does not fully align with the Agile recommendations. • CI/CD pipeline is not fully realized, e.g., proper version controlling, different testing platforms, code review and methodology among different projects, and inter-team syncs. • Additionally, technical capability, competence, and performance of some personnel are not aligned with necessary requirements for the role, which leads to a wider learning curve, thus expensive delays.

In the next section, related knowledge and best practices of the state-of-the-art related to the current weaknesses and challenges arisen from the CSA are reviewed, and the proposal in Section 5 is built around these issues.

4 Existing Knowledge

This section of the thesis explores existing knowledge in the areas of the Agile software development and its best practical framework. Also, this section discusses ways of improving team's capability and performance in the corresponding field. The choice of topics in Section 4 is informed by the findings of current weaknesses and challenges in the case company (identified in Section 3), and this section selects and presents the relevant knowledge and state-of-the-art practices to address these identified weaknesses and consequently build the proposal (in Section 5).

4.1 Software Development

Software development evolution over the past decades has not been an easy and smooth journey. There has been enormous amount of learning and retrospective experiences. From the early ages of computation and programming, it was foreseen that that the best the way to produce high quality software was to prevent potential bugs and error from happening in the first place instead of later investing heavily in various resources on the debugging process, which is still a key feature of today's test-driven development. (Poppendieck & Poppendieck, 2009) Later, it became clear that a complex software process should be developed in different stages, each of which having a unique responsibility. Each component should have also an acceptance criteria and threshold to validate its output. An example of such method, the three staged processes, consisted of design, coding, and testing phases. Later, this method was further improved to give birth to the waterfall development model. (Poppendieck & Poppendieck, 2009.)

Software development has used various approaches to organize its working practices, of which one of the most popular modes has been the waterfall approach that was prevalent for many years. The Waterfall model, which is depicted in Figure 3, is a plan-driven development model, in which different

stages are followed to develop the required software. These stages are briefly explained as follows:

1. Requirement: In this stage, requirements for the to-be-developed software system are gathered through communication with involved stakeholders.

2. Design: The gathered requirement data are on both the hardware and software levels to construct the technical architecture.

3. Implementation: In this stage, the actual coding attempts occur, which outputs the application constructed according to required specifications.

4. Verification: In this cycle, the developed program is thoroughly tested, that is, various test cases are developed according to the general and corner case functionalities of the whole system. These tests usually include unit testing, acceptance testing, and integration testing.

5. Deployment: In this stage, the developed system is deployed to the target customer, and the maintenance phase starts. (TechRepublic, 2006.)

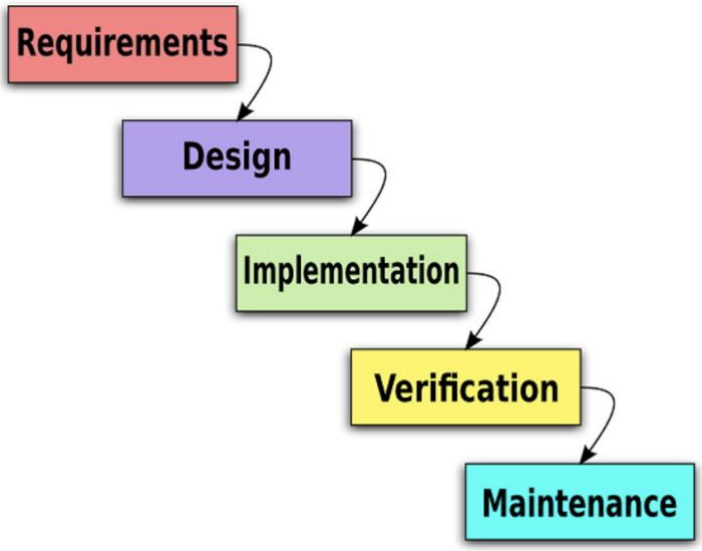

Figure 3. Schematic of the waterfall (traditional) software development method (SEODesign, 2012).

The main factor, which greatly contributes to the popularity of the waterfall model, shown in Figure 3, is its simplicity through a layered deterministic structure aiming to remove the unknowns and unpredictable incidents from the system. This alleviate the overall understanding of the developers as it is layered in different phases. (Martin, 2021.)

On the other hand, this can also be accounted as a major drawback of this model, since it does not have adequate remedies, when unpredictable issues happen in the development life cycle, which is usually the case in the real-world software development process, and thus dynamic changes along the development process are impractical to be addressed. Furthermore, the customer has basically no information about the product and does not receive the value until the very late phases of the development. (Blankenship et al., 2011.)

Contrary to the staged development process, e.g., waterfall model, the iterative development method has been established to deliver the value as soon as possible through frequent interactions with the stakeholders to address the dynamic requirements in the software development process. (Blankenship et al., 2011.)

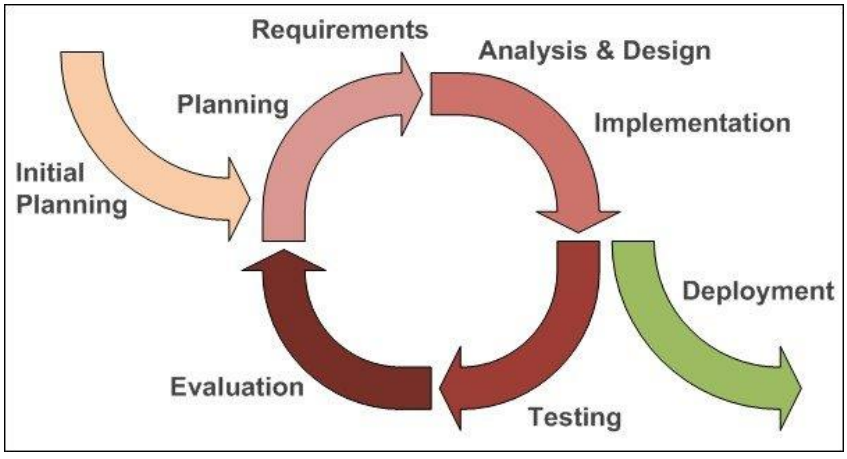

Figure 4. Schematic of the iterative model (Moniruzzaman & Hossain, 2013).

As shown in Figure 4, in the iterative model, the prior knowledge collected is fed back in successive cycles to enhance the product quality and deployment.

There have also been other alternatives developed as approached to address the drawbacks of the classic waterfall approach. One of the most popular alternatives is the Agile, which is discussed next.

4.2 Agile

Agile in a broader term, is an incremental and iterative approach in the software development, which is achieved through various development cycles. The cornerstone of Agile principles in the literature lies in the Manifesto for Agile Software Development (2001) authored by several professional software developers, in which a set of guidelines based on their own best practices and experience in software development is advocated, among which collaboration between business and development teams, enhanced communication, incremental delivery and deployment, and flexible preparation for requirement changes, are highlighted.

- *Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.*
- *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*
- *Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.*
- *Business people and developers must work together daily throughout the project.*
- *Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.*

- *Give them the environment and support they need, and trust them to get the job done.*
- *The most efficient and effective method of conveying information to and within a development team is a face-to-face conversation.*
- *Working software is the primary measure of progress. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.*
- *Continuous attention to technical excellence and good design enhances agility.*
- *Simplicity--the art of maximizing the amount of work not done--is essential.*
- *The best architectures, requirements, and designs emerge from self-organizing teams.*
- *At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.* (Beck et al., 2001.)

The main motivation of the Agile framework/methodology inception for the software development generally stems from the observation that the traditional software development process is not responding fast and well enough to the dynamic environment, i.e., frequently changing requirement from the customer side. In other words, Agile methodologies enable the dynamic adjustment to changing requirements. (Misra et al., 2012.)

Nowadays, numerous industries have adapted the Agile methodology in their way of working including various fields of technology, financial, and professional services. The main reasons that such organizations tend to adapt Agile methodology include accelerating software delivery pipeline, improving software quality, enhanced management capability in changing priorities, better productivity, and improving business and information technology alignment. (State of Agile Report, 2018.)

In the modern software development, Agile methods are usually accompanied with DevOps. There are various definitions of DevOps, but according to Hüttermann (2012)

> "*DevOps is a blend of development (representing software developers, including programmers, testers, and quality assurance personnel) and operations (representing the experts who put software into production and manage the production infrastructure, including system administrators, database administrators, and network technicians). DevOps describes practices that streamline the software delivery process, emphasizing the learning by streaming feedback from production to development and improving the cycle time (i.e., the time from inception to delivery*)." (Hüttermann, 2012, p.p. 4.)

The inception of the concept of DevOps was the recognition of the fact that the implementation of Agile methods would affect the infrastructure, on which the software development and delivery activities took place. DevOps is currently a combination of process culture meaning that the final software quality depends on every member in the development and deployment team, process automation meaning the necessary support for the development and deployment process, action measurement, and information sharing, e.g., source codes, principles. (Fitzgerald & Stol, 2015.)

Some of DevOps best practices include following the CI/CD approach, implementation of testing automation, keeping all stakeholders (development, operation, and deployment) in the information loop, and proactively monitoring the software performance. (Chowdhury, 2019.) Therefore, automation is a key ingredient in a continuous software development.

In Agile, as well as in the waterfall, the tasks are mostly handles as projects. It allows the management and teams to approach each product or service

development as a separate task and thus manage it more effectively. The project application in Agile are discussed next.

4.3 Agile Project Management

A project is a temporary and unique action, i.e., a defined beginning and end, defined scope and resources, and not a routine operation designed to accomplish a singular goal (Project Management Institute [PMI], n.d.). A project team may consist of people from different teams and organizations and across multiple locations, and the project management can be defined as the process of applying of skills, knowledge and technical tools on a project to meet the common requirement and goal.

Project management process typically consists of five cycle phases, namely initiating, planning, execution, monitoring/control, and closing, in which following areas need specific attention, integration, scope, time, cost, quality, procurement, human resources, communications, risk management, and stakeholder management. (PMI, n.d.)

In a traditional waterfall approach, the sequential process to project management, these phases coincide with the life cycle of a project. First, all the requirements of a project are defined and signed off by project stakeholders in the initiation phase. Next, the implementation of the project is designed in the planning phase. After the planning, the designed implementation is built, tested, integrated and finally set up for approval by the customer. (Dolan, 2007.)

Regardless of which project management framework is followed, Agile or traditional, they follow the core components of the project management which consist of

- Defining the necessity of the project
- Specifying project requirements, the deliverables and their quality
- Estimating project resources and timeframes

- Preparing a business case, securing stakeholder agreement and funding
- Developing a management plan
- Leading the project delivery team
- Monitoring progress
- Managing budget, communication, risks and changes in the project
- Provider management
- Closing the project (APM, n.d.).

These core components and the principles can be scaled based on different projects' complexity and significance.

As mentioned, the Agile way of working and its principles have been adopted into numerous fields and practices. Agile project management is defined similarly to the Agile software development counterpart, that is, an incremental and iterative approach to manage projects throughout its life cycle. Therefore, Agile project management basically relies on values and principles defined in the Agile manifesto. (Cervone, 2011)

Agile was included in the Project Management Body of Knowledge in 2018 recognizing the importance of this methodology in the project management practices (Ereiz & Music, 2019).

Several frameworks exist through which Agile principles can be applied including in a project, e.g., Scrum, Kanban, Lean, extreme programming, adaptive software development, and etc. The difference between these frameworks mostly comes from how Agile principles are incorporated. However, Scrum has usually been the main focus in the Agile project framework. Scrum, an acronym taken from the rugby, is a development framework that applies Agile principles to successfully implement and deliver a software project. (Grimme, 2009.) It was firstly cited on the analysis of few successful companies that were founded on the self-organized team and scalability. In addition to abiding to the Agile principles, Scrum has also a set of core principles, that is,

> "*Scrum is refreshingly simple, people-centric framework based on the values of honesty, openness, courage, respect, focus, trust, empowerment, and collaboration"* (Rubin, 2012, p.p. 13.)

Practically, in the Scrum methodology, the focus is on regular team meetings to iteratively and incrementally develop and deliver a specific product or service. A schematic of Scrum framework is depicted in Figure 5.

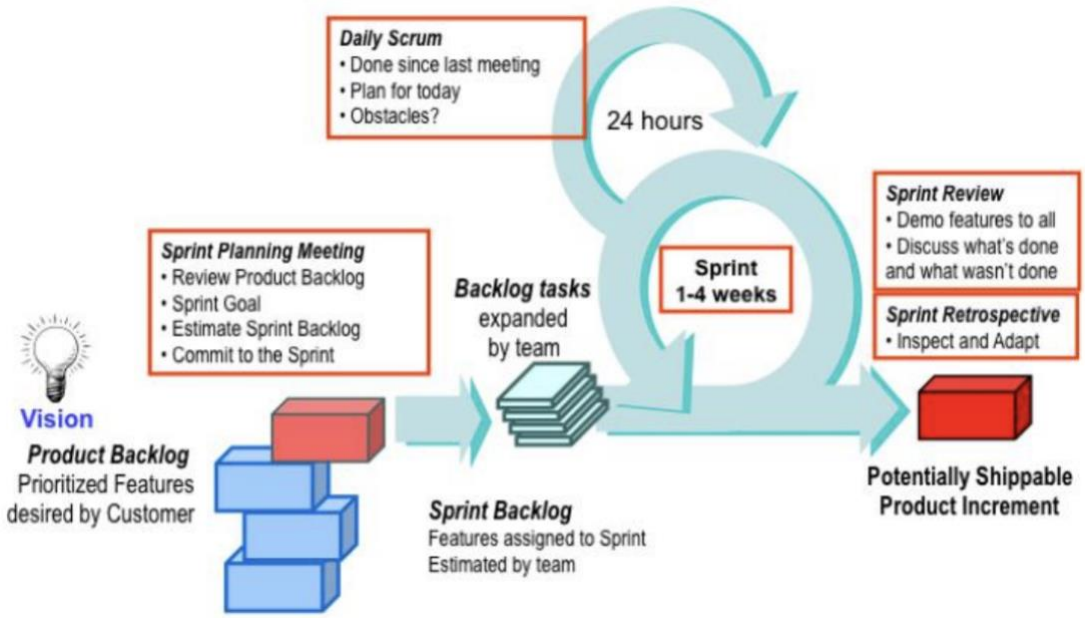

Figure 5. The original Scrum framework (Sliger, 2011).

Each Scrum team has few roles and responsibilities assigned to certain members. The team consists of Scrum master, product owner, development and testing members. Product owner is responsible for managing the product backlog, therefore it is also its duty to clarify tasks for the development team. Scrum master, which has a close meaning to a team leader, is responsible for guiding the team to better follow the Scrum practice. Its leadership style however is different from the traditional command and control, that is, it is more like lead by example. The development team is the group of professional, who are responsible for developing and delivering increments at each cycle. It is a self-

organizing team, which handles the planning, implementation, and testing together in a collective ownership of the implementation. (Sutherland & Schwaber, 2020).

In any approach, the goal of the project team is to reach certain predefined objectives. Following the progress on the road to reaching these objectives is often considered as part of performance management that helps to define, support and trace the team's or project's performance. To achieve high levels of performance, teams critically need certain capabilities, which is discussed next under the topic of capability and preference management.

4.4 Capability and Performance Management

In practice, many Agile teams are temporary, that is, professionals who work in the team for a single project or move to different projects on a rotational basis. However, the same fundamental team forming principles, Tuckman's team development model depicted in Figure 6 (forming, storming, norming, and performing) may still be applicable (Tuckman,1965). However, in these temporary teams, minimizing distractions via improving the individual's capability and performance is crucial for the team's success.

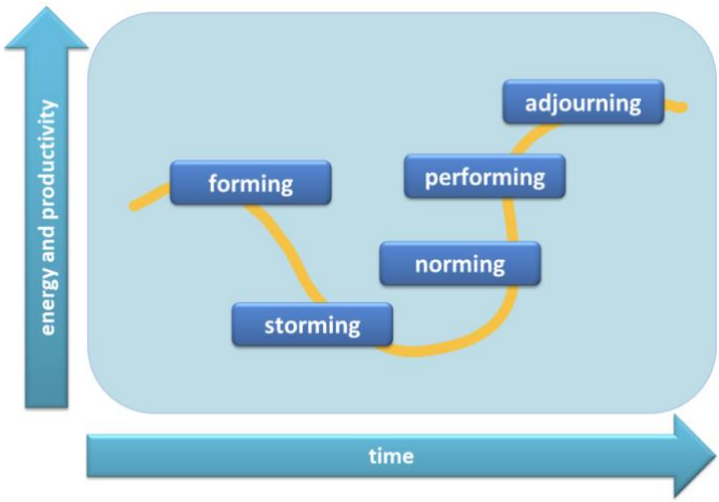

Figure 6. Tuckman's team development model (Wilson, 2020).

Performance management is a critical capability for organizations, however it is not necessarily well defined or integrated. It basically aims at answering the question how the performance of individuals, teams, processes, information, and materials contribute to the organization's strategic goals and key performance indicators (KPIs). In other words, its goal is to illustrate a clear understanding of current performance, target levels, and the gap between.

Thus, performance management can be thought of an ongoing communication process, where managers help employees by clarifying expectations, setting goals, providing feedback, reviewing results, and determining developmental opportunities and compensations. (Stange, 2020.)

Therefore, identification of domains of interest and corresponding actions is required to move towards the target levels in the shortest possible time period considering the current available resources. Therefore, a continuous awareness and assessments of the progress is crucial. To this end, management by objectives, a sequential and hierarchical approach, is the most used process, however agility has been evolved to be integrated to its features. (Kavanagh, 2017). The hierarchical model for management by objectives method is illustrated in Figure 7.

Figure 7. Schematic of the hierarchical model (Kavanagh, 2017).

It is found out that organizations, in which employee goals to business priorities are linked, managers' capabilities are invested in, and rewards for the extreme performance are set, are 84% more likely to have performance management approaches that their employees perceive as fair. (Darino et al., 2019).

As mentioned, Agile organizations, which are made up of a network of various people-centric teams with rapid learning and fast decision cycles to eventually co-create value for all stakeholders, take stability and dynamism into the account. Therefore, Agile models allow for quick and efficient reconfigurations of strategy, structure, processes, people, and technology toward value-creating opportunities. In this scope, linking employees' goals to business priorities and maintaining a strong element of flexibility are core elements in such ways of working. It is also significant, if employees feel sense of meaning and purpose in their work. On the other hand, it may rise to challenges of how the strong emphasis on individual goals can be integrated in the autonomous teams featured with agility. According to Darino et al. (2019), there are three approaches

to help Agile organizations to accordingly adapt and ensure that goals remain meaningful and linked to business priorities, which are listed in Table 3 below.

Table 3. Three approaches to help Agile organizations to adapt (Darino et al., 2019).

- Linking goals to business priorities
  - Introduce team objectives in addition to (or instead of) individual targets
  - Set objectives as a team, discuss results frequently, pivot as required
  - Create transparency of targets and performance
- Investing in managers' coaching skills
  - Clarify the roles that leaders play in development and evaluation
  - Focus on continuous feedback and ongoing development conversations
  - Frequently collect input from multiple sources when evaluating performance
- Differentiating consequences
  - Differentiate individual contribution to team performance based on desired values, mindsets, and behaviors
  - Increase the emphasis on intrinsic motivation and nonmonetary rewards.

In the Agile performance management, in contrast to the traditional approach, in which goals were set in the beginning and the employer's work throughout the year was aligned to achieve them, and at the end of the year, performance review session was held to specify the bonus based on the corresponding performance, frequent measurement, continuous performance feedback and incremental improvement, are the key elements. Continuous communication is crucial as it

develops meaningful relationships through ongoing dialogues between managers and staff and highlights potential problems and challenges ahead before they can negatively affect the productivity in the later stages. (Williscroft & Uddeen, 2021.)

Overall, Agile performance management consists of four components, continuous learning, frequent check-ins, building trust, and connection to the work community (Mosley, 2021).

Figure 8 summarizes some aspects of transformational elements in the Agile performance management.

The Transformation of Performance Management		
Traditional Approach		**New Approach**
Annual Reviews	→	Continuous 1:1 Conversations
Individual Meritocracy	→	Team-Based Goals
Pay for Performance	→	Market-Based Pay
Ratings and Rankings	→	Progress Feedback
Manage by Objectives	→	Shared Goal Setting
Individual Goals	→	Team and Organizationally Aligned Goals
Rigid and Inflexible	→	Agile and Evolving
Improvement Plans	→	Feedback and Coaching
Performance-Focused	→	Engagement-Focused

Figure 8. Transformation of performance management (Stange, 2020).

Considering these key elements of the Agile approach directs the construction of the conceptual framework for next steps in conducting this thesis.

4.5 Conceptual Framework of This Thesis

The basis of existing knowledge exploration and information gathering was to build a foundation for the initial proposal co-creation with the corresponding

stakeholder to improve the current way of working in the hardware modelling team within the case company. Therefore, the aim was to build a connection between the relevant knowledge and best practices and the objective of this thesis study.

The conceptual framework addresses two focus areas for improvement identified in this study, that is, first, identification of the key elements of *the Agile software development framework* to improve the current hardware modelling practices in the case organization; and second, utilization of best practices of improving the capability and performance management in order to alleviate the weaknesses and challenges currently present in the team.

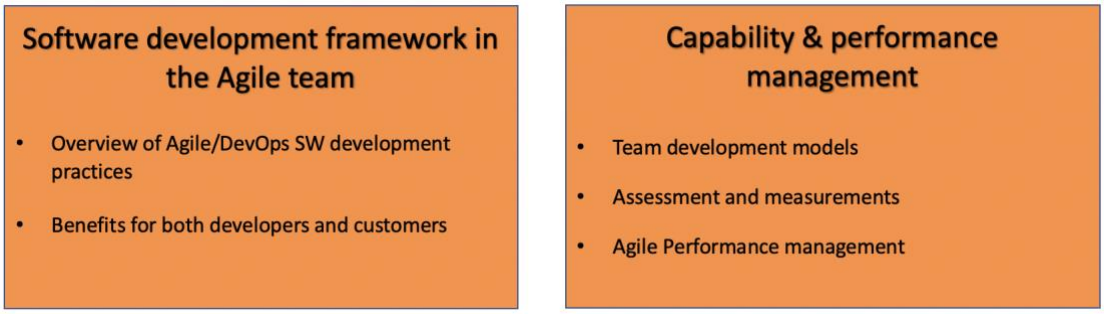

Figure 9. Conceptual framework of this thesis.

Based on key results of the CSA, specifically weaknesses of the current framework, two distinct domains were identified as the main scope for the existing knowledge exploration. The first one mainly focused on exploring DevOps practices in the Agile software development team, and the second area mainly dived into the performance and capability management in the Agile team. Figure 9 shows the conceptual framework, which aims to build a connection between the relevant knowledge and best practices and the objective of this thesis study.

The next chapter describes the proposal based on the findings of the current state analysis, selected elements of state-of-the-art knowledge and best practices, and co-creation with the case organization's team.

5 Proposal Building of the Framework for Hardware Modelling Practices

This section of the thesis presents the proposal co-creation process for improving the current hardware modelling in the target team within the case company. The proposal was based on the key findings from the current state analysis, the available literature in the selected areas, and the data collected from the second round of interviews (Data 2). The section first gives an overview on the proposal building, then continues with the corresponding propositions, and finally summarizes the whole process in the last subsection.

5.1 Overview of the Proposal Building Stage

The objective of this thesis was to propose ways to improve the hardware modelling framework and related software development practices in the case organization. It is important to note that during the time of writing this thesis, efforts to enhance such practices were also progressing in the case company, and therefore the proposal is based on the information available at the time of writing this thesis.

In order to build a proposal for improving the hardware modelling framework and related software development practices at the case organization, the following steps were taken.

First, the current state of the hardware modelling in the case company was investigated for establishing a clear understanding of the current practices and their state. These findings then highlighted relevant areas to be improved, which informed the search for best practice from the literature, after which, interviews with relevant stakeholders were held in the Data 2 phase to formulate the proposal outline. Figure 10 depicts the logic behind the proposal co-creation.

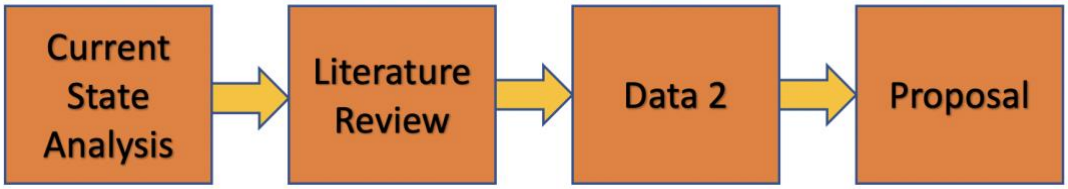

Figure 10. The logic of the proposal building.

Based on findings from the previous section, an initial proposal with two distinct set of focal points was drafted, one focusing on the Agile framework and relevant DevOps practices, and the other one concentrating on the enhanced performance management within the team.

Second, based on the findings from the current state analysis in the Data 1 phase along with points from the exiting knowledge, the team reviewed and discussed these inputs in the informal meetings with relevant stakeholders (the Data 2 phase). It confirmed consensus in the team about the shortcomings of the current framework, and clarified what kind of improvement target should be set to establish a better framework.

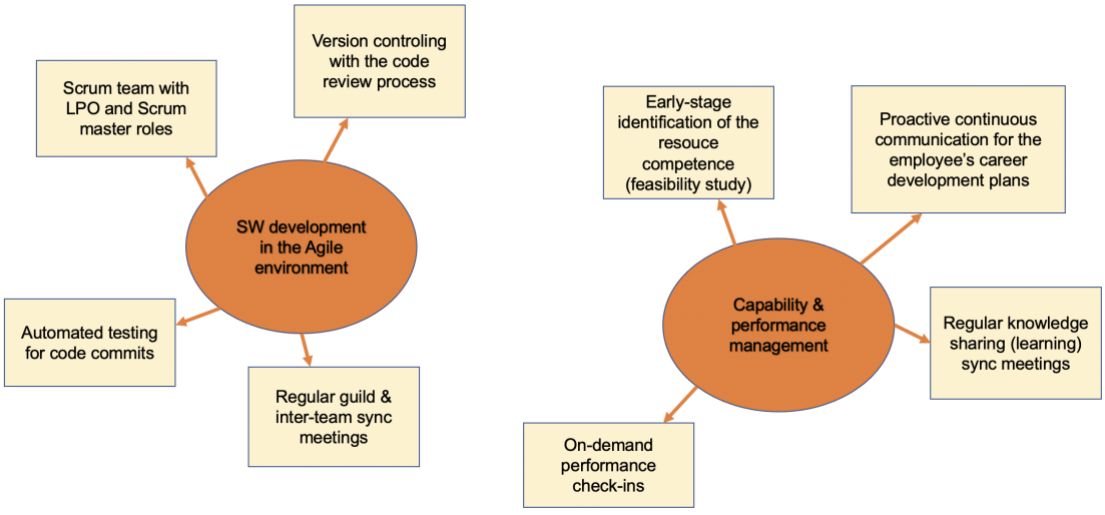

Figure 11. The initial proposal.

Third, it has led to an iterative proposal co-creation process based on received feedbacks. In other words, the team continued discussions until the draft of the proposal was formulated. The draft of the proposal is presented in Figure 11.

5.2 Inputs to the Proposal, Element 1: Software Development in Agile Environment

Based on the CSA results, a major weakness to address was related to the alignment of the current team structure to the Agile setup and also implementation of proper CI/CD tools. To address this challenge, a few proposal points were created, which are presented below.

- **Scrum team**: In the current framework, Scrum master and the team's product owner (local) are not distinguished, that is, one person acts as the project manager while no local product owner, neither a Scrum master role exist. The project manager should oversee the project to ensure the timeliness of the project delivery, which requires coordinating complex work across different teams and dependencies. The product owner on the other hand, supports the development team by prioritizing and managing the product backlog. A product owner acts as an internal customer expert for the development team to clarify requirements. (Haaff, 2019.) Therefore, a clear distinction is observed between these two roles.

  Scrum master is the crucial part of the Scrum as one of the most accepted forms of Agile. A Scrum master is responsible for teaching the Scrum theory, practices, rules, and its values, coaching, and mentoring the team. However, it is fairly a common practice in companies to regard the role of the Scrum master as an additional work and to assign it to the project manager. However, specifically when team members are not very experienced and are not fully familiar with Scrum practices, a Scrum master is a requirement for the team's success. (Ereiz & Music, 2019.) It is also worth mentioning that there are other Agile methodologies, in which

a Scrum master role is not required, however a Scrum project without a dedicated Scrum master is more likely to fail. (Ereiz & Music, 2019.)

- **Automated testing**: The major benefit of the testing automation is decreasing the amount of human intervention in the software testing process. Utilizing the DevOps test automation, bugs in the code can be detected in shorter periods of time, which potentially renders a shorter development and deployment, while ensuring the minimization of human error in the final product meaning more reliability. DevOps makes testing a shared responsibility of the team, while test automation helps developers to deliver code changes faster with higher confidence in quality. (Hristov, 2021.)

- **Version controlling with the code review process**: Version control (source control), is the practice of tracking and managing changes to software code over time. Version control software keeps track of every change and modification made to the code in a database, therefore developers can compare earlier versions of the code to help fix possible mistakes, while minimizing disruption to the ongoing code development activities of other team members. It is crucial for DevOps teams as they help them to reduce development time and increase successful deployments. (Atlassian, 2021.) In the code review process, when a developer completes a task, another developer looks over the code to make sure that the requirements are fully implemented, there is no obvious logic error, the code is aligned with existing guidelines, and the test coverage is acceptable for that specific application. In addition to the immediate benefit of increasing the reliability of the product, this decentralized task indeed enables the knowledge sharing across members. (Radigan, 2021.)

- **Inter-team sync meetings**: A noticeable issue that was found in the CSA was the lack of a regular communication link, e.g., regular sync meetings

between the modelling team and its customers and vendors. Modelling activities are based on several inputs from other teams and the code delivered serves as a reference to other teams. Therefore, establishing regular sync meetings with different stakeholder teams within the scope of the hardware modelling activities from the early stages of the project was proposed during the co-creation process, which can improve the mutual understanding of the members involved, which in return can reduce the delivery time as everyone establishes a common conception of what they are expected to deliver and receive.

5.3 Inputs to the Proposal, Element 2: Agile Capability and Performance Management

Another category of weaknesses, which was found during the CSA and addressed in the proposal building, was capability and performance management in the Agile development team. A few proposal points were also created to address these issues.

- **Early-stage identification of resource competence**: Competency is referred to the combination of knowledge, skills, abilities, and personal attributes contributing to the employee's performance to ultimately render the organizational success (Viswanathan, 2019). Therefore, identifying and mapping the required competencies to operate effectively in a specific job position is fundamentally important. This is even more crucial, if this identification is performed in the very early stages of the project (feasibility study period) to ensure the enough available time for learning and competency development for the specific project.

- **Proactive continuous communication for the employee's career development plans**: As it was already highlighted in Section 4, continuous communication between the managers and staff is a key element in the Agile performance management (Stange, 2020). These interactions require trusted two-way connections. This is the opposite of

the traditional approach, in which goals, performance, and development plans are reviewed and discussed in the beginning and the end of the year and frequent and continuous feedbacks are the main differentiations.

- **On-demand performance check-ins**: As described in the previous section, frequent performance check-ins are one of the main components of the Agile performance management. In fact, a Harvard study found that frequent check-ins with managers could improve the performance (Mosley, 2021).

- **Regular knowledge sharing (learning) sync meetings**: As found in the CSA, one of major strengths of the current framework was availability of abundant learning material and helpful experts. Thus, it can be readily utilized to overcome the possible lack of expertise or skill in modelling team members. To this end, organizing knowledge sharing session on specific topics of interest is very practical and useful. However, a successful implementation of knowledge sharing between Agile teams requires three elements, which have to be facilitated. Namely, the adoption of practices (collective meetings, pair programming across teams and projects, technical presentations, etc.), organisational support (which includes strategy, structure, culture, environment, and leadership support), and appropriate stimuli (for instance common goals or incentives). (Kuusinen et al., 2017.)

5.4 Draft of the Proposal

Proposal co-creation was an interactive process, which included, in addition to the thesis researcher, several other stakeholders. The process started by reviewing the findings of the CSA and the corresponding existing knowledge and relevant evidence of best practices during the Data 2 stage.

Table 4 summarizes the proposed improvements to (a) the current practices of software development in an Agile environment (framework), and (b) the current capability and performance management.

Table 4. Initial proposal of the framework for hardware modelling practices at the case organization.

Initial proposal of the framework for the improved hardware modelling practices at the case organization	
A. Improvements to the practices in software development in Agile environment	**B. Improvements related to capability and performance management in Agile environment**
1. Scrum team. The Scrum master and the team's product owner (local) should be distinguished from the project manager. • **The project manager** should oversee the project to ensure the timeliness of the project delivery and coordinates complex work across different teams and dependencies. • **The product owner** should support the development team by prioritizing and managing the product backlog, acts as an internal customer expert.	**1. Early-stage identification of resource competence.** The team needs to know very early (ideally, at the feasibility study period) about the knowledge, skills, abilities, and attributes of members contributing to the project. • **If competency is missing**, the team needs time for learning and competence development for that specific project.

• **The Scrum master** should be responsible for coaching and mentoring the team.	
2. Automated testing. To decrease the amount of human intervention in the software testing process, the DevOps test automation should be utilized for detecting bugs in the code detected in short period of time. • DevOps makes testing a shared responsibility of the team, thus automated testing platform is in everybody´s interest. It helps developers to deliver code changes faster with higher confidence in quality.	**2. Proactive continuous communication for the employee's career development plans**. Such communication is a key element in the Agile performance management. • Interactions need to be trusted **two-way connections**. This is the opposite to the traditional approach, in which goals, performance, and development plans are usually reviewed and discussed in the beginning and the end of the year. • **Frequent and continuous feedback** is one of the main differentiations in Agile performance management.
3. Version controlling with the code review process. Version control means tracking and managing changes to the software code **over time** and code review process significantly increases the code **reliability**.	**3. On-demand performance check-ins.** Frequent performance check-ins are one of the main components of the Agile performance management.

• **Version control software** keeps track of every change and modification made to the code in a database. Thus, developers can **compare earlier versions** of the code to help fix possible mistakes and **minimize disruption** to the ongoing activities of other team members. It significantly **reduces the development time** and **increases quality**. • **The code review process increases the reliability** of the end product, and enables **knowledge sharing across members.**	• These **frequent check-ins** should be done on demand by the employee **with managers.** • **Frequent check-ins** aim to **improve the performance**, which is supported by academic studies.
4. Inter-team sync meetings. Currently, lack of regular communication (e.g., regular sync meetings) between the modelling team and its customers and vendors reflects in terms of decreased quality and increased time for development. • **Modelling activities** are based on **inputs from other teams** and the code delivered serves as a **reference to other teams**. • Thus, it is a "must" to establish **regular sync meetings** with	**4. Regular knowledge sharing (learning) meetings.** One major strength of the current practice is availability of abundant learning material and helpful experts. • **Existing learning materials** in the case organization can be utilized to overcome the lack of expertise or skills in team members. • In addition, the team can **organize knowledge sharing session on specific topics**.

different stakeholder teams within the scope of the hardware modelling activities **from the early stages** of the project. It is needed for **mutual understanding** of the members involved. In return, it can reduce the delivery time as everyone establishes a common conception of what they are expected to deliver and receive.	• Best practice in Agile prove that successful knowledge sharing between Agile teams requires three elements: (a) **adoption of practices** (collective meetings, pair programming across teams and projects, technical presentations, etc.), (b) **organisational support** (which includes strategy, structure, culture, environment, and leadership support), and (c) **appropriate stimuli** (for instance common goals or incentives).

The proposed improvements merge into the framework for the hardware modelling practices at the case organization.

As seen from Table 4, several important proposals were made to address specific challenges and weaknesses of the current framework for the hardware modelling practices under two distinct categories that were identified during the CSA, the practices for software development, and the improvements related to the capability and performance management. When merged, they form the framework of improved practices for the hardware modelling at the case organization.

6 Validation of the Proposal

This section reports on the results of the validation of the proposal, which aims to improve the current framework in the hardware modelling practices within the case organization.

6.1 Overview of the Validation Stage

The initial proposal was validated in two ways: first, through internal development that took part in the course of the thesis, when the proposed improvements were implemented without waiting for the Final version of the proposal; and second, through an expert discussion with a key stakeholder and collecting feedback.

First, many of the points included in the initial proposal were immediately addressed after the interviews and discussions for the current state analysis along with the tribe's internal measures. These changes to the initial proposal that were supported and directly implemented are discussed next.

Second, after completion of the initial proposal, which was co-created during the Data 2 phase, and to ensure that the proposed solutions and points meet the company's requirements and higher-level agenda, a stakeholder session for the proposal validation was organized.

During this discussion (Data 3 phase), first, a summary of the CSA and corresponding existing knowledge and best practices in those specific area were presented for a better comprehension of the proposal context. Then, the draft of the proposal (shown in Figure 11) was presented to the key stakeholder, the Quality Manager of the tribe.

6.2 Inputs from Validation

During the course of the thesis and co-creation process, many challenges and shortcomings of the existing framework in the modelling teams and their practices

identified in the current state analysis have been recognized and confirmed also through the tribe's internal review process and project quality assessments. Moreover, several of the proposal points from the Initial proposal in Section 5, which aimed to alleviate the weaknesses, got implemented without waiting for the final proposal, or appropriate actions in that regard were initiated.

Namely, **automated testing** platform was launched, an advanced **version controlling software** along with **code review process** was put into place, and **regular guild meetings and inter-team sync** sessions were scheduled. On the capability and performance category, **early-stage feasibility studies** and also **regular domain knowledge sharing** session were also scheduled.

These developments confirmed in practice the reliability of findings in the current state analysis and stressed the team involvement in the internal improvement of the Agile practices.

Second, based on the validation discussion with the Quality Manager, the scope of the further necessary changes to the initial proposal was specified, considering both the internal development that happened in the teams during the course of the Thesis, as well as further development needs. Both are summarized below as they mark the changes to the Initial proposal.

The elements of the changes to the Initial proposal are shown in Figure 12, where implemented proposals are highlighted in green.

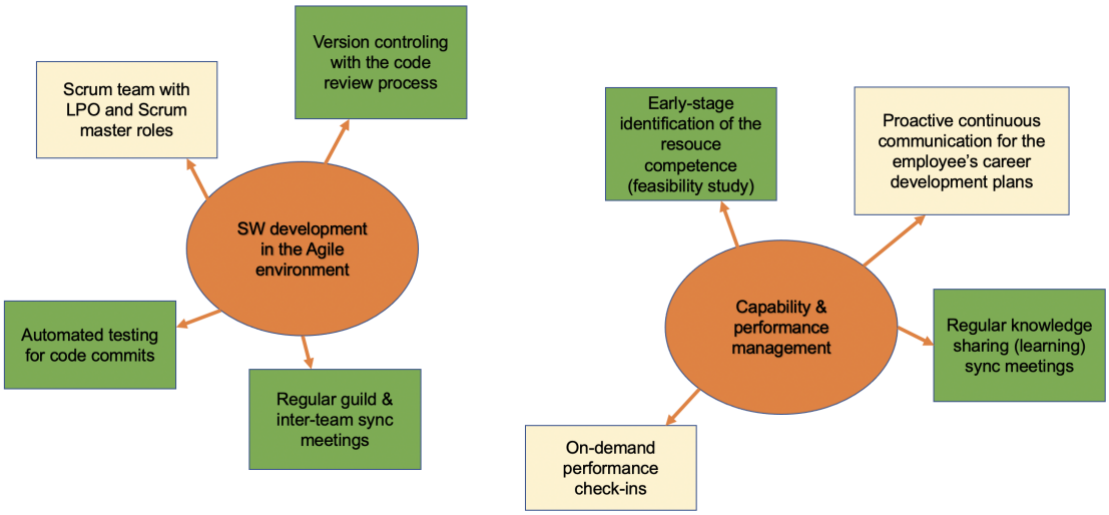

Figure 12. The validated proposal (green boxes show the proposals that were addressed and implemented).

In the validation discussion, the validity of other proposal and their potentials to improve the current framework were also acknowledged, but it was stressed that their implementation would require higher-level discussions and approvals.

6.3 Final Proposal after Validation

The proposal was validated by pro-active implementation of some selected key proposals, as well as collecting feedback from the key stakeholder to ensure about the validity and applicability of the proposals, and also their alignment with the tribe's requirements and vision.

Table 5 shows the final proposal that includes both planned and implemented improvement proposals to present the full framework for the future reference.

Table 5. Final proposal of the framework for improved hardware modelling practices at the case organization.

Final proposal of the framework for the improved hardware modelling practices at the case organization	
A. Improvements to the practices in software development in Agile environment	**B. Improvements related to capability and performance management in Agile environment**
1. Scrum team. The Scrum master and the team's product owner (local) should be distinguished from the project manager. • **The project manager** should oversee the project to ensure the timeliness of the project delivery and coordinates complex work across different teams and dependencies. • **The product owner** should support the development team by prioritizing and managing the product backlog, acts as an internal customer expert.	**1. Early-stage identification of resource competence.** The team needs to know very early (ideally, at the feasibility study period) about the knowledge, skills, abilities, and attributes of members contributing to the project. • **If competency is missing**, the team needs time for learning and competence development for that specific project.

• **The Scrum master** should be responsible for coaching and mentoring the team.	
2. Automated testing. To decrease the amount of human intervention in the software testing process, the DevOps test automation should be utilized for detecting bugs in the code detected in short period of time. • DevOps makes testing a shared responsibility of the team, thus automated testing platform is in everybody´s interest. It helps developers to deliver code changes faster with higher confidence in quality.	**2. Proactive continuous communication for the employee's career development plans**. Such communication is a key element in the Agile performance management. • Interactions need to be trusted **two-way connections**. This is the opposite to the traditional approach, in which goals, performance, and development plans are usually reviewed and discussed in the beginning and the end of the year. • **Frequent and continuous feedback** is one of the main differentiations in Agile performance management.
3. Version controlling with the code review process. Version control means tracking and managing changes to the software code **over time** and code review process significantly increases the code **reliability**.	**3. On-demand performance check-ins.** Frequent performance check-ins are one of the main components of the Agile performance management.

• **Version control software** keeps track of every change and modification made to the code in a database. Thus, developers can **compare earlier versions** of the code to help fix possible mistakes and **minimize disruption** to the ongoing activities of other team members. It significantly **reduces the development time** and **increases quality**. • **The code review process increases the reliability** of the end product, and enables **knowledge sharing across members.**	• These **frequent check-ins** should be done on demand by the employee **with managers.** • **Frequent check-ins** aim to **improve the performance**, which is supported by academic studies.
4. Inter-team sync meetings. Currently, lack of regular communication (e.g., regular sync meetings) between the modelling team and its customers and vendors reflects in terms of decreased quality and increased time for development. • **Modelling activities** are based on **inputs from other teams** and the code delivered serves as a **reference to other teams**. • Thus, it is a "must" to establish **regular sync meetings** with	**4. Regular knowledge sharing (learning) meetings.** One major strength of the current practice is availability of abundant learning material and helpful experts. • **Existing learning materials** in the case organization can be utilized to overcome the lack of expertise or skills in team members. • In addition, the team can **organize knowledge sharing session on specific topics**.

different stakeholder teams within the scope of the hardware modelling activities **from the early stages** of the project. It is needed for **mutual understanding** of the members involved. In return, it can reduce the delivery time as everyone establishes a common conception of what they are expected to deliver and receive.	• Best practice in Agile prove that successful knowledge sharing between Agile teams requires three elements: (a) **adoption of practices** (collective meetings, pair programming across teams and projects, technical presentations, etc.), (b) **organisational support** (which includes strategy, structure, culture, environment, and leadership support), and (c) **appropriate stimuli** (for instance common goals or incentives).

As seen from Table 5, most of the weaknesses and their proposal counterparts were also identified during the internal quality checking process and actions to implement several of proposal points were started, which in fact reinforces the validity of the findings of this research meaning that weaknesses identified during the CSA and corresponding proposals were crucial to ensure maintaining a high-performing software development environment. These improvements are marked blue in Table 5. Other proposals to improve the current framework were also acknowledged, but as stressed in the validation discussion, their implementation would require higher-level discussions and approvals.

7 Discussion and Conclusion

In this section, the thesis is concluded and a summary is presented. It covers the overview of the business challenge, utilized research methods, analysis of the current state, co-creation process of the proposal to address the business challenge, and the final outcome.

7.1 Summary

The objective of this thesis was to propose an improved framework for the hardware modelling team in the case company. The outcome of this thesis consists of a set of proposals under two different categories, the proposals for the software development practices in the Agile environment and the proposals relevant to the capability and performance management at the case organization.

The thesis was carried out in seven stages. In the first stage, the thesis established the business challenge and the objective of this study. In the second stage, the research approach along with the research design were specified. The data collection plan and its time schedule were also set here. Furthermore, research quality criteria for this qualitative study were presented.

In the third stage, the current state analysis was conducted to explore the current state of the team's way of working and software development practices. This stage included the Data 1 stage of the data collection plan, in which several interviews with modelling personnel were conducted to achieve a wide perception of the current state.

In the fourth stage, existing knowledge and available best practices were explored, which were investigated based on the main areas highlighted in the current state analysis. It covered domain such as Agile methodology, project management, DevOps, capability and performance management in an Agile environment.

In the fifth stage, the initial proposal was co-created through the mixture of the information from the current state analysis, existing knowledge and best practices, and interview and discussions in the second data collection phase. The proposals included guidelines for two distinct domains. The first one proposed improved practices for the software development in an Agile team based on the DevOps best practices, and the second category proposed key actions in the Agile performance management to improve the team's performance and competency.

In the sixth stage, the initial proposal, which was partly implemented by the team at that point, was reviewed by the key stakeholder. Based on the implementation results and received feedbacks, validity of the initial proposal was acknowledged and the final proposal was created.

7.2 Evaluation

The objective of this study was to find a solution to a specific business problem in the case company through co-creating a proposal to improve the existing hardware modelling framework. The outcome of the study, the final proposal, was the initial step towards the improved way of working and a better performing team. The proposal can potentially be further developed, for instance through data collection form a larger audience from different teams working in other geographical sites and offices within the case company, to better address the challenges.

However, even the implementation of the current proposal will render major improvements in terms of the quality of the software, shorter delivery time, and high-competence personnel. Additionally, this study might also benefit the case company through the learnings and its findings that may not readily be available via the internal tools and measures.

Throughout the study, and based on the nature of the research design, several professionals and experts from the company were interviewed for the data collection. Furthermore, the proposal was co-created, in addition to the input from the literature review and evidence exploration, through consultation and received feedbacks. Besides, the objective determined in the beginning of the study align with the outcome of this thesis. Thus, all in all, the reliability of this study can be confirmed.

7.3 Future

Most of the proposals in this study were addressed and implemented, or appropriate actions to implement those are ongoing.

Therefore, for these proposals, the final step to mention here is to quantitatively measure their impact on the hardware modelling framework improvement, which can consist of, for instance, the delivery time compared to previous projects. Implementation of the remaining points in the proposal, which are not currently implemented, is another future direction for the company, if proper approvals are granted within the case company.

References

Association for Project Management [APM]. (n.d.) *What Is Agile Project Management?* Available at: <https://www.apm.org.uk/resources/find-a-resource/agile-project-management> [Accessed: 25 November 2021].

Aspers, P. & Corte, U. (2019). What is Qualitative in Qualitative Research. *Qualitative Sociology*. Vol. 42, pp. 139-160.

Atlassian. (2021) *What Is Version Control?* Available at: <https://www.atlassian.com/git/tutorials/what-is-version-control> [Accessed: 25 November 2021].

Beck K., et al. (2001). *Manifesto for Agile Software Development*. [Online]. Available at: <http://Agilemanifesto.org> [Accessed: 25 November 2021].

Blankenship J., & Bussa M., & Millett S. (2011). *Pro .NET Agile Development with Scrum*. 1st. ed. Apress.

Bryman, A. & Bell, E. (2011). *Business Research Methods*. 3rd edition, Oxford University Press

Cervone, H. F. (2011). Understanding Agile Project Management Methods Using Scrum. *OCLC Systems & Services: International digital library perspectives*, 27(1), 18-22.

Chowdhury A. R. (2019). *DevOps Best Practices: A Complete Guide*. Available at: <https://stackify.com/devops-best-practices-a-complete-guide> [Accessed: 25 November 2021].

Darino L., Sieberer M., Vos A., & Williams O. (2019). *Performance Management in Agile Organizations*. Available at: <https://www.mckinsey.com/business-functions/people-and-organizational-performance/our-insights/performance-management-in-Agile-organizations> [Accessed: 25 November 2021].

Dolan, S. (2007). Project Management for Agile Software Projects. *PMI® Global Congress. North America*. Project Management Institute.

Ereiz Z. and Music D. (2019). Scrum Without a Scrum Master. *Proceedings of IEEE International Conference on Computer Science and Educational Informatization*. pp. 325-328.

Fitzgerald, B. and Stol, K. J. (2015). Continuous Software Engineering: A Roadmap and Agenda. *The Journal of Systems and Software*, 123, pp.176-189.

Grimme A. (2009). *Rugby Vs. Development: What Is Scrum?*. Available at: <https://www.projecttimes.com/articles/rugby-vs-development-what-is-Scrum> [Accessed: 25 November 2021].

Haaff B. (2019). *Product Manager vs. Product Owner vs. Project Manager*. Available at: <https://www.aha.io/blog/product-manager-vs-product-owner-vs-project-manager> [Accessed: 25 November 2021].

Hristov A. (2021). *How Automated Testing Enables DevOps*. Available at: <https://www.atlassian.com/devops/devops-tools/test-automation> [Accessed: 25 November 2021].

Hüttermann, M. (2012). *DevOps for Developers, The Expert's Voice in Web Development*. New York. Apress.

Kananen, J. (2017). *Design Research (Applied Action Research) as Thesis Research. A Practical Guide for Thesis Research*. Publications of JAMK University of Applied Sciences.

Kavanagh D. (2017). *Performance Management Capability: Critical Success Factors?* Available at: <https://www.wispapp.com/blog/2017/04/07/performance-management-capability-critical-success-factors-1> [Accessed: 25 November 2021].

Kuusinen K., et al. (2017). Knowledge Sharing in a Large Agile Organisation: A Survey Study. *Proceedings of Agile Processes in Software Engineering and Extreme Programming*.

Martin M. (2021). *What is Waterfall Model in SDLC? Advantages & Disadvantages*. Available at: <https://www.guru99.com/what-is-sdlc-or-waterfall-model.html> [Accessed: 25 November 2021].

Misra, S., Kumar, V., Kumar, U., Fantazy, K., & Akhter, M. (2012). Agile Software Development Pactices: Evolution, Principles, and Criticisms. *International Journal of Quality & Reliability Management*, 29(9), 972–980.

Moniruzzaman A. B. M., & Hossain S. A. (2013). Comparative Study on Agile Software Development Methodologies. *arXiv:1307.3356*.

Mosley E. (2021). *This Year, Performance Management Must Become Agile*. Available at: <https://www.forbes.com/sites/ericmosley/2021/03/25/this-year-performance-management-must-become-agile/?sh=732dd12a6031> [Accessed: 25 November 2021].

Project Management Institute [PMI] (n.d.). *What is Project Management?* Available at: <https://www.pmi.org/about/learn-about-pmi/what-is-project-management> [Accessed: 25 November 2021].

Poppendieck M., & Poppendieck T. (2009). *Leading Lean Software Development: Results Are not the Point*. 1st. ed. Addison-Wesley Professional.

PR Newswire (2020). *5G Technology Market Size.* Available at: <https://www.prnewswire.com/news-releases/5g-technology-market-size-is-expected-to-reach-usd-667-90-billion-by-2026--valuates-reports-301007406.html> [Accessed: 25 November 2021].

Radigan D. (2021). *Why Code Reviews Matter (and Actually Save Time!)*. Available at: <https://www.atlassian.com/agile/software-development/code-reviews> [Accessed: 25 November 2021].

Rubin K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. 1st ed. Addison-Wesley.

Sacolick I. (2020). *What is CI/CD? Continuous Integration and Continuous Delivery Explained*. Available at: <https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html> [Accessed: 25 November 2021].

SEODesign (2012). *The Waterfall Model of Software Development*. Available at: <https://www.seowebsitedesign.com/the-waterfall-model-of-software-development> [Accessed: 25 November 2021].

Shafi M. et al., (2017). 5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice. *IEEE Journal on Selected Areas in Communications*. Vol. 35, Issue 6, pp. 1201-1221.

Shenton, A. (2004). Strategies for Ensuring Trustworthiness in Qualitative Research Projects. *Education for Information*, Vol. 22, Issue 2004, pp. 63-75.

Sliger, M. (2011). Agile Project Management with Scrum. *PMI® Global Congress*. North. Project Management Institute.

Stange J. (2020). *The Who, What, Why, and How of Performance Management*. Available at: <https://www.quantumworkplace.com/future-of-work/performance-management> [Accessed: 25 November 2021].

State of Agile Report (2018). *12th Annual State of Agile Report*. Available at: <https://www.qagile.pl/wp-content/uploads/2018/04/versionone-12th-annual-state-of-agile-report.pdf> [Accessed: 25 November 2021].

Sutherland J., & Schwaber K. (2020). *What is Scrum?*. Available at: <https://Scrumguides.org/index.html> [Accessed: 25 November 2021].

TechRepublic (2006). *Understanding the Pros and Cons of the Waterfall Model of Software Development*. Available at: <https://www.techrepublic.com/article/understanding-the-pros-and-cons-of-the-waterfall-model-of-software-development> [Accessed: 25 November 2021].

Telecoms (2020). *Nokia, Ericsson and Huawei Dominance Beginning to Fade*. Available at: <https://telecoms.com/505872/nokia-ericsson-and-huawei-dominance-beginning-to-fade-analyst> [Accessed: 25 November 2021].

Tuckman, B. W. (1965). Developmental Sequence in Small Groups. *Psychological Bulletin*, 63(6), 384–399

Viswanathan S. (2019). *Why Competency Mapping Is Important for Your Company*. <https://www.forbes.com/sites/theyec/2019/10/14/why-competency-mapping-is-important-for-your-company/?sh=6256ce996e05> [Accessed: 25 November 2021].

Williscroft P., & Uddeen. H. (2021). *Agile Performance Management: How Smart Managers Implement It*. Available at: <https://www.totaralearning.com/en/node/1633> [Accessed: 25 November 2021].

Wilson C. (2020). *Bruce Tuckman: The Forming, Storming, Norming & Performing Team Development Model*. Available at: <https://www.coachingcultureatwork.com/bruce-tuckman-team-development-model> [Accessed: 25 November 2021].