

**Overcoming Challenges in Software Development Projects: An Integrated Approach  
for Small and Medium Enterprises (SMEs)**

Robin A. Lagerburg

Master Business Administration

Behavioural Management and Social Sciences, University of Twente

194100040 - Master Thesis BA

Dr. Ir. A.A.M. Spil

Dr. M.R. Strienstra

August 5, 2023

## Abstract

The study at hand zeroes in on the distinctive challenges that small and medium-sized enterprises (SMEs) confront during their software development processes, with particular attention on process management and knowledge transfer. Among the numerous obstacles are limited resources, frequent staff changes, an absence of established processes, difficulties in adapting traditional methodologies, and complications related to knowledge management, including the transfer and retention of technical knowledge.

To fully explore the nuances of process management and knowledge management within the SME context, the selected research method is qualitative. Semi-structured interviews became the preferred mode of data collection, and the data analysis relied on the Gioia technique. This methodological structure underscores the significance of understanding specificities in process methodologies and knowledge management for SMEs, aiming for a balanced view that can be generalized across different business contexts.

The study's findings propose a comprehensive action plan tailored to SMEs, addressing not just operational barriers but providing in-depth solutions for process methodologies, such as the potential adoption or blending of Agile and Lean strategies, and innovative knowledge management techniques that focus on continuous learning and technology-supported collaboration. The research highlights the need for change in how process management is approached and how knowledge is shared and retained within SMEs.

These tailored solutions can pave the way for transforming impediments into growth opportunities, leading to an increase in the efficiency and effectiveness of software development processes in SMEs. The insights have the potential to result in higher quality end products and an improved market position. The findings also serve as a foundation for future research and can be immensely useful for guiding software development practices within SMEs.

**Keywords:** Challenges in Software Development Projects, Small and Medium-sized Enterprises (SMEs), Software Process Improvement, Resource Constraints, Agile and Lean Practices.

## Table of Contents

1.	Introduction-----	6
2.	Literature Method -----	8
3.	Theoretical Framework -----	10
3.1.	Literature Analysis -----	10
3.2.	Practical Implications of Strengths and Challenges-----	16
3.3.	SMEs' Unique Market Position-----	17
4.	Methodology -----	19
4.1.	Research design -----	19
4.2.	Research Method -----	20
4.3.	Sampling Procedure-----	21
4.4.	Interview Content -----	22
4.5.	Data Analysis Technique-----	23
4.6.	Ethical Considerations -----	25
5.	Findings and Results-----	26
5.1.	Overview of Study Population-----	26
5.2.	Data Analysis Procedure -----	27
5.3.	Development of Data Structure -----	28
5.4.	Overview of Main Concepts -----	30
5.5.	Explanation of Data Structure -----	32
5.6.	Analysis and Validation of Results-----	33
5.7.	New Gradation of Challenges -----	35

5.8. Interrelationship of Challenges .....	38
6. Discussion .....	40
6.1. Summary of Results .....	40
6.2. Interpretations .....	40
6.3. Practical Implications and Recommendations .....	42
7. Conclusion .....	46
8. Limitations and Recommendations .....	47
8.1. Limitations .....	47
8.2. Recommendations for Future Research .....	48
9. Bibliography .....	50
Appendix A: Questionnaire I .....	57
Appendix B: Questionnaire II .....	60
Appendix C: Exemplary Citations from Interviews .....	61

## 1. Introduction

Scholarly work on software development predominantly focuses on large-scale projects within expansive organizations, resulting in methods and techniques designed for the capacities of larger entities (Sambamurthy et al., 2003). Nevertheless, Small to Medium Enterprises (SMEs) represent a major part of the economy in numerous countries, providing a significant source of employment (Muller et al., 2021).

SMEs, distinct from larger organizations, often grapple with resource limitations, a dearth of specialized expertise, and diverse operational and managerial challenges. These constraints can hamper their ability to deliver software projects timely and within budget, potentially leading to business failures (al-Tarawneh et al., 2011).

While considerable research exists on agile and lean methodologies (Alahyari et al., 2019; Dybå & Dingsøyr, 2008), the focus remains on larger organizations and the associated challenges in implementing these methodologies. Less research concentrates on the difficulties SMEs encounter when adopting such methodologies.

Previous work emphasizes the economic and employment significance of SMEs, particularly in the software development sector (Brodny & Tutak, 2022). However, they highlight SME-specific challenges in assimilating usability engineering and agile methodologies into software development processes (Hajjdiab & Al Shaima Taleb, 2011; Hering et al., 2015).

Other studies accentuate the alignment difficulties between requirements and verification in software development processes and the need for measurement and improvement of agile processes in SMEs (Bjarnason et al., 2014; Choras et al., 2020). However, these research endeavors primarily concentrate on specific software development aspects, lacking a comprehensive understanding of SME software development challenges and potential solutions.

Given the evident research gap, this thesis investigates the overall software development challenges within SMEs in-depth. The main goal of this study is to conduct an

in-depth study of the barriers and challenges faced by small and medium-sized enterprises (SMEs) during their software development processes. This research draws on existing academic sources and individual interviews with SME stakeholders. The findings of this two-staged approach are then used to create a coherent action plan tailored to the specificities and challenges of SMEs. The ultimate goal is to provide a comprehensive, realistic, and productive integrated approach that can be used by SMEs to optimize the effectiveness and efficiency of their software development processes.

This study's pressing research question is: **"How can small and medium-sized enterprises (SMEs) effectively address the specific challenges in their software development projects using an integrated and prioritized approach?"**

It is essential to understand these SME-related challenges in the software development context. Despite being smaller, SMEs compete in the same technology-driven markets as their larger counterparts. The need for efficient and effective software development practices becomes urgent, given their unique challenges—from resource limitations to technical constraints, and the absence of structured processes and methodologies. A deeper understanding of these challenges can pave the way for practical, SME-specific solutions and strategies.

The insights derived from this research are transferable across various domains. They can aid SMEs in identifying pain points and developing effective software development management strategies. These insights also serve as valuable tools for consultants, coaches, and organizational development professionals who advise and support SMEs. Additionally, policymakers can leverage these insights to create a supportive framework that aids SMEs in improving their software development processes, thus enhancing their competitiveness.

This thesis consists of five sections. It starts with a detailed literature review on software development in small and medium-sized enterprises (SMEs), followed by an explanation of the research methodology (Chapters 1-4). The findings, obtained from in-depth interviews with various stakeholders, are then presented and placed within the existing

literature, with a discussion of both theoretical and practical implications (Chapters 5-6). The thesis is concluded with a summary, a discussion of the research limitations, and recommendations for future research (Chapters 7-8).

## **2. Literature Method**

This chapter provides an overview of the literature search methodology used to investigate project delays and process optimization in software development teams in small and medium-sized enterprises. The literature search method is based on the grounded theory approach described by Wolfswinkel et al. (2013). This chapter explains the four steps of the literature search process, namely defining the search criteria, selecting the research areas, selecting the sources, and formulating the search terms. By following the literature search method described in this chapter, the research aims to provide a thorough and rigorous analysis of the relevant literature on project delays and process optimization (topic) in software development teams in small to medium enterprises (scope).

In the first step (1.1), the criteria for whether to include literature in the research must be determined, as outlined by (Wolfswinkel et al., 2013). Given the constantly evolving nature of the IT industry, a scope from 2015 to the present (2023) was chosen to search for relevant literature. For basic concepts or comparisons, reference can be made to earlier works.

Appropriate research areas must be selected for the second part of step 1.2 (Wolfswinkel et al., 2013). To achieve this, existing literature was searched related to the previously defined topic, scope, and search criteria. The relevance of this literature was assessed based on characteristics such as title, summary, highlights, and keywords. By identifying which research areas, the literature corresponds to, the final research areas were defined as Business, Management, and Accounting; Computer Technology; and



Engineering. The MBA in Digital Business and Analytics was also considered during this process.

For step 1.3, selecting appropriate sources/databases that allow filtering based on the previously defined criteria from steps 1.1 and 1.2 is crucial. In this study, sources from the University of Twente were used because they are largely open-access and contain all the necessary filter options. The functions and design of the filters are also similar between the sources. Scopus, ScienceDirect, and FindUT were identified as the best available sources after cross-testing for the broadest possible search terms and careful selection in research areas.

The fourth and final step of section 1.4 defines the precise formulation of the specific search terms (Wolfswinkel et al., 2013). These search terms are the result of steps 1.1, 1.2, and 1.3 performed earlier. The following search terms were chosen: SME software development project, SME software development challenges, software development challenges, software development project delays, agile software development, and software process improvement.

### **3. Theoretical Framework**

#### **3.1. Literature Analysis**

This literature review examines the challenges small and medium-sized enterprises (SMEs) face in software development. Compiling a list of challenges and categorizing them according to their frequency in the literature reveals insights into SMEs' experiences and struggles. The analysis highlights the interrelationships between challenges and the complexity of the software development landscape, presenting challenges sorted by their prevalence in the literature. Table 1 displays the final product of the literature analysis, compiling and grading the most prevalent challenges for SMEs in software development mentioned in scientific literature. The consecutive paragraphs will explain the challenges from Table 1 per theme and the corresponding theory.

**Table 1***Main Challenges Derived From Literature*

<b>Theme</b>	<b>Number</b>	<b>Challenge</b>	<b>Source(s)</b>
Process Adoption and Improvement	1	Inadequate and unstructured software development processes.	(Almomani et al., 2015; da Silva & Carneiro, 2016)
	2	Difficulties in managing and controlling the software development lifecycle.	(Almomani et al., 2015; Choras et al., 2020)
	3	Challenges in adopting and implementing agile and lean methodologies.	(Choras et al., 2020; Hu et al., 2015)
	4	Obstacles in adopting the Scaled Agile Framework (SAFe) and aligning it with organizational goals.	(Alahyari et al., 2019; Choras et al., 2020; Turetken et al., 2017)
Resource Constraints and Project Management	5	Insufficient resources and expertise for adopting industry best practices and standards.	(Almomani et al., 2015; Hu et al., 2015)
	6	Project delays due to poor planning, limited resources, and ineffective communication and collaboration.	(Yogaantara & Fajar, 2022)
	7	The need for significant investments in infrastructure, training, and personnel.	(Turetken et al., 2017)
Requirements Engineering and Quality Assurance	8	Challenges in aligning software requirements with verification and validation processes.	(Bjarnason et al., 2014; Turetken et al., 2017)
	9	Absence of systematic approaches to requirements engineering.	(Bjarnason et al., 2014)
	10	Difficulty in controlling code complexity, maintaining quality, and optimizing development processes.	(Turetken et al., 2017)
	11	Challenges in integrating usability engineering into the software development process.	(Hering et al., 2015)
Organizational Culture, Communication, and Change Management	12	The need to change organizational culture to embrace agile development practices.	(Choras et al., 2020)
	13	Resistance to change and lack of management support for implementing lean principles.	(Hu et al., 2015)
	14	Ambiguity arising from informal communication and ad-hoc decision-making.	(Almomani et al., 2015)

### ***Process Adoption and Improvement***

The Process Adoption and Improvement category is the most cited challenge in the software development SME. This notoriety emphasizes the importance of efficient processes for small and medium-sized businesses navigating the competitive landscape of the software industry. The underlying challenges within this category, such as inadequate and unstructured software development processes, indicate that SMEs often struggle to establish and maintain clearly defined workflows (Almomani et al., 2015; da Silva & Carneiro, 2016). This can lead to inefficiencies and hinder their ability to deliver high-quality software products. In addition, SMEs face difficulties in managing and controlling the software development lifecycle (Choras et al., 2020), exacerbating these inefficiencies.

The literature also discusses the challenges of adopting and implementing agile and lean methodologies, which are widely regarded as best practices for improving software development processes (Choras et al., 2020; Hu et al., 2015). These methodologies focus on rapid iteration, collaboration with customers, and continuous improvement, but SMEs may encounter obstacles in embracing them, such as limited resources or a lack of expertise (Hu et al., 2015). Another challenge SMEs face is the adoption of a Scaled Agile Framework (SAFe), which aims to scale agile practices to the needs of the company size (Alahyari et al., 2019; Turetken et al., 2017). Aligning SAFe with organizational goals can be especially difficult for SMEs, as it often requires significant investments in infrastructure, training, and human resources (Turetken et al., 2017). The prevalence of these challenges in the literature highlights the critical role that process adoption and improvement play in the success of SMEs in the software development industry.

### ***Requirements Engineering and Quality Assurance***

The second most cited challenge in the software development literature for SMEs relates to Requirements Engineering and Quality Assurance (Bjarnason et al., 2014; Turetken et al., 2017). This emphasis underscores the critical role that accurately defining, managing, and validating requirements plays in the software development process,

particularly for small and medium-sized enterprises striving to compete in a rapidly changing industry. One of the main challenges in this category is aligning software requirements with verification and validation processes (Bjarnason et al., 2014; Turetken et al., 2017). SMEs often struggle to ensure their software products meet their customers' needs and meet necessary quality standards, which can lead to project delays and unsatisfactory results.

The literature also points to the lack of a systematic approach to requirements engineering in SMEs (Bjarnason et al., 2014). Without a structured approach, these enterprises can experience difficulties in gathering, organizing, and managing requirements, which can result in inconsistencies and inefficiencies during the development process. In addition, SMEs often face challenges in managing code complexity, maintaining quality, and optimizing development processes (Turetken et al., 2017). These factors can contribute to higher costs, longer development times, and reduced customer satisfaction.

### ***Resource Constraints and Project Management***

Resource Constraints and Project Management ranked as the third most mentioned challenge category in the literature, sheds light on the difficulties encountered by Small and Medium Enterprises (SMEs) in effectively managing their limited resources, handling project-related tasks, and addressing the requirements for investments in infrastructure, training, and personnel. The significant presence of this category in the literature indicates that SMEs frequently face challenges in allocating and managing resources efficiently, and such challenges can directly influence the outcomes of their software development projects.

The underlying challenges, such as insufficient resources and expertise to adopt industry best practices and standards (Almomani et al., 2015; Hu et al., 2015), project delays due to poor planning, limited resources, and ineffective communication and collaboration (Yogaantara & Fajar, 2022), and the need for significant investment in infrastructure, training, and human resources (Turetken et al., 2017), all highlight how limited resources can be a root cause of several problems.

For example, insufficient resources and expertise can make it difficult for SMEs to keep up with technological progress, resulting in sub-optimal software development practices (Almomani et al., 2015; Hu et al., 2015). Similarly, project delays can result from a lack of resources, leading to challenges in maintaining project timelines, meeting customer expectations, and ultimately affecting the competitiveness of the organization (Yogaantara & Fajar, 2022). In addition, the need for significant investments in infrastructure, training, and human resources can place a heavy financial burden on SMEs, making it even more difficult to adopt and implement new practices and technologies (Turetken et al., 2017).

In summary, limited resources can exacerbate existing challenges, leading to a ripple effect on software development processes and overall project management in SMEs. So addressing these constraints should be a priority for SMEs to improve their software development practices and achieve better project outcomes.

### ***Organizational culture, communication, and change management***

Organizational culture, communication, and change management is the fourth most frequently mentioned challenge category in the literature. This category includes several underlying challenges faced by SMEs in software development (Almomani et al., 2015; Choras et al., 2020; Hu et al., 2015). A primary challenge, as identified by (Almomani et al. (2015); Choras et al. (2020), is the need to change organizational culture to embrace agile development practices, which requires a mindset of flexibility and adaptability among team members. Another challenge is the resistance to change and the lack of management support for implementing lean principles, which can hinder process improvement initiatives (Hu et al., 2015). Ambiguity due to informal communication and ad-hoc decision-making can also lead to confusion and inefficiencies in the development process (Almomani et al., 2015).

Resource constraints can be a root cause of these challenges, as resource-constrained organizations often struggle to invest in the necessary training, tools, and infrastructure to support new processes and practices. For example, the need to change organizational culture may be hampered by the inability to allocate resources for training and

coaching on agile methodologies (Choras et al., 2020). Similarly, the lack of managerial support for lean principles, as discussed by Hu et al. (2015) may be a result of resource constraints, as managers are hesitant to allocate resources to change initiatives without a clear understanding of the potential benefits. In addition, informal communication, and ad-hoc decision-making, as mentioned by Almomani et al. (2015), can be exacerbated by limited resources, as organizations may not have the resources to establish and maintain structured communication channels and decision-making processes. Thus, addressing resource constraints is critical to overcoming the challenges in this category and facilitating successful transformation in SMEs' software development processes.

### **Conclusion**

In the field of software development for Small and Medium Enterprises (SMEs), various challenges have been identified through literature. These challenges differ in terms of their frequency of discussion and the level of urgency attached to them. Several factors contribute to this variation, including the significance of a particular challenge, its prevalence in SMEs, and the availability of research on specific aspects of software development.

A challenge that is frequently mentioned in the literature can be considered more pressing due to its potential impact on the overall success of software projects and the competitiveness of SMEs. For instance, the most commonly cited challenge is process adoption and improvement. This challenge is critical because the adoption and refinement of software development processes directly influence the quality, efficiency, and predictability of project outcomes. However, some challenges receive less attention in the literature because they are more context-specific or have not been extensively explored in research. Usability and user-centered design, for example, may be mentioned less frequently as they are relatively specialized areas within software development.

Understanding the interrelationships between these challenges is crucial as it emphasizes the need for a holistic approach to addressing them in practice. Tackling one challenge can have positive or negative effects on other challenges. For instance, investing

in infrastructure, training, and personnel to improve processes can alleviate limited resources, but it can also introduce new challenges related to change management and organizational culture.

Examining these challenges in real-world contexts and observing how SMEs attempt to address them can yield new findings and uncover gaps in the literature. This exploration can help researchers identify previously unknown connections, dependencies, or nuances between challenges, thus providing valuable insights for the development of targeted and effective software development strategies for SMEs.

Understanding the gradation and interrelationships of challenges in the literature is essential for researchers and practitioners alike. It enables them to prioritize their efforts and allocate resources effectively. By investigating these challenges in practice, new insights can be gained and existing gaps in the literature can be addressed, ultimately leading to the development of more effective software development solutions for SMEs.

### **3.2. Practical Implications of Strengths and Challenges**

Software development within Small and Medium Enterprises (SMEs) has a unique mix of benefits and challenges. These characteristics directly influence the operational efficiency, growth prospects, and ultimate survivability of these companies in the software market.

Demanding situations inclusive of constrained sources, loss of established techniques, and difficulties in enforcing Agile and Lean standards can drastically jeopardize the operation, reputation, and economic achievement of SMEs. An instance of that is the observation conducted by Almomani et al. (2015), which shows that the absence of properly described techniques can cause delays and lower-exceptional products. These deficiencies can result in customer dissatisfaction and damage the firm's recognition inside the marketplace, making it extra challenging for them to obtain new projects. In addition, these quality issues can lead to higher costs in the long run, as correcting errors after release is often more expensive than preventing them during the development process.



SMEs are also prone to market fluctuations, given their commonly smaller monetary reserves as compared to large organizations. According to a study by Raghuvanshi et al. (2017), changes in economics, which include financial recessions or disruptions in the supply chain, may have a giant impact on SMEs and jeopardize their survival.

However, if the strengths of SMEs, together with their flexibility, fast decision-making, and creative innovation, are efficaciously harnessed, they can offer substantial competitive advantages. As highlighted by means of Serrador & Pinto (2015), SMEs can leverage their flexibility and agility to quickly respond to market changes and embrace new technologies ahead of large, much less nimble competition.

Furthermore, due to their smaller size, SMEs can provide an extra personalized and consumer-focused carrier, that may help foster customer loyalty and retention. In line with an observation by Frishammar (2003), SMEs that keep strong customer relationships can outperform their competition.

But these strengths additionally come with their set of challenges. The agility and short decision-making that symbolize SMEs can bring about a loss of direction and consistent approaches, that may in the end impact the performance and quality of software development (Almomani et al., 2015).

Therefore, SMEs must strike a balance between leveraging their strengths and addressing their challenges to succeed in the competitive software industry. This requires a detailed understanding of their operational context and the ability to adapt best practices and tools to their specific needs.

### **3.3. SMEs' Unique Market Position**

The unique market position of small and medium-sized enterprises (SMEs) in software development is characterized by their flexibility, proximity to customers, innovative potential, distinctive corporate culture, and ability to quickly adopt new technologies. These factors give SMEs a significant edge in the competitive software development market and differentiate them from larger enterprises.

SMEs often have flat organizational structures and less bureaucracy, which enable them to quickly respond to market changes and implement new technologies (Bititci et al., 2015). This flexibility is crucial in the software development industry, where technologies and consumer needs are constantly evolving. In addition, the more cohesive and collaborative culture within SMEs, due to their smaller size, can contribute to teamwork and employee engagement, which can lead to higher productivity and creativity (Cardon & Stevens, 2004).

SMEs can also interact more closely with their customers due to their smaller size. This enables them to better understand and respond to their needs (Lee & Cavusgil, 2006). This direct interaction with customers can lead to more customized and customer-oriented software solutions (Morgan et al., 2015), which can strengthen their market position. Moreover, their flexibility often allows SMEs to respond more quickly to new technology trends, as they are less constrained by lengthy approval processes and established routines typical of larger organizations (Cragg et al., 2011).

Finally, the innovative potential of SMEs can strongly influence their market position. SMEs are often less constrained by formal structures and procedures, leaving them more room for creativity and innovation (Tripathi & Agrawal, 2014). This innovative potential can help them develop unique software solutions that differentiate them from larger competitors.

While SMEs and large companies each have their unique strengths, the combination of flexibility, customer focus, innovative potential, distinctive corporate culture, and the ability to quickly adopt new technologies gives SMEs a unique market position in software development. By taking full advantage of these features, SMEs can compete with larger companies and be successful in the software market.

## **4. Methodology**

This chapter provides an in-depth description of the research design, data collection methods, sampling techniques, and data analysis procedures. By providing a clear and transparent account of the methodology, this chapter aims to demonstrate the study's validity and reliability while enabling future researchers to replicate the research if necessary. In this chapter, the research approach will be outlined, the chosen methods will be justified, and any limitations or challenges encountered during the research process will be discussed.

### **4.1. Research design**

The overarching aim of this study was to articulate the characteristics and dynamics of elements that create constraints and opportunities in the software development processes of SMEs. The pursuit was not solely to identify these factors, but also to fathom how and why they materialize, and how they impact the performance of the process and the success of projects. Such comprehension called for an in-depth understanding of the context, perceptions, and experiences of involved individuals, a depth often challenging to obtain with quantitative strategies. Consequently, a qualitative research method was adopted (Creswell & Creswell, 2017).

To further enhance the reliability and validity of the research, the study was structured into two main parts: a primary case study and a subsequent validation case study. The primary case study allowed for initial data gathering and exploration of the factors impacting software development in SMEs, while the validation study served to test the robustness of these initial findings and ensure their reproducibility and applicability across different contexts (Yin, 2009). This methodological choice underlines the pursuit of a balance between in-depth investigation and generalizable results.

In qualitative studies, and particularly in ethnography, the focus is on understanding the research object from the standpoint of the subjects involved (Hammersley, 2006). In this

context, it aimed to apprehend how software process improvement techniques are perceived and experienced by those who execute and manage them in SMEs. This method afforded the opportunity to collect rich, contextual, and precise data, critical for fulfilling the research objectives (Bryman, 2016).

Semi-structured interviews were chosen as the primary data collection method. These interviews strike a balance between structure and flexibility, enabling the asking of specific questions tied to the research goals while also allowing new topics and ideas to spontaneously emerge during the interviews (Cassell & Gillian, 2004). This flexibility facilitated an immersive exploration into the narratives and perceptions of the participants, thus fully illuminating the complexity and dynamics of software development processes in SMEs.

The Gioia technique was deliberately employed for the analysis of the collected data (Gioia et al., 2013). This method is purpose-built for analyzing qualitative data and proves effective in identifying, analyzing, and presenting complex and abstract concepts. Application of this technique assured a systematic, transparent, and reliable analysis process, producing results that are clear and comprehensible to both academic and practitioner audiences.

All methodological choices targeted the primary objective: providing a comprehensive and detailed understanding of the factors that engender constraints and opportunities for growth in the software development processes of SMEs, and how these factors can be addressed to optimize process efficiency and improve overall project success (Creswell & Creswell, 2017).

#### **4.2. Research Method**

Semi-structured interviews were conducted with software development professionals working in SMEs. The aim was to involve participants from all hierarchical levels in the software development process, such as product owners, project managers, and developers. Numerous scientific studies have shown that semi-structured/in-depth interviews require a

minimum sample size of 5 to 25 participants to obtain representative results (Creswell, 2007; Symon & Cassel, 2012; Townsend, 2013).

Ultimately, seven participants were interviewed to gather rich and varied insights into the characteristics and dynamics of the factors that create constraints and opportunities in software development processes. The participants came from a software development SME called LogicTrade, based in the Netherlands, with less than 50 employees. The entire LogicTrade software development team was interviewed, as each member was able to provide a unique perspective and insight into the complexities of software development within an SME context.

In addition, an interview was conducted with the Chief Product Officer of the contact provided by LogicTrade, named Every Day. The reason for this choice was that this officer was responsible for the overall product strategy and development, and thus could provide an in-depth and yet broad understanding of the challenges and opportunities of software development within an SME.

The interviews lasted approximately one hour each and were recorded and summarized for analysis. The semi-structured interview format allowed participants to provide detailed and nuanced responses to the questions, as well as flexibility to expand on interesting or relevant points.

#### **4.3. Sampling Procedure**

All key people involved in the software development process of the two specific SMEs were surveyed. This included product owners, software developers, and project managers. The selection of different SMEs was intended to increase the usability and validity of the results and to gain a broader understanding of the specific challenges faced by these SMEs (Bryman, 2016).

Due to the limited size of these companies, it was necessary to follow this approach. This was crucial to get a representative group and get an in-depth view of the software development process (Creswell & Creswell, 2017). This approach allowed for a deeper and

more nuanced exploration of the research questions as all relevant viewpoints were included.

It is important to note that the selection of participants may not have been representative of all SMEs, as participants may come from certain industries or regions (Bryman, 2016). In addition, the selection of participants based on their involvement in the software development process may have resulted in a biased group. Despite this, attempts have been made to include a diverse range of participants with different levels of experience and viewpoints on the software development process (Creswell & Creswell, 2017). Furthermore, the limitations of the sampling method were mentioned in the Limitations and Recommendations chapter.

#### **4.4. Interview Content**

The original interview questions were carefully crafted based on critical challenges identified through the literature review of SME software development. With these themes as a guideline, ten main questions have been formulated, each with two sub-questions. These sub-questions served as an impetus for a more in-depth discussion of each main topic, while also following a chronological sequence inherent in a typical software development process (Bryman, 2016).

After analyzing the results of the primary case study, the interview questions for the validation case study were modified. This was done to further explore and test the newly discovered issues and challenges that emerged during the first case study. It was a necessary step to validate the results of the first study and ensure the consistency and reliability of the findings (Creswell & Creswell, 2017).

This iterative approach, in which the interview questions were adjusted based on the results of the first research phase, provided a deeper and more focused insight into the problem. It also allowed for open responses from the participants, leading to the discovery of new challenges not previously identified in the literature. This approach promoted authentic

insights and a better understanding of the factors influencing software development within SMEs (Cassell & Gillian, 2004).

This structured and iterative approach proved especially valuable during the data analysis phase of the study. The structured format and chronological order of the questions, combined with the space for open answers, streamlined the organization and analysis of the interview data. This made it easier to identify patterns, trends, and relationships between different factors and strategies, ultimately supporting the research objective of providing valuable insights and recommendations to SMEs in the software development industry (Gioia et al., 2013).

#### **4.5. Data Analysis Technique**

The method adopted for dissecting the responses collected during the interview process was the Gioia approach as suggested in the work of Gioia and his collaborators (2013). This multifaceted procedure comprised a sequence of steps - formulation of a coding manual, encoding of the acquired data, and the discovery of significant concepts and prevailing themes. The themes that emerged were subjected to analysis, forming an umbrella of findings aimed at answering the primary research question.

With the goal to maintain an organized and transparent approach toward data analysis, certain measures were put into effect. Initially, a coding manual was constructed, which was founded upon the main research question, the primary goal of the study, and the theoretical framework. This manual was scrutinized by a project manager and a mentor, ensuring its comprehensiveness and clarity. This process is reinforced by the research of Gioia et al. (2013). Next, the acquired data was subject to coding, this coding was reassessed periodically to ensure precision and uniformity Gioia et al. (2013). Lastly, the emergent themes from the data were reassessed and validated through a process termed 'member checking'. This process involved sharing findings with the participants and gathering their feedback. This step is also advocated by Gioia et al. (2013).

The Gioia methodology was selected for this research because of its potential to provide a structure that ensures the validity and reliability of data analysis while permitting inductive research. Notably, the use of a coding manual and multiple coders ensured consistency across coders, and the process of 'member checking' allowed for the validation of findings with the participants. Furthermore, the key features and steps described by Gioia et al. (2013) provided a robust and grounded strategy for conducting research. The overall approach, communication with participants, and methodology for data collection and analysis were appreciated by the participants for their logical sequence and natural flow.

### Figure 1

#### *Features of the Methodology That Enhance Grounded Theory Development*

Step <sup>a</sup>	Key Features
Research Design	<ul style="list-style-type: none"> <li>• Articulate a well-defined phenomenon of interest and research question(s) (research question[s] framed in “how” terms aimed at surfacing concepts and their inter-relationships)</li> <li>• Initially consult with existing literature, with suspension of judgment about its conclusions to allow discovery of new insights</li> </ul>
Data Collection	<ul style="list-style-type: none"> <li>• Give extraordinary voice to informants, who are treated as knowledgeable agents</li> <li>• Preserve flexibility to adjust interview protocol based on informant responses</li> <li>• “Backtrack” to prior informants to ask questions that arise from subsequent interviews</li> </ul>
Data Analysis	<ul style="list-style-type: none"> <li>• Perform initial data coding, maintaining the integrity of 1st-order (informant-centric) terms</li> <li>• Develop a comprehensive compendium of 1st-order terms</li> <li>• Organize 1st-order codes into 2nd-order (theory-centric) themes</li> <li>• Distill 2nd-order themes into overarching theoretical dimensions (if appropriate)</li> <li>• Assemble terms, themes, and dimensions into a “data structure”</li> </ul>
Grounded Theory Articulation	<ul style="list-style-type: none"> <li>• Formulate dynamic relationships among the 2nd-order concepts in data structure</li> <li>• Transform static data structure into dynamic grounded theory model</li> <li>• Conduct additional consultations with the literature to refine articulation of emergent concepts and relationships</li> </ul>

<sup>a</sup>The Research Design and Data Collection steps are moderate variations on traditional grounded theory approaches. The Data Analysis and Grounded Theory Articulation steps constitute the main distinctive features of the approach.

Reprinted from “Seeking Qualitative Rigor in Inductive Research“, by Gioia, D. Corley, K. Hamilton, A., 2013, *Organizational Research Methods*, 16, p. 26.



#### **4.6. Ethical Considerations**

This research, which was centered on conversations with staff from all levels of the business hierarchy discussing difficulties in their profession, took into account key ethical guidelines. Securing informed agreement from each individual was the first move, guaranteeing the participants understood fully what the study entailed, the types of inquiries, as well as any potential gains and risks of their involvement.

The need to preserve confidentiality necessitated the implementation of rigorous measures to protect participants' identities and any confidential information shared during discussions. Offering anonymity was an important factor, as it encouraged participants to express their thoughts and experiences openly without any fear of backlash.

The research upheld high levels of professionalism and respect, treating all those involved with impartiality, regardless of their rank in the organization. A commitment to fairness was maintained throughout the research process, with a keen eye ensuring no person or group was unjustly singled out or focused on.

Upon completion of discussions, individuals were offered the chance to share any thoughts or questions they might have had in a debriefing session. Additionally, they were briefed on how their shared data would be utilized. This level of transparency helped establish a sense of trust and build a positive rapport with participants.

Incorporating these ethical considerations was key in ensuring the research was carried out with due diligence and respect, and that the participants felt comfortable sharing their experiences and insights. By strictly adhering to these ethical guidelines, the integrity and reliability of the research findings were maintained.

## 5. Findings and Results

This chapter presents the findings and results of the study, applying the Gioia method to the analyzed interview data. The aim is to provide a detailed insight into the main findings that emerged from the interviews.

Using the structured Gioia method, the collected data has been carefully analyzed and coded. This method provides an in-depth understanding of the underlying patterns and meanings in the data. By applying this method, an attempt is made to reveal the perceptions, views, and experiences of the interviewed individuals regarding the software development process within their organization.

This chapter includes an overview of the study population and the individuals interviewed. In addition, the data analysis procedure, the coding scheme and the coding process are briefly explained.

The findings are presented through identified themes and categories that emerged during the analysis. These are described in detail and linked to existing literature. This emphasizes the relevance and similarities with previous studies.

### 5.1. Overview of Study Population

**Table 2**

*Overview of Study Population*

<b>Occupation</b>	<b>Number of subjects</b>
Product Manager	2
Chief Product Officer	2
Product Engineer	3

## **5.2. Data Analysis Procedure**

The analysis of the collected data proceeded in several phases, in accordance with the method developed by Gioia and his colleagues.

First, the interviews were recorded, and the answers were elaborated per individual interview question. This contributed to a detailed representation of each participant's response.

After this initial step, the elaborated interview data was coded to 1st order codes, in accordance with Gioia's method. This systematic coding process led to the organization and analysis of the collected raw data in a structured way.

Subsequently, the initial codes were transformed into 2nd order themes. These themes were partly derived from the theoretical framework, but new themes also emerged during this phase of the analysis. Incorporating both existing and new themes provided a rich and diverse picture of participants' perceptions and opinions.

In the next phase of the analysis, the relationships between the themes were examined, resulting in the identification of overarching theoretical dimensions. By recognizing patterns and connections between the various themes reveals underlying dynamics and mechanisms.

In the final phase of the data analysis, all 1st order codes, 2nd order themes and theoretical dimensions were integrated into a coherent data structure. This methodical approach provided a clear overview of the findings, making it possible to formulate detailed and global conclusions based on the results.

Coding reliability and validity were key concerns throughout the analytical process. In this regard, multiple measures were taken to ensure the thoroughness and accuracy of the analysis. One crucial step to ensure reliability was the involvement of both the internal and external research coordinator in reaching consensus on the codes and themes. This procedure increased the intercoder reliability and ensured that the coding and thematization were not purely subjective interpretations but based on intersubjective agreement.

### 5.3. Development of Data Structure

The deployment of the Gioia methodology, outlined by Magnani and Gioia (2023), has played a crucial role in developing the current data structure. This approach aims at facilitating the translation of individual experiences and perspectives, as expressed in 1st order concepts, into broader theoretical constructs and dimensions.

Each 1st order concept, identified and extracted from the collected interview data, reflects the unique insights and experiences of the participants. To illustrate, the concept of "Excessive dependence on individual team members for critical tasks" highlights a core problem within many organizations by emphasizing the problem of critical tasks resting on just a few individuals. This first-order notion, arising directly from the respondents' experiences, forms the basis for further theoretical interpretation.

Subsequently, by applying an abductive process of 'systematically combining' (Magnani & Gioia, 2023), these 1st order concepts were brought together in 2nd order themes. Here the theory-driven nature of the methodology becomes apparent. For example, the theme "Incomplete and unstructured software development processes" was formulated as an overarching concept for various 1st order insights, as they collectively highlight the gaps and unstructured aspects of software development processes.

However, considerations had to be made here. Take, for example, the concept of 'Resistance to change and the difficulty of fully transitioning to these methodologies due to the existing work structure and limited resources'. This concept could fall under different themes, such as 'Challenges in controlling software development processes' and 'Challenges in the implementation of agile and lean methodologies'. Here a sound and thoughtful decision was made to link it to the latter theme, as it most closely connected to the wider discussion about implementing Agile and Lean principles.

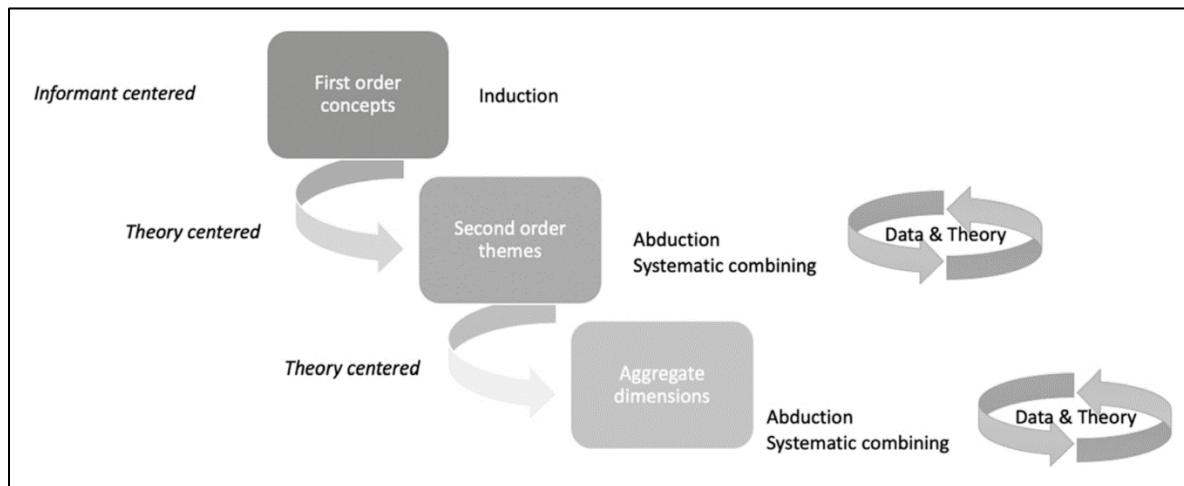
The final step in the process of data analysis concerns the formation of the aggregate dimensions. The formulated 2nd order themes are further abstracted into overarching dimensions, which together reflect the complex and multifaceted nature of the research findings. For example, the "Software development process management" dimension

encapsulates the broader theoretical perspective on the management strategies and processes that influence software development in SMEs.

The formation of an aggregate dimension from multiple 2nd order themes requires a combination of deductive and abductive thinking, with the researcher looking for overarching patterns and trends that connect these themes. For example, in this study, the themes 'Incomplete and unstructured software development processes', 'Challenges in controlling software development processes' and 'Challenges in the implementation of agile and lean methodologies' were combined into the dimension 'Software development process management'. This decision was made because these themes shared a common focus on different aspects of software development processes and their management within SMEs.

On the other hand, some dimensions were formed from a single 2nd order theme, such as 'Resource and capacity management'. This reflects the significant distinctiveness of this theme and its importance within the wider research field. This choice was made because the theme was sufficiently substantial and unique to be considered a separate category. This decision was made based on both the empirical data and the relevance of the theme within the existing theoretical framework of software development within SMEs.

This process of abduction and systematic combining, evolving from 1st order concepts to 2nd order themes and finally to aggregate dimensions, illustrates the rigorous and thorough nature of the Gioia methodology. This methodology enables us to generate in-depth and nuanced insights to provide a coherent and structured overview of the complex world of software development within SMEs, further contributing to the existing literature (Magnani & Gioia, 2023).

**Figure 2***The Inferential Process in Developing a Data Structure*

Reprinted from "Using the Gioia Methodology in international business and entrepreneurship research", by Magnani G., Gioia D., 2023, *International Business Review*, 32, p. 4.

#### 5.4. Overview of Main Concepts

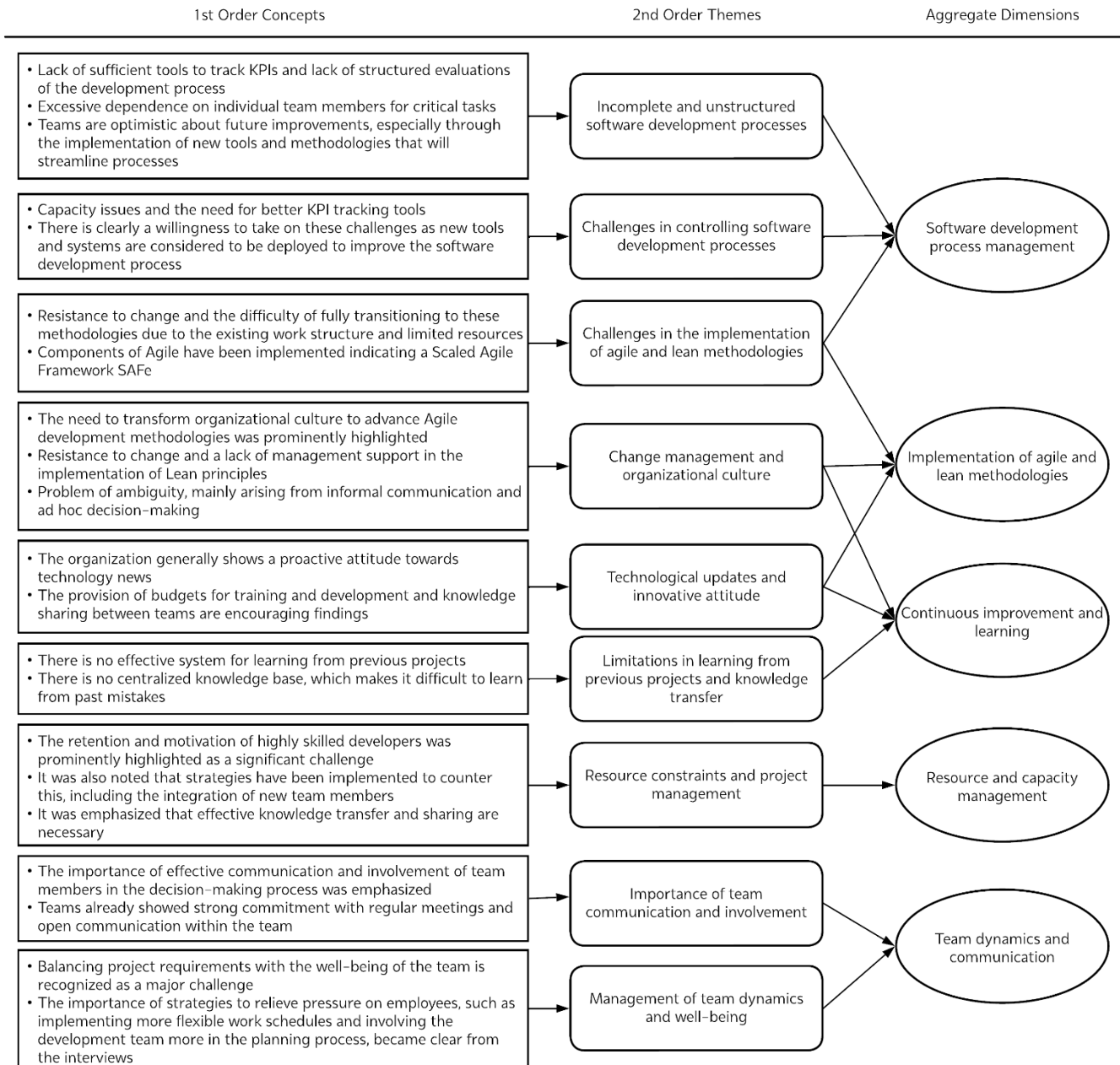
**1st Order Concepts.** This is the starting point of the data structure and arises from the direct interpretations and statements of the interviewed participants. These first-order concepts are expressed in terms of the participants themselves, and form the basis for the further development of the data structure. In this study, the first-order concepts were identified by thematically coding the interview transcripts, with concepts such as "imperfect software development processes", "team management challenges", "knowledge transfer limitations", etc.

**2nd Order Themes.** These themes are the interpretations of first order concepts. They form a second level of abstraction, grouping first-order concepts into broader, overarching themes. Examples of second-order themes in this study include "unstructured software development processes", "management of team dynamics and well-being", "change management and organizational culture" and so on.

**Aggregate Dimensions.** These are the highest level of abstraction in the data structure and bring together multiple second-order themes under even broader categories. These aggregate dimensions are "Software development process management",

“Implementation of agile and lean methodologies”, Continuous improvement and learning”,  
 “Resource and capacity management”, “Team dynamics and communication”.

**Figure 3**  
*Data Structure*



## 5.5. Explanation of Data Structure

In the light of the research methods employed, in particular, the Gioia methodology, five fundamental dimensions have been identified within the context of software development for small and medium-sized enterprises (SMEs). These dimensions show similarities with the existing academic literature but also offer new perspectives and insights.

First, managing the software development process is critical. Challenges arising from inadequate and unstructured management have already been extensively highlighted in the literature (Almomani et al., 2015; Choras et al., 2020; da Silva & Carneiro, 2016). Research by Pino et al. (2008) and Laporte et al. (2008) support the idea that efficient processes play a determining role in software development performance in SMEs. However, the current analysis also underlines that the unstructured nature of the processes is a major obstacle to the implementation of agile and lean methodologies. In addition, the problem of distributed software development, where teams work remotely, exacerbates already existing challenges.

The second dimension concerns the implementation of Agile and Lean methodologies. Previous studies have highlighted the challenges in adopting and implementing these methodologies (Choras et al., 2020; Hu et al., 2015; Turetken et al., 2017), while other studies demonstrate their effectiveness in improving software development processes (Dybå & Dingsøy, 2008; Hajjdiab & Al Shaima Taleb, 2011). This analysis confirms these findings and further states that more research is needed on the challenges faced by SMEs in scaling up these methodologies.

The third dimension, continuous improvement and learning, supports the philosophy of Agile and Lean. This dimension is supported by studies pointing to the need for a systematic approach to improvement (Bjarnason et al., 2014; Kruchten, 2013; Turetken et al., 2017), indicating that continuous improvement and learning are essential for the adaptation and growth of SMEs in a rapidly evolving industry.

The management of resources and capacity, the fourth dimension, also finds parallels in the literature. Studies have identified resource and project management



constraints as significant challenges for SMEs (Almomani et al., 2015; Fowler & Highsmith, 2001; Hu et al., 2015). This dimension confirms the literature and improves understanding of the impact of these constraints on the adoption of industry standards and best practices.

Team dynamics and communication, the fifth identified dimension, resonates with findings in the literature pointing to the central role of organizational culture and communication in change management (Almomani et al., 2015; Choras et al., 2020; Hu et al., 2015). This dimension confirms and reinforces the importance of this aspect, highlighting the importance of managing team dynamics and facilitating effective communication within the context of SMEs.

Although the continuous improvement and learning dimension has been less extensively covered in the literature, this dimension emphasizes the importance of continuous education and training for the implementation of the latest methods and techniques in software development. In this light, the limitations in learning from previous projects and knowledge transfer, and the challenges in change management and organizational culture are closely linked.

In a broader perspective, team dynamics and well-being are closely linked to the level of team communication and engagement. Effective communication and involvement of team members can positively influence team dynamics and contribute to a better overall work environment.

Finally, the extent to which an organization keeps up with technology updates and maintains an innovative attitude impacts the effectiveness of change management, organizational culture, and the implementation of agile and lean methodologies. An organizational culture open to change can pave the way for successful implementation of these methodologies.

## **5.6. Analysis and Validation of Results**

In the secondary case study, an interview was conducted with a product owner of another SME in software development. The results of this secondary case study and the

primary case study present interesting similarities and differences between each other and scientific literature.

The results of both studies reflect the considerable focus on project management and efficiency. For example, the respondent from the validation study emphasized the use of internal project management tools, as did the respondents from the initial study who mentioned the importance of well-structured management of the software development process (Almomani et al., 2015; Choras et al., 2020; da Silva & Carneiro, 2016). Also, the emphasis on compliance with project size, the difficulties in estimating project size and costs, and monitoring work-hour efficiency resonate with the findings of the first study and the existing literature (Hron & Obwegeser, 2018; Schön et al., 2017).

The identified dimension of continuous improvement and learning was reconfirmed in the validation study. The respondent mentioned the use of technology for continuous improvement and learning and development opportunities through webinars. This confirms the literature that continuous improvement and learning are essential for the adaptation and growth of SMEs in a rapidly changing industry (Bjarnason et al., 2014; Kruchten, 2013; Turetken et al., 2017).

With regard to organizational culture and team dynamics, the finding from the first study is supported by the respondent in the validation study, who emphasizes the need to maintain team dynamics and implement a structured internship program. This resonates with literature citing the role of organizational culture and communication in change management (Almomani et al., 2015; Choras et al., 2020; Hu et al., 2015).

A unique dimension emerging from the validation study is the strategic development focus. This includes themes related to the organizational shift to digital development and the pursuit of project scope compliance. This indicates that SMEs are flexible and can strategically adapt to market changes, which is consistent with previous research (Deakins & Freel, 2012).

Finally, risk management is another new dimension identified in the validation study. This uses documentation and archiving as risk management measures, which is consistent

with studies highlighting the impact of such practices on managing risk in project management (Chouki et al., 2020; Stettina & Hörz, 2014).

### 5.7. New Gradation of Challenges

In this paragraph the focus will be on identified challenges of small and medium-sized software development companies, resulting from interviews conducted. These seven challenges form a revised hierarchy of hindrance these companies face.

Each challenge is given a detailed discussion and contextual placement, ensuring a clear understanding of the barriers they pose for SMEs. In addition, each challenge is compared to existing academic literature to provide a comprehensive evaluation of the problems and possible solutions. (Complete interview citations can be found in Appendix C)

**Table 3**

*Top Mentioned Challenges Derived From Interviews*

<b>Challenge</b>	<b>Times Mentioned (out of 7 interviews)</b>	<b>Examples</b>
Resource Constraints	6	Problems with employing enough developers Moving resources from one project to another
Lack of Formal Processes	5	No standardized bug tracking methods Lack of formal communication protocols
Knowledge Management Issues	5	Difficulty keeping up with the latest technologies and methods Loss of valuable knowledge when employees leave
High Employee Turnover	4	Problems retaining developers Difficulty staying competitive with larger companies in attracting technical talent
Inadequate Project Management	4	Issues with scope creep and not being able to deliver projects on schedule Difficulties in managing customer expectations
Poor Communication	3	Miscommunication with clients about project requirements Inefficient internal communication between teams
Technical Debt Management	3	Accumulation of technical debt by prioritizing new features over code refactoring Codebase is becoming increasingly difficult to maintain due to the implementation of "quick fixes"

### ***Resource Constraints***

Resource constraints, mentioned in 6 of the 7 interviews, encompass a series of problems that SMEs (SMEs) may encounter. As one interviewee said, "We often don't have enough hands to work on all the projects." This illustrates the struggle of SMEs to find and retain the right amount of qualified personnel. In addition, they may struggle with balancing limited financial resources between different projects and other operating costs. In the literature, these issues are recognized and emphasized as limiting factors for the growth and success of SMEs (Almomani et al., 2015; Cardon & Stevens, 2004).

### ***Lack of Formal Processes***

Lack of formal processes, mentioned in 5 of the 7 interviews, is another challenge. One interviewee indicated: "We often work on an ad hoc basis, which can lead to errors and inefficiencies." This indicates the tendency of SMEs to operate without well-defined processes and systems. In the literature, this lack of formality is identified as a potential barrier to effective and efficient software development, and as a possible cause of problems such as bug tracking and project management (Laporte et al., 2008; Pino et al., 2008).

### ***Knowledge Management Issues***

Knowledge management issues, mentioned in 5 out of 7 interviews, are another challenge. One interviewee explained, "Transferring knowledge between team members can be difficult, especially when people leave the company." This highlights the challenges of capturing, storing and sharing knowledge within an organization. It is recognized in the literature that knowledge management problems can lead to inefficiencies, such as reinventing solutions and losing business-critical knowledge (Cragg et al., 2011).

### ***High Employee Turnover***

High employee turnover, mentioned in 4 out of 7 interviews, is a fourth challenge. As one interviewee said, "We often lose good developers to bigger companies with better salary

packages." This demonstrates the challenges SMEs face in attracting and retaining talent in a competitive market. The literature highlights how employee turnover can lead to the loss of critical skills and business knowledge, which can reduce efficiency. (Cardon & Stevens, 2004; Mäkelä et al., 2010).

### ***Inadequate Project Management***

Inadequate project management, mentioned in 4 of the 7 interviews, is a fifth challenge. One interviewee explained, "It's hard to keep all projects on track, especially when we're working on multiple projects simultaneously." This highlights the challenges SMEs can face in effectively managing multiple projects, managing client expectations and delivering projects on time and within budget. The literature highlights how inadequate project management can lead to inefficiencies, customer dissatisfaction and loss of corporate reputation (Serrador & Pinto, 2015; Yogaantara & Fajar, 2022)

### ***Poor Communication***

Poor communication, mentioned in 3 of the 7 interviews, is a sixth challenge. One interviewee said: "It is sometimes difficult to communicate clearly with customers about their expectations and requirements." This illustrates the challenges of effective communication, both internally between team members and externally with customers. The literature suggests that effective communication is crucial to project management and customer satisfaction and that poor communication can lead to misunderstandings and inefficiencies (Bjarnason et al., 2014; Lee & Cavusgil, 2006).

### ***Technical Debt Management***

Technical debt management, mentioned in 3 of the 7 interviews, is the last challenge. One interviewee explained, "We often deal with old code that needs updating, but there isn't always time for that." This illustrates the challenges of managing technical debt, that is, balancing delivering projects on time with maintaining and updating the code base. The

literature emphasizes that technical debt management is a critical factor in software development and that an accumulation of technical debt can lead to increased maintenance costs and degradation of code quality (Alahyari et al., 2019; Kruchten, 2013).

### **5.8. Interrelationship of Challenges**

The analysis of the seven challenges in SME software development highlights the interdependence of these issues, with the impact of one challenge often exacerbating the severity of the others. This complex network of challenges is deeply embedded in the literature on software development and SME management (Höst, 2000; Staples & Niazi, 2007).

Central to this network of challenges is the lack of formal processes. The literature repeatedly emphasizes the importance of efficient processes in software development (Laporte et al., 2008; Niazi, 2012; Pino et al., 2008). Studies suggest that inadequate and unstructured management can be a significant barrier to implementing Agile and Lean methodologies (Almomani et al., 2015; Choras et al., 2020; da Silva & Carneiro, 2016; Hajjdiab & Al Shaima Taleb, 2011). The primary case study showed that a lack of structured processes is a dominant challenge for SMEs. But this begs the question: Why do some SMEs fail to implement formalized processes? Is it a matter of limited resources, a lack of knowledge, or is it a conscious choice to remain flexible and adaptive (Fitzgerald et al., 2006; Fowler & Highsmith, 2001)?

Resource constraints and the lack of formal processes are closely linked. Both challenges can hinder the adoption of best practices and industry standards, as illustrated by the fourth dimension of the primary case study. Previous literature has identified resource and project management constraints as significant challenges for SMEs (Almomani et al., 2015; Fowler & Highsmith, 2001; Hu et al., 2015; Lee & Xia, 2010). While this confirms the link, it remains unclear how SMEs can overcome these challenges in practice.

Insufficient project management and poor communication are two more challenges that are closely linked to the shortcomings of formal processes and resource constraints.

Poor project management can lead to project delays, inaccurate cost estimates and an overall drop in efficiency (Rodríguez et al., 2012). A lack of clear communication can lead to misunderstandings, loss of essential information and a drop in team morale (Šmite et al., 2010). While this is often emphasized in the literature, there seem to be few practical solutions to overcome these challenges.

Managing technical debt is a challenge highlighted by both the primary and secondary case studies (Kruchten, 2013). The accumulation and maintenance of technical debt can lead to significant long-term costs, which can hinder SMEs in their growth and development (Cunningham, 1992; Nugroho et al., 2011). But it's also an inherent part of software development - even with the best planning and execution, technical debt will arise at some point. This raises the question of how SMEs can manage this inevitable challenge.

Finally, knowledge management issues and high turnover are closely linked. When employees leave an organization, they often take with them valuable knowledge and experience (Deakins & Freel, 2012; Menezes et al., 2019). This highlights the need for SMEs to establish robust knowledge transfer and retention processes.

In summary, while some of these challenges may seem bigger than others, they are all interrelated and affect each other. Addressing these challenges therefore requires a holistic approach that takes into account the full context of SMEs and how these challenges are intertwined (Laporte et al., 2008; Staples & Niazi, 2007). It is also necessary to remember that these challenges are not necessarily 'problems' to be 'solved' - in many cases they are simply aspects of running an SME that need to be managed and addressed (Kruchten, 2013; Özkan & Mishra, 2019).

## 6. Discussion

### 6.1. Summary of Results

This research aimed to identify the specific challenges in the software development processes and projects of small and medium-sized enterprises (SMEs). The central research question this addressed was: "How can small and medium-sized enterprises (SMEs) effectively address the specific challenges in their software development projects using an integrated and prioritized approach?"

The data collected showed that several factors influence the software development processes in SMEs. Key elements included managing the development process, implementing Agile and Lean methodologies, continuous improvement and learning, resource and capacity management, and team dynamics and communication. In addition, seven key challenges were identified, namely resource constraints, lack of formal processes, knowledge management issues, high staff turnover, inadequate project management, communication issues, and technical debt management.

### 6.2. Interpretations

The examination of the software development landscape within small and medium-sized enterprises (SMEs) unveils intriguing findings that dispute existing literature and supply fresh insights for future studies.

Managing resources within SMEs represents a significant challenge. Results reveal the issue in question arises not just from financial constraints, but also from a lack of manpower and technological capabilities. Even though the problem often seems peripheral in scientific studies (Fitzgerald et al., 2006), it constitutes a central concern in daily operations. This reflects a knowledge gap that calls for further exploration. The outcomes suggest effective resource management stands to gain from an enhanced comprehension of the unique circumstances of SMEs, including financial limitations, workforce dynamics, and technological capabilities.



Moreover, the analysis questions the effectiveness of Agile and Lean methodologies in SMEs. In stark contrast to the findings of Dybå & Dingsøy (2008) and Hajjdiab & Al Shaima Taleb (2011), which laud these methodologies' efficiency, this research points to a need for more structured processes within these enterprises. This indicates that Agile and Lean methodologies might not be as beneficial for SMEs as traditionally thought. Given the distinctive challenges, SMEs might need to adopt these methodologies differently or blend them into a more systematic approach.

Knowledge management remains a pertinent issue for SMEs, in contrast to academic literature suggesting significant advancements in this field (Cragg et al., 2011). The disparity between these findings and existing literature indicates that current theories and models of knowledge management may not cater to the unique requirements of SMEs. Such organizations may grapple with issues like rapid technological advancements and insufficient resources to keep pace, culminating in numerous knowledge management challenges.

High employee turnover in SMEs features prominently in the findings, which contradicts scientific literature about SMEs' ability to draw and retain employees via flexibility and growth opportunities (Cardon & Stevens, 2004; Mäkelä et al., 2010). This discrepancy invites questions about the efficacy of existing strategies for workforce management and retention within SMEs. They may fail to communicate or utilize their benefits effectively or encounter unique challenges in employee retention and management that existing strategies and practices do not fully address.

The investigation into team dynamics and communication within SMEs shows these areas face specific challenges and are not adequately covered by existing literature. Prior studies imply open communication and positive team dynamics are essential for efficient software development management (Bjarnason et al., 2014; Lee & Cavusgil, 2006). However, in SMEs, these aspects prove more complex than in larger organizations. Findings show that the casual and flat structures often seen in SMEs can both help and hinder communication and team dynamics. These structures can enhance communication flow by

dismantling hierarchical barriers, but can also create role and responsibility confusion, leading to unclear communication and disrupted team dynamics.

The analysis also identifies technical debt as a significant concern for SMEs, a detail frequently neglected in academic works (Kruchten, 2013; Nugroho et al., 2011). Technical debt, the implicit cost of added work from opting for a quick, easy solution instead of a superior long-term approach, can curb software development productivity. SMEs may lack the resources to manage the extra workload that ensues when technical debt is not properly handled. This necessitates more attention in future studies on managing technical debt in SMEs, along with the creation of effective strategies for its management in this setting.

This comprehensive discourse underscores the need to understand and address SMEs' unique software development challenges. It offers valuable insights that question existing literature and form a foundation for future research and practice. By tackling these challenges, SMEs can enhance the effectiveness and efficiency of software development processes, leading to improved outcomes and a stronger market presence.

### **6.3. Practical Implications and Recommendations**

Based on the extensive research of current software development within small and medium-sized enterprises (SMEs), a series of important implications and recommendations can be formulated into a coherent action plan. This plan proposes an integrated approach to navigate the specific challenges these companies face. The order of the parts is structured by priority, focusing first on immediate operational barriers such as resource management, and then on more complex aspects such as process methodologies and knowledge management. A guiding model of this integrated approach is shown in Figure 4.

The action plan starts with the critical topic of Resource Management, as it forms a fundamental basis for all other activities within a company. Research has shown that managing resources within SMEs poses challenges not only due to financial constraints, but also due to a lack of human resources and technological capabilities (Fitzgerald et al., 2006). In response, it is crucial to invest in customized management training that explicitly

addresses these unique challenges. These trainings can be enhanced by integrating technologies that increase operational efficiency, such as cloud-based solutions or machine learning tools (Armbrust et al., 2010; Jordan & Mitchell, 2015). In addition, deploying AI tools can help solve resource shortages. For example, AI can help automate routine tasks, freeing employees to focus on more complex tasks (Chui et al., 2018).

Next, the plan covers the implementation of Agile and Lean methodologies. This is the second step because these methods directly relate to how software development is done after mobilizing the necessary resources. Here it is essential for SMEs to flexibly adapt to their specific context, which may mean mixing Agile principles with more structured and sequential development processes, such as waterfall methods, leading to a hybrid model that offers both stability and flexibility (Larman & Basili, 2003).

Knowledge management is the next topic on the agenda, given the importance of up-to-date technological expertise in today's digital age. To address this issue, the action plan could suggest setting up regular training and workshops to educate employees on the latest technological developments. Fostering a culture of knowledge sharing through the use of collaborative tools such as Microsoft Teams or Slack can help capture and share business knowledge (Leonardi & Meyer, 2015).

With regard to the issue of employee turnover, this follows as the fourth step because, despite the importance of the topic, retaining employees is closely linked to the earlier steps of effectively managing resources and ensuring adequate training and technological competence. SMEs could review their retention strategies by adopting effective HR practices such as comprehensive onboarding programs, attractive benefits and employee satisfaction surveys to collect valuable feedback (Holtom et al., 2008).

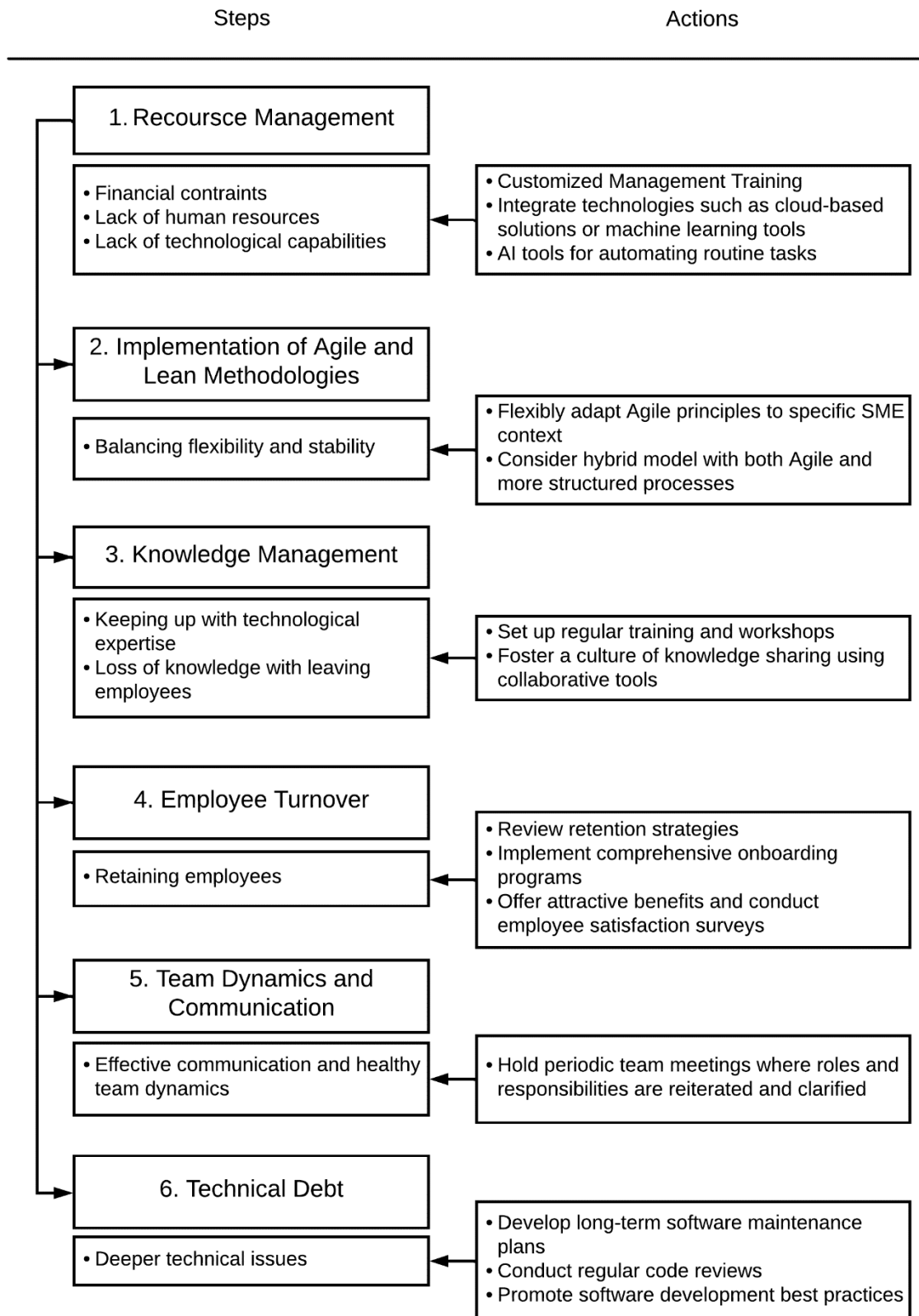
In the context of team dynamics and communication, this point is emphasized later in the action plan because effective communication and team dynamics often result from the correct application of the previous steps, such as a clear division of roles and responsibilities, as well as a healthy company culture. It can be helpful to have periodic team meetings where roles and responsibilities are reiterated and clarified (Moe et al., 2012).

Finally, in the area of technical debt, it is mentioned last in the Action Plan as it is a deeper technical issue that is most effectively addressed when other operational challenges have already been addressed. The action plan proposes to develop long-term software maintenance plans and conduct regular code reviews (Kruchten, 2013). Promoting software development best practices can help reduce technical debt and increase overall productivity (McConnel, 2004).

By following these steps in this order, SMEs can create a holistic approach to addressing the unique challenges of software development, which can result in improved processes, better outcomes, and ultimately a stronger market position.

Figure 4

## Integrated Approach for Overcoming Main Challenges



## 7. Conclusion

The central aim of this research was to uncover and understand the unique challenges small and medium-sized enterprises (SMEs) experience during their software development processes. Specifically, the research question: "How can small and medium-sized enterprises (SMEs) effectively address the specific challenges in their software development projects with an integrated and prioritized approach?"

Collecting and studying extensive data has unlocked a series of important insights. The spectrum of challenges facing SMEs is significant. Resource constraints, high frequency of personnel changes, lack of established processes, problems in knowledge transfer, and handling technical debt are just some of these hurdles. Nevertheless, these challenges are not insurmountable. With the help of a thoughtful, integrated approach, these stumbling blocks can be turned into opportunities for progress and development.

This study has led to a reconsideration of some established assumptions in the academic literature, in particular the effectiveness of Agile and Lean practices in the SME domain. The data suggest the greater importance of structured processes within SMEs, emphasizing that the agility of Agile and Lean techniques is highly dependent on the organization. This suggests the possibility of alternative methodologies or a unique integration of existing techniques in SMEs.

In addition, this research also provides insightful insights into employee retention within SMEs. It has shown that retention challenges stem not only from resource constraints but also from broad organizational and cultural factors. This suggests that effective employee retention strategies should extend to addressing these broader aspects.

In the field of knowledge management, this research has added new knowledge to the existing corpus. Although progress has already been made in this field, the research shows that the practical implementation of knowledge management within SMEs still faces major challenges. This implies that existing theories and models may not be fully adequate to meet the unique needs of SMEs.

In response to these identified challenges, this study proposes a concrete action plan specifically tailored to the priorities and needs of SMEs. This action plan provides an integrated and prioritized approach that starts by addressing operational hurdles such as resource management and then extends to more complex aspects such as process methodologies and knowledge management.

This study has provided valuable additions to the existing literature on software development in SMEs. It sheds light on the complexity and dynamics of software development in such organizations and offers renewed insights that contribute to the optimization of processes and outcomes.

The research has contributed to a deeper understanding of the unique challenges of software development in SMEs and presents a useful and realistic plan of action to overcome these challenges. The findings can result in increased efficiency and effectiveness of software development processes in SMEs, leading to higher quality end products and an improved market position. These insights are useful for software development practice within SMEs and may also provide guidance for future research in this field.

## **8. Limitations and Recommendations**

### **8.1. Limitations**

This study has some notable limitations. First of all, despite the confirmation and verification of the results by the participants involved, the reliability of the study is naturally limited by the type of research design, based on subjective interpretations arising from semi-structured conversations. This approach can give rise to bias in interpretation, even when there is confirmation from participants.

Secondly, the size of the participant group, limited to seven individuals from only two organizations, may affect the overall relevance of the conclusions. Despite the profound

insights these conversations provided, the selective bias inherent in the limited group size may limit the ability to extrapolate these results to a larger, more diverse population.

Third, despite dedicated attention to identifying and studying as many relevant aspects as possible, it is unfeasible to isolate and control absolutely every possible variable that could impact software development processes within SMEs. This leaves the possibility that unidentified or unmeasured confounding factors influence the results.

In view of these limitations, this research nevertheless yielded important insights that both expanded the scientific knowledge of software development processes within SMEs and offered practical suggestions for the improvement of these processes. This research serves as a springboard for future studies, with subsequent studies addressing these limitations by using more diverse participant groups, cross-sector comparisons, and more diverse methodological approaches.

## **8.2. Recommendations for Future Research**

Results of this study and the identified limitations suggest a number of directions for future studies. Involving a larger and more diverse group of participants in additional studies can promote a more comprehensive analysis of elements that influence software development processes in small and medium-sized companies. Involvement of people from different layers of the organization and various geographical and industrial areas can support a diversified understanding of the issue.

Future research could also benefit from the integration of quantitative methods, in addition to the existing qualitative framework. This approach can validate the findings of this study and allow an objective measurement of the influence of the identified factors. In addition, quantitative methods can provide additional insights that did not emerge from the qualitative interviews.

In subsequent studies it might be relevant to consider other potential influencing factors. Additional research into the impact of corporate culture, management methods and technology capabilities can contribute to a richer understanding of software development



processes within SMEs. This allows more effective strategies for improvement to be developed.

Based on the findings of this study and the factors that emerged during the validation interview, it is recommended that subsequent studies give specific attention to risk management and strategic development focus. Although these elements were only mentioned by a few interviewees and were not included in the discussion and conclusion, they can be potentially important.

Risk management appeared to be a relevant aspect in the validation interview. Although this element was not covered in detail in this study, initial findings suggest that successful risk management can play a critical role in the outcomes of SMB software development projects. Additional studies can explore this further, such as the techniques SMEs use to identify, evaluate and manage risk and how these practices affect project outcomes.

In addition, strategic development focus was touched upon in the validation interview. The integration of strategic development initiatives into SMB software development processes can have significant impacts on process efficiency and project success. Future research could look at the correlation between strategic development focus and project outcomes, and how this focus is influenced by the specific context and challenges SMEs experience.

In conclusion, while this study has provided valuable insights, there is still much scope for additional research to gain a more complete and detailed understanding of the factors influencing software development processes within SMEs. These recommendations aim to stimulate and guide future research in this direction.

## 9. Bibliography

- Alahyari, H., Gorschek, T., & Berntsson Svensson, R. (2019). An exploratory study of waste in software development organizations using agile or lean approaches: A multiple case study at 14 organizations. *Information and Software Technology*, 105, 78–94. <https://doi.org/10.1016/j.infsof.2018.08.006>
- Almomani, M. A. T., Basri, S., Mahmood, A. K. B., & Bajeh, A. O. (2015). Software Development Practices and Problems in Malaysian Small and Medium Software Enterprises: A Pilot Study. *2015 5th International Conference on IT Convergence and Security (ICITCS)*, 1–5. <https://doi.org/10.1109/ICITCS.2015.7293018>
- al-Tarawneh, M. Y., Abdullah, M. S., & Ali, A. B. M. (2011). A proposed methodology for establishing software process development improvement for small software development firms. *Procedia Computer Science*, 3, 893–897. <https://doi.org/10.1016/j.procs.2010.12.146>
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. <https://doi.org/10.1145/1721654.1721672>
- Bititci, U. S., Garengo, P., Ates, A., & Nudurupati, S. S. (2015). Value of maturity models in performance measurement. *International Journal of Production Research*, 53(10), 3062–3085. <https://doi.org/10.1080/00207543.2014.970709>
- Bjarnason, E., Runeson, P., Borg, M., Unterkalmsteiner, M., Engström, E., Regnell, B., Sabaliauskaite, G., Loconsole, A., Gorschek, T., Feldt, R., Bjarnason, E., Runeson, P., Borg, M., Engström, E., Regnell, B., Unterkalmsteiner, M., Gorschek, T., & Feldt, R. (2014). *Challenges and practices in aligning requirements with verification and validation: a case study of six companies*. 19, 1809–1855. <https://doi.org/10.1007/s10664-013-9263-y>

- Brodny, J., & Tutak, M. (2022). Digitalization of Small and Medium-Sized Enterprises and Economic Growth: Evidence for the EU-27 Countries. *Journal of Open Innovation: Technology, Market, and Complexity*, 8(2), 67. <https://doi.org/10.3390/joitmc8020067>
- Bryman, A. (2016). *Social Research Methods* (5th ed.). Oxford University Press.
- Cardon, M. S., & Stevens, C. E. (2004). Managing human resources in small organizations: What do we know? *Human Resource Management Review*, 14(3), 295–323. <https://doi.org/10.1016/j.hrmr.2004.06.001>
- Cassell, C., & Gillian, S. (2004). *Essential Guide to Qualitative Methods in Organizational Research*. Sage Publications.
- Choras, M., Springer, T., Kozik, R., Lopez, L., Martinez-Fernandez, S., Ram, P., Rodriguez, P., & Franch, X. (2020). Measuring and Improving Agile Processes in a Small-Size Software Development Company. *IEEE Access*, 8, 78452–78466. <https://doi.org/10.1109/ACCESS.2020.2990117>
- Chouki, M., Talea, M., Okar, C., & Chroqui, R. (2020). Barriers to Information Technology Adoption Within Small and Medium Enterprises: A Systematic Literature Review. *International Journal of Innovation and Technology Management*, 17(01). <https://doi.org/10.1142/S0219877020500078>
- Chui, M., Manyka, J., Miremadi, M., Henke, N., Chung, R., Nel, P., & Malhotra, S. (2018). Notes from the AI frontier: Insights from hundreds of use cases. *McKinsey Global Institute*, 2.
- Cragg, P., Caldeira, M., & Ward, J. (2011). Organizational information systems competences in small and medium-sized enterprises. *Information & Management*, 48(8), 353–363. <https://doi.org/10.1016/j.im.2011.08.003>
- Creswell, J. W. (2007). *Qualitative Inquiry and Research Design: Choosing among five approaches* (2nd ed.). Sage Publications, Inc.
- Creswell, J. W., & Creswell, D. J. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.
- Cunningham, W. (1992). *The WyCash portfolio management system, experience report*.

- da Silva, G. C., & Carneiro, G. de F. (2016). *Software Process Improvement in Small and Medium Enterprises: A Systematic Literature Review* (pp. 603–613).  
[https://doi.org/10.1007/978-3-319-32467-8\\_53](https://doi.org/10.1007/978-3-319-32467-8_53)
- Deakins, D., & Freel, M. (2012). *Entrepreneurship and small firms 6e. McGraw Hill.*
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology, 50*(9–10), 833–859.  
<https://doi.org/10.1016/j.infsof.2008.01.006>
- Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems, 15*(2), 200–213.  
<https://doi.org/10.1057/palgrave.ejis.3000605>
- Fowler, M., & Highsmith, J. (2001). *The Agile Manifesto.*  
[www.martinfowler.com/articles/newMethodology.html](http://www.martinfowler.com/articles/newMethodology.html)
- Frishammar, J. (2003). Information use in strategic decision making. *Management Decision, 41*(4), 318–326. <https://doi.org/10.1108/00251740310468090>
- Gioia, D. A., Corley, K. G., & Hamilton, A. L. (2013). Seeking Qualitative Rigor in Inductive Research. *Organizational Research Methods, 16*(1), 15–31.  
<https://doi.org/10.1177/1094428112452151>
- Hajjdiab, H., & Al Shaima Taleb. (2011). Adopting Agile Software Development: Issues and Challenges. *International Journal of Managing Value and Supply Chains, 2*(3), 1–10.  
<https://doi.org/10.5121/ijmvsc.2011.2301>
- Hering, D., Schwartz, T., Boden, A., & Wulf, V. (2015). Integrating Usability-Engineering into the Software Developing Processes of SME: A Case Study of Software Developing SME in Germany. *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering, 121–122.*  
<https://doi.org/10.1109/CHASE.2015.22>
- Holtom, B. C., Mitchell, T. R., Lee, T. W., & Eberly, M. B. (2008). Turnover and Retention Research: A Glance at the Past, a Closer Review of the Present, and a Venture into the

- Future. *The Academy of Management Annals*, 2(1), 231–274.  
<https://doi.org/10.1080/19416520802211552>
- Höst, M. (2000). Using Students as Subjects—A Comparative Study of Students and Professionals in Lead-Time Impact Assessment. *Empirical Software Engineering*, 5(3), 201–214. <https://doi.org/10.1023/A:1026586415054>
- Hron, M., & Obwegeser, N. (2018). Scrum in Practice: an Overview of Scrum Adaptations. *Hawaii International Conference on System Sciences*, 5445–5454.
- Hu, Q., Mason, R., Williams, S. J., & Found, P. (2015). Lean implementation within SMEs: a literature review. *Journal of Manufacturing Technology Management*, 26(7), 980–1012.  
<https://doi.org/10.1108/JMTM-02-2014-0013>
- Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260. <https://doi.org/10.1126/science.aaa8415>
- Kruchten, P. (2013). Contextualizing agile software development. *Journal of Software: Evolution and Process*, 25(4), 351–361. <https://doi.org/10.1002/smr.572>
- Laporte, C. Y., Alexandre, S., & O'Connor, R. V. (2008). A Software Engineering Lifecycle Standard for Very Small Enterprises. In *Software Process Improvement* (pp. 129–141). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-85936-9\\_12](https://doi.org/10.1007/978-3-540-85936-9_12)
- Larman, C., & Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer*, 36(6), 47–56. <https://doi.org/10.1109/MC.2003.1204375>
- Lee, & Xia. (2010). Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility. *MIS Quarterly*, 34(1), 87.  
<https://doi.org/10.2307/20721416>
- Lee, Y., & Cavusgil, S. T. (2006). Enhancing alliance performance: The effects of contractual-based versus relational-based governance. *Journal of Business Research*, 59(8), 896–905. <https://doi.org/10.1016/j.jbusres.2006.03.003>
- Leonardi, P. M., & Meyer, S. R. (2015). Social Media as Social Lubricant. *American Behavioral Scientist*, 59(1), 10–34. <https://doi.org/10.1177/0002764214540509>

- Magnani, G., & Gioia, D. (2023). Using the Gioia Methodology in international business and entrepreneurship research. *International Business Review*, 32(2), 102097. <https://doi.org/10.1016/j.ibusrev.2022.102097>
- Mäkelä, K., Björkman, I., & Ehrnrooth, M. (2010). How do MNCs establish their talent pools? Influences on individuals' likelihood of being labeled as talent. *Journal of World Business*, 45(2), 134–142. <https://doi.org/10.1016/j.jwb.2009.09.020>
- McConnel, S. (2004). *Code complete*. Pearson Education.
- Menezes, J., Gusmão, C., & Moura, H. (2019). Risk factors in software development projects: a systematic literature review. *Software Quality Journal*, 27(3), 1149–1174. <https://doi.org/10.1007/s11219-018-9427-5>
- Moe, N. B., Aurum, A., & Dybå, T. (2012). Challenges of shared decision-making: A multiple case study of agile software development. *Information and Software Technology*, 54(8), 853–865. <https://doi.org/10.1016/j.infsof.2011.11.006>
- Morgan, T., Anokhin, S., Kretinin, A., & Frishammar, J. (2015). The dark side of the entrepreneurial orientation and market orientation interplay: A new product development perspective. *International Small Business Journal: Researching Entrepreneurship*, 33(7), 731–751. <https://doi.org/10.1177/0266242614521054>
- Muller, P., Devnani, S., Ladher, R., Cannings, J., Murphy, E., & Robin, N. (2021). *Annual Report on European SMEs 2020/2021*.
- Niazi, M. (2012). An exploratory study of software process improvement implementation risks. *Journal of Software: Evolution and Process*, 24(8), 877–894. <https://doi.org/10.1002/smr.543>
- Nugroho, A., Visser, J., & Kuipers, T. (2011). An empirical model of technical debt and interest. *Proceedings of the 2nd Workshop on Managing Technical Debt*, 1–8. <https://doi.org/10.1145/1985362.1985364>
- Özkan, D., & Mishra, A. (2019). Agile Project Management Tools: A Brief Comparative View. *Cybernetics and Information Technologies*, 19(4), 17–25. <https://doi.org/10.2478/cait-2019-0033>

- Pino, F. J., García, F., & Piattini, M. (2008). Software process improvement in small and medium software enterprises: a systematic review. *Software Quality Journal*, 16(2), 237–261. <https://doi.org/10.1007/s11219-007-9038-z>
- Raghuvanshi, J., Ghosh, P. K., Agrawal, R., & Gupta, H. (2017). Hierarchical structure for enhancing the innovation in the MSME sector of India. *International Journal of Business Excellence*, 13(2), 181. <https://doi.org/10.1504/IJBEX.2017.086327>
- Rodríguez, P., Markkula, J., Oivo, M., & Turula, K. (2012). Survey on agile and lean usage in finnish software industry. *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 139–148. <https://doi.org/10.1145/2372251.2372275>
- Sambamurthy, Bharadwaj, & Grover. (2003). Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms. *MIS Quarterly*, 27(2), 237. <https://doi.org/10.2307/30036530>
- Schön, E.-M., Thomaschewski, J., & Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces*, 49, 79–91. <https://doi.org/10.1016/j.csi.2016.08.011>
- Serrador, P., & Pinto, J. K. (2015). Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040–1051. <https://doi.org/10.1016/j.ijproman.2015.01.006>
- Šmite, D., Wohlin, C., Gorschek, T., & Feldt, R. (2010). Empirical evidence in global software engineering: a systematic review. *Empirical Software Engineering*, 15(1), 91–118. <https://doi.org/10.1007/s10664-009-9123-y>
- Staples, M., & Niazi, M. (2007). Experiences using systematic review guidelines. *Journal of Systems and Software*, 80(9), 1425–1437. <https://doi.org/10.1016/j.jss.2006.09.046>
- Stettina, C. J., & Hörz, J. (2014). *Agile portfolio management: An empirical perspective on the practice in use*. <https://doi.org/10.1016/j.ijproman.2014.03.008>
- Symon, G., & Cassel, C. (2012). *The Practice of Qualitative Organizational Research: Core Methods and Current Challenges* (1st ed.). SAGE Publications Inc.

- Townsend, K. (2013). *Saturation And Run Off: How Many Interviews Are Required In Qualitative Research?* [Curtin Business School]. Griffith University, Curtin University.
- Tripathi, K., & Agrawal, M. (2014). Competency Based Management In Organizational Context: A Literature Review. *Global Journal of Finance and Management*, 6(4), 349–356.
- Turetken, O., Stojanov, I., & Trienekens, J. J. M. (2017). Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework. *Journal of Software: Evolution and Process*, 29(6). <https://doi.org/10.1002/SMR.1796>
- Wolfswinkel, J. F., Furtmueller, E., & Wilderom, C. P. M. (2013). Using grounded theory as a method for rigorously reviewing literature. *European Journal of Information Systems*, 22(1), 45–55. <https://doi.org/10.1057/ejis.2011.51>
- Yin, R. K. (2009). *Case study research: Designing methods* (5th ed.). Sage Publications.
- Yogaantara, H., & Fajar, A. N. (2022). ANALYSIS OF FACTORS CAUSING INFORMATION SYSTEMS PROJECTS DELAYS IN IT CONSULTING COMPANY. *Journal of Theoretical and Applied Information Technology*, 100(23), 7017–7024.



## Appendix A: Questionnaire I

UNIVERSITY  
OF TWENTE.

### Interview

Investigating factors and mitigation strategies for delays in software development projects in small to medium-sized enterprises.

*Dear interview participant,*

*The purpose of this semi-structured interview is to gather valuable insights into the challenges and strategies associated with delays in software development projects in small and medium-sized enterprises (SMEs). My research aims to identify factors contributing to these delays, compare the results with those of other companies to identify differences in factors and strategies between SMEs at different stages of development.*

*In this semi-structured interview format, I have prepared a series of leading questions to focus our discussion on various aspects of software development processes, such as planning, resource allocation, requirements engineering, and organizational culture. However, I encourage open and honest answers, allowing for flexibility and the opportunity to explore additional topics based on your experiences and perspectives.*

*Your input as a software developer or project manager is crucial for this study. While the following questions will guide our discussion, feel free to share other relevant insights related to the challenges and strategies used within your organization.*

*Thank you for your time and valuable contribution to this research. I look forward to a productive and insightful interview.*

*Respectfully*

*Robin Lagerburg*

**1. How do you approach the initial planning and prioritization phase of your software development projects while ensuring effective resource allocation?**

- A. What tools or methodologies do you use in the planning process to align expectations and engage stakeholders?
- B. How do you prioritize projects and manage stakeholder expectations?

**2. How do you integrate a systematic approach to requirements engineering and align it with verification and validation processes within your organization?**

- A. What methodologies or tools do you think are most effective in managing software requirements?
- B. How do you ensure effective knowledge sharing within your development teams?

**3. How do you ensure that your software development process is well structured, effective, and continuously improving within your organization?**

- A. What techniques or methodologies have you implemented to improve the structure and effectiveness of your development process?
- B. How do you foster a culture of continuous improvement and measure the success of these improvements?

**4. How have you integrated agile and lean methodologies into your software development process, while addressing resistance to change and fostering a supportive organizational culture?**

- A. What were the main factors that influenced your decision to use these methods?
- B. What initiatives have you taken to promote cultural shifts and provide management support?

**5. How do you deal with limited resources, gaps of expertise and risks within your organization while applying best practices and standards?**

- A. What strategies do you use to overcome these limitations and mitigate risk?
- B. How do you identify and prioritize areas that require additional resources or expertise?

**6. How do you deal with code complexity, maintain quality, and optimize your software development processes?**

- A. What tools or techniques do you use to manage code complexity and maintain quality?
- B. How do you measure the effectiveness of these optimization efforts?

**7. How do you integrate usability engineering and customer feedback into your software development process?**

- A. What steps have you taken to ensure usability is considered during the development process?
- B. How do you collect and process customer or end-user feedback and measure the success of your usability engineering efforts?

**8. How do you maintain control over the different stages of the software development lifecycle and reduce project delays due to poor planning, limited resources, or ineffective communication and collaboration?**

- A. What tools or processes do you use to monitor and manage the development lifecycle and foster communication and collaboration within your development teams?
- B. How do you address any issues that arise during the life cycle to ensure the success of the project?

**9. How do you deal with ambiguities arising from informal communication and ad hoc decision-making within your organization?**

- A. What processes have you implemented to ensure clear communication and structured decision-making?
- B. How do you measure the impact of these processes on your software development projects?

**10. How do you stay on top of the latest trends, technologies, and best practices in software development while investing in infrastructure, training, and staffing?**

- A. What sources, channels, or approaches do you rely on to stay informed, make informed investment decisions, and maintain a competitive advantage?
- B. Can you give examples of successful investments or application of new trends or technologies within your organization?

## Appendix B: Questionnaire II

### General Information

1. How would you describe the size of your company and what is your role within the company?
2. How many people work on your software development team?

### Software Development Process Management

3. How does your organization manage its software development processes?
4. What are the biggest challenges you face when managing software development processes?
  1. Requirements engineering?
  2. Monitoring?
5. What strategies has your organization implemented to address these challenges?

### Implementation of Agile and Lean Methodologies

6. Which Agile and/or Lean methodologies does your organization use in the software development processes?
7. What was the process of implementing these methodologies like?
8. What are the biggest challenges you face in implementing and handling these methodologies?

### Continuous improvement and learning

9. How does your organization promote continuous improvement and learning within the team?
10. How is knowledge shared within your team and your organization?
11. What tools or processes do you use to measure progress and improvement?

### Resource and Capacity Management

12. How does your organization manage capacity and resource allocation within software development projects?
13. How does your organization deal with situations of understaffing or overload?
14. What strategies has your organization implemented to maintain a balance between workload and available capacity?

### Team Dynamics and Communication

15. How would you describe the team dynamics within your software development teams?
16. What is your organization doing to promote effective communication within and between teams?
17. How are conflicts or disagreements within the team addressed and resolved?

### Appendix C: Exemplary Citations from Interviews

<p><i>Dealing with limited resources, expertise gaps and risks within our organization is a complex challenge. It's a constant reminder that we don't always have enough people for all the work.</i></p>	<p>Limited Resources</p>
<p><i>A challenge here is that some tasks have been performed by only one person, while ideally we want to achieve a team effort. In the past, retrospectives took place during, for example, the Tuesday plan meeting, but we recognize that we should organize these types of evaluations more often.</i></p>	
<p><i>Dealing with resource constraints, expertise gaps and risk within the organization while applying best practices and standards can be challenging. Since we currently don't have dedicated testers, we try to divide the responsibility for testing between the development team and encourage everyone to work carefully and meticulously.</i></p>	
<p><i>Our organization struggles to motivate team members for specific roles and methodologies. For the past two years, we've been trying to understand how best to work with the people available. This has led to processes that are mainly based on consultation and practical experience rather than on theoretical methodologies. As a result, we often work ad hoc, which can lead to errors or inefficiencies.</i></p>	<p>Lack of Formal Processes</p>
<p><i>An additional problem is that the necessary knowledge to work in a certain way is often lacking within the team. We try to exploit the benefits of an advisory approach, adapting to the dynamics and capabilities of our team. However, this sometimes leads to ambiguities and challenges that can hinder our processes. That is why it remains necessary to work on improving our communication and reducing ambiguities within the team, although the results are not always optimal.</i></p>	
<p><i>However, there are some aspects that can still be improved. We currently don't have the right tools to track KPIs based on project hours, and we lack structured retrospectives to thoroughly evaluate and improve our development process. It is important that we make time to do this reflection and thus continuously improve our software development process.</i></p>	
<p><i>In addition, performing root cause analysis in identifying errors in our processes and systems would help us better understand and correct the underlying problems. By analyzing the causes of errors and making improvements, we can increase the overall quality and reliability of our software while reducing risk.</i></p>	
<p><i>We deal with limited resources and risks in our organization by sharing as much knowledge as possible. However, transferring knowledge between team members can be difficult, especially when people leave the company. Moreover, the preparation of documentation proves difficult and does not always receive sufficient attention, especially due to understaffing. The limited availability of staff makes it challenging to find the time and resources to create and maintain documentation.</i></p>	<p>Knowledge Management Issues</p>

<p><i>One of them is knowledge sharing. At the moment we are not very active in stimulating knowledge sharing. We rely on developers to support each other informally, which is not ideal. We should consider more structured methods to promote knowledge sharing. We also do code reviews, but the process is not always consistent or thorough. To further improve code quality and best practice sharing, we need to pay more attention to regular and thorough code reviews. Our current testing and code review process can be further optimized to improve efficiency and effectiveness. We must be prepared to critically evaluate our processes and make adjustments where necessary. While we try to maintain code standards and best practices, this is not always the case. The risk of knowledge loss when an employee leaves remains and we need to work towards greater consistency in our code quality. Finally, we have not given enough time and priority to prepare extensive documentation. Improving the documentation would help us better retain and share knowledge within the team.</i></p>	
<p><i>Although we as an organization invest in infrastructure, training and personnel, we recognize that there is room for improvement in knowledge sharing within teams. We currently participate in trade shows and events to stay up to date with the industry, but taking targeted courses and training can further contribute to the development of our employees and the quality of our software development processes.</i></p>	
<p><i>We try to guarantee as much as possible by documenting and archiving everything. However, customization remains an exciting field. For example, when one person is responsible for the entire development and all knowledge lies with that one person. Sharing knowledge remains a difficult challenge. A lot of our dev-ops knowledge also lies with one person, and this is an area we need to work on.</i></p>	
<p><i>When identifying and prioritizing areas that require additional resources or expertise, we look at the knowledge and interests of our team members. Based on this, we can determine who can best perform which tasks. A disadvantage of this approach is that a lot of knowledge can remain with one person, which can pose a risk to the continuity of projects.</i></p> <p><i>The challenge here is compounded by the fact that we often lose good developers to larger companies, which can offer better salary packages. This increases the importance of facilitating effective knowledge transfer within our team.</i></p> <p><i>To combat this, we organize, for example, debriefings on Friday afternoon. Although these do not provide much knowledge transfer at the moment, they do help in the learning process of new team members such as, who has to ask questions and thus gain more insight into different aspects of the project.</i></p> <p><i>In addition, we are always looking for new employees with additional expertise to strengthen the team and broaden the knowledge base.</i></p>	<p>High Employee Turnover</p>
<p><i>We use a review slip on our board. Here we place pull requests that need to be checked for code quality and style, using tools like Bitbucket. However, it is challenging to keep all projects on track, especially when we are working on multiple projects at the same time. This complexity, combined with our understaffing, sometimes makes it</i></p>	<p>Inadequate Project Management</p>

<p><i>difficult to conduct thorough assessments.</i></p> <p><i>While feedback is sometimes provided on Fridays over weekends, this is not the ideal time for extended discussions. Instead, points for improvement and suggestions are more often included in the planning meeting on Tuesday. It's an ongoing balancing act to ensure the quality of our work as we try to move multiple projects forward.</i></p>	
<p><i>As a team, we plan our work every morning. We try to plan based on priority, but often the work that comes our way is older and difficult to schedule. It is difficult to estimate accurately how long we will be working on something. Usually we are pretty close to it, but often it turns out differently. We avoid estimating hours in detail because this is often unrealistic and takes more time than it saves. Instead, we focus on prioritizing tasks and estimating the duration of the work as accurately as possible.</i></p>	
<p><i>In the sales phase, a number of basic requirements are already laid down that relate to the fundamental functionalities of the project. Nevertheless, it is sometimes difficult to communicate clearly with customers about their expectations and requirements, which can lead to misunderstandings and complications during the execution of the project.</i></p> <p><i>I then go deeper into the specific and technical requirements, by investigating, among other things, whether the project fits within the standard and which techniques will be used. This involves looking at whether this is a completely new technique or whether we build on existing solutions within LogicTrade.</i></p> <p><i>In addition to mapping out the requirements, we sometimes also make a functional and technical design. However, we notice that sometimes this is not enough to capture all aspects of the project. It would be nice if there was more documentation to fall back on.</i></p>	<p>Poor Communication</p>
<p><i>However, we recognize that there is room for improvement in this area as we currently only have one person (product owner) with whom we can spar on a deeper level. Adding more people with similar expertise could help address ambiguities even more effectively and provide a broader basis for decision making and idea sharing.</i></p>	
<p><i>To keep abreast of the latest trends, technologies and best practices in software development, we try to learn from consultants' experiences and share knowledge between teams. However, the communication between the teams can be improved to exchange knowledge more effectively and to learn from each other.</i></p>	
<p><i>Dealing with code complexity, maintaining quality and optimizing software development processes within our organization is a challenge given the size of the product in relation to the size of the company. The product has become so large that it is sometimes difficult to maintain. We often have to deal with old code that needs to be updated, but there is not always time for that.</i></p> <p><i>We've already taken steps to simplify the code, making testing and setting up the system easier. However, we recognize that this is an intensive and technical process and that it takes time and effort to apply all aspects of the product. This is not easy to achieve.</i></p>	<p>Technical Debt Management</p>

<p><i>First of all, we strive to keep the software standardized and maintainable as much as possible. Nevertheless, time constraints can sometimes lead to less than optimal development choices, such as adding additional licensing options that reduce the maintainability of the software.</i></p>	
--	--