



VILNIUS UNIVERSITY
BUSINESS SCHOOL

INTERNATIONAL PROJECT MANAGEMENT PROGRAMME

Povilas Reinotas

THE FINAL MASTER'S THESIS

<i>Projektų valdymo saviorganizavimo praktikos</i>	<i>Self-organizing practices in project management</i>
--	--

Supervisor - associate. prof Eglė Dauniė

Vilnius, 2024

Summary in English

The master thesis focuses on self-organization within IT product development project teams, through the theoretical foundations of self-organization and its practical application, with the aim to bridge the gap between theory and real-world implementation of these practices. The research uses an extensive literature review and qualitative research methods, including a case study conducted through semi-structured interviews with IT product management professionals, as well as thematic analysis to delve into the practical implementations of self-organizing practices.

The empirical research on self-organization in IT project management establishes that self-organizing environments in IT projects can highly improve the adaptability and responsiveness of such teams which is crucial within this fast paced industry. Furthermore, effective communication, transparent information flow and strong interpersonal relationships are essential for the effectiveness of self-organizing teams, and the establishment of a shared vision and alignment on project goals, foster such critical aspects as autonomous decision-making. To structure these processes within these teams, the leadership is most effective when it blends aspects of adaptive, administrative and enabling styles, providing a balance between guidance and autonomy. Finally, it was established that along with the challenges of introducing Agile methodologies, and through the offering of flexibility and structured autonomy, play a significant role in providing a solid background of further development of self-organizing practices within project teams. These findings, as well as the derived concrete practices of self-organization, contribute to a deeper understanding of the dynamics of self-organization in IT product development project teams, highlighting its potential to enhance the success of IT projects, while also underscoring the necessity of customized implementation in various team environments.

The thesis contributes to both academic and practical understanding of implementing self-organization in IT project management and suggests that self-organizing practices vary across different team environments, indicating the need for further empirical research in diverse contexts and industries.

Summary in Lithuanian

Šis magistro darbas sutelkia dėmesį į saviorganizavimo praktikas IT produktų kūrimo projektų komandose, o nagrinėjant saviorganizacijos teorinius pagrindus ir jų praktinį taikymą, siekiama užpildyti spragą tarp teorijos ir šių praktikų realaus įgyvendinimo. Tyrimas apima išsamią literatūros apžvalgą ir kokybinius tyrimo metodus, įskaitant atvejo analizę, atliekamą per pusiau struktūrizuotus interviu su IT produktų projektų valdymo specialistais, taip pat teminę analizę, kurios metu gilinamasi į praktinius saviorganizacijos praktikų taikymus.

Atliktas empirinis tyrimas parodė, jog saviorganizuojanti aplinka IT projektuose gali labai pagerinti tokių komandų prisitaikomumą bei reaktyvumą, kas yra būtina šioje sparčiai kintančioje rinkoje. Be to, efektyvi komunikacija, netrikdomas informacijos srautas ir stiprūs tarpasmeniniai santykiai yra būtini saviorganizuojančių komandų veiksmingumui, o bendros vizijos sukūrimas ir susitarimas dėl projekto tikslų reikšmingai skatina tokius svarbius aspektus kaip autonomiškas sprendimų priėmimas. Šių procesų struktūrizavimui komandose itin efektyvus vadovavimo stilius, derinantis skirtingus adaptyvios, administratyvios ir įgalinančios lyderysčių aspektus, balansuojančios tarp vadovavimo ir autonomijos. Galiausiai nustatyta, kad šalia Agile metodologijų diegimo iššūkių, šios lankstumą ir struktūrizuotą autonomiją skatinančios priemonės suteikia tvirtą pagrindą tolesniam saviorganizavimo praktikų vystymuisi projektų komandose. Šios išvados, kartu su pateiktomis konkrečiomis saviorganizavimo praktikomis, prisideda prie gilesnio skirtingų saviorganizacijos aspektų supratimo bei jų įtakos sėkmei IT produktų kūrimo projektų komandų kontekste.

Apibendrinant, šis darbas prisideda prie teorinio bei praktinio saviorganizavimo praktikų supratimo ir įgyvendinimo IT projektų valdyme. Tuo pačiu būtina pažymėti, kad saviorganizavimo praktikos skiriasi priklausomai nuo komandų aplinkos, todėl tolimesni empiriniai tyrimai turėtų fokusuotis į šių praktikų įgyvendinimą įvairesniuose komandų bei rinkose.

Table of contents

Introduction.....	5
2. Roots of self-organization theory.....	7
3. Complexity theory.....	9
4. Triple loop learning.....	10
5. Application in project management.....	11
5.1 Complexity models.....	11
5.2 Importance to project management.....	14
6. The specifics of the IT product development projects.....	15
6.1. Technological complexity in IT product development projects.....	15
6.2 Uncertainty.....	17
6.3 Flexibility or/and adaptability.....	18
7. Self-organization in project management.....	20
7.1 Conditions for self-organization.....	20
7.1.2 Identity.....	20
7.1.3 Information.....	21
7.1.4 Relationships.....	22
7.1.5 Fusion of these elements.....	23
8. Leadership as another essential part of self-organization.....	24
8.1 Complexity Leadership Theory.....	25
8.2 Different Leadership styles.....	26
8.2.1 Adaptive Leadership.....	26
8.2.2 Administrative leadership.....	27
8.2.3 Enabling leadership.....	28
9. Self-organization and agile methodologies.....	30
9.1 Self-organizing practices and teams in agile methodologies.....	31
10. Research framework.....	34
11. Methodology.....	36
11.1 Qualitative methods.....	36
11.2 Case study.....	37
11.3 Source and sampling.....	38
11.4 Thematic analysis.....	40
12. Results.....	41
12.1 Adaptability and response to change.....	41
12.2 Communication dynamics.....	47
12.3 Alignment on vision and team goals.....	56
12.4 Leadership.....	61
12.5 Influence of Agile methodologies.....	67
13. Conclusions and recommendations.....	78
Bibliography and the list of reference.....	83
Annexes:.....	89

Introduction

A concept of self-organization nowadays is most commonly referred to in the discourse tied to such project management and organizational frameworks as Agile, whose methodologies which have gained huge popularity in recent years among the project and product management professionals all over the world. The idea of self-organization has also become important in various fields, helping us understand complex systems and how they work on their own. Whether we're looking at nature, society, or modern organizations, self-organization provides new ways to think about how things organize themselves without external control. In a time marked by swift technological progress, globalization, and growing complex issues, the usual top-down organizational structures are proving to be less and less efficient, as well as struggling to keep up with the complexities of modern environments. In contrast, self-organizing practices provide an alternative path to handle this complexity effectively. These principles promote adaptability, innovation, and resilience, potentially transforming how organizations function, making them more responsive to change and better prepared to succeed in an uncertain world.

On the other hand, despite the growing interest in self-organizing practices, there exists a significant gap between the theoretical foundations of self-organization and their practical implementation within organizational and project management contexts. While the theoretical literature provides a valuable deep dive into the history and principles of self-organization, translating it into actionable practices for real-world organizations remains a challenge. The gap between theory and practical application in project management necessitates a thorough investigation to link these poles closer and understand how self-organizing practices work in practice within the context of IT product development project teams.

Therefore, the aim of this thesis is to investigate the applicability of self-organizing practices, as well as to explore and substantiate the theoretical ideas of self-organization dynamics in the real-life context of IT project management within project teams, engaged in building various IT products. This attempt will be guided by the following objectives:

1. Theoretical exploration - based on the comprehensive review of the theoretical foundations and analysis of the different concepts and ideas of self-organization, to identify key research areas relating to the dynamics of self-organizing environments.

2. Research method - to establish an appropriate research method to collect and analyze the necessary data through deep diving into the derived research areas on the different dynamics of self-organizing practices within the context of certain IT product development project teams.
3. Practical review - to identify and elaborate on different themes relating to the practical usage and dynamics of self-organizing practices in IT product development projects.

To fulfill the objectives set above, this research of a case study will employ a diverse research approach, utilizing an extensive literature review which will offer a framework for further empirical analysis. It will be carried out by employing the qualitative research methods, including a semi-structured interview model to gain insights from multiple IT product management project professionals about the usage of self-organizing practices in their line of work. A further thematic analysis will be carried out to investigate the derived research areas and delve deeper into the practical implementations of self-organizing practices.

This thesis is structured into different chapters, beginning with an overview of the literature and theoretical background behind different aspects of self-organizing practices. Further chapters elaborate on the research frameworks of this thesis and present the results of the analysis on application and dynamics of the real-world implementation of self-organizing principles.

Throughout this research, several difficulties and limitations were encountered, namely in aligning the theoretical concepts with practical implementations of self-organizing practices from different contexts from the chosen organization, as well as acknowledging the inherent limitations of qualitative research, such as the potential for subjective bias and the non-generalizability of findings. Despite these challenges, by applying the chosen research method and aligning the theoretical concepts with practical implementation experiences from a diverse range of professionals, this study aims to provide valuable insights for both scholars and practitioners, enhancing understanding and practical application of self-organizing practices in IT project management environments.

2. Roots of self-organization theory

While self-organization is indeed one of the main pillars of these “modern day” frameworks, the conceptual roots of self-organization in contemporary organizational and management discourse actually lie in the natural sciences. It finds its origins in the fields of various disciplines such as physics where it emerged as a fundamental principle in explaining and governing the behavior of various complex systems. To be more specific, such natural systems, including physical, chemical, biological, and ecological systems, have been investigated and have provided vast, compelling examples of self-organizing processes. Such work was firstly carried out by W. Ross Ashby in the 1940s leading to a first recognition of the initial concept of self-organization (Ashby, 2017). The central idea suggests that complex systems, existing or pertaining to the natural world or an artificial structure, have a capacity for autonomous organization and structured behavior, driving the system towards stability and some sort of a stable situation referred to as equilibrium (Heylighen, 2001). Meaning that under the circumstances where there is an absence of concrete external control or intervention, separate parts (agents) of the system spontaneously and autonomously transition into states of order or organization while basing their operation on local interactions with one another and dealing with fluctuations in their surroundings or shifting environmental conditions (Ashby, 2017). This concept provided a solid background and was a catalyst for further research in the academic field on self-organization theory carried out by various colleagues of W.R. Ashby, most notably Heinz von Foerster, whose ideas on self-organization were shaped by a more interdisciplinary approach drawing from fields beyond cybernetics, including biology and the sciences of complexity (Von Foerster & von Foerster, 2003). The Austrian-American scientist also puts an emphasis on the autonomy of self-organizing systems, highlighting their capacity to organize themselves without central control and also creating the “order from noise” principle. The latter noted that, paradoxically, the more there are random perturbations (“noise”) that the system encounters, the more quickly it will self-organize and get back into the “order” (Von Foerster & von Foerster, 2003). In other words, the idea is straightforward - the more widely a system is made to move through its state space, the more quickly it will end up in equilibrium versus the situation where if it would just stay in place, no equilibrium would be reached, and no self-organization could take place (Heylighen, 2001).

With these ideas setting a background for an academic self-organization field, further scholarly discourse gravitated towards the domain of physics and chemistry. During the epochs of the 1960s and 1970s, a particular focus was set on phenomena pertaining to phase transitions and the spontaneous ordering of molecules and particles. Within this scholarly trajectory, academics such as Ilya Prigogine (Prigogine & Stengers, 1984) and Hermann Haken (Haken, 2006), whose significant research into self-organizing "dissipative structures" merited the Nobel Prize, and whose research produced the term "synergetics" to define his approach, played pivotal roles in advancing the theoretical frontiers of self-organization. Soon after, the study of self-organization started blending with the growing fields of non-linear dynamics and chaos mathematics. This convergence later on brought in a more precise and mathematical approach mainly led by various physicists. Simultaneously, another academic tradition called "complex adaptive systems" emerged (Waldrop, 1993). This tradition had a philosophical connection with the original ideas of self-organization, particularly those associated with cybernetics. It gained prominence with the founding of the Santa Fe Institute, which further expanded the academic research on different aspects of complexity theory. Building upon these dual traditions, the turn to the 21st century witnessed a significant surge in scholarly interest in complex networks, sanctioned by the introduction of the internet. Additionally, it was further enhanced by the work of researchers like Watts and Strogatz (1998) as well as Barabási and Albert (2002), whose contributions bore significant influence within this domain. Looking at it perspectively, such a combination of developments of this academic field comprise a rich part of academic discussions about self-organization within the branch of natural sciences.

3. Complexity theory

Moving forward closer to the subject of this thesis, the connection between the already described origins of self-organization and social sciences lie within the mentioned complexity theory. Step by step, it emerged as an interdisciplinary perspective to studying and analyzing various complex systems. It explores how different components or parts of a system interact with each other as well as their surroundings. In particular, complexity science studies a type of complex system known as "complex adaptive systems," which can simply be described as systems comprised of a large number of objects that display a high level of interactivity (Richardson, Cilliers, & Lissack, 2000; Wood & Gidado, 2008).

Such systems are open and always changing, with their parts strongly connected and capable of self-organizing (Wood & Gidado, 2008). To simplify, various constructs like rainforests, societies, immune systems, the Internet, and the global economy should be considered as the most probable examples of such systems. Thus, complexity theory helps us to dive deeper into them and analyze how these systems are structured, how they and their separate parts behave, and how they change over time, given all the irregularities, complexity and different circumstances of the real world (Wood & Gidado, 2008).

Furthermore, complexity theory argues that complex behaviors often arise in a system because its parts interact in specific ways, even though each part follows simple rules. The components within a system are guided by those fundamental rules, and according to Merry & Kassavin (1995), it is important to emphasize that understanding these behaviors isn't a matter of simply knowing everything about the individual parts. Instead, it involves studying how these parts interact among themselves and how the system changes over time in reaction to those interactions (Merry & Kassavin, 1995). Even when things seem chaotic initially, they might become predictable if we can uncover the underlying patterns and rules governing complex behavior (Merry & Kassavin, 1995). A bit of a different perspective is offered by Stacey (2001), who argues that all the individual agents that comprise a system interact with each other based on their own set of rules. These rules govern how one agent interacts with others and these interactions are "local," meaning that there are no universal rules dictating how interactions should happen. Each agent follows its own set of rules, making its interactions repetitive as well

as forcing each agent to adapt to each other, creating an environment for various mutations and different variations of those rules over time (Stacey, 2001).

To summarize, complexity theory provides a framework to deconstruct various systems to a single-agent level, offering to investigate the interactions between these parts. However, the set of rules sanctioning those interactions can be looked at from a few different perspectives, that involve concentrating on the individual level of agent rules, or looking at the bigger picture and attempting to understand the patterns of agent interactions and their effect on the system itself.

4. Triple loop learning

A further adaptation of self-organization theory in project and organization management is evident in the practices of circular organizing and triple loop learning. It is emphasized, that all the agents within the complex organization or system must develop and sustain a capability to learn, otherwise it is inevitable that in the absence of learning, it will fail to reflect on the major challenges or issues that may arise (de Waard, Rietjens, Romme, & van Fenema, 2021). To avoid this, a framework of triple loop learning was introduced so that more complex challenges and transformations within the organizations could be addressed:

1. Single-Loop - the primary focus here lies on reviewing and making corrective actions on current processes without critically examining and changing the fundamental assumptions or strategies (Gardner, 2022).
2. Double-Loop - it entails questioning not just actions but also the underlying assumptions and strategies. Instead of simply making corrections, the aim is to challenge and potentially transform the organization's knowledge and competency (Georges, Romme, & Van Witteloostuijn, 1999).
3. Triple-Loop - goes beyond actions, strategies, and goals. It focuses on the very underlying values and principles of the decision-making process. At its core, it attempts to improve how deeply and comprehensively we understand highly complex problems and challenges that arise (de Waard et al., 2021).

In essence, triple-loop learning offers a method based on a deeper level of reflection and transformation, aiming for not just surface-level changes but fundamental and sustainable change

within an organization. Therefore, it is particularly relevant and useful in today's fast-paced and complex environment, where organizations need to be agile and ready for any significant shifts in their approach and strategy when dealing with such wide challenges as COVID-19 (Asawo & Ogbonda, 2022).

5. Application in project management

The nature of projects has evolved to become more and more intricate and complex, leading to a situation where significant concerns about effectively managing this heightened complexity arose. It has become evident that the conventional project management approaches and tools, originally created for more straightforward and less complicated projects, no longer suffice in addressing the various challenges posed by the increasing level of complexity (San Cristóbal et al., 2018). This is especially valid for different IT products which tend to reach high-level complexities and are prone to significant and unpredictable changes. As technological advancement is constant, it will also inevitably become increasingly complex - the next generation of computers, space shuttles, phones, watches, and cars will incorporate greater intelligence and complexity (Morcov, Pintelon, & Kusters, 2020). Thus it is essential for companies to offer enhanced functionality, benefits, and value, in order to allow these technologies to stay competitive and relevant in the market. Therefore, these recognitions have sparked a further need for innovative and advanced strategies to deal with the arising issues and successfully execute different multi-layered projects, thus turning the attention to the complexity theory field.

5.1 Complexity models

Having in mind the different streams and ideas of complexity theory itself, adaptations of it to the project management field have also resulted in different approaches among the scholars, resulting in a lack of consensus on determining what complexity really is (San Cristóbal et al., 2018). This divergence in perspectives arises from the complex nature of projects and the various factors that can contribute to their complexity. Different scholars or experts may approach and describe complexity in different ways, depending on their specific contexts and areas of focus.

For some, complexity in project management may focus around the intricacy of the project's objectives, the number of stakeholders involved, or the uncertainty around the project outcomes. Others might put an emphasis on the complexity arising from various technological challenges, restrictions coming from resource limitations, or the interactions between multiple components within the project. As a result, there exists a spectrum of viewpoints on what elements and attributes define complexity in projects. Thus, a few complexity models that are the most relevant ones in the project management context have to be looked at:

1. Goals and Methods Matrix developed by Turner and Cochrane composes a categorization system for projects based on the clarity of project goals and the methods used to achieve them (Turner & Cochrane, 1993). Accordingly to these two criteria, four project types are outlined:

- Type 1 projects are those with well-defined and thoroughly understood goals. Here, the project manager assumes the role of a conductor.
- Type 2 projects possess well-defined goals but lack clarity in terms of activities that are used to reach them. In such cases, the project manager functions as a coach, and project planning is carried out using the "rolling wave" approach as information becomes available.
- Type 3 projects involve poorly defined goals but well-established methods. These projects are structured in stages throughout their lifecycle, and the project manager has to take on the role of a craftsman.
- Type 4 projects lack both defined goals and established methods.

2. Stacey's Agreement and Certainty Matrix combines two main factors for the analysis of the complexity: how certain we are about things related to the project and how much agreement there is among stakeholders (Stacey, 1996). Based on these factors, he divides projects into four zones:

- Zone 1 - there's a lot of agreement and certainty, thus simple projects are found here. In such cases, the traditional project management works well, and the emphasis is put on finding the most efficient and effective processes.

- Zone 2 - projects that are far from agreement but have a high level of certainty around them. To handle these coalitions, compromises, and negotiations take place, Game Theory or Hypergames can also be useful in resolving such situations (San Cristóbal, 2017).
- Zone 3 - when there's some agreement but a lot of uncertainty. Traditional project management techniques might not be sufficient here, thus leadership-based approaches can be helpful instead.
- Zone 4 - is where there's very little agreement and a high level of uncertainty. This is like a zone of chaos and anarchy. In such cases, traditional management methods don't work, and one of the options is to rely on leadership approaches to deal with the situation.

3. Williams' complexity model expands upon Baccarini's model by adding a new element to it (Williams, 2002; San Cristóbal, 2017). While Baccarini's approach to complexity was based on the number of elements and their interdependence, Williams introduces a new dimension: uncertainty. Scholar argues that the growing complexity in projects can be attributed to two main factors :

- Firstly, he points to the relationship between the complexity of the product or outcome the project aims to deliver and the overall complexity of the project itself. When the product is highly complex, it often leads to the project itself being more intricate.
- Secondly, Williams highlights the increasing time constraint in modern day projects. Those have become increasingly time-sensitive, which adds another layer to their complexity, making the management part of a project more challenging, especially when dealing with intricate products.

To summarize, it is evident that there isn't a single agreed-upon definition among scholars due to the many factors that can make projects complex. However, it is undebatable, that complexity theory has been adapted to social sciences and does indeed offer different frameworks to delve deeper into the discourse of project complexities.

5.2 Importance to project management

The significance of complexity in the context of the project management processes soon became a widely acknowledged proposition, exerting influence over some critical aspects. According to various different “classical” scholars, such as Morris, Gidado, or Wozniak, and later supported by Prince & Said and Morcov et al., it governs various important parts such as project planning, coordination and control, as well as may even determine the overall success of the project (Morris, 1994; Gidado, 1993; Wozniak, 1993; Princes & Said, 2022; Morcov, Pintelon, & Kusters, 2020; Joseph & Marnewick, 2022). Secondly, it complicates the clear definition of goals and objectives for large projects. Thirdly, it has influence on determining the suitable organizational structure for a project and the expertise needed among the team members and management personnel. Finally, complexity can serve as a significant criterion in the selection process of an appropriate project management framework, determining the outcomes of the projects including but not limited to time, cost, quality and safety (San Cristóbal, 2017). With that said, it's evident that grasping the concept of project complexity and figuring out how to handle it properly is vital for the success of project managers. This importance stems from the fact that the level of complexity in a project is closely linked to how decisions are made throughout the project's lifecycle. Additionally, it influences the probability of successfully fulfilling the project's objectives, considering that as projects become more intricate, the way decisions are taken and the path to reaching project goals tend to differ significantly (San Cristóbal, Carral, Diaz, Fraguela, & Iglesias, 2018). In addition, getting hold of complexity and navigation of it may unlock a number of surprisingly positive features of the system, such as flexibility, autonomy and robustness, that traditional mechanistic and stubborn systems usually lack. These qualities can all be seen as aspects of the process of self-organization that determines complex systems: these systems spontaneously organize themselves so as to better cope with various internal and external perturbations and conflicts (Heylighen, 2008). This allows them to evolve and adapt to a constantly changing environment and develops an ability to take advantage of arising potential opportunities. Therefore, effectively managing complexity and understanding the importance of self-organization is a crucial skill for project managers in product development field to ensure project success.

6. The specifics of the IT product development projects

It's now time to shift our focus towards the object of this thesis - the IT product development projects - where self-organizing practices will be researched. In today's very technology-driven world, the realm of IT product development has emerged as one of the key driving forces behind economic growth and innovations that are constantly changing our everyday lives. Different IT products, encompassing software applications, various hardware components and extremely wide network infrastructure, have now been integrated into nearly every aspect of our lives, transforming the way we work, communicate, and interact with the world around us. And while these products are focused on making the activities of our daily routines easier, the development of these products themselves is a very complex and multifaceted venture, requiring a combination of high level technical expertise, business knowledge, and creative vision. In comparison to traditional project management, IT product development projects face significant challenges of technological complexity, necessitating specialized skills and knowledge to address the dynamic interactions of different software, hardware, and system integration solutions (Iyer, 2021). In addition to the technological complexity, such projects are also unique in other aspects relevant to this thesis, especially focusing flexibility when facing uncertainty.

6.1. Technological complexity in IT product development projects

Complexity is considered to be the coherent characteristic of IT product development projects, comprising the already mentioned fusion of challenges between the technological advancements, evolving user demands, and competitive market environment. The latter is actually one of the key drivers here as the software industry is characterized by an exceptionally high degree of complexity, primarily due to the rapid pace of change in project nature, distinguishing it from other industries where changes occur more gradually (Muhammad et al., 2021).

As already suggested, one of the main contributors to the overall complexity of IT project development projects is the wide web of technological challenges that are crucial and have to be addressed during the whole lifecycle of the project. The vast number of systems and subsystems

integrated within the project, the different methodological and philosophical assumptions across these systems, the cross-organizational and schedule interdependencies between activities, the upgrading and maintenance efforts and the sheer size and entanglement in the project are all key factors influencing complexity (San Cristóbal et al., 2019). Having to deal with unique architecture of individual components, different functionality, data requirements and other aspects of modern IT products creates a completely new level of large complex network of interdependencies, where changes in one system can have influence on every other part of the entire project. Furthermore, complexity in such systems grows even at the individual level of each component, with emerging generations of software and devices becoming intelligent and advanced, making it harder to maintain a holistic understanding of the project itself.

Consequently, the organization and processes within it, such as marketing, sales and delivery departments are also becoming more complex to introduce these innovative products to new users and markets (Morcov, Pintelon, & Kusters, 2020). Different teams contributing their expertise to the delivery of the project thus face a high level of interdependence and interrelations, creating another challenge of maintaining effective collaboration among them. They have to stay aligned on task sequence, ensure timely and unobstructed flow of information and effective resource allocation, with all of these dependencies positively contributing to the success of their shared goal (Zhang & Kwan, 2019). On the other hand, the challenge of delivering projects swiftly could be approached by the introduction of cross-functional teams, which are composed of members with all the necessary expertise backgrounds, thus avoiding the barriers of functional and departmental barriers (Mateo, Formoso, & Iglesias, 2021). While having these self-managing groups that involve the people from all the relevant departments should make the management of the discussed dependencies easier, the effectiveness of such teams does not guarantee success and still require a high level of coordination, appropriate communication and collaboration efforts.

Naturally, such a high level of complexity in different aspects of IT product development projects presents substantial risks that can critically impact the outcomes and the whole phase of the project, and if not managed properly, could jeopardize the existence of an organization itself (Muhammad et al., 2021). Such problems as delays, cost overruns or compromised product quality can arise, when the complex structure of the project is not embraced and consistency across complex development processes is not maintained. On the other hand, while the

traditional project management literature and its evidence is focused only on the negative effects of complexity, modern project management scholars also emphasize the positive potential it holds (Morcov, Pintelon, & Kusters, 2020). A rather counterintuitive perspective is offered, suggesting that complexity is not an obstacle but rather somewhat of a driving force behind innovation and adaptability capabilities. As by embracing complexity, IT projects are operating further from stable order and embrace irregularity which introduces a level of intricacy fostering the environment for the development of novel solutions as well as adaptability to the ever-changing demands of the technological landscape (Morcov, Pintelon, & Kusters, 2020). Complexity, therefore, may become a source of competitive advantage, enabling project teams to efficiently deal with different dynamic environments and fulfill the changing market demands. Moreover, the very nature of IT systems development and software engineering is very complex and focused on delivering valuable outcomes through intricate processes, thus embracing complexity and leveraging its creative potential, can unlock transformative innovations and drive exceptional value creation. Especially given the rapid emergence and incorporation of the likes of Artificial Intelligence, Data Science and Machine Learning, the challenges being tackled in both industrial and societal contexts are also growing in complexity, often surpassing current levels of human understanding (Morcov, Pintelon, & Kusters, 2020).

6.2 Uncertainty

The second aspect that greatly distinguishes IT product development projects is the amount of uncertainty that they have to face and cope with in order to be successful. This uncertainty arises from various different factors, including the evolving technology trends, changing market demands, and unpredictable user preferences. To begin with, there is really no introduction needed to the contributions of the technological part here, which is greatly associated with the already covered topic of technological complexity evident in IT projects. With technological sophistication of IT systems increasing rapidly, the uncertainty brought with it does as well, steaming from the usage of various novel technologies, the intricacy of their interactions, and the potential for unexpected outcomes, arising from sometimes unpredictable correlation of technical aspects (San Cristóbal et al., 2019). In other words, the negative side of complexity and the challenges that are brought by it towards the management of the project,

contributes to having to embrace speculative results and unclear outcomes of the decisions made, thus possibly adding another layer of risk to the delivery of the project itself.

A different aspect of uncertainty in IT product development projects is brought by the specifics of the IT product market, characterized by such aspects as rapid change and unclear customer requirements. As it has already been suggested, the IT sector is extremely high paced, pushed forward by the constant development of innovative products and services, increased feature functionality and especially the rising complexity in all of these aspects. Operating in such an environment raises the challenges of guessing what awaits around the corner, how to grasp the opportunities, what to build and how to do it - working with such information which is not “knowable”, is something that all firms developing IT products face at one point in time (MacCormack & Verganti, 2003). A further layer of uncertainty here arises from growing and less predictable requirements from the customers, which go hand in hand with the pace of the market. With all the innovations at hand, companies still need to address the ambiguity that concerns not knowing what the customer really desires in terms of new technology (Chen, Reilly, & Lynn, 2005). Meaning, that as customer needs or market trends become increasingly personalized, diversified, and complex, it becomes more and more difficult for organizations to accurately predict and satisfy the full spectrum of customer needs, posing a significant challenge in navigating market uncertainty. Therefore, it is necessary for companies to constantly and operatively engage in research, learning and iteration activities in order to capture those evolving customer needs and deliver value (Chen, Reilly, & Lynn, 2005).

6.3 Flexibility or/and adaptability

Finally, with the above mentioned complexity and uncertainty specifics of IT product development projects being major challenges to the success of such projects, the natural demand for a response had to be answered. With the traditional project management approaches emphasizing thorough planning, strict control and dependence on predictability, more flexible and adaptive methodologies such as Agile are viewed as a better fit due their iterative and incremental approach (Berényi & Soltész, 2022). In the environment of evolving requirements and unpredictable circumstances, Agiles adaptability comes from its core principles of iterative development, continuous feedback, and collaborative decision-making. As by deconstructing

large tasks or projects into smaller, manageable items delivered in gradual iterations, it enables teams to embrace the challenges of complexity and uncertainty step by step, incorporating gained knowledge into subsequent phases, as well as minimizing the impact that decisions and activities in one area have on other ones (San Cristóbal et al., 2019). Such an approach stimulates a much more focused handling of each project segment and problems there, enables progressive improvements and facilitates a deeper understanding and more effective management of the complexities involved. Furthermore, Agile benefits project teams with its focus on open communication and collaboration among team members, stakeholders, and customers, ensuring constant flow of feedback and timely adjustments to the project deliverables. This feedback loop and the iterative nature of Agile allows teams to make adjustments to the project in real-time, reflecting the fast pace of the market and its requirements. Finally, such a collaborative approach promoting creativity and productivity empowers teams to respond effectively to changing requirements or unforeseen challenges, and, most importantly, better identify the needs of the customers and bring value to them (Agbejule & Lehtineva, 2022).

On the other hand, the discussed literature on the new and more adaptable methodologies provides a very solid background for the benefits of these practices to face uncertainty and be more agile in this environment, but it does not emphasize its practical implications. While the concluded research outlines the theoretical advantages in the application of general ideas of these methods in complex environments, it so far lacks a more specific detailing on how these benefits can be captured in the real-world scenarios. As it stays at a high level, it overlooks the practical challenges that teams may encounter when adopting these methodologies, as well as does not provide more guidance on how to actually take the most out of these methodologies, illustrating how these methodologies are applied and the outcomes they produce in terms of dealing with complexity.

7. Self-organization in project management

Moving the focus even more towards self-organization, represents a significant departure from conventional top-down approaches, offering a novel perspective on project execution. Here it defines the intrinsic ability of different project teams to autonomously structure their work, make decisions collectively, and adapt dynamically to changing internal or external circumstances. It pivots away from the “old school” idea of rigid, centralized control and instead embraces the idea that, given the right conditions and guidance, project teams can efficiently organize their own efforts. This shift recognizes the complex interactions among variables, stakeholders, and uncertainties that define the modern project environment, moving away from a mistaken assumption that organizations operate like machines (Wheatley & Kellner-Rogers, 1996). Dating back to the 17th century, a period when philosophers began likening the universe to a grand clockwork mechanism.

7.1 Conditions for self-organization

Moving away from the ancient assumptions, nowadays, organizations can and should be compared and treated as proper living systems. Like living systems, they have the ability to self-organize, sustain their existence, as well as grow towards higher levels of complexity and order when needed (Wheatley & Kellner-Rogers, 1996). They can intelligently adapt to change by reorganizing their structures and patterns without external direction. However, accordingly to the works of Wheatley and Kellner-Rogers (1996), the emphasis here must be put on the fact that there are three conditions that support an organization's capacity to access its intelligence and change appropriately to the circumstances.

7.1.2 Identity

The process of organizing always starts with an intention, or a belief that there is a possibility to create something when people come together. This organizing process focuses around the initial development of some sort of an identity – a sense of self for the organization

(Wheatley & Kellner-Rogers, 1996). Once this identity takes shape, it becomes the primary view-point through which the organization is perceived, as well as operates and makes decisions, always referring to its self-concept. Furthermore, an organization's self-concept encompasses its vision, mission, values, as well as its interpretations of its history, current decisions, activities, and future aspirations. It represents what we wish to believe and what our actions demonstrate about us (Wheatley & Kellner-Rogers, 1996).

Most organizing approaches overlook the importance of establishing a rational identity. Yet, it is this clarity and understanding of the uniqueness of the organization that enables individuals to be involved creatively and autonomously (Cotrel, 2022). When there's alignment around principles and purposes of the activities, maximum autonomy is possible, as people organize their unique contributions based on their shared sense of identity. It is the organizational identity that should remain the most stable aspect of the enterprise when facing various problems. Structures, programs and people may change, but organizations with a clear core identity can bear the turbulence with confidence, which comes from clarity of understanding who they are and what their vision is (Wheatley & Kellner-Rogers, 1996).

7.1.3 Information

Information is one of the key aspects in the life of organizations. It circulates throughout the various levels of it, providing the necessary inputs for its functions and processes (Ismael, 2011). Without this information flow, the organization would struggle to function effectively, much like a body without blood would cease to operate. Although it might seem chaotic when it circulates freely, it's actually what fuels the organization's ability to self-organize and adapt.

Real organizational agility is sanctioned to happen only when information is shared openly and is available to everyone, even in the cases where there is no direct need for it at the time. To ensure the organization's vitality and continuous operation, information should be readily available and accessible throughout all its departments and levels. However, it must be emphasized that this does not mean that when everyone has access to information, decision-making shifts to small local groups (Wheatley & Kellner-Rogers, 1996). When information is accessible to everyone, different people notice different things, meaning that those with a big-picture view might spot opportunities that others might not see. Therefore, being open

to a multitude of viewpoints helps members to access and interpret the information available, aiding organizations to self-organize more effectively (Wheatley & Kellner-Rogers, 1996). It's the presence and availability of abundant, unprocessed information that facilitates swift, well-coordinated, and effective responses.

7.1.4 Relationships

Relationships within an organization serve as the intricate pathways through which the collective intelligence of the system thrives (Wheatley & Kellner-Rogers, 1996). Through connections and interactions between individuals, information is generated and adapted in different ways, causing the organization's identity to include a wider scope of stakeholders, as well as contributing to the organization's overall intelligence. The greater the availability of interactions among individuals, the more diverse and promising the possibilities become, people are empowered to collaborate and get things done. Therefore, in self-organizing systems, an essential requirement is the free and open access individuals have to one another (Wheatley & Kellner-Rogers, 1996). This unrestricted connectivity empowers individuals to cross any part of the organization to fulfill their roles and responsibilities.

In addition, having the ability to meet the challenges of rapidly evolving situations efficiently, is also sanctioned by having an unobstructed access to the entirety of the organization's collective (human) intelligence (Wheatley & Kellner-Rogers, 1996). Individuals need to be aware of who is available, the extent of their knowledge, and the appropriate methods of getting in touch. Notably, these interactions often lead to the organization's reach expanding and supporting a wider number of stakeholders. Furthermore, an environment focused on interactions also enhances tolerance and acceptance among colleagues working on shared projects, despite their diverse backgrounds and lives outside of work. Over time it may also translate to having more trust in co-workers, which could then turn into a natural commitment to the goals of the team as well as the psychological commitment leading to loyalty to one's colleagues (Larsen, 2004).

In conclusion, the organic expansion of connections and the increasing engagement of stakeholders play a vital role in shaping the dynamic and adaptive qualities of self-organizing systems. This continual growth in relationships and the incorporation of diverse viewpoints

enable the organization to consistently adapt and react efficiently to evolving situations and complexities.

7.1.5 Fusion of these elements

The interplay of identity, information, and relationships forms a dynamic and intricate cycle in which these elements are deeply interconnected, often making it challenging to separate one from the others (Wheatley & Kellner-Rogers, 1996). As new connections and relationships emerge within the organization and its daily activities, the network of individuals involved expands, consequently generating fresh information. This influx of information, in turn, influences the organization's identity, shaping how it perceives itself, its role, its vision and processes.

Conversely, the unrestricted circulation and availability of information enhances the creation of new business opportunities as well as enables individuals to establish new relationships, both within and beyond the organization. As the organization responds and adapts to these changes and developments, its identity simultaneously evolves. While this transformation provides greater clarity about the organization's purpose and direction, it also introduces shifts in how the organization defines itself. Thus, this continuous fusion between identity, information, and relationships forms an environment of a dynamic cycle, enabling self-organization to happen as well as driving the organization forward (Wheatley & Kellner-Rogers, 1996).

However, one of the key problems of these findings is that it focuses on the theoretical level of these communication and relationship elements and only considers this from a conceptual point of view. It limits itself with an understanding that different groups will be able to find their own ways of ensuring the seamless flow of information and efficient communication channels, as well as manage it on different levels of project teams. Thus while providing an in-depth exploration of the theoretical frameworks of identity, information and relationships, the literature does not sufficiently address the practical implementation steps or the challenges that organizations might face when trying to support these conditions. Furthermore, there is a lack of specific practices or concrete insights from different teams and environments that could be used as a background framework to ensure the benefits of these elements towards self-organizing.

8. Leadership as another essential part of self-organization

Another essential part of self-organizing teams that must be discussed here is leadership. While self-organizing teams are capable of delivering their objectives with its separate parts working autonomously, leaders play a vital role in firstly, creating such an environment and secondly, actually making it work.

Traditional leadership theories see the leader as responsible for setting goals and directing the organization to achieve those goals, basing their view on the already mentioned mechanistic view of organizations (Uhl-Bien & Marion, 2009). It defines the organization as a very structured and clear machine, run by a very clear system of hierarchy, power and authority. Given such an approach the focus here is put on the leader as the central figure who controls and directs the organization to the clearly presumed target, controlling the team and guiding them to desired outcomes. In other words, there is a belief that leaders can plan and execute their plans in order to achieve the goals by having the ability to predict the future as well as being in control of the present situation (Plowman et al., 2007).

However, a more modern approach has been offered by the advocates of complexity science (Chauhan, Crewe, & Mowles, 2022) who suggest that organizations nowadays are complex systems operating in a less stable environment, and that their outcomes are delivered from the interactions of many different people and factors (Marion & Uhl-Bien, 2020). Suggesting that it is very unlikely for anyone to predict the long-term future or control all of the factors that will affect the outcome of a current situation, thus the question of leadership should be approached from a perspective that embraces complexity, rather than tries to simplify it (Geerlof & van Beckhoven, 2016). This means that leaders cannot simply just set goals and expect the organization to achieve them by following the strict plan, rather they need to be able to understand and manage the complex systems they are involved in, and to create the conditions for self-organization to thrive.

8.1 Complexity Leadership Theory

Taking such an approach into account, a new leadership framework called Complexity Leadership was proposed in the late 20th century. The latter moves away from the “older school” view of leadership, restricted by hierarchical structure, direct accountability, centralized decision-making and the general view of the world as a predictable environment where most of the outcomes can be prepared for and controlled. According to the scholars, rigid frameworks created to provide solutions to past problems are by anyway flexible enough to provide alternative solutions to organizational challenges in nowadays more chaotic environments (Uhl-Bien & Marion, 2011). Thus, basing its view on the already discussed complexity theory and its view of unpredictability, complexity leadership is focused on creating the environment for self-organization and emergence to evolve. This means providing team members with the resources and support they need, but also giving them the autonomy to make decisions and solve problems.

Lichtenstein and Plowman argue that complexity leadership allows for problem-solving to emerge, instead of controlling the outcome of a situation (Lichtenstein & Plowman, 2009). They contend that complexity leadership doesn't seek to dictate predetermined solutions or control the outcome of situations. Instead, it embraces the notion that, within complex and dynamic organizational environments, problems and challenges are better addressed by allowing solutions to emerge organically. In other words, complexity leadership is not about trying to control everything and predict every outcome. Instead, it is about creating an environment where leaders enable people to work together to welcome experimentation, adapt to changes as well as solve emergent problems through the collective intelligence and interactions of individuals within the organization (Lichtenstein & Plowman, 2009). Thus, if leaders and members are open to new ideas and ways of doing things, and if they are willing to experiment and make mistakes, then the system will be more likely to identify and implement positive changes.

Further emphasis in the Complexity Leadership theory is also put on welcoming disorder as a partner and an opportunity, rather than trying to achieve, maintain and operate in a stable equilibrium (Rosenhead, Franco, Grint, & Friedland, 2019). Leaders must position themselves and their teams to embrace a controlled level of chaos or disorder. Only by doing so can the organization foster a controlled environment where a diversity of ideas have the ability to

flourish and the emergence of new potential pathways for the organization could be welcomed. Rather than adhering to a strictly planned corporate strategy, where individuals are stuck within the repeated patterns and approaches, this open-mindedness results in an organization that continually reinvents itself (Rosenhead et al., 2019). Such an organization does not attempt to bring itself back into the “stable” state before facing a challenge, but tries constantly to evolve and find new approaches to operate and develop itself further in accordance with the ever changing conditions of its environment.

8.2 Different Leadership styles

Having discussed the general ideas of complexity leadership theory it's now very important to delve deeper into the way different leadership styles are interconnected within the theory. As already suggested previously, one of the main missions of complexity leaders is to create the best possible environment for self-organization to thrive, which revolves around striking a balance between making team members feel comfortable taking risks while making informed decisions and supporting them with the necessary assistance they need to be successful. Taking this goal into account, complexity leadership comes as a joint, resultant product of three different types of leadership, including adaptive, administrative and enabling leaderships (Baltacı & Balcı, 2017).

8.2.1 Adaptive Leadership

The seeds of adaptive leadership lie within the tension that could arise from different interactions between agents that operate within the system - through the interplay of separate individuals new information and ideas can emerge. And when this novelty and innovation is used to spark a positive change of the system, then there is evidence that adaptive leadership has occurred (Lichtenstein et al., 2006). In other words, adaptive leadership takes attempts to leverage the tension and use it as a driver for change, through which interacting agents address complex challenges in ways that produce new patterns of thought and behavior. As such, adaptive leadership represents an interactive, dynamic process which focuses on having the ability to ensure resonance with rapidly changing new organizational conditions through learning

and evolving (Baltacı & Balcı, 2017). Therefore, adaptive leadership is not about pushing followers to strictly follow the leader's plan, but it is rather about creating the conditions for groups of people to solve problems and adapt to change, about “transforming, rather than providing technical fixes” (Lichtenstein & Plowman, 2009). Given this, scholars suggest, that adaptive leadership can happen anywhere in the system or organization, and it does not require authority or a position of power as the traditional leadership would, instead, it is defined as a complex and dynamic process that is sparked by adaptive challenges, which are situations where the old ways of doing things no longer work (Lichtenstein et al., 2006). In these situations, individuals have the chance to take up the role of a leader by mobilizing others to seize new opportunities and tackle complex problems, while maintaining their adaptive environment and giving others the chance to step up in cases where different skills or experience would be desired.

Another significant part of influence played by adaptive leadership is visible in the course of identity formation, which has already been defined as a significant aspect of self organization. A new social identity is usually formed over time through different interactions of the agents, especially those connected to leadership events, as participants together define "who we are" and what we are doing through their interactions (Lichtenstein et al., 2006). The emergence of a shared set of values or goals occurs through the formation of a joint social identity, arising from different processes of collaborations and interactions of the individuals involved. However, this driver of collective identity formation can be forgotten as soon as the participants create a common-sense conception of a formal leader "out there," with themselves embracing in complementary follower roles (Kahneman & Tversky, 1972). In the context of complexity leadership, this can lead to a situation where group members forget that they played a role in creating the group's identity and gradually pertain themselves as followers of the leader. Therefore, it's a must to keep individuals in the environment where they can claim the ownership of their roles and where creativity, learning and resonance is ensured on a large-scale platform through the guidance of adaptive leadership (Baltacı & Balcı, 2017).

8.2.2 Administrative leadership

Another important and often described as the key part of complexity leadership theory is the administrative leadership. According to scholars, it is the managerial form of leadership that

addresses and controls the bureaucratic functions of the organization, while not restraining the organization's ability to operate in complex dynamics and producing adaptive changes (Uhl-Bien & Marion, 2009). It focuses on coordination of the activities that individual organizational members and groups perform in their formal, managerial roles to achieve organizational goals, including planning, task delegation, defining organizational vision, providing resources, managing crises and conflicts, and deciding on survival strategies or policies (Holland, 2006). Administrative leaders play a vital role in implementing organizational policies and procedures, and oversee the day-to-day operations of the organization. In the context of complexity, administrative leadership provides a foundation for a certain level of stability and order, which is completely essential in complex and unpredictable environments, where change is constant. Therefore, administrative leaders in complex organizations must balance the need for stability and order with the need for change and adaptation. While a certain degree of regulation of workflow and other bureaucratic functions have to be maintained, leaders have to be able to adapt to change and to create an environment where team members remain empowered to innovate, take risks and make autonomous decisions which are essential for success in today's complex and unpredictable world (Baltaci & Balci, 2017).

8.2.3 Enabling leadership

Finally, complexity leadership theory draws upon the importance of enabling leadership, which is seen as a critical strategy in the face of ever-changing organizational environments. However, the approach to it is slightly different to the distinctive adaptive and administrative leadership theories discussed above. While administrative leadership occurs in the administrative structure of the organization and adaptive leadership tends to emerge in the general environment of complex structures, enabling leadership actually acts in the interface between the two (Uhl-Bien & Marion, 2009). It emerges as this crucial mediator bridging the gaps between the set boundaries of organizational structure, usually characterized by bureaucratic procedures, and the interactive dynamics of responsive adaptation (Uhl-Bien & Marion, 2009). Such balancing involves fostering an environment which nurtures the purpose of the adaptive function to promote innovation while at the same time ensuring the alignment with the administrative part of following the organization's goals, mission and set procedures. In other words, the function of

enabling leadership could be described as safeguarding the adaptive and innovative abilities of the organization from the potentially constraining control mechanisms brought in by administrative systems.

Furthermore, enabling leadership plays an important role by fostering innovation in the essence of connecting the locally generated novel solutions with influential individuals and necessary resources within the organization, enabling these innovations to gain traction and advance into the formal organizational structure (Arena et al., 2017). It centers around the idea that such leaders not just allow or tolerate new ideas being translated into the operational side but actively support the construct and development of new ideas from the entrepreneurial space and inject them into the operational side of the organization (Arena et al., 2017). Without such direct and formal support, the innovative responses to complex challenges may lack the necessary effort to integrate them into the operational systems and structures for implementation (Rideout, 2023).

In this sense, the enabling function of complexity leadership theory can reduce this challenge by facilitating the implementation of ideas generated by the adaptive function and fostering the creation of new information. This practical approach enables administrative leaders to recognize and validate these ideas, making new information available, which as emphasized earlier, is one of the key contributors to the effective self-organization.

On the other hand, the mentioned research focuses on leaders creating the environment for self-organization, but it doesn't address the role of leaders in building trust, fostering collaboration, and developing team skills teams, which are crucial aspects of enabling successful self-organization. Furthermore, it emphasizes the conceptual understanding of leadership in self-organizing teams and its possible benefits towards fostering these environments without providing practical guidance on how leaders can actually create and nurture such environments. Therefore, while the current academic research extensively discusses leadership theories and their evolution from traditional to complexity-based approaches, it lacks a detailed exploration of how these theories are applied in real-world organizational settings, particularly in the context of self-organizing IT product development project teams.

9. Self-organization and agile methodologies

Moving our focus towards the actual implementation of self-organization practices in reality points us straight towards a very popular methodology used nowadays - Agile. While it has been multiple decades now since the initial Agile manifesto was drafted in 2001, the popularity of the framework rose significantly since the 2010s as the constantly changing field of software development began to demand new solutions to its less and less stable environment. Up until today Agile methodologies seem to have fulfilled this demand for an adaptable, change embracing and improvement oriented method, by offering a project management approach based on iterative and incremental progress of delivering value to the customer.

The core of Agile is based on a more segmented approach to the deliverables, breaking them down into smaller pieces, which are then easier to manage, develop and release, expediting the delivery of business value to customers. To accomplish these objectives, agile methodologies incorporate precise iteration planning procedures, focused on outlining and detailing the tasks that have to be completed within each iteration, along with their allocated cost and time estimates (Matook, Soltani, & Maruping, 2016). The customer also plays an important role here by participating in each of the iterations, identifying, clarifying, and prioritizing the requirements for the desired outcomes, therefore, such an approach of constant input ensures that the project remains aligned with the evolving needs. These ideas are the core pillars in Scrum framework, which is best characterized by its form of working in cycles known as Sprints, with each of them lasting from two to four weeks (Hoda, 2011). During each cycle, self-organizing teams determine and work on tasks from a previously developed and prioritized backlog of requirements, ensuring that the features developed first are those that deliver the greatest value for the customer, while at the same time aligning the team's efforts with the evolving needs of the project's stakeholders. At the conclusion of each Sprint, a potentially shippable product or an incremental piece of it is delivered, marking a step forward in the project's development (Hoda, 2011). Scrum's iterative design also ensures a continuous learning cycle, enabling teams to get and take note of the feedback, as well as adapt their approach based on insights gained from each Sprint, promoting adaptability and making sure that the final product aligns with the evolving needs of its users. Furthermore, coordinating work in such way that the work is split and knowledge is gained gradually, allows the complexity of the project to be managed easier, as

well as deal with persistent issues that have been affecting the software development and service delivery in the IT industry, such as budget overruns, missed deadlines, low-quality outputs, and dissatisfied users (Cooke, 2012).

9.1 Self-organizing practices and teams in agile methodologies

Having emphasized the core principles and practices of Agile, it now must be discussed how the self-organizing teams are viewed and incorporated into the framework. According to scholars, self-organizing teams are actually considered to be at the heart of Agile software development (Hoda, 2011), being highly adaptive and responsive to the emerging challenges without having to bear with the constraints of traditional hierarchical structures of teams within the organization. They are comprised of many diverse individuals, that are able to manage their own workload, collaborate among themselves, while at the same time performing highly related or interdependent jobs and participating in a collective team decision-making processes (Hoda, 2011). The whole team, including developers, designers, testers and other team members participate and contribute their efforts to mutual planning activities, working together to estimate and prioritize tasks to deliver in each iteration, as well as setting and aligning on the desired goals. Only then various practices are employed to distribute the workload effectively, including self-assignment via storyboards (Kanban), self-monitoring conducted during daily stand-up meetings and the application of information radiators to ensure the availability of information (Hoda, 2011). One of the best examples of self-organization in agile is actually Kanban, meaning “card or sign board” in Japanese and adapting its meaning directly into practice by the usage of physical or virtual boards. These teams note down and visualize the tasks which have to be completed, as well as the workflow of the stages that the tasks go through during the project, allowing team members to self-organize their work by assigning and completing the tasks (Hoda, 2011). This way monitoring and controlling of the processes involved are improved, responsiveness to change is unlocked, as well as production time is reduced. With such collaboration practice among different teams, the necessary balance between the combination of cross-functional skills and specialized expertise is further achieved through methods of embracing diverse viewpoints, engaging in group programming activities, or even implementing role rotation. Finally, a very important part of the iterative approach - continuous

learning - is conducted and managed through practices like conducting retrospectives, integrating learning spikes, and adopting a 'pair-in-need' approach when necessary (Hoda, 2011).

On the other hand, such self-organizing teams are dependent on a few key factors, including autonomy and shared identity. In an Agile environment, self-organizing teams operate independently from the usual top-down management structure and are not micro-managed, which is crucial for collaborative learning, enabling the team to come up with new and innovative ideas (Rigby, Sutherland, & Hirotaka, 2016). When team members embrace the freedom they have been given to make decisions and act on their own sense, they are set to become more involved and committed, leading to a greater care for their work, sparking creativity, willingness to help others, higher productivity, and better quality of service (Moe, Dingsøyr, & Dybå, 2008). Moreover, the autonomy of self-managed teams contributes to the higher effectiveness level - decision-making in such teams is brought to the operational level, thus can be made quickly and accurately, right where problems or uncertainties arise (Tata & Prasad, 2004). However, it's natural that such freedom comes with a certain degree of responsibility, requiring the team to balance independence with accountability, which is partially achieved through the already mentioned collective planning and alignment activities. Overall, creating autonomy for self-organizing teams in Agile isn't just about freedom from management, but it's rather more of a key driver towards better teamwork, innovation, and a strong sense of empowerment among team members.

The second key component of such self-organizing team setup is its shared identity and the alignment on goals they are trying to reach. As in its nature the team is supposed to work autonomously and organize itself without external guidance, it's necessary for it to have a common understanding of its desired end goals, so that all the individual contributions still end up directed towards it. Therefore, firstly self-organizing teams tend to collectively come up with the set of norms and principles that guide the way they work is conducted, including mutual agreements and understanding of working hours, team velocity, policy on defect tolerance etc (Hoda, 2011). This collective decision-making process is further crucial in everyone's self-commitment to the cooperative environment where the responsibilities, goals, and challenges are collectively embraced, as well as mutual sense of trust and respect is developed, committing to open exchange of ideas and the facilitation of effective collaboration. In addition, shared identity also plays a significant role in conflict resolution and decision-making processes

within these teams, where multiple different viewpoints are not only tolerated but valued as an essential component in the collective understanding of the bigger picture (Hoda, 2011). When team members share a common understanding of the project's mission, values, and expectations, they can communicate between each other more effectively, resolve conflicts constructively, and make decisions that derive from the mutual ideas. Finally, the sense of belonging and commitment sparked by a common purpose also motivates the team members. It facilitates engagement and support, as well as a sense of ownership and accountability for the delivery of both the individual tasks and collective outcomes, thus enhancing the overall productivity and success of the self-organizing team (Hoda, 2011).

To conclude this part, self-organizing teams within Agile frameworks suggest a dynamic and efficient approach to project development, characterized by adaptability, collaborative learning, and a sense of shared accountability to be successful. Such teams embrace the high level autonomy and a common identity, which enables them to respond to changing requirements swiftly and innovate continuously. The usage of practices like Kanban, collective planning or mutual decision-making not only enhances the overall team productivity but also fosters a culture of trust, mutual respect, and alignment on the collective goals. This environment, where diverse viewpoints are valued and team members are empowered to contribute actively on an individual level, results in a more effective, united, and innovative team output. Ultimately, the success of self-organizing teams in Agile methodologies hinges on their ability to balance between given autonomy and collaborative alignment in order to make the most of the collective strengths and creativity of their diverse members, thereby pushing the project towards its strategic objectives.

On the other hand, to touch upon the shortcomings of the mentioned research on Agile methodologies, it does as well provide more of a theoretical look into how these practices could contribute to the fostering of self-organizing environments. While the multiple benefits are mentioned of these methodologies, there still is a gap of a more detailed exploration of how those practices and details of them can be altered or customized to fit the particular IT product development project team environments in terms of contributing to different aspects of self-organization. In addition, it does not sufficiently address and elaborate on the practical challenges or solutions to them that particular teams might face when attempting to develop and maintain such environments in the real-world.

10. Research framework

In this critical section of the thesis, we delve into five research areas that are the core of our research into self-organizing practices in IT product development projects. The extensive literature and theory review above showed the existing gap between the theoretical ideas and practical applications of self-organizing practices, thus five different research areas were distinguished to unravel the practical experiences of self-organization in IT product development projects, with each of them aiming to gain insights on specific aspects of self-organization, from adaptability and responsiveness to the intricacies of leadership and team dynamics within self-organizing teams :

1. The impact of self-organizing practices on adaptability and responsiveness of IT product development project teams - This area of our research aims to explore how self-organizing practices influence a team's ability to adapt and respond to changes in the fast-evolving IT industry. It offers a deep dive into practical applicability of these practices and seeks to understand their impact on enhancing the team's agility and responsiveness within the context of IT product development projects.

2. The role of communication, information flow, and relationships in the success of self-organizing teams - Research on this area will investigate how different communication dynamics within self-organizing teams can impact their effectiveness. It will focus on how different practices of effective communication, seamless flow of information and enhancement of interpersonal relationships between team members contribute to the efficiency and success of these teams within the context of IT product development projects.

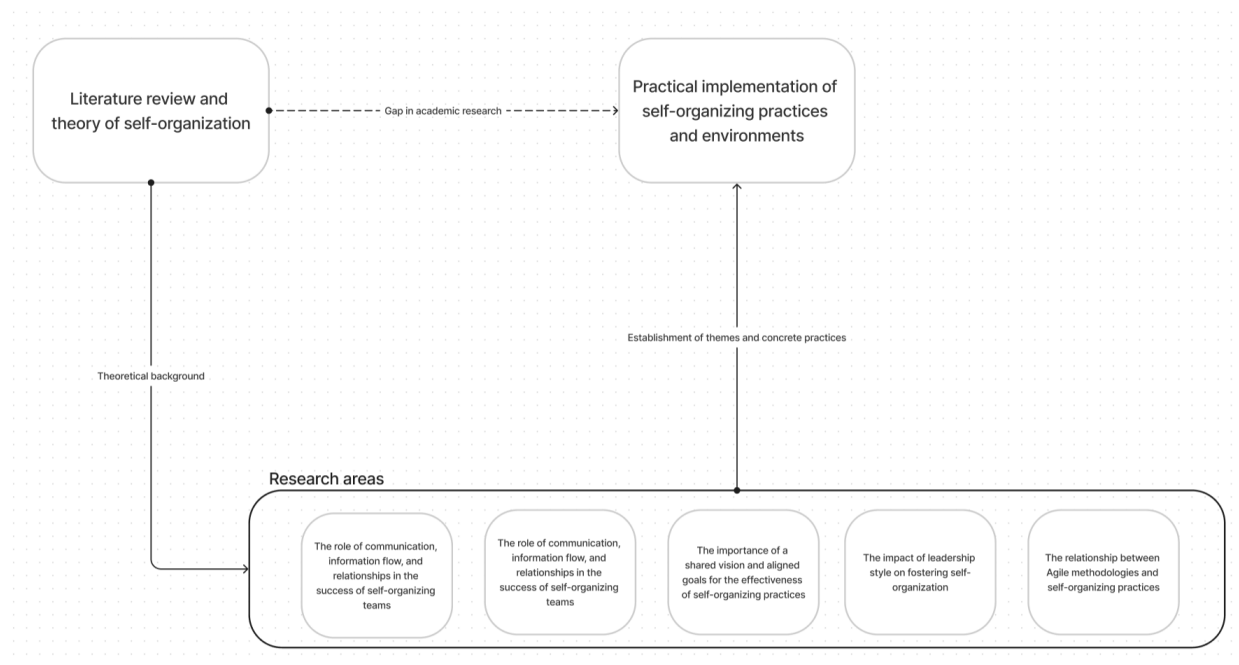
3. The importance of a shared vision and aligned goals for the effectiveness of self-organizing practices - This research area will examine how a common understanding of project objectives can help self-organizing teams to function more effectively. It will delve into the practices of maintaining shared objectives and alignment on the goals of the team within its members that are vital for the autonomous and effective functioning of self-organizing IT product development project teams.

4. The impact of leadership style on fostering self-organization - The examination of this research area will investigate how different forms and styles of leadership, particularly those emphasizing autonomy and decentralized decision-making, affect the development of self-organizing environments. It aims to identify the practices of leaders that cultivate such environments within IT product development project teams.

5. The relationship between Agile methodologies and self-organizing practices - This area focuses on the research whether the adoption of different Agile methodologies helps to foster a self-organizing environment. It seeks to determine if and how different and diverse approaches of Agile practices contribute to the successful implementation of self-organization in the context of IT product development project teams.

Figure 1.

Conceptual research framework



Source: The author of the thesis, 2024.

The methodology to explore these research areas and bridge the gap between the theory and practices, as well as to gain insights into the practical application and effectiveness of

self-organizing practices in the IT product development project team is described in the following section.

11. Methodology

The primary objective of this research is to explore and understand the empirical realization of self-organizing practices within the IT product development teams and provide further insights as well as bridge the gap of this academic field. The following research method was chosen to fulfill it:

1. Literature review was conducted to determine the background of self-organizing practices and their origin.
2. Based on the literature of self-organizing environments, teams and practices in different contexts, the described five research areas were derived.
3. To investigate the derived research areas and explore the practices of self-organizing teams in IT product development projects the qualitative research method was applied.

11.1 Qualitative methods

This thesis employs a qualitative research method by the usage of semi-structured interviews with directors/leads of product development projects in the IT industry to collect the necessary data focused on the use of self-organization in their teams. This approach is suitable here due to its alignment with the perspective that individuals tend to construct their reality through different interactions in their professional environments (Merriam & Tisdell, 2016). As the aim of this research is to explore the subjective experiences and perceptions of individuals regarding self-organizing practices in their work environment, it involves a detailed research of how these professionals understand and apply the principles of self-organization in their day-to-day activities within IT product development teams. The focus through semi-structured interviews is on understanding the nuances of their experiences, the self-organizing practices they use, their adaptation and the impact these practices have on the overall functioning of their teams. Furthermore, the qualitative research method, when realized through interviews, particularly enables the in-depth exploration of complex topics (Merriam & Tisdell, 2016), as it

emphasizes descriptive, detailed insights into individual experiences and perceptions of the respondents. In our context of IT product development project management, such interviews are essential for delving deeper into the practical experiences of using self-organizing practices, which may be pretty unique or context-specific. In addition, this method is also beneficial in this research because of its less strict structure, enabling the use of follow-up questions and generating more authentic responses, thus fostering an understanding and bond between the researcher and participants (Melles, 2005).

11.2 Case study

In academic research, particularly within the social sciences, the case study method has emerged as a valuable tool for in-depth investigation of real-life events such as organizational and managerial processes (Yin, 1994). To elaborate, a case study method is a unique way of observing any natural phenomenon which exists in a set of data, where unique is described to be a small area or number of subjects of interest that come from a particular context (Yin, 1994). Differently to various quantitative analysis which observes patterns within the data at the macro level, case studies observe the data at the micro level, focusing on the phenomena in particular settings. Furthermore, this method is particularly suited for situations where the boundary between the phenomenon and context is not clearly evident, allowing researchers to examine the 'what' and 'how' questions of the particular issues or relationships (Yin, 1994). Its strength lies in the ability to provide detailed insights and in-depth understanding of such phenomena like self-organization, which are often unattainable through other research methods. Characterized by its focus on exploring and understanding complex issues in their real-life settings, the case study method enables researchers to gain a comprehensive understanding of the intricacies involved in a particular case, often through qualitative methods like interviews.

Due to the goal of this research being to investigate and gain insights on the usage and various dynamics of self-organizing practices in certain IT product development project contexts, the case study method was chosen as the most appropriate one to be used. The certain IT product development context here was chosen to be a notable organization within the IT industry, focused on the delivery of different high-value online cybersecurity and privacy products. It was deemed fit for this purpose due the amount of different companies and products that this

organization offers to its B2B and B2C clients. Furthermore, all of the products within the organization are completely separate and have their own distinct teams, thus allowing to gather the data from different segments on self-organizing practices within the environment of this organization.

11.3 Source and sampling

As the main data collection source, the semi-structured interviews were conducted remotely through the Google Meets platform and video-recorded. All recorded interviews were only accessible and transcribed by the researcher to ensure a thorough and accurate analysis of the data.

The interviews were based on three main parts - introduction, questions and closing. For this, the list/guide of questions was compiled to be followed during the interviews themselves (annex 1). During the introductory part, every respondent was introduced to the topic as well as the aim of the research, and gave the consent for the responses to be used in the thesis. The main part of the interview is divided into three parts - during the first one, a few background questions were asked to understand the role of the respondent within the IT product development project team as well as gain an overall insight of how the respondent feels about the shift towards self-organizing practices. The second part combines multiple open questions to get the information relevant to the derived research areas, it must be also mentioned that questions were grouped in a way that focus of the conversation would thematically go from one area to another. To conclude the main part a few closing questions were asked in order to get the overall thoughts from the respondent about the positive and negative aspects of self-organization in their experience. Every interview then finished with thanking the respondent for their time and participation in this research.

To select the participants for the interview, purposive sampling method was used, following the recommendations of Dörnyei (2007), which emphasized that qualitative research involves participants who can provide rich and varied insights, with an ideal sample size of six to ten informants. As further explained by Cohen et al. (2007), non-probability purposive sampling does not aim for complete randomness in the selection, but rather focuses on selecting a specific portion of the population particularly relevant to the research topic. Therefore, not everyone

within the broader population had an equal chance of being included in the study. The respondents within the mentioned organization were selected based on the researchers' knowledge and public information of the fact that the participants possessed relevant and diverse experience on self-organizing practices within the IT product development projects. The selected individuals shared several characteristics:

- **Project Management** - Each participant managed a project team responsible for the development of a successful IT product. The success of their respective products was determined by their ability to be stand alone products in the market due the sufficient user base, generated revenue and user demand. Unfortunately, further details regarding these criterias of the selected products are confidential information, therefore could not have been disclosed.
- **Self-organization** - All participants worked in an environment which is not based on traditional management practices with strict hierarchy and long term planning, but rather in an environment that emphasizes self-organization, cross function, collaboration and autonomy.

To finalize, by employing purposive sampling, this research ensured access to a group of 7 individuals with relevant knowledge and experience about self-organizing practices within the context of IT product development projects:

1. **Respondent no. 1** - 7 years of experience in managing IT product development projects; Has been working in the current self-organizing environment for 2 years.
2. **Respondent no. 2** - 9 years of experience in managing IT product development projects; Has been working in the current self-organizing environment for 2 years.
3. **Respondent no. 3** - 10 years of experience in managing IT product development projects; Has been working in the current self-organizing environment for 2.5 years.
4. **Respondent no. 4** - 12 years of experience in managing IT product development projects; Has been working in the current self-organizing environment for 2 years.
5. **Respondent no. 5** - 5 years of experience in managing IT product development projects; Has been working in the current self-organizing environment for 2.5 years.

6. **Respondent no. 6** - 13 years of experience in managing IT product development projects; Has been working in the current self-organizing environment for 5 years.
7. **Respondent no. 7** - 6 years of experience in managing IT product development projects; Has been working in the current self-organizing environment for 3.5 years.

This approach is aligned with Dörnyei's (2007) concept of homogeneous sampling, where participants share specific experience to the study, in this case, their leadership roles in successful product development projects within the fast growing groups products.

Furthermore, the author of the thesis followed the research ethics in a few ways. People's consent to participate in the study and engage in recorded interviews was received by personal LinkedIn messages to ensure the confidentiality of the respondents. Before asking any interview-related questions, every participant would be reminded that they may not answer some questions, if they were not feeling comfortable, however, no such case was recorded. Further anonymity was guaranteed by coding interviewees' names with numbers, as provided above. Also, if any other sensitive information, like company or specific product names were disclosed throughout the conversation, it was coded with phrases like 'the company', 'the product' or 'the organization'. It was also guaranteed that the research data (recordings and transcripts) would be accessible only to the author of the thesis and not shared with anyone else, except the obligatory anonymized transcriptions, which were added as an annex to the final paper.

11.4 Thematic analysis

After the interviews were conducted and transcribed using an online tool - Notta (transcriptions provided in Annex no. 2), a further qualitative research method of thematic analysis was employed to analyze the data collected from directors/leads of IT product development projects within the chosen organization. Thematic analysis allowed for an in-depth exploration of the participants' perspectives and experiences regarding self-organizing practices in their project management environments, through the development of common themes and practices in the responses. The data grouping and analysis was done in accordance with its relevance to the interview questions and investigation of the research areas raised for this thesis, as well as the development of themes concerning practical application of self-organizing practices within IT product development projects.

12. Results

In the following sections of the thesis the data collected from the interviews with IT product development leads on self-organizing practices from different companies and products of the chosen organization will be analyzed, interpreted and presented. Each section will provide the findings of each research area regarding different aspects of self-organizing practices, as well as their descriptions and insights on certain practices derived from the collected data.

12.1 Adaptability and response to change

Having already elaborated on the landscape of IT product development projects characterized by their high level and variety of complexities, the theoretical part emphasized the agility and responsiveness of teams to be crucial for the success in such projects. It drew this conclusion on the fact that businesses in the IT sector face ever-changing market conditions and project requirements, thus the need for adaptable and efficient team structures were noted to be paramount to survive in this environment. Such a necessity brought us to the research on the **impact of self-organizing practices on adaptability and responsiveness of IT product development project teams**. The analysis of the responses from the conducted interviews provided multiple aspects related to this research question.

First of all, flexibility in methodology came up as a key concept brought by self-organization of the IT product development project teams that may contribute to the success of the project itself. The underlying idea here is that the teams themselves are firstly flexible in the way they organize their work. Different teams are given the opportunity to try, adapt and switch between different approaches of how the tasks are carried out, throw out practices that don't work for particular groups and decide on what practices suit their specific needs, circumstances or challenges.

Respondent no.1: *“I worked with multiple teams, for example, we started working in Sprints, we saw that it's not working here and we moved to Kanban, which made us work way faster, for example. And with another team we worked with Kanban and we were seeing that it's not working for us so we moved to Sprints and that optimized our development, our delivery. So it really depends on each team to find the way for you to organize your work.”*

Giving this freedom to the teams to find their own way of working in a self-organizing way leads to a more efficient teamwork in general, as teams are able to “play to their strengths” - the workflows are built or adjusted accordingly with the particular dynamics and strengths of the teams. In this way, the setup is oriented towards being able to gain maximum advantage from the processes where the team is actually good at and where its core expertise lies. In addition, a level of continuous learning is ensured within such teams, as by going through various methods and refining their ways of working, the group gathers different experiences which can later help to deal with certain scenarios and make the necessary adaptations in order to reach the goal.

Respondent no.2: *“Yeah, so what it (self-organization) gives, it gives a bit of a freedom to change what you do and how. So it's not set for them, you know, you must do this and this and this and this and this. But there's some sort of a north star set that we're trying to go with our team towards.”*

In general, by allowing the team to choose methodologies and practices that best fit their current group or project situation, teams are enabled to work more efficiently, as this flexibility allows quicker deliveries because the team can avoid being stuck to a method that may not be the most effective for their specific situation.

Respondent no.7: *“I think this is very important as well because at no point during the development or planning, there should be an argument that, hey, we always used to do that, or hey, we agreed to do that, and that's why we will do it. Like if we have new information that shows that a particular approach is no longer the most optimal one, or no longer makes sense, then it means that we need to adjust accordingly”.*

Another aspect in the research of how self-organization helps to increase the teams adaptability and responsiveness to various changes lies within the autonomous characteristic of such teams. As already established in the theoretical part of the thesis, autonomy is a key pillar behind a self-organizing environment, where its members are empowered to make decisions on an individual level and deliver their work successfully without strict overview or external guidance. A very similar connection was also emphasized by the respondents of this research,

where autonomy was mentioned to be a direct source of empowerment, as every respondent suggested, that the teams that they lead have their full trust to organize their work and make decisions. Delving deeper into such an approach, it became apparent that the key thing behind this is the understanding that the given autonomy also inflicts a certain degree of accountability and people naturally become more involved to do things faster and better.

Respondent no.1: *“So that's the key thing, that when you feel that you're accountable and responsible for that thing, so you are more eager to solve it in a way that you feel that it's going to work or you do more analysis on that, rather than when somebody comes and say you need to do it according to this, and they don't even know if it's the right way or not.”*

With this rising responsibility and power to organize your line of work, a sense of ownership is developed, but it is not just about controlling the tasks that have to be done, but also about being deeply invested in the quality of the outcomes as well. Meaning, that team members who feel a sense of ownership are more engaged, more motivated to deliver the work and collaborate, and this is a crucial boost towards the success of any project.

Respondent no. 4: *“So that is by enabling and involving the team in all that process. So that they make the decision as well the investigation and they understand from the very beginning, they are aware of the new requirements coming in. <...> So it's not like the development team is just working on moving the tickets from to due to done, but as well being involved in all the research and making the decisions or contributing as well to the decisions how to be implemented.”*

Respondent no. 1: *“Because only then the team can find the best way forward. So if you have a problem, everybody has their own specialty, technical specialty, business specialty, process speciality. And if every person can contribute, it's way, way faster or easier to solve the problem.”*

In addition, autonomy also encourages self-organizing teams to develop and enhance their problem-solving skills, as when faced with different challenges, they need to navigate the issues,

find solutions, and make decisions without relying on directions or instructions from above. In a longer term this highly contributes to the responsiveness of the group, as they develop their abilities to deal with the problems on the individual level, allowing for a more swift response to the emerging challenges or capture of opportunities relating to the project.

Respondent no.3: *“It's that usually employees take responsibility on their own and they don't always ask the managers for help with this and that. It's within their sort of responsibility. It's sort of relying on that expertise on the individual level.”*

Respondent no.5: *“And they in some cases are pretty autonomous. For example, if the Apple team finds some new possibilities which Apple introduced itself and we can adapt to our product. They are trying to do that. So that's a good thing because in some cases I as a manager don't need to catch up with everything around. The teams just come to me and they say look we have this new stuff that we can implement maybe we can do that, so we agree on that.”*

While naturally not all of the problems can be solved individually, such situations also foster a stronger sense of team cohesion, which is also a very important aspect in the adaptability of self-organizing teams. Here team members have a greater need to collaborate in order to make decisions and solve the problems, thus they build trust and a sense of camaraderie among the group. Putting this into play where teams that are autonomous are also directly accountable for the results of their work, can really act as a powerful motivator, as team members know they will be answerable not just for their actions but also for how these actions impact the project's success.

Respondent no.4 : *“So if there is change, then I would say our team is still building, but it is already in that pretty much self-organizing part where they have lots of open discussions. And yeah, basically that helps to solve those uncertainties, all those changes quite fast and to agree how we move forward.”*

Respondent no. 7: *“But I don't think I've been very much involved with the last, I don't know, six to nine months. And it's been essentially just happening and getting it solved. People*

responsible for it now have my trust to do it and that's it."

Therefore, it is clear that fostering autonomy within the group often leads to higher levels of team engagement and job satisfaction, as people embrace the fact that they have a say in their work and can see the direct impact of their decisions. People are more encouraged to discuss, debate, and constantly brainstorm different ideas on how to self organize their work and how to arrive at the best possible decisions.

To summarize, the research on adaptability and responsiveness fostered by self-organizing IT product development project teams led to a discovery of two themes and their sub themes that collectively contribute to the successful and efficient execution of IT projects in an ever-changing and demanding technological landscape.

1. Flexibility in Methodology. Self-organizing IT project teams exhibit a high degree of flexibility in their work methodologies, which allows teams to adapt and switch between different approaches based on what suits their specific needs. By granting teams the freedom to find their own ways of working, self-organization leads to more efficient teamwork, as teams can play to their strengths, building workflows that maximize their expertise and proficiency. This setup not only ensures efficiency but also fosters continuous learning, as teams go through various methods and refine their ways of working, gathering experiences that aid in adapting to different scenarios.

Firstly, this can be realized through giving teams full freedom to use different methodologies, try to customize them and switch between different approaches of how the tasks are carried out, without any burdens to throw out practices that don't work for them. Such practices involve allowing particular groups to experiment with their workflows and decide on what practices suit their specific needs, circumstances or challenges. Secondly, this could also mean adopting one agile methodology in one phase and shifting to another in a different phase, depending on what maximizes their efficiency and effectiveness, as well as leveraging their unique skills and expertise. Finally, such practices fosters an environment of continuous learning as teams experiment with various methods, they accumulate a vast amount of experiences, which becomes a valuable resource for tackling future projects and challenges.

2. Empowerment through autonomy. Self-organizing teams are characterized by a significant level of autonomy, which is not just about controlling tasks but also about being

deeply invested in the quality of outcomes. Team members who feel a sense of ownership over their work are more engaged, motivated to deliver high-quality work, and collaborate effectively. This sense of ownership is a crucial factor in the success of projects, as it empowers team members to take initiative and contribute meaningfully to the project goals. Furthermore, embracing autonomy and raising the sense of ownership and responsibility of the teams allows teams to adapt more quickly to changes, whether they are market-driven or internal. As the decision-making is decentralized, the responses to new information, changing customer needs, or technical challenges can happen on the team or even at the individual level without having to wait for approval from higher-ups.

Creating such an autonomous environment contains a combination of practices that firstly involves defining clear responsibilities of team members. People have to be made aware of what outcomes they are responsible for, defining clear quality requirements for them and ensuring them of their capabilities or expertise to deliver them. Secondly, they have to be involved in decision making and research processes, so that they are aware of the direction, changes or possible upcoming challenges as early as possible, as well as deciding on how it will be implemented. Providing this power to control their work is necessary to further enhance their commitment and ownership of their deliverables and contributions towards the team goals, which in a longer team can lead to fully autonomous decision-making. In addition, all the necessary team resources and support should be provided, including tools, information, and assistance or expertise they need from other colleagues to make informed decisions and carry out their tasks effectively. Finally, the practice of experimenting should be welcomed, so that individuals could look for innovative solutions to emerging challenges or take initiative and capture any occurring opportunities that contribute to the overall success of the team.

In conclusion, the following themes and sub themes, consisting concrete practices were derived in this research area:

1. Flexibility in Methodology
 - a. Freedom to use, customize and get rid of different methodologies
 - b. Freedom to adopt one agile methodology in one phase and shift to another in a different phase
2. Empowerment through autonomy
 - a. Clear definition of responsibilities of team members

- b. Direct involvement in decision making, research and other processes
- c. Necessary support of autonomy through information, tools and expertise
- d. Freedom to innovate

12.2 Communication dynamics

In the rapidly evolving industry of IT product development projects, the dynamics of team organization and management play a critical role in determining the success of these endeavors. Taking into account the discussed high level of autonomy and decentralization that a self-organizing team is based on, the team dynamics here become a major cornerstone for the efficiency of such a setup. The theoretical part of this paper established three significant components - effective communication, flow of information and relationships between team members - which all together contribute to the constant development of new ideas and approaches, thus allowing for self-organization to happen within the team as well as driving the organization forward. The research on the **role of communication, information flow, and relationships in the success of self-organizing teams** allows for a deeper look into how different communication dynamics within the self-organizing teams contribute to the efficiency and effectiveness of self-organizing practices.

To begin with, the significance of employing a variety of digital tools and platforms to enhance communication and information flow within their self-organizing teams was highlighted. All interviewees mentioned the use of multiple communication tools like Slack, Jira, and Confluence, which are critical for ensuring effective communication and unrestricted flow of information. To streamline this, a three layer system could be visible within these self-organizing teams. Firstly, tools such as Slack here serve as the foundational or the individual layer for instant messaging, quick and informal communication, which facilitates daily real-time interactions and fast information exchange.

Respondent no.6: *“We use slack for our daily communications and for quick status updates.”*

Respondent no. 3: “<...> we use Slack also mostly for quick communications, like, oh, who can take this task? Or is there like a quick question, can we do this? So quick questions like that...<...>.”

However, while having a convenient tool for quick messaging ensures that information flows efficiently among team members and fosters quick decision-making, it can also foster a challenge to maintain transparency and availability of information. A Tendency to engage in private or direct chats can easily create situations where specific information is kept only there and not available to other relevant team members, thus creating disparities in knowledge and understanding within the team. In addition, a further risk of miscommunication arises, as people might later on receive different or insufficient versions of information, creating confusion and misalignment. Therefore, an important practice was underlined here, which revolves around putting emphasis on using public channels and keeping information visible for everyone, rather than trapping it in private conversations.

Respondent no. 6: “*What I always encourage my team to do is to use public channels instead of direct messages. Avoid direct chats one on one, so that information is public and available for everyone. Just to avoid any miscommunication.* “

Respondent no. 7 : “*So ideal scenario would be to mostly use channels and we do have channels for different topics. So again, this makes sure that people are aware like what's going on and sometimes they just need an FYI.*”

Respondent no.4 : “*So basically by the end of the day, we had the situation solved. So it really helped that all the conversation was happening under the same thread, sales people seeing the same information, support teams seeing the same information, developers, QA, etc. There were no, you know, hidden conversations in private or whatsoever.*”

And while using public channels where more people are involved might not be the most convenient practice for some people, but it really contributes to efficiency of the team, especially in the cases where a certain problem has to be dealt with. Having this openness ensures that all

team members have access to the same information, which is crucial for self-organizing teams where autonomous decision-making is key.

Moving forward, employment of similar platforms like Jira, forms the middle layer which focuses on the overall project and task management part, which is essential for tracking progress of the work, assigning responsibilities and maintaining a more structured overview of the project. It brings a level of clarity and a more systematic approach to the way the work is actually organized which is crucial for maintaining a constant oversight of the project's trajectory and ensuring that all team members are aware of their roles and deadlines.

Respondent no. 5: *“Another thing is JIRA, of course, like different project or product management tools, I mean online tools for example.”*

Respondent no. 6: *“We use Jira for the whole product and feature implementation and planning and delivering and QA. So we have a process in Jira”*

Finally, Confluence and similar tools constitute the top layer which acts as a comprehensive knowledge base and repository for different documentation, plans, and other vital information, ensuring organized storage and long-term accessibility of compiled knowledge. It allows teams to be a few clicks away from accessing information about the work concluded in the past, ongoing tasks and thoughts or research behind them, as well as be aware of the future plans and ideas of the project.

Respondent no. 4: *“First of all, we use Confluence for all of our documentation. So, as I mentioned, we do the research, then we move based on the research results, we decide do we want to continue with the development or we pause or we say that we will not look into that at that moment. So, already within the research stage all the documentation is available for the team members.”*

Respondent no. 6: *“So that element and I think another thing that's proven to be quite a bit of success is a lot of written down communication that we use in Confluence right now. And I*

think this is a practice that has been successful in sort of making transparent and liberally use of information that's where everyone can access what's important.”

In general, this tiered approach creates a sort of an ecosystem where information is able to flow seamlessly from immediate communication to more structured task management processes, and finally to a long-term knowledge storage. It helps to address the core needs of such IT project teams for smooth communication, organized task management, and accessible documentation, thereby enhancing overall team efficiency, transparency, and cohesion. This system not only streamlines the workflow of the team but also contributes significantly to the project's success by helping to ensure that all team members are informed, aligned, and efficiently contributing to the common goals.

Respondent no. 1 : *“I would say that if you don't have descriptions and this information about the project, it's very hard to build something. <...> So that was the key of some of our problems, that we didn't know how the product worked in the first place. Right now, when something is being built or something is done, we have a clear description so we can come back to it after a year or so. I sometimes come back to some tasks after a year, check how it was built and check if it's working correctly or not. So without that, it's pretty hard to at least function as a team, because you don't know how anything works.”*

On the other hand, while this system already creates a solid framework for the teams to work efficiently due to the availability of information in different layers and contexts of the project, issues on an individual level can occur. In detail, such a setup still requires people to put in individual efforts into looking for this information, but also takes their time to contribute to updating it, preparing different documentations, writing up reports and lastly making others aware of its availability. To tackle this, another pillar supporting this setup in the form of meetings appeared to be of significant influence in the researched self-organizing teams.

Respondent no. 5: *“So in a way it's a setup including meetings that we catch up on stuff and then there is also tools like Confluence that we use to keep basically this knowledge base that has all the information available at any times to anyone needing it.”*

Respondent no. 6: *“And yeah, we have daily standups for reporting, for just sharing what we are working on and if we have any blockers”*

Respondent no. 3: *“We have weekly meetings where we discuss high level stuff, and the team can always create their own meetings.”*

In the dynamics of these teams, meetings provide a regular and consistent place where team members can catch-up, vocally share their current work status, highlight accomplishments, and more importantly, bring up any problems they are facing. Such daily meetings and touch points are essential for maintaining the momentum of the project, because they ensure that any issues are identified and addressed quickly, thus preventing any minor challenges from escalating into major roadblocks. These meetings also helps everyone to be on track with the team's daily objectives, as well as proactively keeps everyone aligned and focused on the ongoing tasks and the roadmap of the project.

Furthermore, given the complexities of IT product development projects, having these brief daily catch-up's, which are focused on ensuring that the team is synchronized in their day-to-day efforts, are not a sufficient platform to get hold of the bigger picture. Therefore, having more strategic meetings offer a different but equally an important dimension to team communication. These sessions happen on different periods and are typically more in-depth to discuss broader project goals, long-term objectives or specific problems on different levels, including different or more specific groups of people.

Respondent no. 7: *“Then we have a team level. So we have team level check-up's on a weekly basis mostly. <...> So like within the teams, they can organize whatever makes sense for them, but on a whole level, the team level sort of checkups, again to see how we progress through our OKR's. <...> And then we have the weekly organization level catchups. So this is again to bring everyone just a bit more context of what's going on in different parts of the organization because like if you're working on product, you might not necessarily know what's going on, marketing side, vice versa.”*

Respondent no. 2: *“So every two weeks we have product managers assembly where we share what the topics are, maybe there's some shared project that we need to take care of, there's maybe an issue on one squad field causing an issue on another squad, then other processes how*

we release things together and all that. These are all needs to be aligned. So we have these product assemblies every two weeks and then we have every month assembly with product engineering managers. So both development parts and product parts comes together. Then on the whole product level, every two weeks we have all hands meeting where one of those meetings, like every month, we take the two types of meetings. <...> So everyone sees the same picture, you know, and then we try to connect everything what's done in the squads, towards what we actually trying to achieve here in general.”

This wider approach to meetings is a crucial addition to the communication that happens via different online tools, making sure that people have a platform to discuss and align on things vocally or even face-to-face. In addition, it’s an important aspect of self-organizing teams, where autonomy and collaborative decision-making are key, as it ensures that while team members have the independence to manage their daily tasks, they remain attached to the collective objectives and strategies of the project.

The final part of team dynamics that influence the effectiveness of self-organization is the enhancement of relationships between team members. It’s apparent that building strong interpersonal relationships between team members firstly facilitates easier collaboration and support, as when those connections are established, team members are more inclined to assist each other.

Respondent no. 1 : *“I think it's way easier to work with people which you build the relationship. Otherwise, if you have some issues, if they don't know you, people tend to either ignore you or they're not very keen to help you in a way because they don't know you”.*

Respondent no. 2 : *“I think it's the key. There's some team members who sits together and the more this close relation between team members exists, the better, because they can understand each other from the first sentence or know exactly the person where to go to consult.”*

Having these relationships fosters a supportive environment, where people help and understand each other a lot better, communicate more efficiently, thus also accelerating the problem-solving and decision-making processes.

Furthermore, when team members build relationships, they get to know each other beyond their professional capacities, a mutual understanding and empathy can be developed, thus making it easier for team members to trust one another. As they better understand each other's strengths, weaknesses, and working styles, there's an increased confidence in each other's abilities.

Respondent no. 5: *“So I think that it's very important also when you're developing something and you're especially talking a lot with engineers from what I see it's quite important that people, engineers especially that you're working with would have some trust in you. That they would trust that your decisions are good and of course in some cases they need to raise some questions, but if they do not trust you not only as an expert but as a person as well it can be very complicated because then developers may have lots of questions why we need to do this, this won't work and sometimes the outcome can be not as good as in the case where you have that relationship.”*

Also, having this trust based environment is crucial for fostering the autonomy which as already established, is an integral part of the self-organizing teams. Therefore, team members who trust each other are more comfortable to delegate tasks, make decisions independently, while at the same time empowering each other to take ownership of their work.

Respondent no. 6: *“Relationships are well built, then your delivery will be effective and you will probably notice any blockers or issues happening sooner, like earlier in the stage. And of course this unofficial relationships, like having team buildings, similar hobbies or interests is also contributing a lot because it's just a better atmosphere in the team, less stress and yeah, I think it helps a lot. Sometimes if you have a difficult moment, something is not happening as we want to be. And the team is built in a good way that they do not stress and they even find the way how they can make a laugh out of that, it helps a lot”.*

Respondent no. 7: *“So like questioning, sharing feedback and so on, it's a lot more difficult when you don't have this sort of personal relationship built with the people because they*

might take it as a sort of challenge or like being called out and so on. And once you have this relationship, we can actually discuss on the ideas and on the topics and not take it personally.”

Finally, in the environment where trust prevails, conflicts and feedback are handled more constructively, people are enabled to give and receive feedback without taking offense, understanding that it is aimed at improvement rather than criticism. This aspect of trust is crucial for teams to evolve and improve continuously, as well foster a nicer team environment. When team members trust each other, there's less worry about potential conflicts, misunderstandings, or the reliability of others' work, which can reduce the stress levels and contribute to higher team morale or job satisfaction.

To summarize, the investigation of the research question on whether the fusion of effective communication, flow of information and enhancement of relationships are critical factors for the success of self-organizing teams in IT projects two key themes were developed:

1. Importance of efficient communication and the flow of information - it was emphasized that communication and the seamless flow of information are crucial in boosting the efficiency of self-organizing teams. Ensuring that information is able to efficiently flow among team members, fosters quick decision-making and responsiveness on the individual level within the team. In addition, having the ability to track progress of different ongoing initiatives brings more clarity and structure to the way the work is organized, which is crucial for maintaining a continuous review of the project's trajectory and ensuring that all team members are aware of their needed contributions. Finally, it was established that equal and open access to vital information at any time to all team members, is a crucial part of such a setup, creating an environment of transparency and ensuring that people are able to quickly find details about past work, current tasks, and other collected knowledge.

To put this into practice, a three layer system was derived that utilizes a diverse array of digital tools like Slack, Jira, and Confluence, that ensure all aspects of information flow. These tools are strategically integrated to optimize the communication processes: tools like Slack facilitate instant messaging and quick exchanges between individuals, usage of Jira enables constant project and task management to happen, and platforms like Confluence serves as a comprehensive and accessible knowledge base. This system establishes a structure that enables self-organizing teams to work more effectively, thanks to the accessibility of information across various levels and aspects of the project. Furthermore, a practice of using public channels was

highlighted, to counter the risks of transparency loss and information silos that private communications might foster. Finally, the combination of various types of meetings - from daily stand-ups to strategic sessions - complements this system by providing regular platforms for synchronization, problem identification, and alignment on project objectives, further bolstering the team's efficiency.

2. Enhancement of relationships between team members - the second theme involves the enhancement of interpersonal relationships within teams. Being engaged in strong relationships with team members facilitate easier collaboration and support, as team members are more inclined to assist those they have a good connection with. These relationships foster a supportive environment, more fluent communication, and more efficient problem-solving, as well as quicker decision-making processes. The role of mutual understanding and empathy developed through these relationships in building trust is also crucial in fostering team autonomy and empowerment, as it makes team members more comfortable in delegating tasks and making independent decisions. Additionally, trust-based environments are beneficial for handling conflicts and feedback constructively, reducing stress levels, and contributing to higher team morale.

Firstly, the enhancement of these relationships can be achieved through time with more official practices of working together or actually having the team to physically work next to each other, which naturally sparks a closer relationship as they have to deal with each on a daily basis. On the other hand, unofficial relationships play a significant role as well, such as those formed through team-building activities, shared hobbies, or interests. These elements play a critical role in fostering a cohesive and positive work environment, especially in self-organizing teams, and they significantly contribute to team dynamics and more efficient workflow.

In conclusion, the following themes and sub themes, consisting concrete practices were derived in this research area:

1. Importance of efficient communication and the flow of information
 - a. Development of a three layer system to ensure the information flow through instant messaging, progress tracking and accessible knowledge base.
 - b. Use of public channels instead of practicing private messages
 - c. Use of different types of meetings for face-to-face synchronization

2. Enhancement of relationships between team members
 - a. Physical placement of team members next to each other
 - b. Involvement in team-building activities and shared hobbies

12.3 Alignment on vision and team goals

Previous parts of this thesis have already emphasized some of the important aspects and practices of self-organizing teams and different practices employed to ensure their efficiency. Therefore, in this part the focus will be put on the idea that sort of lies above the actual self-organizing teams and keeps them together. In detail, the **importance of a shared vision and aligned goals for the effectiveness of self-organizing practices** will be discussed. This part will examine and provide insights on the established idea that a common understanding and agreement on project objectives are crucial for self-organizing teams and its members to function autonomously and effectively.

The development of the shared vision and project goals was emphasized to be a pretty structured process within the research teams, as most of them employed different goal setting and tracking frameworks such as OKR's or KPI's, as well as inclusion of roadmaps. With the vision being set by the company for further years, these frameworks are put in place to break down the actual steps that will have to be taken in order to achieve it.

Respondent no. 7: *“So I think OKR's framework does a lot of the things here. It is that we kind of decide what's most important for us about the next year so it's usually being aligned with financial goals and based on that we then kind of break it down through impact mapping and what needs to happen for us to make those financial goals possible. Yeah I think that that does a lot of heavy lifting <...>.”*

Respondent no.1 : *“I would say through the roadmap. Our roadmap is connected and comes from the vision and to higher level OKR goals. So every team member, if they work on the task, they at least know from which initiative it's coming from. And that initiative is basically always contributing to some sort of an OKR or even the bigger vision or the goal of the project.”*

Having the set goals and building this organizational roadmap in IT product development projects is essentially a strategic plan that outlines the key steps or milestones needed to achieve the project's goals, and it's not a schedule of tasks, but rather a visualization of the journey towards the project's and the whole organization objectives. Most importantly, further breakdown happens with the help of OKR's framework, which revolves around setting a few important Objectives that will very directly contribute to the desired high level goal and the vision of the company or the product. Each Objective is then further broken down into smaller Key Results that different teams need to achieve in order to contribute to the established Objectives. Finally, different small initiatives are established and have to be completed so that the desired results are achieved. Such a breakdown structure is key behind actually getting everyone on board with the higher goals

Respondent no. 2: *“That is the vision part that and OKR's helps to connect those points as well. So, okay, in order to go to here, when we think that we need to achieve this objectives, become this, become this, become this. You have to have that structure of goals and KPIs on top of it, or next to it, it is really important.”*

Respondent no. 6: *“And yeah, we constantly talk about the same KPIs or our goals. So the vision is quite clear that in our case, it's basically happy and loyal existing customers. In a way it's a constant process of always reminding them through business goals and KPI's.”*

In the context of these IT projects, this linkage ensures that every task or initiative undertaken by the team or individual members of it are contributing towards achieving these higher-level goals. It helps people to remain on track with themselves as well as each other and actually see how and if their individual tasks fit into the bigger picture.

Respondent no. 1: *“Also, because all the tasks are connected, team can always look into the goal it is connected to and which key result is contributing to it.”*

Respondent no.2 : *“But there's also multiple things like, when there is these two squads which are working on separate things, but eventually it comes down to the very similar thing and connects. So there are some moments where you feel that, it's like, "oh nice", where they were*

working separately, but suddenly the features that they produced, connect, because it shares the same values, the same vision.”

As team members understand the significance of the work they are doing in the context of the project’s broader objectives, not only are they more focused, but it also contributes to a sense of ownership and accountability towards the overall success.

Respondent no. 6: *“So it has a very important role, not only because of the reminding constantly and sharing, but probably mostly because of these team members having the context to know what value will they bring to the users, to the company, so that they can feel maybe proud of it, so they understand that they did something, not just a piece of code, but actually they delivered the value to someone. So yeah, it has a huge role.”*

Furthermore, this approach not only clarifies the purpose behind each task but also helps in prioritizing work based on its impact on the overarching goals, as linking tasks to OKRs on the roadmap ensures that each team member's work is not seen in isolation but as part of everyone's efforts towards the shared purpose.

However, maintaining a shared vision through a roadmap and OKR’s is not a one-time activity but actually a continuous process.

Respondent no. 2: *“But again, as you say, you need to constantly remind, because people forget, there's new people joining, things are changing, so it's a constant work.”*

Respondent no. 3: *“No no, every two weeks during our weekly meeting, I spend about five minutes saying most important business topics, sort of general topics of where we want to go with the product and just reminding them.”*

Respondent no. 5: *“Well, we had, but not as much as I would like to see, but now the situation is getting better because we have a vision, we have clear goals. Also, we have achieved the thing that some people in some teams, were raising those questions themselves, for example, how does this initiative align with the goals? What OKR will be improved with this one? So, yeah, I think it's just continuous communication and continuously reminding about the stuff that we have <...>.”*

Teams have to constantly be reminded and communicated about the goals that the team is heading towards, review their progress against the OKRs and adjust their efforts as needed. These reviews allow teams to assess if they are on track, understand if certain approaches or initiatives are working or not, and make necessary adjustments to stay aligned with the overall vision and goals.

Finally, the most important benefit of having this alignment and understanding of the goals and how individual effort contributes to them is that it fosters autonomy and creates shortcuts in decision-making processes.

Respondent no. 3: *“Well, they usually know the things that we're doing, why are we doing it, and which sort of metric or user experience improvement that will contribute to and having sort of a goal of why are we doing this and why it will be useful, or why it was unsuccessful as well. Yeah, they have, I think, a bit more, how should I put it, they don't have to communicate as much with the rest of the team because everyone sort of understands where we're going. And if issues arise, it's a lot easier to solve them because this is our business goal, this is where we want to go.”*

Respondent no. 7: *“Again, it's a sort of shortcut in decision making, so it saves time. You don't need to cross-check, double-check before making decisions. You know that if this is aligned, then this is where we're going, and even if it's wrong, it probably won't be very much off.”*

Respondent no. 3: *“This agreement of a certain development topic is going to help in achieving that business goal. And it helps solve issues a lot faster and not go deep dive into meaningless conversations, long chats and so on. So it's just a lot faster if everyone, technical people know business goals as well, which is not that easy to do, but they at least have an idea.”*

Having this clarity in purpose of the team's work streamlines the problem-solving and development processes, as team members which understand the shared vision and goals, can align more swiftly and efficiently on different decisions. Additionally, a shared vision simplifies decision-making by acting as sort of a reference point that everyone relates to, which contributes to reducing the time spent in checking, aligning and discussing the need. In such an environment

team members are enabled to work with confidence that their actions are in alignment with the team's goals, enabling a more effective and successful self-organization.

To summarize the research of the development and presence of a shared vision and aligned goals within IT project teams and its members in the context of self-organizing IT product development project teams, three key themes emerge as critical for enhancing team effectiveness and efficiency:

1. Structured development of shared vision and goals. The structured development of a shared vision and goals stands out as a critical factor for team success. This process revolves around breaking down high-level goals into smaller, actionable segments, which is fundamental for aligning team and individual efforts with the overall project vision. It also transforms broad objectives into clear, manageable tasks, enhancing team focus and setting a direction, as well as simplifying the measurement and tracking of progress, making goals more tangible and achievable. This approach in goal setting and alignment clarifies individual roles and responsibilities, fostering a sense of accountability and ownership among team members. By ensuring that each smaller task contributes to the larger goals, this method makes it easier to maintain consistency and focus towards the common objectives on team and individual level, which is crucial for the project's success.

To achieve this, a practice of employing robust frameworks like Objectives and Key Results (OKRs) and Key Performance Indicators (KPIs) was established, which fosters a clear and detailed roadmap that outlines key milestones and objectives. The use of OKRs, in particular, provides a clear structure for setting, tracking and breaking down goals, as well as narrowing down the needed efforts to small initiatives, which directly contribute to the desired success. This structured approach not only aligns the team's efforts but also makes the shared vision more tangible and achievable, thereby enhancing overall project coordination and direction.

2. Constant communication about goals is essential to maintain a shared vision within IT project teams. It became apparent that the establishment of set goals and adopting certain frameworks by themselves is not a sufficient guarantee of success in these teams. Therefore, continuous communication emerged as a necessity to ensure that team members are not only made aware of the objectives at the beginning, but are also constantly reminded about them to make sure they understand their role in achieving them and remain focused.

Such practices as weekly or monthly team meetings, status updates, and regular reviews of OKRs progress serve as platforms for this continuous process, reinforcing the shared vision and goals. Keeping these goals on top of the team's mind, ensures that everyone remains on the same page, contributing to the team's effort towards the project's success.

3. Shortcut in decision-making. Finally, the well-defined and constantly communicated shared vision acts as a shortcut in the team's decision-making processes. When team members have clarity about the end goals and their individual role in achieving them, they can make decisions more efficiently and autonomously. Employing these practices of clarity and alignment on the shared vision reduces the need for extensive discussions or approval for every decision, allowing team members to quickly solve problems and self-organize. Such an approach not only speeds up project execution and delivery of the tasks but also empowers team members by fostering a sense of ownership and responsibility, as by being aware of the impact of their decisions on the project's success, team members can operate with greater confidence and efficiency, contributing significantly to the project's progress and overall success.

In conclusion, the following themes and sub themes, consisting concrete practices were derived in this research area:

1. Structured development of shared vision and goals
 - a. Employment of OKR or KPI frameworks
2. Constant communication about goals is essential to maintain a shared vision within IT project teams
 - a. Regular weekly or monthly meetings to track progress within the OKR/KPI frameworks
3. Shortcut in decision-making
 - a. Autonomous decision-making

12.4 Leadership

Moving forward, the creation of a self-organizing environment requires a specific and complex leadership approach, which as established in the theoretical part of the thesis, is composed from different aspects of adaptive, administrative and enabling leadership styles.

Those include fostering an environment of innovation through change and adaptability, a level of stability and order through set procedures, as well as ensuring that the adaptive function remains in alignment with the organizational level. Therefore this part of the research focuses on unraveling **the impact of leadership style on fostering self-organization**.

To begin with, the interviews highlighted the importance of leadership that empower teams by fostering their autonomy and trust, creating a purpose and guiding team members is their main priority, rather than micromanaging tasks. This approach involves leaders being integral parts of the team, contributing to problem-solving and making sure the team faces the right direction with their efforts.

Respondent no. 2: *“So if talking about my leadership style, it encourages a lot of autonomy and trust. It encourages a lot of creativity. So I trust people, I trust those who are leading different squads. And then I really encourage challenging, as it does not matter if I say the goal is this, we are feeling safe enough to be challenged, and those who are challenging feel safe enough to challenge, it's really an important place to be.”*

Respondent no. 3: *“So I try to give them as much autonomy as possible and I always try to allow them to make the final decision, although I will strongly give my opinion as well. I know that they're better experts and I do allow them to make decisions and sometimes make mistakes because I think they learn a bit better from their mistakes instead of me saying from my expertise.”*

Such an approach encompasses the traits of adaptive leadership style, where people are given the trust with their decision-making power, where attempts to innovate are welcomed and fostered. However, it does not mean that individuals are just thrown at problems and have to figure out everything by themselves. The leader here plays an even more important role in nurturing that given autonomy with some guidance of where the solution might lie or how the strengths of the team can be grasped.

Respondent no. 1 : *“So usually the problems they come to me with, it's something that they cannot solve themselves or there is something that they need support on, but sometimes there are problems that they can solve themselves. So you just guide them in a way so they could do that themselves.”*

Respondent no. 7: *“So they need to accept that there's the positive. Like the wins and the losses are theirs, but mine as well. So at the end of the day, I'm kind of responsible for everything. But at the same time, I think it's important to have trust in your team that they can own like an individual aspect. They can make better decisions. So from my side, I think a lot more aspect goes into sort of, not necessarily mentorship, but trying to help guide the team based on what their strengths and the sort of areas to improve.”*

Respondent no. 3: *“So I try to give them as much autonomy as possible and I always try to allow them to make the final decision, although I will strongly give my opinion as well. I know that they're better experts and I do allow them to make decisions and sometimes make mistakes because I think they learn a bit better from their mistakes instead of me saying from my expertise, which is not always right, to make decisions and make mistakes.”*

In this adaptive environment a fundamental trust in team members' abilities is shown, granting them autonomy in decision-making and recognizing their direct involvement in project's success. Furthermore, such leaders view mistakes not as failures, but as valuable learning opportunities, fostering an environment for experimentation and innovation, as well as a culture of continuous improvement and personal growth within the self-organizing team.

However, as these teams operate within a certain organization, appropriate level of structure and control still has to be deployed in order not to lose control of the project. This is achieved through soft implications of administrative leadership, as teams are bound not by strict rules or procedures of how things have to be delivered, but rather through the development of goals and targets.

Respondent no. 2: *“On that side, I'm not managing to predict and say that this has to be this way or another way, but sometimes we have to have that, those OKR's, those organizational sort of things. You can choose the path, you go in through the goal, but the goal needs to stay there. And we do constant revisions of you, if we're not achieving something, so why we didn't do, what could have helped.”*

Respondent no. 6: *“Yeah, this is quite easy (balancing between autonomy and guidance). We have our quarterly goals that we define before each quarter. We also have regular checkups*

on how we progress towards these quarterly goals. Everybody seems to be on the same page. Of course there are times when we are behind our plan and it's very common practice, I guess, but usually it's happening not because of our fault.”

By coming up with these well-defined quarterly objectives and goals, leaders provide that needed structured sense of direction and organization, as breaking down long-term goals into manageable segments also helps to maintain focus and clarity. In this way self-organizing teams have a clearer path and boundaries of where and how they are going with their work. In addition, the setting of these goals also provides a platform for innovations to occur, as teams can include their experiments and testing of different ideas into those desired objectives. Finally, an important part falls on concurrent and regular check-ins on the progress through overview meetings, which facilitate the ongoing communication and feedback, allowing the monitoring of progress, addressing occurring challenges and making necessary adjustments to the work.

Having this set up that balances autonomy with necessary guidance and support, as well as ensures the alignment of individual efforts with project goals while empowering team members to take initiative, be creative, and fully own their work processes suggests about the presence of enabling leadership.

Respondent no. 1: *“We have one example that our front end team needs to build a technical solution, new technical infrastructure for the front end application. And our goal is to guide them in a way so they would build a plan, be responsible for that plan, tell us what benefit it will bring, what cons it can bring, or maybe what happens if we don't do it. So they have full autonomy and full responsibility to build this plan so we could align with our product initiatives.”*

Respondent no. 2: *“Well, the goals really help you. So if you have a goal, then you have that relationship, then you explain really clearly what your squad is responsible to do, and then your performance is linked to that goal. When you are kind of clear on that side, then you can go and say, I trust you and can you find the best decision. I'm also constantly double checking if the person is not lost in those goals and all that.”*

Such leadership structure not only fosters a sense of collective responsibility towards achieving goals but also respects and encourages the autonomy of team members. This balanced approach ensures that while individuals have autonomy to organize and deliver their work, the efforts remain aligned and coordinated with the team's overall objectives and efforts. This highly contributes to the creation and fostering of effective self-organization which is vital for the success of IT product development projects.

In conclusion, research on how different forms of leadership facilitate the environment of self-organization within IT product development project teams, established the presence of aspects from all three discussed leadership styles - adaptive, administrative and enabling. The fusion of these leadership elements lead to an establishment of two key themes that are crucial to self-organization in this context:

1. Guided autonomy. The concept is derived from a leadership approach which is centered on empowering team members with the authority to make decisions, while also providing necessary guidance and support. Emphasis here is put on developing a significant level of trust in the expertise and skills of the team members, allowing them the freedom to embrace problems and innovate independently. However, this autonomy does not imply a lack of leadership involvement, but rather revolves around having a layer of support, as leaders are available to offer direction and assistance when challenges exceed the team's or individual capabilities. By fostering an environment where team members can make decisions, learn from their experiences, and even their mistakes, leaders encourage a culture of continuous improvement and autonomy. This environment is crucial for the development of individual problem-solving skills and enhances the team's capacity for effective self-organization.

In practice, such an environment is fostered through time and is definitely not something that works straight away. Firstly, it is important to spend time with people, observing their working preferences and understanding the level of guidance needed. While some people will need an extensive amount of support in understanding of what needs to be done and how, others are able to proceed towards the shared goal without further detailed explanations. Secondly, there is no better way of learning rather than doing it by yourself, therefore, people need to be put into positions where they have to make certain decisions by themselves and be responsible for their line of work. It is crucial for individual members of the team to feel the leaders trust in their expertise to make correct decisions, take risks and learn from their mistakes without harsh

punishments. Finally, an environment for challenging has to be created by the leader, in order to ensure that creativity is fostered and hierarchical structure doesn't prevail. Team members have to be empowered to constructively challenge any decisions that are being made, voice their opinions on the issues and their solutions, as that allows for new perspectives to occur and people to be more involved in the steering of the project itself.

2. Structure through alignment. This aspect of leadership involves establishing clear goals and being engaged in regular reviews to track progress, as well as maintaining the team's focus on the broader project objectives. The setting of well-defined objectives and consistent monitoring of progress ensures that while team members work autonomously, their individual efforts remain aligned with the team's vision and goals. This sort of a structure provides a clear sense of direction and boundaries for the team's work, essential in integrating individual and team efforts towards the organization's objectives. It allows for innovation and experimentation to happen within a set framework that aligns with the project's overall direction, ensuring that self-organization does not lead to divergence from the intended goals. In addition, individual efforts are thus systematically directed towards achieving the project's objectives, balancing the freedom of self-organization with the necessity of maintaining team consistency and alignment within the project's context.

Practical implementation of such setup can be achieved through the already emphasized adoption of OKR or KPI frameworks. These approaches to goal setting highly involve team members into developing their objectives, roadmaps and detailing their deliverables for the limited time period (f.e. quarter). In doing so, teams are bound by their own agreed structure and plan of what needs to be done in that short period of time, however, they do not only have their say in the formation of those goals, but also the autonomy is fostered to choose the methods of how that work will be carried out and what further decisions will be made. To support this structure, another practice of tracking the progress has to be implemented, as it is not viable to set and forget the objectives. Regular catch-up's and progress review meetings have to be put in place, so that the alignment is reassured or corrected, and any blockers can be moved out of the way for the team to efficiently self-organize.

In conclusion, the following themes and sub themes, consisting concrete practices were derived in this research area:

1. Guided autonomy
 - a. Understanding the level of guidance needed through observation and familiarizing with team members
 - b. Responsibility for decisions in practice
 - c. Safe environment to challenge decisions
2. Structure through alignment
 - a. Involvement in planning through OKR/KPI frameworks
 - b. Autonomy in method to deliver tasks
 - c. Constant meetings and catch-up's for progress review

12.5 Influence of Agile methodologies

This final part of the results is focused on how IT product development project teams use Agile methodologies, which were established to be a major adoption of different self-organizing practices. Therefore, **the relationship between Agile methodologies and self-organizing practices** is analyzed. This part of research will determine how the use of various Agile methodologies can lay a solid background and contribute to further establishment of self-organizing practices within the teams.

One of the key advantages of using Agile methodologies was established to be the flexibility and autonomy that this approach brings. Different practices of Agile are very customizable and can be adopted to fit the particular setups and needs of different teams.

Respondent no. 1: *“So for each team you just find a way what it's best suited for. Our kanban currently with, for example, one of the teams is not even a Kanban itself because it doesn't even have specific, let's call it restrictions of the board. Because in Kanban, you need to assign how many tasks per column should be in that column. And we don't have that. So we adjust our way of working according how the team wants to work and how it's the most efficient for them”.*

Respondent no.5: *“We have, as I said, some sprints, different teams in our company work in different sprints duration. One team is working two week sprints, other teams work in*

three week sprints, because it's more comfortable for them. We also have retrospectives, retrospectives for different stuff as well. We have weekly retrospectives, for example, for app development. We have project retrospectives when we finish some project.”

As each team usually has its unique composition, strengths, weaknesses and ways of working in general, standard practices can be tailored to meet the dynamics of individual teams, as well as their goals. Teams might choose to modify the frequency, duration, or format of such ceremonies as Sprints to better align with their work rhythms, or artifacts like user stories, backlogs, and task boards can also be modified to suit the team's way of tracking and managing work. Having this ability already creates a solid background for teams to self-organize.

Furthermore, Agile practices themselves foster autonomy, as such tools like Kanban boards allow for the tasks to be visible for anyone and leave the opportunity for team members to pick them individually.

Respondent no.6 : *“And then we saw that there is a lot of bureaucracy in it and now we are just working in a kanban way and we like it. So everybody seems to be happy. Everybody knows what needs to be done, what are the highest priorities. And yeah, we are managing this roadmap in different ways.”*

Respondent no.2: *“Although we are working in the same model, but we're not dragging all of the developers from different platforms to talk about. Maybe they have a new feature, introduction, they do it all together. But then you go in a specific platform to adjust one or another thing. They don't need to be in the same room, so we allow them to create their own rhythm around that.”*

This setup offers a platform where every member can be aware of the project's status and can autonomously choose tasks based on their expertise and current workload, which ensures a more balanced and efficient work pace. It not only prevents tasks from piling up or burnouts, but also promotes a sense of ownership and responsibility among the team members as everyone is able to see their contributions to the entire objective. Additionally, this flexibility allows teams to

adapt quickly to changing priorities, maintaining agility and responsiveness, as team members are encouraged to take initiative and engage in problem-solving, fostering a collaborative yet independent work culture.

It must be elaborated, that in the face of this freedom, agile methodologies still provide a necessary level of alignment, as teams have to be engaged in planning their line of work, whether it's updating their Kanban boards, planning and presenting their Sprint goals or participating in mandatory ceremonies like daily catch-up's or retrospectives.

Respondent no. 4: *“Basically I would say following the agile practices, scrum practices, having those really alignments between the team, following the sprint goals. Are we actually focusing our delivery into the sprint goal, right? Are we adding certain tickets if it is needed to the sprint board to actually achieve the goal if something unknown happened? Are we actually communicating the blockers and removing them?”*

Respondent no. 2: *“As an example, like a legal don't need to go to each of the squads and see what new features they've released. But they go into that one pulse meeting, they see what people are working on or planning to work on. And they say, oh, this might have legal implications that we need to take care of. And then after that meeting, they go to product manager and say, okay, we need to talk and find out if my consensus is there or not or what we need to prepare for.”*

Respondent no. 5: *“And some of those totally make sense, for example, to have daily stand-ups where the teams could gather and discuss what we've done yesterday, what we'll do today, or tomorrow, or etc. So these are a must.”*

This approach not only maintains the necessary structure but at the same time allows for individual autonomy within the agreed framework, ensuring that team efforts are cohesive and effectively aligned with the project's goals. In addition, it keeps everyone on the same page and provides an opportunity to quickly address any issues that could slow down progress or review the completed work and make necessary adjustments to the workflows. Finally, while individual team members have the autonomy in determining how they execute tasks, major decisions regarding the project's strategy and priorities are often made collectively, which ensures that all

team members with different expertise have a say in this direction, fostering a sense of shared responsibility and alignment.

Another crucial impact of adopting Agile ideas towards the enhancement of self-organizing practices derives from its focus on iterative work delivery and constant feedback loop.

Respondent no.7: *“I think just in general, rapid planning and short-term planning, that makes sense for us. Limiting scope in development, so building features, you want to start with smaller scale testable solutions versus just building the end result up front. Yeah, I think that for me, those are the most important elements.”*

Respondent no.5: *“So it's hard to say, because if you're meaning by agile that we can quickly adapt, yes, we have teams that can work on separate things on different set of things at the same moment. We can also quickly adapt to some changes. We can deliver quickly if we need something.”*

Respondent no.6: *“So because we are working in iterations, basically this is what enables these changes to be made before actually delivering the feature. If we wouldn't have this way, probably then it would cost more money, more time and less value for the business and users. So yeah, I think that it enables quite a lot.”*

By employing this philosophy of breaking down projects into smaller, manageable units of work, and delivering them step by step, teams are more capable of changing things quickly, thus enhancing flexibility towards changing goals or requirements. In addition, this approach allows teams to plan in shorter capacity, evaluate outcomes, and adjust their strategies based on immediate feedback.

Respondent no.1: *“You work on an iteration, get feedback, work on another iteration, get feedback. And that's basically where the autonomy of the team happens because they get instant feedback of their delivery.”*

The iterative nature of Agile practices, coupled with regular feedback loops, enables teams to be more responsive and fosters autonomous decision-making, as people are in position to constantly react and shift priorities or adapt to new challenges, re-prioritize their tasks and ensure that the

team remains agile and can effectively adapt. This methodology not only reduces the time to delivery but also boosts overall efficiency, as teams can continuously refine their processes and approaches in response to real-time project dynamics.

On the other hand, the integration of Agile methodologies into teams can as well bring major challenges to the team and not foster self-organization. While the customization and adaptability of the discussed methodologies brings of a variety of mentioned benefits, it also has its drawback in the actual effort needed to find the one that fits.

Respondent no.2: *“Yeah, it's a constant looking for the most effective way, what would be faster, and there's different developers, different leaders who use it to work in different ways. And each episode, maybe this week we should work on this or that, so there's a constant battle between arguments about which way of working is better. One kind of improvement should make everyone work better, but that improvement may be good for this team, but not good for that team. So this is a constant challenge, what processes and what practices to make global, you know, mandatory for everyone, and which of those should be kept in the team level, because you want to give that autonomy. But you also have to have those processes where, you know, you manage to release something that compiles from different squads work into one thing, you know. So it's a constant challenge to format. It took years for us to shape something that works, and then we constantly are looking out to improve it, with some automation or something, it's really a huge job.”*

The major challenge lies in the fact that the adoption and customization of different practices is not a one-off task but rather a constantly ongoing process which involves understanding the unique characteristics, strengths, weaknesses, as well as the specific working styles of each member within the team. It requires constant monitoring, feedback gathering and making adjustments, consuming considerable time, effort and patience from both team leaders and individuals. On top of that, with the evolving scale of the projects and teams, a further question of standardization versus adaptation tends to appear, where leaders must navigate and find the balance between standardizing certain practices across the organization and leaving the freedom to adapt others to meet the needs of specific teams. This issue becomes more complex when more people or even multiple teams work on interconnected aspects of the project, as it requires

more in depth planning, constant oversight and a higher level of communication between everyone involved. All of these aspects become essential to ensure smooth integration of workflows and maintaining overall project alignment as well as the support of self-organization.

Moving forward, it was established that further issues with introducing certain Agile methodologies which foster self-organization can arise from the mindset and different experiences of people within the teams.

Respondent no. 4: *“I would say that the main blockers might be the habit of how people were previously working. If they were working on one of them, they don't really understand how the agile will work. You know, I have onboarded quite a lot of developers on what we are trying to achieve here. And it was like, I was really seeing some mind blowing reactions, questioning does it work, how come, how we will actually achieve that? And when they really see the benefit, it's really nice. So yeah, working on waterfall previously, it's really hard to switch the mind in the agile way. And I would say that is the main. If you do not have that previous experience, then it's rather waterfall or agile to adopt, it is still something that you will be learning.”*

Respondent no. 7: *“I think the main challenge is that different people have worked in different organizations, they have different experiences, different biases. So someone that had good experience with a agile, they might think that that's how we want to work and so on. Those that had bad experience with that, will say that this is the worst thing ever, let's not overthink, why do we need so many meetings and blah blah blah.”*

As Agile methodologies employ a variety of different and unique practices and methods of organizing the workflow of the teams, it is not easy for some people to onboard these ideas and get used to the particular workflows. This challenge is especially present in environments where people have to make transitions from more traditional project management like Waterfall, as a significant shift from a fixed and structured approach to a more dynamic and fluid one is needed. It involves not just teaching them the mechanics of Agile practices, but also helping understand and embrace the underlying principles and values behind them, which requires patience and tailored individual approaches, as each person's experience and level of adaptability often varies. Speaking of experiences, it's also something that could either be a positive factor behind people's motivation and enthusiasm to get up to speed faster, or it can be another major blocker in

people's mindsets towards these practices in case of negative past experiences or lack of familiarity with the principles behind them. This occurring diversity can lead to lack of commitment or even conflicting views on how projects should be managed and delivered, posing a challenge in creating a working self-organizing environment.

Finally, a significant challenge can arise even after adopting and engaging in certain Agile practices or its workflows, as keeping up with the procedures and ceremonies of those also requires a high level of commitment and engagement.

Respondent no.3: *“Requirements change and a lot of people don't look at the updates anymore within stories or Jira. But if you constantly update those stories and people get notifications, then they stop reading as well. So that's too much communication which is informal, it leads for them not to read. So you create sometimes instead of ten updates, you create a meeting, which again is a bit more time consuming than reading a line. But if developers don't read it, then you have to physically create a meeting and go and talk it through, which usually takes 15 or 20 minutes because there's more discussion.”*

Respondent no.6: *“You need to close the sprint, review what was delivered and what we saw that we usually don't deliver half of what we put into the sprint. Then we tried to put less into the sprint, but again we realized that then at the second half of the sprint a lot of people are just sitting without work or taking the tasks from their future sprints. So what is the value of that sprint?”*

Teams that work with Agile practices are not just about being flexible in planning and iterative development, they also have to be involved in various procedures and ceremonies like daily stand-ups, sprint planning, and retrospectives. Each of these activities requires active participation and a significant amount of engagement from team members, which can definitely be very demanding, especially in fast-paced IT product development project environments. Furthermore, Agile environments, especially those that involve utilization of frameworks like Scrum and Kanban, pose a significant reliance on tools such as Jira for updating user stories, tasks, and tracking progress of the project deliverables. While these commitments are essential for maintaining transparency and coordination within groups, they can also lead to an overload of information and potentially overwhelming team members with continuous notifications and

status updates. Further effects on team performance and effectiveness can occur as people might become disengaged and lose focus on important aspects of team objectives. Therefore, it is crucial to strike the right balance in communication practices, in order to keep everyone aligned without forcing the excess of information and remain everyone productive in the environment of a self-organizing team.

To conclude, investigation of different Agile usage experiences and its contributions towards the fostering of self-organizing environments established a few key themes that elaborate on the positive and negative aspects of this relationship:

1. Customization and flexibility. Agile approach brings a lot of flexibility to the workflows of the teams through the high level of customization that different methodologies such as Kanban or Scrum embody. This customization allows each team to find the most efficient way of working, as various aspects can be tailored to meet the dynamics and goals of particular teams. Through this adaptability of the methods teams can work in a way that maximizes their efficiency and forms a solid background for teams to foster their self-organization.

To be more in detail, several practices were established that embody this adaptiveness of Agile methods. Firstly, to accommodate the specific pace and workflow of certain teams, different durations and frequencies of Sprints can be used or modified. This allows teams to leverage between a fast-paced environment where tasks are delivered in a smaller and faster capacity, allowing to gather instant feedback, and a slower deployment of bigger tasks, allowing for more complex work and slower changes. Beyond the adjustment of length, teams can also adjust the frequency of their Sprints to have more alignment with their project timelines and delivery expectations. Some teams might find that a continuous, back-to-back Sprint cycle is most effective, while others might take advantage of breaks between Sprints for other activities like planning or research. Another occurring practice that shows the adaptability of these frameworks includes the adjustment of Kanban board boundaries. As these boards are a visual tool used to track work being done at various stages of a process, by adjusting the boundaries or columns of these boards teams can tailor them to reflect their specific workflows. This can involve creating additional stages specific to the team's process, such as separate columns for design, development, testing and deployment or even merging stages for smaller and simpler

tasks. Such customization allows teams to prevent or deal with bottlenecks and ensure a steady flow of work, thus contributing to the creation of a more effective self-organizing environment.

2. Structured autonomy and agility. Agile methodologies enable a level of structured autonomy and agility to prevail, by maintaining a balance between allowing teams to deliver their work individually and maintaining them between the set line of time, scope and priority boundaries. By employment of Kanban boards, delivering tasks in Sprints and breaking down deliverables into smaller batches, teams can facilitate their agility to address any changes or needs.

Practical implementations of this firstly comes through the fostering of individual autonomy emphasized in Kanban boards. As these boards provide a visual representation of all the tasks within a project, it allows every team member to see the status of its various stages, thus promoting transparency and alignment of the project's current state. Even more importantly, it empowers individuals to manage their work autonomously, as team members can choose tasks based on their current workload, expertise and project priorities. By doing so not only is self-organization encouraged, but also a continuous flow of work and progress is maintained in alignment with the direction of the project. Moving forward, employing Sprint methodology allows self-organization to happen from a different point of view. Such ceremonies as Sprint planning always involves all the team members that will be working on the delivery of the tasks, thus everyone has their say and in deciding what and when will be done. This autonomy in task selection enables individuals to choose tasks that match their skills, interests, and goals, yet allowing for a more balanced and aligned distribution of work. Through such involvement, a higher level of ownership and commitment is fostered, as people are more invested in the outcomes because they played a part in defining what those outcomes at the end should be. Finally, an underlying idea of breaking down bigger deliverables into smaller tasks and working on them in separate iterations forms an environment where individuals can be more autonomous and less reliant on others. Individuals can thus focus on their assigned tasks without being overly dependent on others for progress, as well as take ownership of their part, contributing to the common goal in a more controlled and efficient way, delivering progress in parallel rather than sequentially. Moreover, having control of smaller tasks allows team members to make decisions at the individual level, instead of constantly having to find consensus or approval from higher-ups or other peers. This empowerment is crucial in Agile environments, where team

members are expected to be proactive and take responsibility for their work, encouraging a culture of ownership and accountability, which are key drivers of effective self-organization.

3. Challenges in adoption and commitment. Despite the mentioned benefits, the integration of Agile methodologies into teams also can bring significant challenges and thus make self-organization burdensome. These include the effort, time and resources needed to find and customize the right Agile approaches to fit the particular needs of specific teams, as well as actually making them work in the contexts of different team dynamics. With this usually requiring a significant shift in people's mindsets and understanding of new ideas about the workflows, other blockers can arise due to major differences of methodologies like Waterfall and Agile, as well as the diversity in team members' experiences or even biases towards employing these methodologies. Furthermore, certain Agile environments rely on active involvement and commitment to various procedures and ceremonies, which ensures that all team members are aligned, informed, and involved in the decision-making process. However, the frequency and capacity of those can be very demanding, requiring significant time and attention to detail from team members, which can easily lead to a communication overload, loss of focus on key information or even productivity.

To tackle and manage these possible issues a few key practices have to be put in place. Most importantly, constant feedback loops, retrospective sessions and trainings are crucial in this process, as they help in identifying the most effective practices, adjusting and learning how to make the most out of them. Through regular meetings with team members, opportunities for individuals to provide feedback or suggest particular changes can be created, allowing for continuous improvement and more tailored communication flow within the teams to happen. Secondly, as the required engagement levels are pretty high within these methodologies, it becomes more likely for team members to get overwhelmed by the amount of different information, status updates and notifications in different tools or platforms they have to keep up with. Therefore, it is crucial to find a balance that optimizes the efficiency and effectiveness of communication within the teams, which can involve consolidating information to focus on key updates, setting up catch-up meetings instead of providing multiple written updates, streamlining meetings by setting clear agendas for them, making most of asynchronous communication methods for non-urgent updates or even removing certain ceremonies if they are not adding value. Finally, adopting and embracing iterative development workflow and delivering tasks in

smaller and faster capacities can contribute to smoother mindset shift and involvement of individuals into team workflows. Delivering work in smaller batches allows people to acclimate to new or particular practices in stages, with each iteration offering learning opportunities and enabling team members to build their understanding and skills step by step. In addition, team members can also see the results of their work more quickly and recognize the impact of their contributions to the overall objectives of the project, thus helping individuals to embrace the benefits and make the most of Agile practices.

In conclusion, the following themes and sub themes, consisting concrete practices were derived in this research area:

1. Customization and flexibility
 - a. Adjustment of duration and frequency of Sprints
 - b. Customization of Kanban boards
2. Structured autonomy and agility
 - a. Autonomy through Kanban boards
 - b. Autonomy through involvement in Sprint planning
 - c. Autonomy through breakdown of tasks
 - d. Empowerment through control of tasks
3. Challenges in adoption and commitment
 - a. Tailoring through feedback sessions and retrospectives
 - b. Balance of communication flow through consolidation of information, effective meetings and asynchronous channels.
 - c. Smooth integration and recognition of impact through iterative development

13. Conclusions and recommendations

This final section of the thesis aims to summarize the key findings and insights gained from the study on the application of self-organization principles in IT product development projects within the chosen organization. This investigation focused around exploring the theoretical foundations of self-organization and different dynamics of such an environment, as well as examining their practical implementation and effectiveness in actual IT project teams. Through a comprehensive case study analysis this thesis aimed to bridge the gap between theoretical ideas and practical application of self-organizing practices. The following conclusions and recommendations are grounded in the research objectives set out in the introduction and reflect a thorough examination of the dynamics of self-organization within the context of IT product development project teams.

1. The comprehensive review of literature and theoretical foundations of self-organization has elaborated the evolution of self-organization concepts from their inception in natural sciences to their current applicability in modern organizational structures, especially within Agile frameworks. This theoretical exploration provided a solid foundation for understanding the principles and history of self-organization, highlighting its benefits in fostering such aspects as adaptability, innovation, and responsiveness of the teams in the face of rapidly changing technological and market conditions especially present in the IT industry. On the other hand, the concluded research also highlighted the significant lack of alignment of the theoretical concepts with actual insights into the adaptation, usage and challenges of self-organizing practices from the real-world environments. For this purpose, five key research areas relating to the dynamics of self-organizing environments were derived:

1. The impact of self-organizing practices on adaptability and responsiveness of IT product development project teams.

2. The role of communication, information flow, and relationships in the success of self-organizing teams.

3. The importance of a shared vision and aligned goals for the effectiveness of self-organizing practices.

4. The impact of leadership style on fostering self-organization.

5. The relationship between Agile methodologies and self-organizing practices

2. To successfully conclude the research into the derived areas about different dynamics of self-organizing practices and their implications within specific IT product development projects, a qualitative research method was developed. It included the adoption of a case study method, which fulfilled the need of this research to gain actual insights into the phenomena of self-organization within the particular area - the specific teams of different products and companies within the concrete organization, focused on the delivery of high value IT products in the cybersecurity and privacy industry. The purposive sampling method was then applied to select the participants, that poses relevant knowledge and experience of IT project management and self-organizing environments for the participation in the semi-structured interviews. A further qualitative research method of thematic analysis was employed to analyze the data collected from directors/leads of IT product development projects within the chosen organization. The data grouping and analysis was concluded in accordance with its relevance to the research areas raised for this thesis, as well as the development of themes concerning practical application of self-organizing practices within IT product development projects.

3. The practical analysis of data concerning the dynamics and applications of self-organizing practices in IT product development projects provided a deeper understanding of investigated research areas, as well as established concrete practices through the implementation of which, the self-organizing environments could be fostered to fit the unique demands of different IT project environments:

1. Impact of self-organizing practices on adaptability and responsiveness - it was established that self-organizing IT project teams display a significant amount of adaptability and responsiveness towards the occurring internal or external challenges. These abilities were explored through the development of two themes here - flexibility in methodologies and empowerment through autonomy, which allows teams to tailor their approach to specific project needs that also contributes to fostering the culture of ownership and proactive decision-making. Furthermore, the following possible practical implementations for these themes were derived:

- Freedom in experimentation and adoption of methodologies
- Clear definition of roles and responsibilities of team members
- Direct involvement in decision making, research and other processes

- Support of autonomy through necessary availability of information, tools and expertise

2. Role of communication, information flow, and relationships - all these elements were emphasized to be fundamental for the success of self-organizing teams, as they enhance team efficiency, foster quick decision-making on an individual level and support a transparent yet collaborative work environment. The following possible practices were established for implementation that lay a solid foundation for efficient communication and flow of information, as well as the enhancement of relationships within project teams:

- Development of a three layer system to ensure the information flow through instant messaging, progress tracking and accessible knowledge base
- Use of public channels instead of practicing private messages
- Use of different types of meetings for face-to-face synchronization
- Physical placement of team members next to each other
- Involvement in team-building activities and shared hobbies

3. Importance of a shared vision and aligned goals - the establishment of a shared vision and alignment of goals stood out to be critical for the effectiveness of self-organizing practices within the IT product development project teams. Structured development of these goals and continuous communication about them provide a platform to maintain team focus and facilitate efficient decision-making on an individual level. A few practices have been derived here to

- Employment of OKR or KPI frameworks
- Regular weekly or monthly meetings to track progress within the OKR/KPI frameworks
- Allowing decision-making to happen on an individual level

4. Impact of leadership style on fostering self-organization - the research came to a conclusion that different aspects of all three expected styles of leadership - adaptive, administrative and enabling - were noted to be very important to the proper functioning of the self-organizing environment. Such a complex form of leadership effectively balances guidance with autonomy as well as maintains the necessary level of structure and alignment of individual and team efforts through setting clear trajectory and boundaries of the project delivery. The

establishment of such an autonomous, yet organized and consistent environment can be fostered by the implementation of such practices:

- Understanding the level of guidance needed through observation and familiarizing with team members
- Responsibility for decisions in practice
- Safe platform to challenge decisions
- Involvement in planning and decision-making through OKR/KPI frameworks
- Autonomy in method to deliver tasks
- Constant meetings and catch-up's for progress review

5. Relationship between Agile methodologies and self-organizing practices - it was established that the different implementations of Agile methodologies have a dual influence on the fostering of self-organizing environments. On one hand, multiple aspects of Agile approaches such as their flexibility and customization, as well as the framework to balance a level of structure with agility and autonomy, can highly contribute to further establishment of self-organization. On the other hand, the employment of Agile also can bring significant burdens here, as the successful implementation of it requires a significant amount of effort, time and resources from both teams and individuals. To make the most of the benefits that various Agile methodologies bring and deal with the challenges that they pose towards self-organization, the following practices were derived from the research on this area:

- Adjustment of durations and frequencies of Sprints
- Customization of Kanban boards
- Autonomy through Kanban boards
- Autonomy through involvement in Sprint planning
- Autonomy through breakdown of tasks
- Empowerment through control of tasks
- Tailoring through feedback sessions and retrospectives
- Balance of communication flow through consolidation of information, effective meetings and asynchronous channels.
- Smooth integration and recognition of impact through iterative development

In conclusion, this thesis contributes to the academic and practical understanding of self-organization in IT project management, as it bridges the gap between theoretical knowledge and practical application, offering concrete practices and frameworks that can be instrumental for practitioners and scholars of this industry. The research highlights the potential of using self-organization in enhancing the adaptability and resilience in IT product development projects, while also embracing the challenges in its practical implementation. The findings of this study also suggest that the dynamics and practices of self-organizing teams highly vary between different setups and environments of teams, therefore further empirical research should be directed at investigating different contexts, organizations and industries to further bridge the gap between theoretical and practical implications of self-organizing practices in project management.

Bibliography and the list of reference

1. Agbejule, A., & Lehtineva, L. (2022). The relationship between traditional project management, agile project management, and teamwork quality on project success. *International Journal of Organizational Analysis*, 30(7), 124-136.
2. Albert, R., & Barabási, A. L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1), 47.
3. Arena, M. J., & Uhl-Bien, M. (2016). Complexity leadership theory: Shifting from human capital to social capital. *People and Strategy*, 39(2), 22.
4. Arena, M., Cross, R., Sims, J., & Uhl-Bien, M. (2017). How to catalyze innovation in your organization. *MIT Sloan Management Review*.
5. Asawo, S. P., & Ogbonda, E. H. (2022). Triple-Loop Organizational Learning and Workers' Innovative Behaviour: A Response Mechanism for Pandemic Induced Work Disruptions. *Nigerian Academy of Management Journal*, 17(2), 1-12.
6. Ashby, W. R. (2017). Principles of the self-organizing system. In *Systems research for behavioral science* (pp. 108-118). Routledge.
7. Baltacı, A., & Balcı, A. (2017). Complexity leadership: A theoretical perspective. *International Journal of Educational Leadership and Management*, 30-58.
8. Berényi, L., & Soltész, L. (2022). Evaluation of Product Development Success: A Student Perspective. *Administrative Sciences*, 12(2), 49.
9. Chauhan, K., Crewe, E., & Mowles, C. (Eds.). (2022). *Complexity and leadership*. Taylor & Francis.
10. Chen, J., Reilly, R. R., & Lynn, G. S. (2005). The impacts of speed-to-market on new product success: the moderating effects of uncertainty. *IEEE Transactions on Engineering Management*, 52(2), 199-212.
11. Cohen, L., Manion, L., & Morrison, K. (2007). *Research Methods in Education* (6 ed.). London Taylor & Fancis Group
12. Cooke, J. L. (2012). *Everything you want to know about Agile: How to get Agile results in a less-than-Agile organization*. IT Governance Ltd.
13. Cotrel, B. (2022, July 16). *Self-Organizing Team: How does it work?*. Neatro. Retrieved from <https://www.neatro.io/self-organizing-team/>

14. Dörnyei, Z. (2007). *Research Methods in Applied Linguistics*. Oxford: Oxford University Press.
15. Gardner, N. (2022). Digital Transformation and Organizational Learning: Situated Perspectives on Becoming Digital in Architectural Design Practice. *Frontiers in Built Environment*, 8, 905455.
16. Geerlof, J., & van Beckhoven, A. (2016). Complexity, self-organization, and leadership: Enlivened experiences from The Netherlands. *Integral Leadership Review*.
17. Georges, L., Romme, A., & Van Witteloostuijn, A. (1999). Circular organizing and triple loop learning. *Journal of Organizational Change Management*, 12(5), 439-454.
18. Gidado, K. (1993). Numerical index of complexity in building construction to its effect on production time. (Doctoral dissertation, University of Brighton).
19. Haken, H. (2006). *Information and self-organization: A macroscopic approach to complex systems*. Springer Science & Business Media.
20. Heylighen, F. (2001). The science of self-organization and adaptivity. *The Encyclopedia of Life Support Systems*, 5(3), 253-280.
21. Heylighen, F. (2008). Complexity and self-organization. In LAPLANTE, Phillip (Ed.), *Encyclopedia of Information Systems and Technology*, 1, 250-259.
22. Hoda, R. (2011). *Self-organizing agile teams: A grounded theory*.
23. Holland, J. H. (2006). Studying complex adaptive systems. *Journal of Systems Science and Complexity*, 19, 1-8.
24. Ismael, J. T. (2011). Self-organization and self-governance. *Philosophy of the Social Sciences*, 41(3), 327-351.
25. Iyer, Y. (2021, June 25). *Overcoming the Top Challenges of IT Project Management*. Wrike. Retrieved from <https://www.wrike.com/blog/top-challenges-it-project-management/>
26. Joseph, N., & Marnewick, C. (2022). Investigating the impact of information systems project complexity on project success dimensions. *The Journal of Modern Project Management*, 10(2), 186-205.
27. Kahneman, D., & Tversky, A. (1972). Subjective probability: A judgment of representativeness. *Cognitive psychology*, 3(3), 430-454.
28. Larsen, D. (2004). *Team agility: Exploring self-organizing software development teams*.

Industrial Logic and The Agile Times Newsletter.

29. Lichtenstein, B. B., & Plowman, D. A. (2009). The leadership of emergence: A complex systems leadership theory of emergence at successive organizational levels.
30. Lichtenstein, B. B., Uhl-Bien, M., Marion, R., Seers, A., Orton, J. D., & Schreiber, C. (2006). Complexity leadership theory: An interactive perspective on leading in complex adaptive systems.
31. MacCormack, A., & Verganti, R. (2003). Managing the sources of uncertainty: Matching process and context in software development. *Journal of Product Innovation Management*, 20(3), 217-232.
32. Marion, R., & Uhl-Bien, M. (2020). Complexity leadership in bureaucratic forms of organizing: A meso model. *The Leadership Quarterly*, 20(4), 631-650.
33. Mateo, J. R. S. C., Formoso, J. Á. F., & Iglesias, G. (2021). Complexity and Project Management: Challenges, Opportunities, and Future Research.
34. Matook, S., Soltani, S., & Maruping, L. M. (2016, December). Self-Organization in Agile ISD Teams and the Influence on Exploration and Exploitation. In ICIS.
35. Melles, G. (2005). Beyond the Romantic Impulse for Authentic Data to Coconstruction of Meaning in Interview-Based Educational Research. *Qualitative Research Journal*, 5(2), 21-30. doi:<https://search.informit.org/doi/10.3316/aeipt.160700>
36. Merriam, S. B., & Tisdell, E. J. (2016). *Qualitative research: A Guide to Design and Implementation* (4 ed.). San Francisco: John Wiley & Sons, Inc.
37. Merry, U., & Kassavin, N. (1995). *Coping with uncertainty: Insights from the new sciences of chaos, self-organization, and complexity*. Praeger Publishers/Greenwood Publishing Group.
38. Moe, N. B., Dingsøy, T., & Dybå, T. (2008, March). Understanding self-organizing teams in agile software development. In 19th Australian Conference on Software Engineering (ASWEC 2008) (pp. 76-85). IEEE.
39. Morcov, S., Pintelon, L., & Kusters, R. J. (2020). IT project complexity management based on sources and effects: Positive, appropriate, and negative. *Proceedings Of The Romanian Academy Series A-Mathematics Physics Technical Sciences Information Science*, 21(4), 329-336.
40. Morris, P. W., & Morris, P. W. (1994). *The management of projects*. London, UK: T.

Telford.

41. Muhammad, U., Nazir, T., Muhammad, N., Maqsoom, A., Nawab, S., Fatima, S. T., ... & Butt, F. S. (2021). Impact of agile management on project performance: Evidence from IT sector of Pakistan. *Plos One*, 16(4), e0249311.
42. Plowman, D. A., Solansky, S., Beck, T. E., Baker, L., Kulkarni, M., & Travis, D. V. (2007). The role of leadership in emergent self-organization. *The Leadership Quarterly*, 18(4), 341-356.
43. Prigogine, I., & Stengers, I. (1984). *Order out of chaos: Man's new dialogue with nature*.
44. Princes, E., & Said, A. (2022). The impacts of project complexity, trust in leader, performance readiness, and situational leadership on financial sustainability. *International Journal of Managing Projects in Business*, 15(4), 619-644.
45. Richardson, K. A., Cilliers, P., & Lissack, M. (2000). Complexity science: A 'grey' science for the 'stuff in between'. In *Proceedings of the First International Conference on Systems Thinking in Management* (pp. 532-537).
46. Rideout, B. W. (2023). *Complexity Leadership Theory: The Influence of Enabling Leadership and Experiential Learning on Emergent Outcomes and Organizational Knowhow*.
47. Rigby, D. K., Sutherland, J., & Hirotaka, T. (2016). *Embracing Agile: How to master the process that's transforming management*. Harvard Business Review.
48. Rosenhead, J., Franco, L. A., Grint, K., & Friedland, B. (2019). Complexity theory and leadership practice: A review, a critique, and some recommendations. *The Leadership Quarterly*, 30(5), 101304.
49. San Cristóbal, J. R. (2017). Complexity in project management. *Procedia Computer Science*, 121, 762-766.
50. San Cristóbal, J. R., Carral, L., Diaz, E., Fraguera, J. A., & Iglesias, G. (2018). Complexity and project management: A general overview. *Complexity*, 2018.
51. San Cristóbal, J. R., Diaz, E., Carral, L., Fraguera, J. A., & Iglesias, G. (2019). Complexity and project management: Challenges, opportunities, and future research. *Complexity*, 2019.
52. Stacey, R. (2001). *Complex processes of responding in organizations: Learning and knowledge creation*.

53. Stacey, R. D. (1996). *Complexity and creativity in organizations*. Berrett-Koehler Publishers.
54. Tata, J., & Prasad, S. (2004). Team self-management, organizational structure, and judgments of team effectiveness. *Journal of Managerial Issues*, 16(2), 248-265.
55. Turner, J. R., & Cochrane, R. A. (1993). Goals-and-methods matrix: coping with projects with ill-defined goals and/or methods of achieving them. *International Journal of Project Management*, 11(2), 93-102.
56. Uhl-Bien, M., & Marion, R. (2009). Complexity leadership in bureaucratic forms of organizing: A meso model. *The Leadership Quarterly*, 20(4), 631-650.
57. Uhl-Bien, M., & Marion, R. (2011). Complexity leadership theory. *The SAGE Handbook of Leadership*, 468-482.
58. Von Foerster, H., & von Foerster, H. (2003). On self-organizing systems and their environments. In *Understanding understanding: Essays on cybernetics and cognition* (pp. 1-19).
59. de Waard, E. J., Rietjens, S., Romme, A. G. L., & van Fenema, P. C. (2021). Learning in complex public systems: The case of MINUSMA's intelligence organization. *Public Management Review*, 1-20.
60. Waldrop, M. M. (1993). *Complexity: The emerging science at the edge of order and chaos*. Simon and Schuster.
61. Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684), 440-442.
62. Wheatley, M. J., & Kellner-Rogers, M. (1996). Self-organization: The irresistible future of organizing. *Strategy & Leadership*, 24(4), 18-24.
63. Williams, T. (2002). *Modelling Complex Projects*. John Wiley & Sons Ltd.
64. Wood, H. L., & Gidado, K. (2008, September). An overview of complexity theory and its application to the construction industry. In 24th Annual ARCOM Conference. Cardiff, UK: Association of Researchers in Construction Management (pp. 677-686).
65. Wozniak, T. M. (1993). Significance vs. capability: "Fit for use" project controls. *AACE International Transactions*, A-2.
66. Yin, R. K. (1994). *Case study research: Design and methods*.
67. Zhang, X., & Kwan, H. K. (2019). Team behavioral integration links team

interdependence with team performance: An empirical investigation in R&D teams.
Frontiers of Business Research in China, 13, 1-19.

Annexes:

No 1. Questionnaire used for the semi-structured interviews with directors/leads of IT product development projects:

Questions:

Background:

1. Can you shortly describe your role as an IT product development project lead?
2. Have you experienced a transition from traditional project management to agile/self-organizing practices? If so, can you describe that transition?

Research area number 1 (The impact of self-organizing practices on adaptability and responsiveness of IT product development project teams):

1. Can you describe how your team adapts to changes in project requirements or market conditions?
2. What role does self-organizing practices play in enabling your team's adaptability and responsiveness?
3. Could you specify in what ways have these self-organizing practices improved the team's responsiveness to emerging challenges?

Research area number 2 (The role of communication, information flow, and relationships in the success of self-organizing teams):

1. How are communication and information flow managed within your self-organizing team?
2. Can you share instances where effective communication or the flow/availability of information directly impacted the team's success?
3. In what ways do you think relationship-building within the team members contributes to the team's efficiency?
4. Can you describe a situation where effective communication, information flow, or strong relationships were critical to the project's success?

Research area number 3 (The importance of a shared vision and aligned goals for the effectiveness of self-organizing practices) :

1. How does your team develop and maintain a shared vision and aligned goals?
2. What role does a shared vision play in enabling your team to self-organize effectively?
3. Can you share an example of a time when a shared vision or aligned goals helped your team to overcome a challenge or achieve a significant milestone?

Research area number 4 (The impact of leadership style on fostering self-organization):

1. How would you describe the leadership style in your team, especially in terms of facilitating autonomy and decentralized decision-making?
2. How do you balance providing guidance and support with giving your team the autonomy to make decisions? (If not covered in the answer to the previous question)
3. What role does leadership play in fostering and maintaining a self-organizing environment?

Research area number 5 (The relationship between Agile methodologies and self-organizing practices):

1. Have/How Agile methodologies contributed to the development and implementation of self-organizing practices in your team?
2. Are there specific Agile practices that you find particularly effective in supporting self-organization?
3. What challenges have you faced in integrating self-organizing practices with Agile methodologies?

Closing:

1. What are some of the challenges you've faced while working with self-organizing teams?
2. Can you share an example of a significant success that you attribute to self-organizing practices?

No. 2. Transcriptions of the conducted interviews

Respondent no. 1

Q: Okay, so to begin with, a few background questions. Can you shortly describe your role as an IT product development project lead?

A: In the company, currently I'm product lead for the product, responsible for product vision, product roadmap, basically product discovery and delivery phases.

Q: Okay, so to get more background into your experience with self organizing practices, have you ever experienced sort of a transition from traditional project management towards self organizing practices? And if so, can you describe what that transition looked like?

A: Yeah, so I almost haven't worked on the waterfall method, but I was already transitioning to it before I came here from other company, to the agile teams or agile working style. There were some teams who were working in a waterfall method, where they were planning for two years and basically agreeing what's going to be done in two years. So you would need to align with them when you're working on agile and that was quite complex to do because you were iterating quite fast or rapidly and those teams were working on one or two years planning.

Q: Okay, so then moving forward towards the team that you're leading, can you describe how your team adapts to changes in project requirements and the fast evolving market conditions?

A: I think they're adapting pretty quickly. We had tried quite different methods. So for example, in agile or scrum you can use either sprints or you can use kanban boards. So we're even trying both methods and see which methods fit the team better. I worked with multiple teams, for example, we started working in sprints, we saw that it's not working here and we moved to Kanban, which made us work way faster, for example. And with other team we worked with

Kanban and we were seeing that it's not working for us so we moved to sprints and that optimized our development, our delivery. So it really depends on each team to find the way for you to organize your work. But being adaptive to change is really important and that's what basically help us to make quicker deliveries when we can adapt to different technologies or different methods, how we work. So I would say I worked usually with the teams who are very adaptive to change.

Q: And could you give sort of an insight of how the self- organization really influenced that adaptability? What were the basic reasons why that team was or is adaptable?

A: I think it's important to give not like authority, I think it's not the correct word, but that the team is responsible for the delivery or for the product itself. If you're not responsible or you don't have accountability, you don't care that much, I would say. It's still harder to care when somebody else is basically telling what to do and when you have the ability to choose what you're going to do. So that's the key thing, that when you feel that you're accountable and responsible for that thing, so you are more eager to solve it in a way that you feel that it's going to work or you do more analysis on that, rather than when somebody comes and say you need to do it according to this, and they don't even know if it's the right way or not.

Q: So you believe that mutual decision making process also kind of influences responsiveness?

A: What do you mean by mutual decision making within that team?

Q: So that everyone is involved in deciding what we are going to do. What's the roadmap like, what's the vision and etc?

A: Yeah, for sure. Because then the team can find the best way forward. So if you have a problem, everybody has their own specialty, technical specialty, business specialty, process speciality. And if every person can contribute, it's way, way faster or easier to solve the problem. When somebody comes and just says from his own perspective, for example, business comes and

say some type of the thing you need to build, but they don't know technical limitations, they don't know technical capabilities or best way to solve it. They just say how to solve it and usually it doesn't work. It only works when every person responsible or specializes tells how it could be done or brainstorms how it could be done.

Q: Moving forward, how the flow of communication and information is managed in your self organizing team?

A: So it's chats, which is slack. We also use Jira for the communication, especially building for tasks in itself. We use confluence for more documentation wise, where you need to describe processes or procedures or store some specific documentation which is needed. We do kickoff meetings where the whole teams align and join and understand the problem, and they can later brainstorm how to solve it.

Q: So there's multiple tools, communication and practices so that everyone is pretty much updated almost at all times and aware of the information that's there?

A: Yeah, because if you're not aware, then it's harder to contribute to the project.

Q: Okay, and could you share some instances where this effective communication and the availability of information really directly impacted the success of the team?

A: Good question. I would say that if you don't have descriptions and this information about the project, it's very hard to build something. When we started to work on the products project, we didn't have anything written down. So, for example, tasks were described only in one sentence. So we didn't have any documentation, we didn't know how product worked and we didn't know what was built. So that was the key of some of our problems, that we didn't know how the product worked in the first place. Right now, when something is being built or something is done, we have a clear description so we can come back to it after a year or so. I sometimes come back to some tasks after a year, check how it was built and check if it's working correctly or not.

So without that, it's pretty hard to at least function as a team, because you don't know how anything works.

Q: Do you believe it sort of helps to avoid some past mistakes and reflect on the past experiences?

A: Yeah. Or even to check how something's working. So of course it's hard to remember what you build a year ago, but if you check the task, because usually everything is documented, described and available, so you can check or see how it should work. For example When you do some sort of a testing of the functionality of the product, you can see that something is definitely not working as expected. So you can know where you can refer to it or to even find out which code was merged, because all the code merges are connected to the task so you can find out what may have went wrong basically.

Q: Also about the team and team members, in what ways do you think relationship building between the team members contributes to the self-organising team's success and efficiency?

A: I think it's way easier to work with people which you build the relationship. Otherwise, if you have some issues, if they don't know you, people tend to either ignore you or they're not very keen to help you in a way because they don't know you. But if you have built a relationship, then it's way easier to ask for a favor. Or it's commonly known that if you help someone, the person can ask you for help, you will help, and then you build the relationship, and you know that you can rely on that person. So, for example, I have a colleague from the automation team, with which we even haven't seen each other, but we have built quite a nice relationship when she was helping me with some task. So when she even asks me some questions which is not related to the product or to my expertise we still communicate and I try my best to help because it's easier, because we know each other, so it's making it way faster in that sense.

Q: Then moving on, how does your team develop and maintain the shared vision and alignment on goals?

A: I would say through the roadmap. Our roadmap is connected and comes from the vision and to higher level OKR goals. So every team member, if they work on the task, they at least know from which initiative it's coming from. And that initiative is basically always contributing to some sort of an OKR or even the bigger vision or the goal of the project.

Q: And do you have a constant review of those vision and alignment on the goals? Or is it just pre agreed upon and then it's left for everyone to remember?

A: Yeah, we usually do remind everyone because we have the meetings for that, we call them syncs with the teams, where all team members are usually reviewing our progress. If we see that something is not working well, we can contribute to it and adjust or just leave it as is if we decide that it's not worth investing. So we review the high level OKR's I think quarterly, and also right now I would say more than every month there is a review. But from the quarterly reviews, you can see it better if something works or not. Also, because all the tasks are connected, team can always look into the goal it is connected to and which key result is contributing to it. Also, after each initiative, when it's launched, we have different review of it, so it's either customer satisfaction survey, either some sort of data dashboard review which shows conversion rates, for example, which we weren't going to have, like for example, with our checkout project. So then we know how this initiative contributed and whether it really contribute in the way we wanted or not at all . And also if it goes south at the end, let's call it like that, then we know that we need to adjust it.

Q: Can you elaborate a bit on how does that shared vision contribute to enabling the team to work in a self organizing way and still maintain an effective workflow?

A: I think that's what basically holds them together. That's what we are trying to achieve. I think we're a little bit struggling because our teams are in a different split, let's say it like that. So when you work as a cross functional team, which by the way I'm kind of still working towards and aiming to have in this company, the whole team contributes to some sort of a goal and they all work together to reach it. Right now we have multiple teams which are contributing to the same

goal, but they kind of don't understand that they do that together. So, for example, if one team builds it, but another team doesn't, it doesn't help the user in the end. So cross functional team I think is the best model from my point of view, and usually every cross functional team has the researcher, analyst, an assigned product person, developers from infrastructure, core, back end, front end, tester, designer. So they have full scope of people who can build the feature. So if you have these people assigned, it's way easier that everybody is working to the same goal and there is no split and they don't need to, how to say, scatter around multiple projects or products. They're building one together.

Q: Okay, and could you share an example or a situation where this shared vision and alignment on goals helped you to overcome a challenge or achieve a milestone?

A: Well it's not easy to distinguish one. But when we had about twelve people in one team and everybody was dedicated to migrate products. So the goal was to migrate all products to the a new CMS, like migrate from the old CMS to the new CMS. So we had quite a clear goal, we needed to migrate one website to the new CMS and fully launch it on it. It contained of almost 2000 pages and we needed to support it fully with the page structure, whatever, et cetera, et cetera. But when everybody had this goal, they were so dedicated and everybody even worked overtime, everybody were communicating with one another a lot, everybody even on the launch day stayed up until 07:00 a.m. Because they were dedicated to this launch. And we decided that even though it didn't went very well and in two weeks we were fixing issues, we still decided to keep it and keep it running on a new platform because everybody was so focused and dedicated to the goal and building this because for everybody was quite a huge milestone, which they never had done in their lives to achieve such thing. So everybody were very focused to have it in place.

Q: Then moving towards another topic, how would you describe your leadership style within the team, especially in regards to the autonomy and decentralized decision making?

A: I would say my leadership comes from the fact that I'm part of a team, so I just lead the way. So if my team has problems, with a pipeline or something, I still try to help, even though I'm not maybe a specialist in pipeline, but I will find a person who can help me to fix their pipeline or I

bring up these problems to my manager if it's something above that I could do. My role is to solve these problems, so my team can work without issues, or without any problems or without any blockers. I think for every product and project person, you need to make sure that your team has no blockers to proceed with anything that they're currently building. So if there is a question or they don't understand for example designs, you bring the designer, put them together to figure out. If there is issues with the delivery or some sort of a pipeline you bring in for a person to fix the pipeline, et cetera. I would say leadership is like moving away roadblocks for my team.

Q: And how do you balance providing support and guidance, with also giving them autonomy to make decisions and work on stuff?

A: So usually the problems they come to me with, it's something that they cannot solve themselves or there is something that they need support on, but sometimes there is problems that they can solve themselves. So you just guide them in a way so they could do that themselves. We have one example that our front end team needs to build a technical solution, new technical infrastructure for the front end application. And our goal is to guide them in a way so they would build a plan, be responsible for that plan, tell us what benefit it will bring, what cons it can bring, or maybe what happens if we don't do it. So they have full autonomy and full responsibility to build this plan so we could align with our product initiatives.

Q: In your opinion, what role does leadership play in fostering this and maintaining this completely self organizing team?

A: I think the biggest issue would be if somebody would micromanage the team. For me, I think it's the worst thing that could be done by the leader or whoever. This is not a leader, this is a manager, but leaders should basically, as I said, be a supporting role and basically fix everything, what's possible for the team. So you're a part of a team, and I really like these caricatures which sort of illustrate the situation when the manager is just standing there with the whip and the team is basically pushing the whole carry and the leader, on the other hand, is pushing the carry together with the team. So it's always like this, it has to be a person who's contributing fully and dedicating his full support to the team, rather than just regulating what others should do. And in

self organizing teams, I think that the leader is just encouraging or helping every person to find a way how to do it himself and if he cannot, then he helps to basically remove the roadblock.

Q: Moving forward, how different agile methodologies contributed to the development and implementation of those self-organizing practices that you have in your team?

A: I think using Agile itself is basically saying that team is autonomous because with waterfall teams are basically given a plan for two years. There you go and you work on it. In agile its basically the product person giving a problem and the team should work together with the product owner or a project manager and find a way to solve it. And it shouldn't be that the product owner comes, this is the solution, build me this. It should be more like problem solving. That's what Agile is basically all about. And there's those methodologies like Scrum, which helps basically to organize how the team works, but not limiting their imagination, maybe it's not the correct word, not limiting their brainstorming of innovations and ideas or anything like that. That's why also scrum or Agile has those sprints where the teams are sitting together, building the solutions together. It's also designers, not to forget, it's not only developers, but it's also those researchers who sits together. So everybody sits together and basically contributes and the same goes for product people or any other leader also coming and supporting the team and giving their thoughts, ideas to the team as well because they're specialized in their own field.

Q: In terms of specific practices that you adopted, it would be Scrum, Sprints, Kanban...?

A: Yeah, like scrum itself is one of the ways to work with the development team. And I think it's the simplest way to work with the team. I haven't really worked with different ways, so I cannot comment maybe on those because I haven't worked with enough with waterfall. So I'm not sure how it works in practice, I just know that they are usually given 20- 30 pages document and you need to follow it, but that's it. And with the scrum, you just get iterations. You work on an iteration, get feedback, work on another iteration, get feedback. And that's basically where the autonomy of the team happens because they get instant feedback of their delivery. And with the waterfall you get it after two years and you have no idea what happens.

Q: And with your current team, have you made any particular changes or adjustments to these agile practices so that it fits your team?

A: You always make adjustments for the team. So for example, you can even do story points differently with each team. We use Fibonacci model, because it's easier for scoring for us and it works better. But at the beginning, for example, we worked with T shirt size, which uses XL, L, M etc. So for each team you just find a way what it's best suited for. Our kanban currently with, for example, one of the teams is not even a Kanban itself because it doesn't even have specific, let's call it restrictions of the board. Because in Kanban, you need to assign how many tasks per column should be in that column. And we don't have that. So we adjust our way of working according how the team wants to work and how it's the most efficient for them.

Q: Have you faced any challenges with integrating these self organizing practices in your team?

A: Yeah, sometimes when it's a big project for sure, especially like for example, our payments project which is basically quite a big project so you cannot iterate that much or work in Agile way that much, because the functionality itself has to be built fully in order to work. So very small iteration will still going to be half a year. So yeah, it's harder, way harder. But with I think with Waterfall method, you would still do it in a year. And you don't iterate to see if anything works, with agile, you at least can do the testing, for example, see how it works and then iterate and see how something else works. So that helps a lot in this way.

Q: What are the challenges that you have faced when working with these sort of self organizing teams?

A: I think biggest challenge here is that teams are and feel separated. For me it's still quite a big challenge because every team has their own boards and follow their own roadmaps, and they're not combined in a one basically roadmap, we don't have that. And that's the biggest challenge. If we would work in more of a cross functional team, everybody would work on one board and would be fully concentrated to deliver together. And right now, when every team has their own

separation, it's pretty hard to get that common vision, even though they know that they're working on this vision, but they're still feeling that I delivered this or like my team delivered this, but they don't understand that the other team haven't delivered. So the user do not feel the benefit and that's I think our biggest challenge at the moment, to find a way how to make the teams work more closely. In that sense, they're fully autonomous, but in this sense, they're too autonomous, that they don't understand that they need to be closer to one another.

Q: So it's finding that balance between being fully autonomous and completely separate, right?

A: Yes, I would say so.

Q: And then the final question, can you share an example that you specifically would attribute to working in a self-organizing way?

A: So I would say the best for me is that we can do discovery and delivery, these two phases. So when in discovery, we work together with the specialists to find the best way how to approach the problem or what are the possible ways from business perspective to approach it, or from user experience and design how to build, for example, user journey. And then we find out if it's even possible or not. So, for example, in waterfall, when they have a problem, they write a plan how to fix it, and then they are implementing it for two years and then come back to it and see that, hey, it didn't work or something went wrong. And with self-organisation, you work together with the specialists and maybe you see that it's not possible, for example, to solve it in a discovery phase, and you don't even move to delivery in that sense. So you don't waste your development time, you don't waste anybody's time because you can just close the project before it even starts to go to development.

Q: Okay, so basically in your eyes success here itself is just doing things right and making them right and not building something that's maybe not worth building or won't work at all, right?

A: Exactly. Yes. So building things that are relevant and what's needed. For example with design sprints, we call it double diamond investigation, where you do the design, you also iterate so many times that you understand what basically works for the user itself. In waterfall, again, you just draw a user journey and you forget about it for two years. But in this way, you can iterate so many times and get the feedback so early on that, you will know that you're building at least the most correct way possible. And after the implementation, of course, there could be some adjustments, but it will not be changed drastically.

Q: Okay, well, that's that. Thank's a lot.

Respondent no. 2

Q: To begin with a few background questions, could you shortly describe your role as an IT product development lead?

A: So officially my role is Head of Product of the product of one the organization's products. I lead our squads together with engineering managers. And engineering managers are also led by our Head of Development. I'm leading from the product creation side, vision side, our roadmap, and the way how we're doing things. With other managers within the team we are sort of visioners here. Bye. Bye.

Q: Have you ever experienced a transition from that sort of traditional project management towards the self -organizing environment and if so, can you describe that transition?

A: Yes, very much so. So about six years ago or seven years ago, I started as a process change manager in Danske Bank. And then I transitioned to a project manager role about six years ago. And during that time, I was leading IT related projects. But during that time, the organization was switching from traditional project management practices to Agile practices. And they were putting different kind of skills into one team, into squads. So I was used to work like a separate

organization, project manager, you go with the usual flow, but the development team was completely separate. And this is where my transition to product owner happened. So yeah, there was a time in the career where we transitioned and where we went with the same team, developers, different kind of even legal tax experts and all that. And once we were a squad dedicated for one purpose, so yeah, there was a lot of trainings, how to switch from waterfall to agile methodology . So the short answer, yes, I did.

Q: Can you describe how your team adapts to changes in the requirements and also market conditions?

A: Yes, so those are the squads that we are running, they are quite autonomous, meaning that what they do with the squad is their decision, you know. If we start from me, so I align the whole product goals for the year, for the quarter, you know, the vision, where we want to move as a product together with the board, with my leadership. And then I go down and say to my product managers, this is the changes or different kind of goals that we have, but they are very broad, obviously. And we have squads now dedicated for different areas of the product. So for example, activation squad is responsible for activating users, onboarding, styling the product, introducing. Then we have file management squad because our product is about storage. So everything related to that core functionality, the files, uploads, downloads, then we have, for example, exchange squad, which takes care of sharing experience, throughout all the platforms. So we're working not via platform, but via domains. And each one has a developer, each platform has a developer to support it. So it goes like this, for each squad, we need to make sure that we now increase the sharing functionality and sharing adoption and all that. But the product manager themselves go and investigate what's the cool stuff to do with sharing topic. And they kind of say, I tried to raise the OKR's, or KPIs and all that, saying we will be successful if we raise the proportion of people who use share functionality to this point. And then it's up to the product managers and the teams to figure out what those features are. They can change one color of the button if that achieves the goal, I don't care. But then they say, okay, if we introduce this feature and this feature, we think that these kind of changes and optimizations, we can achieve this, and they go and do it. So I do not control it, no, that's because if there's some project that we say we have to pass through and all that, so there's a little bit of chaos. And it's changing every quarter.

So for example, a couple of quarters, we're focusing on one thing, but then we feel that, okay, this area is kind of served, we need to go and look and support other area more. So the wind starts blowing in my level, saying, okay, we have to optimize, we have to choose different direction, or we have to continue on what we did last quarter. That gives us a start for teams to be planning next quarter. So usually me and product managers, we're working a bit ahead, saying, you know, what would be awesome, what would be good, what would be good for business, what things would bring more value and why. So there's a lot of methodologies around that. And then they go and plan with the squad, work for the next quarter, and then the development teams kind of takes over and say, okay, you want to do this, we can do this, this and this, but we won't be managing to do this because it's too much. So this code planning starts to pick in. And then because we're planning for quarters, it doesn't mean that the middle of quarter, we can say, okay, this seems to be not working or it's not relevant anymore, so we're not afraid just to drop it and exchange it with someone and something more. But then again, there's something that we invest in a couple of quarters and say, let's finish it, and then we go next. So there is constant scope changes, every week, every sprint, every month, every quarter. So it's kind of a down the management chain, how we adapt to these changes. And then I'm worrying myself, we have strategy analysts, we data analysts, I'll think, okay, so what's the healthiness of our product at the moment? Are we increasing adoption of the users? Then the with strategy guy, we look at what's happening in a market to say, oh yeah, there's a couple of products that competitors have and they will make these moves with features, so we better do something about it or we switch to other area. It's constant learning of what's happening and trying to translate those learnings into some strategic actions.

Q: And what does that autonomy give to the team when they have to be rapid in adapting to the changes that you have mentioned?

A: Yeah, so what it (self-organization) gives, it gives a bit of a freedom to change what you do and how. So it's not set for them, you know, you must do this and this and this and this and this. But there's some sort of a north star set that we're trying to go with our team towards. And the better job I do in terms of explaining them what are these north stars, what are these metrics that will be, like the greater the better, that's not always the case of course. But the teams then they

say, okay, we know we can drop this idea, we can experiment with this idea, we manage ourselves how much time we spend for experimenting or learning and how much time we spend for actually developing. Something might work, something we need to put on the shelf. But that happens closest to the user because now I lose that sense of how some functionality really works on the user side. But the team which give it every day which analyzes how users use it, every day they're looking to different products around the market without the specific functionality works, they know best. So I might have an idea but I can always be challenged by product manager and engineering manager. So it's better to do this and that or users don't really need that when we ask users. So if there's arguments against that, I'm sort of okay, there's not a massive questions. But it leaves the team with freedom to plan the time. There's no someone, okay, it must be this, no matter what, don't look at the sort of what is happening around you and just do it. Like standard project management - there is an excellent line to do this, so do this, no matter what.

Q: Could you specify a situation where this self -organization of your team, improved or helped to solve a challenge that you came upon?

A: So it could be from our adoption topic, right? So we started these bundle deals with other products to sell all three products in one combined plan. And the problem that we have is like, it still works as a three separate product, right? And we noticed that it's really hard for user who buys, not for example, our product to go with all the other products. It takes like 30 or 40 screens to onboard all of those products. So the drop rate is really, really low. So instead of coming up with solution for the team and saying you must do this and this and this, we can look, the onboarding rate now is 8%. We need to raise it up to like 12 or 14% in the next few parts, you know, just to start moving into the right direction. And then, and this is, you know, there was like, you know, product manager goes, analyze, they look into data, they go and talk with other product manager, saying, okay, which kind of coordinates we have to, you know, remove, which steps we can avoid, in order to make it quicker. They also went and did analysis with our users, interviews with users saying, you know, what's not working, where's the, you know, why are you not going to the end and all that. So they go and identify really what's happening. So it's not my opinion, it's not like, you know, words, opinion, what's not, is actually going on the field and checking what's not happening. And they did some changes, which constantly start to improve

with that, you know, it was not like massive, double rates, but the quarter after quarter, we could see that half percentage point is increased. So we mean that change goes to the right direction. And they continue doing that. So, so I guess that was that, because when someone from leadership comes and say, they might have great ideas, which could be tested, could be fed, but usually those closest to the user knows best in that particular area. For me, you know, why it's important for us in first place, make the adoption not the problem for us. Maybe there's a different performance issue that we should go for. So there's a bunch of things that I need to take care, you know, and make sure that I set the right priorities. But when we have a problem, sometimes you can leave it just as a problem. And the solution might come from, from a team that's equipped to do that. So this is the way it works for us.

Q: Moving forward to another topic - how is communication and the information flow managed within your team?

A: That's constantly a topic of all the poll surveys and all that. So, yeah, I think we had some issues and we also have some fixes to improve, but in general the communication flow of deciding what to do happens through OKR's planning process. Then I have constant sync's with all the product managers. So every two weeks we have product managers assembly where we share what the topics are, maybe there's some shared project that we need to take care of, there's maybe an issue on one squad field causing an issue on another squad, then other processes how we release things together and all that. These are all needs to be aligned. So we have these product assembly every two weeks and then we have every month assembly with product engineering managers. So both development parts and product parts comes together. But the main communication also happens in the squad. So product manager and engineering manager, they constantly working together in order to clarify requirements, prepare user stories, clarify what kind of path of development will be taking and then we have one-to-one's with product managers, and I'm dealing with each of those angles in more detail. So that's the sort of my side communication line. Then on the whole product level, every two weeks we have all hands meeting where one of those meetings, like every month, we take the two types of meetings. One is product pulse and the other is OKRs. So for product pulse, we go into each squad, usually at the marketing, the people part and all these sorts of parts of product and updating what's

"cooking". What we kind of did, what demo we can demonstrate, and it is all about demonstrating the work that we did. What goals we had these couple of sprints, what we achieved, what we didn't achieve. So we try and talk about that. The next type, OKRs is that we're talking from high level as well. Okay, so these are objectives that we have for this quarter. How we are succeeding towards those? What is lacking behind, what is being achieved already? So everyone sees the same picture, you know, and then we try to connect everything what's done in the squads, towards what we actually trying to achieve here in general. Because usually for communication, it's really hard for us to watch the developers to understand why we do this, what it is contributing towards and they need that. So these OKRs meetings is for that, to clarify f.e. that our goal is to increase our active user base and we have these initiatives focused on onboarding related to performance and quicker opening time. And we believe these are contributing towards that and they are working. Sometimes that connection works, sometimes it doesn't, sometimes we have different topics to talk about, it takes lots of time. So we try to keep that rhythm. I think the communication, the rhythm that we have with our team, the rhythm as we have with each squad, in each one individually, and then the product itself is really important. So we have a rhythm on a product level rhythm and then each squad decides what kind of rhythm they have within their squad. What's work best for them? They start sharing release processes that need to be communicated, like an example, those are formal communications. Another example is that before freeze day, each team has to set the release note which is final. And then every Monday we freeze the development and everything, once ready goes into release candidates. These processes kind of forces that communication and the fact that you must align with each one. So this is important, but again, we try to create this sort of culture where we share what didn't work, what worked, and those post meetings after that are for it. There someone hears that you're trying to do something this way from a developer, and they have similar issue I can go to. Then our engineering also have a flow of how they do that. They are guild meetings, they are specialized meetings, town halls for engineering. So we're trying to limit these meetings as much as possible, not to break too much. So those meetings are kind of cornerstone, but then again, sometimes it doesn't work, but you need to share, different information between managers, the managers have to really communicate quickly in our form of the team. It goes top down, bottom up, you know, all these things. Sometimes it's slower than it should be, sometimes it's

happening at the right pace, but not maybe confirmed on all levels yet. So yeah, communication is a tricky part, but we have fixed things in place to keep it moving.

Q: And on top of this, what you mentioned, constant meetings, constant communication, mandatory meetings, do you have some sort of a knowledge base to keep that information available?

A: Yeah, absolutely, confluence is the key for us. We're documenting ideas about new features, the different POCs, architects have the flow on how they get the idea from squads and prepare different documentation. So, development part has a lot of that, from product side, we document everything, you know, you can talk about the vision and how we imagine, but we try to actually make prototypes. Actually design the thing on Figma and then write descriptions for developers, to show them how we imagine it. And then we start identifying places of trouble right before we go into, creating Jira tickets and all that. So, a lot of that happening but again, not perfect on every corner for sure. And especially when you have to handover the task, you see, that this part is not documented, so we ask the team responsible to go and document it or if the other team documents it. This is important and we do a lot of that thus some practices are really formalized. For example its clear that if we do something new, architect needs to do this and that. But some practices are also dependednt on the team, which sometimes creates little problems. One team is used to writing documentation one way. Another team is more precise or more detailed, but moves slower. And then there is also Agile values, working on product versus, documentation. So, yeah, sometimes balance is there, sometimes it's not.

Q: In what ways do you think this relationship building between team members affects and contributes to the team's efficiency and the way they work?

A: I think, it's the key. There's some team members who sits together and the more this close relation between team members exists, the better, because they can understand each other from the first sentence or know exactly the person where to go to consult. So different team building things and all that are there for the reasons, so people can talk and form that relationship and it's almost a rule that in the beginning that squad is new and just formed there's not a lot of that, but

the better they get to know each other, whether you can identify flaws and conflicts and build that trust. Once that trust is spinning, you can have these really good communication circles. It's really important that I personally invest a lot of time into creating those opportunities to form that culture and talk to each other, be friends with each other and be nice, because if you know the person, if you trust the person, of course you don't need to be friends with everyone, but you're closest team. Squads can form their own subcultures and if it's formed and strong, then it's happening, you know. So it's the key.

Q: How does your team develop and maintain this shared vision and alignment on goals of where the project is heading and where is their direction?

A: So one thing is how the team perceives the vision of the squad. It's, for example, we have a sharing functionality area, right? So product manager needs to do a good job in describing that vision for the squad. It's our share domain, it's the domain that we have, so our vision is to do this, that and this, and then they have to link it with the shared goal of the whole product, which is my part. What I communicate for everyone at the same time usually, or sometimes when we do a bit of changes to the strategy or to the, to the vision, I go around different squads and explaining the shared vision and then what it means for them, but then it's also a product managers part to align their vision with, product vision and then translate it. So this is how you get that breaking down of the key things. What we want to do become this, okay, got it. But then my squad is responsible for this part of this pie. And then we understand what we want to do with that pie, where it connects. That is the vision part that usually also OKR's helps to connect those points as well. So, okay, in order to go to here, when we think that we need to achieve this objectives, become this, become this, become this. You have to have that structure of goals and KPIs on top of it, or next to it, it is really important. And you see that in places where it's not that well connected, you have performance issues and all that. Also, it could be that in parts where everyone knows the vision but there's also some performance, it could be everywhere, but it's really important for teams to really get motivated to work on something. Because you might be working on cool stuff, and it motivates you, but, if it doesn't contribute, then you can detach yourself from the squad. So it's important.

Q: So, if I understand correctly, it's sort of a constant thing of reminding people and keeping them aligned on the vision rather than setting it and then forgetting about it?

A: Well, as I say, vision is something that should be there for one, two, three years. Depending on the product you're working, and the market you're working in, maybe it's will not work out with a five years vision, unless it's really broad like "making internet a better space". So it's a good one, it's a mission, but it needs to be broken down. If we want to go there, so first year we have to do this, next year we have to do this, so one, two years, a vision of what we want to be, how we want to look like in one or two years, is something that I usually want. And then once you go, sometimes some actions and strategy changes, and something that you see in the market makes you change that vision, and then you change it, you adapt it, then you can recommunicate. But again, as you say, you need to constantly remind, because people forget, there's new people joining, things are changing, so it's a constant work. And I myself sometimes go for a long time, I'm not talking about that, and if people start speaking of being detached, the there is need to come back and remind again, but it adapts as well.

Q: And can you share an example of where this alignment and the shared sense helped the team to overcome a challenge or reach a milestone that you are aiming for?

A: Good question. So I think, recently we did a lot of reshuffling in our product, how we need to be organized, where we need to optimize and all that. And we have it that, in our vision, we want to become a really good service for the users, which hold everyone together a bit, no matter how we are organized, or how we are placed in the different teams, or what if we need to maybe reduce focus here or there. But the goal kind of remains the same, and it helps to people not get lost in that organization. So it really helps in our organization to keep that sort of momentum. But there's also multiple things like, when there is these two squads which are working on separate things, but eventually it comes down to the very similar thing and connects. So there are some moments where you feel that, it's like, "oh nice", where they were working separately, but suddenly the features that they produced, connects, because it shares the same values, the same vision. In our case it has to respect privacy. It has to be efficient, but fast. So yeah, it's aligned.

Q: Moving forward, how would you describe the leadership style in your team, especially regarding the approach of facilitating the autonomy and decentralized decision making?

A: So if talking about my leadership style, it encourages a lot of autonomy and trust. It encourages a lot of creativity. So I trust people, I trust those who are leading different squads. And then I really encourage challenging, as it does not matter if I say the goal is this, we are feeling safe enough to be challenged, and those who are challenging feel safe enough to challenge, it's really an important place to be. So yeah, my leadership side, I don't know if you can put it in one word, but it's more like a bit of soft, and trustworthy, encouraging people to be empathic. Empathy is the key thing for me. On that side, I'm not managing to predict and say that this has to be this way or another way, but sometimes we have to have that, those OKR's, those organizational sort of things. You can choose the path, you go in through the goal, but the goal needs to stay there. And we do constant revisions of you, if we're not achieving something, so why we didn't do, what could have helped. The goal maybe was too ambitious, so we are learning through that as well.

Q: And how do you handle or balance the guidance and the support that you provide to your team with that autonomy and full trust in their decision -making process?

A: Well, the goals really help you. So if you have a goal, then you have that relationship, then you explain really clearly what your squad is responsible to do, and then your performance is linked to that goal. When you are kind of clear on that side, then you can go and say, I trust you and can you find the best decision. I'm also constantly double checking if the person is not lost in those goals and all that. And the balance is that you give autonomy and freedom, but people know what needs to be there, not like saying you know best you so you can go do whatever. So that's one part, another part is let people go and do whatever to achieve something the goal, help us. If you see that people don't have their own goals, within their own squad, you help them to define them. Some people have more autonomy, so they have it already So, you're gonna go and double check if the person is just spinning in one place, you know, or they actually connect their work and their efforts to something that the whole product wants to achieve.

Q: Have or how Agile methodologies contributed to the development and implementation of self-organizing practices in your team?

A: Very much so. There's a lot of adjustments and tailoring and customization, but everything kind of goes down into agile ways of working. Those principles, those values, some things are more strict, some things are more based on agile, there's some practices that could be linked with other methodologies like standard, classic project management. Those values and things like it's kind of a cornerstone and DNA, so it's very much linked to that. All the training is spent with teams or product managers, it's based on agile ways of working at different levels, different methodologies and all that. The way we organize our project or product managers development and that competencies matrix is coming out of Agile. And leaders in practice use the idea of agile like the whole developmen in the world, in IT world and tech world especially.

Q: Are there specific Agile practices that you find particularly effective in supporting self-organization?

A: Yeah, so we're working in dedicated squads. We have, cross -functional teams. We have different skills set in our team on the specific area. We're working in sprints. The team is self -organizing, so they decide what kind of practices and ceremonies they want to have with the team. Then I think there's different metrics on the IT side, on the engineering side that they measure, you know, like shipping time, velocity, quality, you know, that those metrics are also there to help us monitor. Then the whole OKRs thing, how we kind of set the goals and then those goals goes into teams and teams tries to refine them and it's also based on some methodologies. So, yeah, I think our whole operating rhythm is set by Agile.

Q: You also mentioned that there is a lot of customization and adjustments. Can you give us a few examples of how you adjusted the Agile methodologies to fit your team setup and the way you work?

A: Yeah, so or example, when we have a sprint, we have certain ceremonies like refinements, grooming, and all that. So we allow some of the squads to decide how they want to do that. Some

of them decide to do refinements based on the platform. Although we are working in the same model, but we're not dragging all of the developers from different platforms to talk about. Maybe they have a new feature, introduction, they do it all together. But then you go in a specific platform to adjust one or another thing. They don't need to be in the same room, so we allow them to create their own rhythm around that. Sometimes product manager goes into a couple of different refinements to talk about different platforms because they don't need comparisons to other platforms because they are not there or they are already ahead. So they're trying to connect those things. We allow them to do quite a lot of customization on how our teams are doing their ceremonies of Sprint. And then we also connect that final ceremony like demos. Each squad and the final day of the sprint, they invite stakeholders and demonstrate what they do. And this really in the bigger teams is that when we have more squads, it's really tricky for the stakeholders. When you have Friday, for example, every hour we have a meeting on different squads and you just go and accept the work, right? Product manager accepts it, it's already commenced, so that's all good. But then we kind of combined those demos into those pulse meetings, that I mentioned to you before. And we figured that this is the place where all the squads present their work and then there's opportunity for people to comment on it. As an example, like a legal don't need to go to each of the squads and see what new features they've released. But they go into that one pulse meeting, they see what people are working on or planning to work on. And they say, oh, this might have legal implications that we need to take care of. And then after that meeting, they go to product manager and say, okay, we need to talk and find out if my consensus is there or not or what we need to prepare for. So it helps to hold that communication line open.

Q: Could you be more detailed about the challenges that you have faced in integrating these agile methodologies and self-organizing practices to your team?

A: Yeah, it's a constant looking for the most effective way, what would be faster, and there's different developers, different leaders who use it to work in different ways. And each that episode, maybe this week we should work on this or that, so there's a constant battle between arguments which way of working is better. One one kind of improvement should make everyone work better, but that improvement may be good for this team, but not good for that team. So this is a constant challenge, what processes and what practices to make global, you know, mandatory

for everyone, and which of those should be kept in the team level, because you want to give that autonomy. But you also have to have those processes where, you know, you manage to release something that compiles from different squads work into one thing, you know. So it's a constant challenge to format. It took years for us to shape something that works, and then we constantly are looking out to improve it, with some automation or something, it's really a huge job. You know, a really particular example was our release load, the when we started working in a bit larger number of squads, different squads working at different base, then they release something. So how should we manage to test everything that its not breaking, how are we going to stage it through the testing and to make sure that we release, how we check the release itself. So we came up with this release train methodology. It's sort of our own thing, but its based on multiple methodologies, like Agile, also some sub-practices that engineers saw around, and the owner of this release train, which takes care of everything, is changing every two weeks. There's different steps that we have to go with testing to follow, improve it, you know, so we'll be collecting approval also. All that was really challenging to build up as this team was just starting to self-organize itself, but now it's working as a clockwork, but again, still a lot of things to improve it. We are trying to do, but so yeah, so that was, that was the example

Q: What's the ideal that you're heading towards with this self -organizing team that you are building now?

A: Well, yeah, I think the team which, is a strong operating team, that can autonomously achieve those goals, and be so good at what they do, so they start leading the visions themselves. So they are not just listening, you know, what is the vision from somewhere above, but they also are so good in what they do. So we are able to understand the goals, also understand how to deliver value for the user, for the company, and we think this is the way, and it's more like, not top down approach, but by teams and people who understand what they do, and know the market around them, they have all the necessary tools and support and knowledge around them to make those decisions, and they are so impactful that, a leader's sort of become as administrators. Obviously, leader, someone to shape that, so product managers, engineering managers, nowadays, different things to be a lead on, but the team is so good that they get, you know, experiment, they get

research, they can do all on its own, and then make a huge impact for the overall roadmap of the project.

Q: And if I understand correctly, the major challenges that are sort of not preventing you, but are making this progress slower is the fact that you're having to figure out this global versus local approaches and practices as well?

A: Yeah, yeah, I mean that once you do a job in particular area, someone needs to be there and think, where is the whole thing going? Money, performance, value for the company, value for the user, and all that so there has to be someone who needs to connect those thoughts. But usually, you know how it is especially in young teams that that someone also brings and knows what to do. The task. Well, I think this needs to be done, this needs to be done, this needs to be done. And then we have more mature teams those teams are they are really known because they collect so much experience, they know what they do and they can have a good sense of feeling and knowledge which tweaks on one or other are will bring bests results. So they really understand how the business performs. They are also invested and usually when you look at the teams are more mature, you go and you talk to them developers usually talk about money, business kind of topics and they are interested and they know how they work connects, they know actually bottom line of how much we we are costing, how much we are spending how much we earning and all that. Those people in mature teams are able to link, their work to business metrics and how it contributes. It's not only like " I completed my two story points, I'm done, I'm completing my metrics. You need that, but you should find that relationship. And if the team finds that relationship the commercial part, it usually naturally happens there.

So that's all from my side and thanks a lot!

Respondent no. 3

Q: To begin with, can you shortly describe your role as an IT product development project lead?

A: Yeah, sure thing. So at the company I'm officially a product owner, but I also cover product managers position. So in essence I work mostly with development team in executing the product roadmap strategy, mostly in the development capacity, but also work of marketing and so on.

Q: So you manage a few cross functional teams?

A: Yes.

Q: And have you ever experienced the transition from traditional project management towards self organizing practices? And if so, can you describe that transition?

A: Yeah, I used to work in a corporate environment for a large company that had like 20,000 people. It's a drastic change, but there are people who like the corporate environment and who don't. So the transition from a corporate environment for me personally was quite amazing, but I had some previous experience working in similar way. I work now at the company, so it wasn't a shock to me because I had already experienced it. The biggest shock was when I was moving from a completely agile environment to a corporate environment. So that was difficult. But yeah, the biggest changes going from the traditional organization is that you are the main driver for absolutely everything and there's no direction of what you should take. So it's not for everyone, I guess. But if you're the sort of person that likes the unknown, then that's good. But if you like to work with processes, everything's clear and structured and so on, then agile and self-organization is chaos for you. And I think a lot of product managers or goods managers couldn't manage with little or no structure. And the decision making process in a traditional way, you have to go somewhere up or you have to talk with a lot of stakeholders, whereas here decisions are made sometimes within minutes or in a single meeting.

Q: Moving forward, can you describe how your team adapts to changes in the requirements and market conditions?

A: At least my team has absolutely no problem with that. Obviously they would like more direction, but if we plan a sprint or two in advance, then that's fully enough for them because like myself, our team still don't have a clear priority for Q1 and it's fine by them, but they will be asking quite soon what's coming up, so I'll have to steer them in the right direction because it's my issue. But again, it depends on the team and most of the team is like that, but the team isn't that big, it's like ten or eleven people. So some of it do like to have a lot more direction and that requires a bit more work from myself.

Q: So not having that sort of a long term plan is what enables your team to be pretty flexible and adaptable?

A: Yes, definitely. They think a lot less about it. Because in a corporate environment you think about X-Y-Z scenarios, and then you just think too much because, you potentially know what the future might hold here. We just think two weeks ahead and don't overthink any other topics. And I also try to encourage that because if I see that this feature should be out of scope, we probably will do it next quarter. So let's stop discussing it at the moment because it might interfere with our current work and that saves time usually, because sometimes problems do happen because we should have thought of this like three months ago when we were doing the first task. But that rarely happens in my experience at least.

Q: And if there are any unforeseen challenges that arise, what role does self-organizing practices play in dealing with them?

A: It's that usually employees take responsibility on their own and they don't always ask the managers for help with this and that. It's within their sort of responsibility. Because a developer and a core developer should communicate directly instead of going through a product manager who is not always technically capable in explaining and discussing stuff. It's sort of relying on that expertise on the individual level.

Q: Could you give an example where this self-organization improved the agility of the team and helped them overcome a challenge?

A: Well, at the company we have a lot of communication between developers and quality assurance testers. They literally sit next to each other. And previously, some time ago, all QA work was sort of reported to the product manager that product managers would prioritize for the development, what should they fix and so on. And that was just extremely time consuming. And product managers not always understand those technical issues. So some time ago our QA started sitting physically next to the developers and solving those problems. We had bugs pretty much every day, but that physically sitting next to each other was quite a huge improvement. They were able to just communicate directly, instead of writing slack messages, emails or whatever, they were able to just say "hey, look at this", then just create a quick Jira ticket, write one sentence, and that saves time. The developer also sees the QA screen so it's a lot more informal, but it's a lot faster because they're physically sitting together. So that's quite a good example, I think, where the process of fixing bugs would drastically improve.

Q: Could you elaborate on how communication and the flow of information is managed within your team?

A: We have weekly meetings where we discuss high level stuff, and the team can always create their own meetings. Invite everyone. I always ask to be invited, because if they doubt should I be there, then I always say, yeah, just involve me. It's better for me to know everything, what's going on. So it's a completely informal environment. And in slack, we use Slack also mostly for quick communications, like, oh, who can take this task? Or is there like a quick question, can we do this? So quick questions like that, but if I see a huge discussion going on, usually if there's more like five replies or messages within the chat, then it requires a meeting. I then quickly go and schedule a meeting and it's a lot easier to solve those issues face-to-face or on an online call. So, yeah, I try to organize or sort of manage this, and not always leave to the technical people to discuss things, because they will write 30 messages and won't meet face to face. So that's one of the things that I always try to do.

Q: In what ways do you think relationship building within team members contributes to the efficiency of the team and how they work?

A: I'm not quite sure how to answer that, because in a corporate environment where I didn't have a good relationship with people. If the work has started, it means that it was decided by 20 people, then you know what you need, and then a developer or a technical person, let's put it that way, is capable of doing the work, then we communicate even without having a proper human connection. Maybe having a well functioning, dynamic team like I do at the company, it adds maybe 10% of the work quality, which is not huge, I wouldn't call it a productive way that it's a lot more work efficient, but it helps people to be more motivated to come into work and do stuff. So I would say having a good relationship contributes to the intangible things like employees wouldn't think of changing their workplace because they have good team members, good managers, and they just enjoy their work.

Q: Could you specify a situation where this flow of communication, the availability of information, and the relationships within the team were critical to the success of the team?

A: Yeah, probably. It's hard to measure because you wouldn't know if it wasn't a functioning team, what would happen. But, yeah, for sure, because we did quite a few projects recently in a very short amount of time, relatively, and those projects were very complex. So migrating from one SDK to another SDK, and that was done quite fast, because there was mainly two team members within the development team that sort of distributed their work very well. There was literally no downtime in the development. It was like "you should do this task or I should do that task, and once you finish that task, I'll start that task". So there was literally none of that. And that absolutely reduced the developments of the new SDK and hopefully we'll be releasing in the coming days.

Q: How your team develops and maintains the shared vision and goals that where you're heading towards with your product that you're building?

A: It's up to me to draw the vision. So I always try to put analogies in place of like, let's compare us to a very well known brand that probably they use or have heard of, and then they have a bit more connection of where do we want to be. Because working at this company, they don't see

company as a brand, they see company as their employer. But if you say that with product, with companies product, we want to be Tesla or whatever, they instantly sort of associate and understand where we're going. I always give some business context to technical people of what's going on and that sort of also involves them in the broader idea with the product. So, yeah, just not going purely hardcore technical with them with product, but also saying of where we're going, some interesting facts about our competition, of what we're doing on a high level. So that always engages them, in my experience, they're a bit more involved and most of the time, they sort of have an idea of where we're going.

Q: And do you have any practices where you remind them of this mission and the goals where you're heading towards? Or is it just before the project starts?

A: No no, every two weeks during our weekly meeting, I spend about five minutes saying most important business topics, sort of general topics of where we want to go with the product and just reminding them. Then we do a small exercise as well. Also, it's not a set date, but about twice a year we do sort of also our product impressions as a new user. So always ask developers to go through our product from start to finish, because that's not what most people do, so it's at least employees, developers interact with the product like on a full lifecycle of our users instead of just smaller task. I always try to do that and they just go one step back and just hopefully see what our users see and that gives them a bit more what our users feel, the product.

Q: What role does having this shared vision play in the way that your team self-organizes itself and works autonomously into building the product?

A: Well, they usually know the things that we're doing, why are we doing it, and which sort of metric or user experience improvement that will contribute to and having sort of a goal of why are we doing this and why it will be useful, or why it was unsuccessful as well. Yeah, they have, I think, a bit more, how should I put it, they don't have to communicate as much with the rest of the team because everyone sort of understands where we're going. And if issues arise, it's a lot easier to solve them because this is our business goal, this is where we want to go. This agreement of a certain development topic is going to help in achieving that business goal. And it

helps solve issues a lot faster and not go deep dive into meaningless conversations, long chats and so on. So it's just a lot faster if everyone, technical people know business goals as well, which is not that easy to do, but they at least have an idea. I dumb it down usually so it's not super advanced business cases with 60 lines and so on. And all of this in a way helps to save time from aligning on topics and ideas that should have been aligned already.

Q: Moving forward, how would you describe your leadership style within the team in terms of autonomy and decentralization?

A: My leadership style is definitely not micromanaging or babysitting and counting time. I give them complete autonomy to do the tasks, but my job is to clearly communicate of what we're trying to achieve and it depends. If it's an easy task to do then it doesn't require much communication and it could be a slack message and if we have to do something a bit more complicated that involves kickoffs involving other people, deciding on the teams, stakeholders and so on. So I try to give them as much autonomy as possible and I always try to allow them to make the final decision, although I will strongly give my opinion as well. I know that they're better experts and I do allow them to make decisions and sometimes make mistakes because I think they learn a bit better from their mistakes instead of me saying from my expertise, which is not always right, to make decisions and make mistakes.

Q: And how do you balance providing guidance and supporting them with giving your team the power to make the decision by themselves?

A: It's probably from my experience, generally it's gut feeling. You just see from people's body language the way they talk. Are they sort of mad, joking or not? It's intangible things again that you look after when you present your idea. Let's say you said that someone's idea is bad and so on, primitive, then just observe the body language. And by that you learn, working with the team month over month over month of how they are, what they like, what they dislike, and you just adapt after a while of how much guidance you need to give them. So yeah, previously I had one team member that left which didn't need guidance, like zero guidance, needed just a general idea where he needs to go. But now that team member left and I got another one and that new team

member needs a lot of guidance of what to do and so on. So it's a massive change. But you pick up on how people work and how much direction they need just by just time wise and observing them. It's pretty much all about reading people and understanding them.

Q: Moving forward how agile methodologies contributed to the development and implementation of self organizing practices within your team?

A: Well, I think agile now is kind of worn down and everyone knows it, but not a lot of people talk about it anymore. Ten years ago it was a hot topic, but now I think everyone's so used to it, it's either a corporate environment or agile environment or somewhere in between. So we don't really talk about it. It just comes in naturally from the person who's driving projects. Well, in that case it's myself. My human nature is approaching uncertainty, so I quite like agile because it correlates with my character of not having proper structure six months down the line and so on. It's working today, adapting to what the client or the user needs and prioritized it from my experience of what things we should do.

Q: Are there any specific practices of agile that you have integrated and you find effective in supporting your team?

A: Well, sprints, we use all the boards of prioritizing certain things, story points. We try to do that, although it's quite unsuccessful in my experience, and that's pretty much it. And just adapting to all the changes week by week pretty much. But also trying to have this is a new thing, having the full quarter of what we want to achieve and we'll try and do some tracking of those metrics, but in my experience those could be a bit demotivating because you will not do a lot of tasks. And if look at the quarter which you had like 30 tasks planned and you only completed ten, that looks bad and it might make the team feel not productive. Whereas me as a product manager, my idea is to do as much task as possible and whatever works, then that's fine.

Q: What have been the challenges with integrating these practices into your teams workflow?

A: Yeah, developers don't read user stories, or they read some of them, or they read once and that's it. Requirements change and a lot of people don't look at the updates anymore within stories or Jira. But if you constantly update those stories and people get notifications, then they stop reading as well. So that's too much communication which is informal, it leads for them not to read. So you create sometimes instead of ten updates, you create a meeting, which again is a bit more time consuming than reading a line. But if developers don't read it, then you have to physically create a meeting and go and talk it through, which usually takes 15 or 20 minutes because there's more discussion. But the upside of that is that developers are more involved in what we're doing and they have more context and people are a bit more involved.

Q: And what have been the challenges that you have faced when you are working with this self organizing team in general?

A: Well, sometimes it happens that priorities might change extremely fast. So that does happen. Like, my manager comes and he says that we have to do this, which is more important and I'll take two of your developers for two weeks. Is that fine? Well, it's not a question, it's a statement. So that happens because someone didn't plan their work on a high level and they have to take some of my developers and that two weeks downtime for one of my teams at least. So, yeah, that's the issue with self organizing, but it's not 100% self organizing. I would say 95% self organizing and 5% not self organizing. And those 5% are a pain in the ass when they do happen, although they happen very rarely, but they do happen a couple of times a year, I guess. Sometimes while you're thinking about it, having some direction from whatever higher ups or the bottom up is nice, because as a product manager working in that kind of environment, you might think eventually that you're doing something wrong. It could be something done better and so on. So you see guys from outside, but if you're self organizing, then that's not always the case. Obviously you have your manager, but he's not involved day to day. So, yeah, that would be some of the drawbacks. A couple of them at least.

Q: Can you share the significant success or a challenge that you've dealt with completely due to the way your team is self organizing?

A: I would say the biggest one is that developers and quality assurance talk directly and very informally.. That's a huge improvement because usually I don't speak now with QA at all. As in they're involved in most of the meetings, and they sort of in the loop of what's going on, so they can plan their time and so on. But during a project, I only get like two or three questions from QA, which takes me 15 minutes. I would say that's an amazing achievement because with quality assurance before, you have to spend quite a lot of time directing them and so on. But that's also proper user story management acceptance criteria that has to be prepared. But, yeah, I would say that's one of the biggest achievements here at the company, at least. From this perspective I'd put it this way that it's not the fact that what you do is a complete success, but rather the fact that you do it in a successful and efficient way, that the team is able to work in this way.

Q: That's that from me, thanks a lot.

Respondent no. 4

Q: To begin with, can you shortly describe your role as an IT product development project lead?

A: Okay, so I work as the manager within the product. So basically recently this role was still counted as the product owner, looking from the agile perspective, but then it was split it up into two separate roles. And now I am the product manager and as well we have the engineering

manager. So from the project management perspective, my main responsibilities is to have the conversations with our customers, with marketing, legal, sales, understanding our existing and potential client needs as well to do all the discovery. So let's say we have within the squad our main responsibilities, topics, and basically I am doing the research and discovery in those specific areas. So that we could build our future roadmap based on that which. And I am also responsible for maintaining that roadmap itself.

Q: Have you ever experienced the transition from the traditional project management to self-organizing environment?

A: Yeah, I would say that it was not in the organization but rather my previous roles. So yeah, I was working as PMO and basically I was working within the company's center in Lithuania and they were trying to adopt agile and moving out more from the Waterfall but I was not there in the full transition and it was very complex, you know, the huge organization, lots of bureaucracy, etc. So I do remember that it was quite a huge struggle at that moment and it was like shifting the mindset from the waterfall to agile approach and it needs to be done not just at the team level but as well at the management level. So yeah, it was challenging as far as I remember.

Q: Can you describe how your team adapts to changes in requirements and also to market conditions as well?

A: Okay, so first of all, changes to requirements. I would say that we have quite little, if I will take our recent year as an example, quite little amount of scope changes or requirement changes. I would say maybe it is because of my squad specifics, the topics that we are taking, they are quite big. We don't have like very small changes or something in comparison to other squads. For our squad, it is not that much. It basically very much depends on the research, then prior to that, you know, understanding all the flows, if we have covered everything, if we did not miss anything, etc. And yeah, if the research is done prior to that from the business side, because I'm working in the enterprise squad. Well, those requirements do not change that quickly or that fast. So if they change, then I would say our team is still building, but it is already in that pretty much self-organizing already part where they have lots of open discussions. And yeah, basically that

helps to solve those uncertainties, all those changes quite fast and to agree how we move forward.

Q: And what role, in your opinion, does that self-organization and its practices play in enabling your team to be this adaptive and responsive?

A: Sorry, could you explain the question?

Q: I want to understand what exactly are those practices that help, and how they help your team to be adaptive even to those small changes or rare changes that still occur?

A: When the new requirement comes, or something unexpected appears, something missed a flow or whatsoever. So it is not that, without not letting the team know I or engineering manager do certain research, do the communication and then provide the results to the team and that's it. this is what you need to do. No, so that is by enabling and involving the team in all that process, right? So that they make the decision they do as well the investigation and they understand from the very beginning, they are aware of the new requirements coming in. So this way they already can think of what we are currently implementing, is that impacting the new requirement? Do we need already to shift how we are implementing whatever we do now, not waiting for the new requirements to actually come in. So it's not like the development team is just working on moving the tickets from to due to done, but as well being involved in all the research and making the decisions or contributing as well to the decisions how to be implemented.

Q: So they're as well included in the thought process of the plan as well, right?

A: Yeah, and myself, engineering manager, we are pushing the team for as much communication. Let's remove all the private messages, let's be open, you don't have anything to hide as a team, right? Everyone needs to be aware of what is happening, if there are any struggles. So just by doing that opening, in the recent few months, we saw such a huge change, by pushing everyone into that and the team feels more comfortable to really express their opinion because they see that we listen to them and it actually really means what they see.

Q: Okay so moving forward, how do you manage communication information and flow of these things within your self-organizing team?

A: So, lots of different communication ways. First of all, we use confluence for all of our documentation. So, as I mentioned, we do the research, then we move based on the research results, we decide do we want to continue with the development or we pause or we say that we will not look into that at that moment. So, already within the research stage all the documentation is available for the team members. They are aware of what kind of research is there. Let's say I was running a series of the user interviews where, I was onboarding the people for the research that are not products past users to gather their insights. And as well, I have been inviting the developers to those interviews to participate as the listeners, to be able to hear what the people are saying, what are the requirements, to allow them to ask the questions. And by including them into those, there were lots of positive feedback because they understand actually what the research is, why we are doing it. So, it's Confluence, open communication, Slack channels. We also have dedicated team channel where it is only the closed loop of the team members for any open communication, development, people management, funny jokes, you know, anything like that. Then as well within the product, we have a new practice of squad asked channels. Every squad has its own public ask channel. And it really helps in a way that, for example I know that another squad is responsible for certain functionality or certain service or whatsoever, but I don't find the answer to my question related to it. So I can go to that channel and ask the question and that team will respond to me. I don't need to go to a specific person within that squad. There is a whole squad inside that channel and they can answer those questions. That really helps and as well, being able to see the other questions, responses to that as well brings the knowledge to the whole community, because everyone is open. What else? Comments within the chat, within the tickets. So, as well, before refinement, we ask our team members to prepare for the refinement to review the tickets, etc. To ask open questions prior to refinement. They have certain technical discussion sessions when they discuss the tickets before going to refinement to understand if there are any gaps from the business requirements or from the technical decisions that they still need to make. So, lots of communication actually happen in very different format.

Q: Could you share an instance, where this availability of information and the constant flow of information directly impacted the success of the team?

A: Okay, so let's see. A recent case, this Monday, right? Monday, Tuesday, we were super busy as a team because it was two full days dedicated, nothing else but to OKR's plan. But in the morning, I come to the office and I see that in our business support channel that there is an issue for one of our EU data center client. So the option is to involve the developers in that issue. QA does this very quick investigation of the situation and the issue was found already. Then developers get involved, they found the external dependency from our other product. They contacted that product by saying that they stopped sending us certain events, you need to fix this one because it was not noticed. And they fixed it instantly. So basically by the end of the day, we had the situation solved. So it really helped that all the conversation was happening under the same thread, sales people seeing the same information, support teams seeing the same information, developers, QA, etc. There were no, you know, hidden conversations in private or whatsoever. And basically at the end, we've identified the issue, it was fixed and it could be reported back to the client that they could proceed and they should no longer face it. So it really helps really quickly react to the situation. All the information was in one place and then basically everyone, every person with different functions could take actions on their own based on that information. Basically, whoever would need to be involved into that, everyone stood up and did not miss that conversation. Even the leader of the support team, he was standing by to closely monitor the situation, because this could repeat during the day. So we agreed what are the steps, in case they would get some kind of similar case. Basically, we are not hiding that there is an issue, but being open and helping not just by fixing the issue, but by allowing the support team to save time being aware of the issue or how to overcome that for our clients.

Q: Could you please elaborate on how do you think relationship building within team members contributes to the team's efficiency and their practices of work?

A: A lot, a lot. I would say that, you know, even few months back, we had that culture within the team, within the development team, where everyone was still in those direct messages, working individually, they were not trying to get together as a team and make the decision as a team. But

we onboarded a new manager who really helped to bring the team together, as I just mentioned. By bringing that culture of really opening all of the conversations, by giving the space for technical discussions, for giving the time for the team members to improve. We were really pushing, at the beginning, really pushing the senior developers to share the knowledge with the junior developers. Because, well, there were certain expectations from seniors that you need to be interested and find it yourself, but it's not that easy to find it yourself if you don't know what you need to find, right? So there is still need for guidance and so on. So we were pushing, you know, for certain peer programming, all the discussions, open discussions during the retro meetings. And, yeah, basically, really dramatically, I would say, we changed all the atmosphere within the team. And the team confidence in committing the new OKRs, they were not that confident in the previous commitments, but now, with all those open discussions and knowing all what is behind those OKRs, it really helps them to commit easier.

Q: And how does your team develop and maintain this shared vision and alignment on goals?

A: Okay, I guess it's a bit different with our squad. As I mentioned, within our squad, the goals and the vision is not fast changing, because in order to achieve the goals, it requires quite extensive time period to achieve the goal itself. So yeah, and this quarter, we will have the second top priority project of the product. So you know, it's not that it is changing the priority as it was top priority, it is still top priority unless something really happens. But I would not see that, you know, something would change in this case. So basically our main focus is this top product priority. Do we do all the communication externally from our squad perspective. Do we engage the required team members, etc. in order to allow us to achieve the goal, right? Because in most of the cases, we are very much dependent on the other teams as well to contribute to achieve that goal. Usually we are the leading squad who is delivering it and then the other squads are contributing or removing the blockers for us. So it is very much important for the team to understand the importance or the business value that this feature is bringing to the product, how it actually will change, let's say, the billings or the tryout conversions of the organizations to the paid customers. And, you know, after completing the EU data center project this autumn, I showed certain numbers for the team, comparing US billings and EU billings and they were

excited because we took a very small period and compared the billings for those two data centers. And, well, EU was winning so now they can see how much that contributes and how well it pays off by really delivering those huge projects. And, yeah, so I hope I answered your question.

Q: Yeah and do you have this practice of constantly reminding people this aligned goal and aligned vision throughout the project itself?

A: I would say all of the cases, during the kickoff meet, even during the starting of the research. What we do is we already can see how many clients we have lost because of not having this functionality within the product. Taking the lost costs of that. So they are coming to us with the amount of licenses needed for this period. So we already know how much money we have lost. That is actually as well one of the big inputs, you know, by setting up the priorities for the new functionalities. And the team can actually see that by not having this functionality, we have lost this huge amount of money. For the product, and if we will implement potentially, it will bring even more money for us. So that is one of the triggers. It can be not just money, but it is usually money from our perspective looking into top priority projects.

Q: How would you describe the leadership style in your team, especially in terms of building this autonomous and decentralized team?

A: Well, I would say a really looking into agile, if we look into the scrum team, there are no leaders. All of them are team members. So I am always pushing for all of those open conversations. I am within the team to answer business questions. To bring the requirements, what our clients actually need, what issues do they have, how we can help to solve them, right? Our UX designer is bringing what is the best way to validated some different flows and this flow works best for our clients. The engineering manager is there to answer any technical questions, to remove the technical blockers. The technical team is there to collaborate and ask any questions to give them the opportunity to make those technical decisions. So I would say that I really like that kind of approach that, the leaders are within the squad, we all are leaders and we all are the

team members, and we are all leaders within our, how to say, competence or the knowledge. But when we gather together, we are team members and we are here to communicate.

Q: So your role is sort of to enable everyone within the team to be their own leaders, to provide their own input, their guidance to them, to do their own stuff, right?

A: I would say that is combined responsibility of myself and engineering.

Q: How Agile methodologies have contributed to the development and the implementation of these self-organizing practices in your team?

A: So we really try to follow the scrum methodology. It's still not perfect, not great, and we are still working towards that, but actually we are, I would say we are much, much closer compared to what we were one year ago. Basically I would say following the agile practices, scrum practices, having those really alignments between the team, following the sprint goals. Are we actually focusing our delivery into the sprint goal, right? Are we adding certain tickets if it is needed to the sprint board to actually achieve the goal if something unknown happened? Are we actually communicating the blockers and removing them? So, yeah, basically by doing that.

Q: Could you give me an insight on what adjustments have you made in your case to adopt these principles to fit your team?

A: I would say that it is very hard to say any specifics. I have been working for quite a number of teams as a Scrum Master previously and I was the Product Manager. So, all of this, we start by setting up the Scrum events, starting the sprints, but actually it never works from day one. The team members, if they did not work in that practice previously, they don't understand the purpose. We have lots of different debates during the retro meetings, etc. So, all those debates, all those small steps that we discussed during the retro meeting, what worked, what did not work, how we can improve, dedicating certain time during the following sprint to make the improvements that we discussed. And actually it takes time for the team to really get closer to a self-organizing team. I would say scrum for example will depend on the team members'

willingness to actually achieve that end goal, or finding the motivation for them to really contribute to them, or for them to find the value of, or see the future value of having that. So, yeah, it will never happen from day one, even if all would be motivated, it works differently for everyone. So, yeah, I would not say that, there is some magic guide how to do it. It is just with every team member and with every team, you work in those small steps to achieve it. And it can be absolutely different steps for different teams. I'm always seeing that impact and saying that you need to allow for the team to become constant, so not change, know. don't remove none of the team members, don't add anyone, because if just some small change you kind of go back a bit, and then you start once again.

Q: Maybe there are any other specific challenges that you have seen over the years that reoccur within your teams in regards to implementing these agile principles?

A: Well, I would say that the main blockers might be the habit of how people were previously working. If they were working on one of them, they don't really understand how the agile will work. You know, I have onboarded quite a lot of developers on what we are trying to achieve here. And it was like, I was really seeing some mind blowing reactions, questioning does it work, how come, how we will actually achieve that? And when they really see the benefit, it's really nice. So yeah, working on waterfall previously, it's really hard to switch the mind in the agile way. And I would say that is the main. If you do not have that previous experience, then it's rather waterfall or agile to adopt, it is still something that you will be learning.

Q: What are some of the main challenges that you have encountered when working with self-organizing teams?

A: So that they would not be disconnected too much from where the whole organization is actually coming. Yes, every teams is self-organizing, but they still need to be on the same route as the whole organization. They should not be going the different way. They have to be apart, but not too far apart, just to stay on the same course. A very nice example is those last two days, the OKRs planning that I mentioned, right? Yesterday the day was to communicate and collaborate between the squads to align on all the dependencies, et cetera. It was the first time we did it, but I

really enjoyed. The whole organization was reviewing everyone's key results. Everyone was giving feedback on if it actually is contributing to the organization goals? If not, why not? Why not at the current moment? When can be done in the future? What are the reasons that this is not the contribution at the current moment? Then the dependencies were captured between the teams. Basically, we got teams coming to us because of their dependencies and we were going to another teams for those dependencies. We were agreeing that, okay, we have here the touch points, how we will work to align for those dependencies. So yeah, in an autonomous team you make your own technical decisions, how you progress further, what you actually are developing, but you need to check that if it is really contributing to the whole organization.

Q: Could you share an example of the significant success that you would specifically attribute to the way your team works in this self-organizing way?

A: Oh, do you have another hour? I will talk very quickly about new data centre project implementation. As I mentioned, at the end of September, we have released the EU data centre for product. Actually, this was a huge and challenging project. There were lots of unknowns. Huge research done prior to that. But when coming to the technical decisions, certain blocker after blocker, certain huge dependencies on the infrastructure and on every single team within the product. So what was done right? First of all, communication. Communication to the leadership that all the squads must contribute, but no one is willing to contribute. They don't have the time because they have their own priorities. But the product's priority is not priority for them. Then basically getting the leadership attention, there was a war room established where every single squad was involved to contribute. Our squad, having those open conversations, really deep diving into the technical topics, we were able to achieve the goal that we were looking forward to.

Q: Okay, perfect. That's the end of the questions, thanks a lot!

Respondent no. 5

Q: So to begin with a few background questions, can you shortly describe your role as an IT product development team lead?

A: Yes, currently I'm a manager for probably the main product at the moment for the company. And my role is basically to form a roadmap and backlog for the product development projects, also to work with the all, practically all of the development teams. And by development teams I mean starting from the programmers and ending with marketing specialists to work with them

towards the development of the product and implementation of various features, marketing campaigns, and all of the other stuff that's necessary to make our product successful.

Q: Have you ever experienced a transition from sort of this traditional project management to self-organizing practices?

Yes, because in the previous company I worked mainly as a Team lead and also a project manager not product manager and it was not an IT company. Everything was done in a waterfall principle so to say, because it was project lead company if I can say like that and in this case this is an entirely IT product development company where we work.

Q: And how was your personal transition happening from that approach towards this?

A: It depends because both of those models I'd say have benefits and drawbacks. For example, working in Waterfall principle and in that other company, there was a lot more clarity. What you need to do, when you will do those things, how those things will be done, how much they will cost and so on. Whereas in an agile company, there's a lot of freedom, but with that freedom comes lots of stress and ambiguities, because you don't always know what's best to do and how the things should be done. So for that thing, you need to do some research, for example, implement some continuous discovery methodologies and so on. So in this case, it's a bit more difficult than working in a project led framework. But it's much more interesting since, as I mentioned, you have more freedom, you have more freedom for your imagination and all of the ideas. So yeah, it's a lot more interesting in that way.

Q: Can you describe how your team adapts to changes in the requirements and also in regards to the market conditions as well?

A: Well, since everyone in this company was already working in that agile principle so people know how to work in this environment. Also, our product is the kind of which very much depends on various situations happening on the market or in different countries. For example, when some countries tend to block our product, for example, so we need to switch everything

over, switch our priorities and move on other things. And I think that our team has adapted quite well to this because they are used this kind of work. It was maybe more difficult for me myself because I was used to work in an environment when everything is planned half a year, one year in advance and you will know what you will do in six months and you can't prepare for that. Whereas in this case, sometimes you're not prepared for some issues that tend to happen at all and you need to adapt very fast. That's a challenge sometimes, but for me it's very motivating to learn fast and to adapt fast. If talking about the teams that are working with us, I think that they are all adapted pretty well because as I mentioned, they are pretty used to it.

Q: So from your perspective, how does the way that these teams are set up, helps to deal with these challenges that you mentioned

A: It also has its own benefits and drawbacks because For example, we have different development teams. We have different different web development teams, but different platform development teams since we have our product on 6 or 7 different platforms we have a separate teams of developers who work on them. And they in some cases are pretty autonomous. For example, if Apple team finds some new possibilities which Apple introduced itself and we can adapt to our product. They are trying to do that. So that's a good thing because in some cases I as a product manager don't need to catch up with everything around. The teams just come to me and they say look we have this new stuff that we can implement maybe we can do that, so we agree on that. Then everything moves on, but in some cases this can also be a drawback, because there is an issue when you need to align different teams between each other because if one app decides to implement something, another app may not have this possibility to implement that. And if we have this thing that's not a problem, but if we have like, I don't know 5 to10 such things, then we have to have a little bit of this balance between different platforms and then some additional alignment needs to be done. Which can also sometimes be challenging because for example on one platform we can implement some cool feature but on other platform we don't have a technical possibility to do that and if we implement that feature on one platform but don't implement it on another then we may get negative feedback from our users on why one platform users can use these features and I as a user from another platform cannot use it. So it can be like

that but it's also a question of management. It's just that you have a little bit more work to do on this.

Q: Moving forward, how is communication and the information flow managed within your team?

A: Well, I think it's quite a traditional way with meetings, depending on the platform. And either we also have squads in our company, and depending if it's a squad or if it's a development team or it's a big feature in development which touches various platforms, it all depends. But for all of those things we have meetings, and depending on those different kinds of implementations, we have meetings of different periods. Let's say for one thing we have weekly meetings, for another things we have bi-weekly meetings, monthly meetings. Also we have daily meetings in some cases if we have some big issues, for example, that need to be solved ASAP. Another thing is JIRA, of course, like different project or product management tools, I mean online tools for example. So it's JIRA, we also have confluence where we write all the information and that everyone could reach it and look through. Slack, Google meets, Excel for example, various ones. So in a way it's a setup including meetings that we catch up on stuff and then there is also tools like Confluence that we use to keep basically this knowledge base that has all the information available at any times to anyone needing it.

Q: Can you share an instance where this effective communication that you have, the flows that you have directly impacted the success of the team?

A: Well, I think there's lots of such cases. For example, it also depends how long the project lasts. Because sometimes the project can be short, an initiative can be very small. And Slack is enough. And you have all of the information, all of the chat in Slack, for example, between different developers, or if you invite a new developer who needs to do something, he can quickly read through all of the chat and see what was the issue, what were the solutions that were already tried out, and maybe didn't work out. So yeah, this is one of the cases, like for some small implementation, small initiative, where you have all of the information in Slack, and you can just move from that. But if we have some bigger projects, for example, previously we didn't use

Confluence. And all of the information was, again, in Slack, in Sheets, in Word files, for example, or in JIRA. And the main source was JIRA. So we had all of the PRP documents for one of the biggest of our new features. And all of the information was written there. We had weekly meetings with the development team, and we had separate meetings with marketing teams, for example, with communication teams, with development teams. But if you would look at development teams, during those weekly meetings, we just discuss what was the progress, what are the issues. If we see that some new stuff needs to be implemented, we add that information to JIRA, to the user story, which we tend to write really big ones with all of the details, because it's also, that user story acts as a requirement document as well. So all of the information is put in there, and everyone who joins the project, if we need someone along the way, can jump in, read all of the information, and move on.

Q: In what ways do you think relationship building within the team members impacts the success and efficiency of the team.

A: I think it depends because in the previous company I was working, I had with some companies that we worked together on some projects. There were some people that I had pretty much no relationships and those relationships were entirely work related. But it worked out pretty well because we were only talking about business. Everything was pretty objective in that case and everything went very smoothly. But in this company for example or when you're working toward the product development I think that relationships between people at least like team leads of the developers, managers in other departments, it's very important because as a product manager to achieve the success in the product sometimes you need to push some initiatives that other departments or other people may not like. And in such cases the relationships with people helps a lot because you can then speak about stuff like less officially and explain in more detail in that less official way why we need some things and talk about that specific initiative in a broader sense and so on. So I think that it's very important also when you're developing something and you're especially talking a lot with engineers from what I see it's quite important that people, engineers especially that you're working with would have some trust in you. That they would trust that your decisions are good and of course in some cases they need to raise some questions, but if they do not trust you not only as an expert but as a person as

well it can be very complicated because then developers may have lots of questions why we need to do this, this won't work and sometimes the outcome can be not as good as in the case where you have that relationship.

Q: How does you team build and maintain shared vision and alignment on the goals that your product is set to achieve?

A: Continuous communication and reminding about the vision, about the goals that we need to achieve. Obviously, we didn't have this at all. Well, we had, but not as much as I would like to see, but now the situation is getting better because we have a vision, we have clear goals. Also, we have achieved the thing that some people in some teams, were raising those questions themselves, for example, how does this initiative align with the goals? What OKR will be improved with this one? So, yeah, I think it's just continuous communication and continuously reminding about the stuff that we have because like a year ago, probably majority of our people couldn't tell what was the vision of the company because it wasn't communicated clearly enough, but now I think that the situation has changed a lot and I think that it also helps a lot because especially when some teams that are working independently in some ways, if they do not know the vision, they are most likely developing the wrong things. But when they are aligned with the vision and all of those initiatives and projects are aligned with the vision, it also contributes to the company's success, so I think that it's very important.

Q: And this change that you have mentioned, did it happen because the vision was there and it was then just presented to the team or was it like a common thing to develop?

A: It was actually quite a long process because we were sitting with all of our top management and deciding on what division should be, what goals, not goals, but what the points in the vision should be pointed out, what are the most important things for our product and for our company in general. I think it took quite a while. It took months to come up with one vision that we were sure that we can present to our people and follow it. And after that we were communicating about it in various meetings, the bigger gatherings where we had more people joining. And step by step we reached a better situation for now. It's not perfect yet. We have still a lot of things to

cover, but at least it's a start. Because implementing the vision and that everyone would follow the same vision is quite a long and difficult process. If it's a small company, it's not that difficult because there are smaller amount of people that you need to envision and enlighten about the vision. But when the company is pretty big, it becomes a bit more complicated to align everyone.

Q: Could share an example where having this alignment on the vision helped the team to overcome a challenge or reach a milestone?

A: Well, I actually have done mistakes on this one as well, myself, because for example, we were developing one new feature, and I've written all of the points, like all of the requirements that needs to be done, and the developers started to challenge me, they said, but hey, these things do not align with our vision. And I, myself, who was like evangelist on this vision thing, caught myself that I was wrong in that case. So we adjusted the requirements a little bit so that it would align with the vision that we have. So that's also one of the cases.

Q: Moving forward, how would you describe your leadership style within your team, especially in terms of having this autonomous approach and sort of less centralized decision-making?

A: At least I personally, I don't know, you should probably ask other people who I'm working with myself, but I would say that I'm very diplomatic. I tend to avoid conflicts. I mean, I do not like avoid conflicts, but I tend to avoid to make people angry or make people sad in some cases. And I always try to be diplomatic to listen to all of the opinions, look through all of the points, evaluate all the data that we have. Like, not only the data, not as data in numbers, but the data that we have from different peoples. I also try to be a good listener, because I think that product management is very important. I don't know if you can call it skill, but trait maybe. Yeah, and also probably a problem solver, because if I see a problem, it starts to bug me and I just need to solve it and have the things done is also another thing. And it it also idepends on the problem. Because there are sometimes problems that I cannot solve myself, especially if those are like technical problems in the code, for example. All I can do is to push some people, talk with some people and look where the issue is and try to talk with them so that people would start doing their

job. But if there are like some problems that I can solve with my own hands, first of all, I try to dedicate the work to the people who should do that work. But if there's no one who I could dedicate it to or the problem is too complex for some people to solve it, I try to do it myself. But I'm not that kind of person that would, I don't know, try to do everything by myself because I would think that I am the one who would solve this the best. No, I'm the one that would more try to dedicate the work to other people.

Q: Okay, and how do you balance between this providing this support and guidance to them with delegating tasks and giving the autonomy to make decisions?

A: Well, as I mentioned previously, lots of the times I try to listen, especially with more complicated technical stuff, I try to ask and listen for the opinions of our senior technical people and then try to reach the consensus between different technical teams who are working on the same project or same initiative at that time. And if you listen to other people, because in lots of the cases which are related to technical stuff, those senior people know the solution a lot better than me myself. So when you get those opinions and all of the people are gathered at the same table and they discuss each other, all of the options out, it's sometimes very easy to make a decision because it's pretty clear what needs to be done. But when the problems are like more abstract, then I just tend to gather as much data as possible so that when we are talking with the other teams or other managers, other leaders, the heads or so on, I just try to prove my point with data. And according to that specific data, I just tend to suggest some solutions for the problem and then we also discuss what that specific solution would work best or not. And if no one has an opinion, well, then I have.

Q: Have Agile methodologies contributed to the development of the way our team works in a self-organizing environment?

A: I don't know, it probably depends on what do you mean by agile, because agile as a framework, as a word, everyone uses agile, but this agile is different in every different company. So it's hard to say, because if you're meaning by agile that we can quickly adapt, yes, we have teams that can work on separate things on different set of things at the same moment. We can

also quickly adapt to some changes. We can deliver quickly if we need something. If you mean like agile, some ceremonies like retrospectives, sprints, daily stand-ups, we also have those, but we don't have all of them, for example, that we would have if we would work in Scrum, because we don't use Scrum, we are simply agile. So yeah, we have those. We have daily stand-ups. We have, as I said, some sprints, different teams in our company work in different sprints duration. One team is working two week sprints, other teams work in three week sprints, because it's more comfortable for them. We also have retrospectives, retrospectives for different stuff as well. We have weekly retrospectives, for example, for app development. We have project retrospectives when we finish some project. We have also post-mortem process when we have some big disaster and we need to look at what was done, why this thing happened and how we could improve so that it wouldn't happen again. We also have grooming sessions, but not all of the teams have this one, because some teams find it working good, other teams don't find it working at all. So yeah, we have different agile principles in different teams. But all in all, I think that not all of those ceremonies, as they're called, not all of those are needed. Well, not like not all needed, but maybe not all of them are crucial. And some of those totally make sense, for example, to have daily stand-ups where the teams could gather and discuss what we've done yesterday, what we'll do today, or tomorrow, or etc. So these are a must. But if it wouldn't be in agile, I think that in either way the teams would have those things, because those are like common stuff, common processes that are used in other companies, not IT companies as well, because when I worked in another companies, which didn't have this agile principles, we still had meetings because we need to communicate. Communication is the main problem in all of the companies, at least from my experience and from other people's experience that I hear. So yeah.

Q: And have you done any customization of these practices in your team?

A: Not for agile stuff. Because we are quite a young company still, and in some places we lack of processes in general, but these are not like agile processes. General process, for example, new IP address introduction to production environment. We didn't have some processes how to do that. Well, we had a process, but it wasn't defined clearly. And if we had some, I don't know, a bit different situation than the chaos came up. This is just a, for example. So yeah, we're also working towards implementing the different processes now, but as I mentioned, these are not like

agile processes. As from my side, not so long ago, we introduced a squad, which I am leading now, and probably all of the agile things that we had before, they remain the same. We just, the only thing that we don't have is daily stand-ups because we have weekly meetings and it's enough. If we have, I don't know, maybe more questions, we can always check and slack or we can have ad hoc meetings for that. But we just decided together with the team that it's not necessary to have daily stand-ups because that's what has like bigger initiatives that they're working on, not like small tasks that they can close every day. Yeah, and it's not needed to have daily meetings for that.

Q: And can you share any kind of a specific example where a significant success was achieved, mainly due to the fact that the teams were self-organizing and were able to deliver things while working autonomously?

A: Probably one of those could be one of the recent ones. Currently we are working to improve the product quality. And a lot of the product quality depends on the bugs that we have. And we have decided with other leads that it's best to leave those specific bugs on each of the specific platforms to deal for those teams individually. The teams I mean developer team, because those developers know best what those bugs are, how they need to be solved. So we left it to deal for them and they actually dealt with it quite successfully. Of course there were some bugs that needed some more detailed or more specific solutions that are related to the product. But more in general they came to me or came to my other colleagues working with the product and asked for the help, let's say. But all of the implementation was mainly done by themselves. Of course this is like more technical thing and it can be shown for them. But yeah I think it can be as an example where the teams worked autonomously as they reached, not maybe a major or like revolutionary success but it greatly greatly adds to the success of the team.

Q: And has there ever been an opposite situation where having these self-organizing teams, posed a challenge where some sort of an issue due to the way the teams are organized?

A: So at least in our company, when different teams start to develop different things, and if you do not jump in at the right moment, those different things can come to production. And then you

have to go back, and then you have to look for what things were implemented, then communicate with those different teams again, so that everything would be aligned. Previously it happened probably a lot more often, but now since we have more people that can overlook those things and ensure that this wouldn't happen, this happens a lot less. But yeah, it was a problem, and it was a challenge, because in this way you tend to waste a lot of time because of that misalignment, because teams have to go back, they need to fix the things, well not fix the things, because maybe the release was good, but it wasn't aligned with other platforms which have the same product. So yeah, in this case we can save more time, more development time, and then use it on other projects.

Q: Okay, that's all of the questions and thanks a lot for the answers!

Respondent no. 6

Q: To begin with, can you shortly describe your role as an IT product development projects lead?

A: Okay, so I'm not sure if the lead here is a correct name, but my official title is Product Owner. It evolved during the time since I started working, but currently I'm focusing most on product itself and its features, on the discovery phase and everything. What relates to development, implementation of the features, delivering actual features and managing the people who delivers these features. So yeah, I'm closely collaborating with the engineering and design team and what

relates to development. I tend not to get involved a lot, but sometimes I am needed there. Yeah, maybe that would be it.

Q: Have you ever experienced that transition from traditional project or product management towards self organizing practices?

A: Well, previously I was working as a project manager in advertising agency. So there probably all the projects were more or less waterfall type of projects. But since I joined the organization, everything here was based on agile, the transition, I wouldn't say that I experienced some transition because it was quite organic change. And yeah, it seems that it's more useful and beneficial to have this type of project management when you are working with products.

Q: And how was that transition for you?

A: As I mentioned, I didn't experience any issues, any struggles because it seemed very organic. It should be this way and it would be maybe harder or more struggle if I had to switch back to the waterfall.

Q: Can you describe how your team adapts to changes in project requirements and also in market conditions?

A: Of course nobody likes it, but when you have a good atmosphere in the team and everybody understands their role and then responsibilities and everybody has same goal, so it doesn't matter if the requirements change or not because it probably changes not because someone wanted to change, but because it is required to be changed. Unless it is obvious mistakes by product owner or another person and it repeats. So then probably people can get frustrated but it doesn't happen for me a lot. But on the other hand, when I see that some requirements from business owners, from marketing owners changes a lot or they do not foresee something, then I usually get frustrated and I don't let my team suffer from it. But yeah, this is the environment we work in and this is what we need to deal with and probably what we can do is just to communicate, talk with each other. It doesn't necessarily need to be pleasant chats, but it has to happen.

Q: And what role does self-organization play in enabling that adaptability and responsiveness that you've just mentioned?

A: So because we are working in iterations, basically this is what enables these changes to be made before actually delivering the feature. If we wouldn't have this way, probably then it would cost more money, more time and less value for the business and users. So yeah, I think that it enables quite a lot.

Q: Can you share a specific instance where it helped to be very responsive to a challenge that emerged?

A: It happens all the time. Maybe in some moments bigger changes and some smaller, but I don't know if I can mention specific features we are working on. But basically it happens. For example, we are introducing new products in our purchase or customer acquisition channel and this new product has to be supported along the whole user journey, starting from the browsing or purchase flow, then onboarding, activation and so on. Especially this product includes a third party. Like it's not our product yet, but we are bundling with third party. This third party company, usually we can't manage them, we can't prevent anything, what they can change. And yeah, it happens in the way. For example, it can be like a small change, like a product naming change, or it can be a whole product activation employee login. So yeah, it happens.

Q: How is information flow and communication flow managed within your team?

A: It depends on how it is managed in the upper level or in the business level. But how I like it to be. We use slack for our daily communications and for quick status updates. We use Jira for the whole product and feature implementation and planning and delivering and QA. So we have a process in Jira. We use confluence for documentation. So a lot of asynchronous communication is happening in confluence in Jira. And yeah, we have daily standups for reporting, for just sharing what we are working on and if we have any blockers. What else? We do not use email, but a lot of people use email as a notifications tool and that would probably be it. What I always

encourage my team to do is to use public channels instead of direct messages. Avoid direct chats one on one, so that information is public and available for everyone. Just to avoid any miscommunication.

Q: Have there ever been cases where this set up where having information in private messages posted some kind of a challenge, where some information was not available and it caused an issue?

A: I don't recall any major issue, but there is usual struggle, especially if we have a new feature which we need to communicate with new people, which they don't know how we prefer to work and to communicate. They then write in private messages and start resending messages. We have to start resending messages all the time and it just prolongs all the process. But yeah, I don't recall any major issues, just that if I notice it, that this is happening, I am stopping it and we are going into the channels.

Q: In what ways do you think relationship-building within the team members contributes to the team's efficiency?

A: Yeah, of course. In what way? So I see two parts here. One is the official relationships and communication. So this is this process, how we deliver. Especially if you have a good product, a lot of people in the team, this process has to be very smooth and improved constantly. So, yeah, if you have process, it's your official process. Relationships are well built, then your delivery will be effective and you will probably notice any blockers or issues happening sooner, like earlier in the stage. And of course this unofficial relationships, like having team buildings, similar hobbies or interests is also contributing a lot because it's just a better atmosphere in the team, less stress and yeah, I think it helps a lot. Sometimes if you have a difficult moment, something is not happening as we want to be. And the team is built in a good way that they do not stress and they even find the way how they can make a laugh out of that, it helps a lot.

Q: And how does your team develop and maintain a shared vision and alignment on the goals?

A: So the vision is most probably developed and it evolves a bit but since we are the platform for existing users where they can manage their account, their subscription services, the vision is quite clear for all. When a new team member is joining, during the interviews, during the onboarding process, we introduce them with what they are going to work with, what are our main KPIs and goals, what are our main stakeholders and how we contribute to the company, to the business, what value do we bring. So this is how everybody gets introduced with it and how do we maintain? Yeah, everything we build, we try to measure, to show and we do not hide anything. We transparently share all the numbers with every team member. And yeah, we constantly talk about the same KPIs or our goals. So the vision is quite clear that in our case, it's basically happy and loyal existing customers. In a way it's a constant process of always reminding them through business goals and KPI's.

Q: What role does having a shared vision play in enabling your team to self-organize effectively?

A: So it has a very important role, not only because of the reminding constantly and sharing, but probably mostly because of these team members having the context to know what value will they bring to the users, to the company, so that they can feel maybe proud of it, so they understand that they did something, not just a piece of code, but actually they delivered the value to someone. So yeah, it has a huge role.

Q: How would you describe your leadership style in the team, especially in terms of the autonomy that you bring?

A: Good question. Maybe someone thinks differently, but I think I'm more of an influencer. I let people do a lot of by their own. I let people do mistakes. As I mentioned before, I tend not to get involved a lot in that development cycle. Also, my team is mainly developers, but yeah, so I like giving a lot of freedom. But when it comes to discussing the problems, the issues of having retros, I'm not staying silent there. And if I see that something is not okay, I raise it. Not everybody likes that. Maybe sometimes I'm too open, but I think that this is the correct way.

Instead of just going every day after each of them and checking what do they do and giving feedback. I tend to give feedback afterwards because they are the specialists. I don't have technical background, I don't know how to develop. But yeah, if I see that something is struggling, I'm raising it. And sometimes it's not pleasant.

Q: And how do you balance this guidance and the autonomy that you give to the team?

A: Yeah, this is quite easy. We have our quarterly goals that we define before each quarter. We also have regular checkups on how we progress towards these quarterly goals. Everybody seems to be on the same page. Of course there are times when we are behind our plan and it's very common practice, I guess, but usually it's happening not because of our fault. Maybe because we have some dependencies from other teams, but yeah, I don't know this OKR's framework and constantly checking that the progress helps.

Q: So you have the set goals and you let people do their work and then you're interrupting if something is not going right.

A: Yeah. So during the check we see that we are behind, so we need to discuss what happened. And it does not necessarily mean that some developer is not working effectively, but maybe some blockers appeared. But that means that we have made the mistake during the planning why we didn't foresee these blockers.

Q: Have Agile methodologies contributed to the development and implementation of self-organizing practices in your team?

A: Yeah, I think when I started I had a different team, but we had one way, we were working in a hybrid kanban way. Then we tried to switch to the strict scrum, but it just brought a lot of confusion. And then we saw that there is a lot of bureaucracy in it and now we are just working in a kanban way and we like it. So everybody seems to be happy. Everybody knows what needs to be done, what are the highest priorities. And yeah, we are managing this roadmap in different ways. We have different tools. We don't need sprints. I guess.

Q: What have been the challenges that you have faced when you were integrating Agile methodologies?

A: I wouldn't say that it didn't work, but it had some, let's say moments which you need to do, like, let's call it bureaucratic moments. You need to close the sprint, review what was delivered and what we saw that we usually don't deliver half of what we put into the sprint. Then we tried to put less into the sprint, but again we realized that then at the second half of the sprint a lot of people are just sitting without work or taking the tasks from their future sprints. So what is the value of that sprint? So maybe this was not the main challenge, but the confusion why we need this print. And another what it was scrum. So yeah, I don't recall any other issues with scrum.

Q: Can you share an example of a significant success that you attribute to self-organizing practices?

A: Yeah. So the success is about the current setup, I like how everybody feels ownership of what they are working on. We have splitted the team into two squads. And these two squads know the domain, know their scope mainly. And they have quite good, quite well defined KPIs what we are measuring in terms of product performance. So I like this. I like the work collaboration way between PO and the engineering manager. So yeah, he takes more of this development cycle and people management and I can focus more on product itself. And I don't know if I have a vision how need to change something but. Maybe what we usually struggle with is the planning phase of a new feature. Yeah, I would like to improve that part. Let's say we have our OKR's defined and we know what we need to deliver. So how we are planning, how we are basically putting into roadmap what needs to actually be done in the epic. I would like to improve this, but it's the setup is okay for this. We need to improve it by ourselves.

Q: That's all the questions, thanks a lot!

Respondent no. 7

Q: To begin with, can you shortly describe your role as an IT product development project lead?

A: Right, so overall I'm responsible for one of our products in the group. In a broader sense, so not just product development but organizational development as well. I'm responsible for the bottom line metrics including PNL for this particular product. In terms of day-to-day operations, I would say that one of the of my priorities from product side is still helping set the direction and the vision for what we want to accomplish. But other than that, when it comes to sort of

day-to-day decision-making on how products should be developed, I think more of an advisory role.

Q: Have you ever experienced a transition from traditional project or product management towards self-organizing practice?

A: Well, what I'm thinking is that when we started in the product, I think the sort of first iteration, it's kind of waterfall-ish because while you still have a lot of unknowns and you cannot sort of make plans or predictions for like two years in advance or even like six months, but because you have very little formation, especially if you're going into sort of unknown market, there are very little competition, so you can't just look over the shoulder and see what everyone else is doing. You kind of need to make those assumptions and make those decisions. Often, some of the decisions, for example, like how do we want users to onboard and the they pay or do we want them to pay and then onboard, like it's a guesstimate because you still don't have the information, so you can just base on your experience and gut feeling and other best practices and so on and so forth, but at the end of the day, you still need to kind of make those decisions. You may need to sort of define the core concept, core product that you can go to production to, which I would say is kind of waterfall-ish and after that, when you start getting users, you start getting feedback, you go into more of the agile development where you're actually getting signals from the market and based on that, you try to respond and react in terms of priorities and what you're trying to accomplish.

Q: Can you describe how your team adapts the changes in requirements and market conditions?

A: I think a few elements are important here. And in terms of adaptability, the first step is mindset of the people. So this is something that we look for when hiring. Because we have been building a new problem from scratch. You know for a fact that a lot of the assumptions and things, they will be wrong. And some of the things that we will develop, which we think is the best thing ever, will actually be useless for the user. Or it will work completely differently than we assumed and we will need to redo it. So for anyone with a very sort of strict mindset, it would

be stressful and painful. So it starts with hiring. So you're looking for adaptability as a mindset. And the other thing in terms of, I would say, cultural aspects is fostering open discussions. I think this is very important as well because at no point during the development or planning, there should be an argument that, hey, we always used to do that, or hey, we agreed to do that, and that's why we will do it. Like if we have new information that shows that a particular approach is no longer the most optimal one, or no longer makes sense, then it means that we need to adjust accordingly. So one of the aspects here is to foster the sort of culture that people can raise objections and say, hey, I don't understand, doesn't make sense for me, why are we doing this?

Q: What role do you think self-organization within the team plays in enabling that adaptability and responsiveness?

A: Absolutely a very important one. Otherwise, you will have this argument that we're doing this not because we think this is the right thing to do, but we're doing this because someone up the chain said that we need to do it like that. So, again, this is not a valid argument, and it doesn't matter if it's me, our CEO, or whatever. We shouldn't be doing something just because someone higher up said it, unless it actually makes sense. And sometimes it does.

Q: Could you specify in what ways the self-organizing practices improved the responsiveness of the team?

A: Hard to say for sure in the sense that since we've been doing it from the start, we can't always do it like that. So we don't really have a sort of before-after comparison to say. So most of the issues that we have solved we done in a self-organizing way. And at this point, for example, for me, where I'm seeing this a lot is handling ad hoc customer support cases or, for example, problematic cases with data brokers. So early on, I had to be more involved in those aspects and help sort of align with how we're the communication and how the decision-making should be happening. But I don't think I've been very much involved with the last, I don't know, six to nine months. And it's been essentially just happening and getting solved it. People responsible for now have my trust to do it and that's it.

Q: How are communications and information flow managed within your self-organizing team?

A: I think it's a constantly evolving challenge. Like I don't think it will ever be like, okay, we're here, we have the perfect communication, it's simply because we started the organization a few years back, and we started this year we had like what 16, 17 people in our team. Right now we're probably close to 43, 44, something like that. So just amount of people involved raises the complexity of communication. That said, I think a few important elements are here. So first and foremost, so different people need to know like what they are responsible for, within the team and within the organization. So we need to be clear on like how we measure success and like what are the priorities. So having this baseline already works as sort of north star, if in doubt, like this is what we need to look at. Then specifically when it comes to communication, so a few things that we do, I would say it's like three layers of communication. So we have one-to-ones with people, so this is to tackle specific cases when something isn't clear. So the whole idea is to identify potential bottlenecks for specific people or specific teams, specific cases and to actually dive into them and see like what we can do better, what we can solve. Then we have a team level. So we have team level check-ups on a weekly basis mostly. So different teams and again, like within the teams, they can organize their own process. So for example, like some engineering teams, they have daily stand-ups, some do it by daily, some do it weekly and so on and so forth. So like within the teams, they can organize whatever makes sense for them, but on a whole level, the team level sort of checkups, again to see how we progress through our OKR's. That's another aspect. And then we have the weekly organization level catchups. So this is again to bring everyone just a bit more context of what's going on in different parts of the organization because like if you're working on product, you might not necessarily know what's going on, marketing side, vice versa. And then nobody knows what's going on on data side and et cetera, et cetera. So that element and I think another thing that's proven to be quite a bit of success is a lot of written down communication that we use in Confluence right now. And I think this is a practice that has been successful in sort of making transparent and liberally use of information that's where everyone can access what's important. And not to forget, we use Slack for daily chats.

Q: And how is that managed? Is it like more of an open channels side of idea, or is it like more for direct private conversations?

A: So ideal scenario would be to mostly use channels and we do have channels for different topics. So again, this makes sure that people are aware like what's going on and sometimes they just need an FYI. And you don't need to include them into like a direct message. But in reality is that still quite a bit of communication happens in direct messaging. So it's not ideal but the reality is that a lot of people just prefer the sort of personal private communication like in Slack. They feel more comfortable to discuss things there.

Q: Can you share an instance where effective communication and the flow or availability of information directly impacted the team's success?

A: I think one example comes to mind is when one of the data brokers had a breach sharing user data, other user data with us. So I think that the whole sort of communication there was quite aligned because we had a case in a channel, in a thread, and we had a discussion with people that needs to be involved. So we have legal, we have PR, we have ourselves. And the sort of talking points were discussed because everyone's clear on their own sort of ownership and responsibilities. So you came in with the prepared message notes from legal and so on approvals, approvals, approvals. And everything's in the line like kind of streamlined. So this is a sort of potential crisis, mini crisis level situation that actually was a result I would say very, very smoothly.

Q: And in what ways do you think relationship building within team members contributes to the team's efficiency and the way they work?

A: I think this part is critical. And it's actually over the last couple of years, one of the reasons why I kind of changed my mind that the sort of benefits of hybrid work versus remote work, that actually like meeting a person, spending time with a person, it's easier to build a relationship. And what I've been noticing is that the teams and the individuals that build a personal relationship, they tend to collaborate and perform better. And this is like, the benefit is that once

you build a relationship, you can be more transparent, and honest, and quick with your communication, you don't need to think about like, okay, so how will this person interpret my message and so on? So you don't need to think, oh, if I make a joke here, like will someone get offended, et cetera, et cetera. So this kind of trust, I think is very important just to open up the communication. And another important aspect is that what we spoke about before. So like questioning, sharing feedback and so on, it's a lot more difficult when you don't have this sort of personal relationship built with the people because they might take it as a sort of challenge or like being called out and so on. And once you have this relationship, we can actually discuss on the ideas and on the topics and not take it personally.

Q: How does your team develop and maintain a shared vision and alignment on goals that the product is heading towards?

A: So I think OKR's framework does a lot of the things here. It is that we kind of decide what's most important for us about the next year so it's usually being aligned with financial goals and based on that we then kind of break it down through impact mapping and what needs to happen for us to make those financial goals possible. Yeah I think that that does a lot of heavy lifting and then the transparency communication. So again transparency, the weekly catch ups where we kind of touch base every week on how we progress. Just kind of keeps everyone in the loop.

Q: And is the vision developed mutually with the team or is it something that's brought down from the top?

A: It's a combination of top-down and bottom-up. So when it comes to the ultimate financial targets, it's more of a top-down. Though, again, now that we're doing more proper financial forecasting, there's a lot of bottom-up input there as well. But still, once we have the top-down ultimate goal, we then work bottom-up to figure out the goal.

Q: What role does shared vision play in enabling your team to self-organize effectively?

A: Again, it's a sort of shortcut in decision making, so it saves time. You don't need to cross-check, double-check before making decisions. You know that if this is aligned, then this is where we're going, and even if it's wrong, it probably won't be very much off.

Q: And is there an example where this alignment on goals helped the team to overcome a challenge or achieve a significant milestone?

A: Nothing specific comes to mind, I think in general, everything we do kind of boils down to it.

Q: How would you describe your leadership style within the team, especially in terms of facilitating the autonomy and decentralized decision making?

A: Not sure if there's a name for a style. So it is naturally evolving. And I have taken a lot of the different elements from my own managers in the past and right now. So in general, I think when it comes to the leadership style, so I try to be clear what's most important. So clear priorities. Then try to foster ownership. So again, I think trust is important here. So if you want people to take ownership, the stakes need to be high. So they need to accept that there's the positive. Like the wins and the losses are theirs, but mine as well. So at the end of the day, I'm kind of responsible for everything. But at the same time, I think it's important to have trust in your team that they can own like an individual aspect. They can make better decisions. So from my side, I think a lot more aspect goes into sort of, not necessarily mentorship, but trying to help guide the team based on what their strengths and the sort of areas to improve.

Q: And how do you balance that guidance and support that you provide to the team with the full autonomy and full power to make their own decisions?

A: I don't think that there's a lot to balance here. Like I think that kind of goes hand in hand. I think essentially the learning opportunities come when something doesn't go well. If everything goes well, there's not a lot of learning opportunities there. So it's just all goes well. There's also not a lot of feedback in that particular case. But most of the learning actually comes from more difficult situations. So if there's a sort of a difficult case that we need to solve. So first and

foremost, a particular person makes the decision. So if in doubt, I can advise and help, but I generally try to refrain from making the final decision. Unless there are cases where there's a sort of disagreement and it starts taking a lot of discussion, just getting back and forth. It just makes sense to intervene and cut it. But this rarely happens. So yeah, I think it's mostly just when something doesn't go as good as it's could, then you can come. I have one friend that has been like a manager at different scales for like 15 years now and he has this sort of saying. That "you sometimes see that a person in your team is walking on a rake and he'll get hit in the face. Should you still let him do it? Probably yes." Like, if it's not business critical, it's usually good investment to give the opportunity for people to make the mistakes, especially in the lower stake environments. So they can actually learn first hand and help guide along the way before the stakes get very serious and business critical. Or if it's something that can be business critical, you probably don't want just to stay aside and look from the side and say that "this is going to be hilarious, let's see how it goes." But yeah, I think often, smaller situations, the lessons are more worthy than trying to prevent upfront.

Q: What role does leadership play in fostering and maintaining such a self-organizing environment?

A: I'm kind of a believer in sort of lead by example and I think like for a self organizing team, like for it to make sense, a few things need to be true. So again, trust is very important here and it's trust that goes from different direction because first you as a manager and your team need to trust their peers, their managers, their support, whatever, everyone in the organization. Like there needs to be trust that people will get, like that they are professionals in their field and they will do a good job. Sometimes there will be mistakes and so on so that's fine. But again, everyone needs trust. Within the team that everyone knows what they're doing more or less. And for this to be true, you as a leader, first and foremost, need to show trust to your team. Because if you're not trusting your team, that they will have trouble trusting others and so on, so forth. So another thing as a leader, you also have to maintain standards. Because one of the issues that can happen and has happened is when you have someone not pulling their weight. Essentially when the pairs are starting losing trust in a colleague as a professional, then you need to, as a leader, to investigate, is the person actually underperforming? And if not, then what's the reason why

there's a sort of misunderstanding? And oftentimes, yeah, it is the case that someone is underperforming. And then you need to intervene and either help get the person up to the level that you need to perform or you need to part ways. So self-organizing teams cannot work if there's multiple weak links at different stages. Because there's not going to be trust. Related to that, again, when there is relationship and there is trust between people, people feel a lot safer to make mistakes as well. And challenge and be challenged. So again, this is something that is important to build. And as a leader, you essentially kind of need to embody and practice what you preach here, because everyone else will be looking at you and taking kind of unconsciously, sometimes subconsciously, if in doubt, they will do what they see you doing.

Q: Have agile methodologies contributed to the development and implementation of self-organizing practices in your team?

A: Kind of sort of. We don't use very textbook level agile development. We don't use it in our product. We don't use it at the company as well. Like, we kind of take the ideas and adapt them to what makes sense to us.

Q: And are there any specific agile practices that you have found effective, that you have adapted, that support self-organization in your team?

A: I think just in general, rapid planning and short-term planning, that makes sense for us. Limiting scope in development, so building features, you want to start with smaller scale testable solutions versus just building the end result up front. Yeah, I think that for me, those are the most important elements. I'm sure that people working more on the day-to-day operations will say that maybe retros tend to be valuable and so on, but for me it's just sort of regular check-up's and clear understanding what we want to accomplish in a short amount of time and limiting scope and adjusting scope so we can ship sooner rather than later.

Q: And what were the challenges that you have faced in terms of introducing agile practices?

A: I think the main challenge is that different people have worked in different organizations, they have different experiences, different biases. So someone that had good experience with a agile, they might think that that's how we want to work and so on. Those that had bad experience with that, will say that this is the worst thing ever, let's not overthink, why do we need so many meetings and blah blah blah. So the thing is that when you're kind of pushing for the self-organizing environment, you need to be careful about top-down, strict way of saying that we all work like that. So it should come as a sort of natural growth or evolution process wise when we work through the problems which we want to accomplish and which are like the challenges that you have in development and comit, so maybe this is the answer for us.

Q: And what are some of the challenges that you have faced in general while working with the way that your team is set up in a self-organizing way?

A: I would say that getting people aligned, this is something that requires a lot of time, a lot of effort, especially people from different mindsets, different backgrounds, because again, when building a team, you kind of want people from different attitudes. You don't want everyone like super single-minded because that's how you get best decisions. But at the same time, when you have different people, there's like built-in conflict and essentially coaching people how to handle those conflict situations. You get to the better answer versus getting offended or whatever. Like this takes quite a bit of effort.

Q: Do you attribute the success that you have managed to reach so far with your product to the way that the team is working in a self-organizing way?

A: Probably, hard to say for sure, like we don't have an AB test running to say. I think there's many ways to succeed and it's probably depends on the leader and organization, like what kind of organization you want to build and that kind of organization that needs to be aligned with the leadership style. And based on that you will hire the teams and build the process and so on. So for me, I probably couldn't imagine building it in a different way, but just because I personally have a sort of disdain for micromanagement and sort of dictatorship style approach. So hard to say for sure, I think there's like a million ways to succeed and there's organizations with very

different styles that accomplish as much and even more. So I think it's just a matter of personal preference.

Q: Okay, that's all of the questions. Thanks a lot.