Master of Science in Software Engineering
June 2024

Harmonizing Agile transformation: Navigating software startup success through a hybrid journey of Scrum and Kanban methods

a Case Study analysis

Omid Mojabi

Faculty of Engineering, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full-time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author:
Omid Mojabi
E-mail: ommo22@student.bth.se

University advisor:
Dr. Mikael Svahnberg
Department of Software Engineering

Faculty of Engineering
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

This research identifies shared strengths of Scrum and Kanban, such as iterative and incremental work and effective workload management, which are particularly beneficial in startup contexts characterized by high uncertainty. By systematically comparing the fundamental elements of both methodologies, the study delineates key dimensions and components that inform the development of a hybrid approach. This study aimed to explore which components of Kanban and Scrum are best suited for navigating uncertainty and ambiguity in a fast-paced environment. To achieve this objective, our first research question delves into the productive attributes of both Scrum and Kanban methods, laying the groundwork for our investigation. Additionally, we sought to identify the key factors driving adaptations in the application of these methods within software startups, as well as pinpoint the highlighted concepts and elements that can be effectively utilized. This formed the basis of our second research question, with the ultimate aim of developing a hybrid model that integrates the most beneficial aspects of both methodologies, thus addressing the challenges posed by uncertainty and rapid change in the initial phase of the study. We employed a mixed-methods approach, utilizing a literature review, survey, and case study analysis. The theoretical foundation was established through a comprehensive review of existing literature on Scrum and Kanban, supported by the construction of a comparative mind map. Survey data was collected from industry professionals to gather insights into the practical application of these methodologies. Additionally, a case study was conducted to provide a deeper understanding of the challenges and successes in implementing agile practices within a startup environment. Through our research, we identified key dimensions and components of Scrum and Kanban relevant to software startups, including values, principles/practices, team structure, events/cadences, and artifacts. By analyzing survey responses and conducting in-depth interviews, we gained valuable insights into the strengths and limitations of each methodology. Moreover, our case study provided real-world examples of the challenges faced and strategies employed in a startup setting. The findings suggest that while Scrum and Kanban offer distinct strengths, a hybrid approach integrating both methodologies can better address the dynamic needs of software startups. The proposed hybrid model emphasizes flexibility and adaptability, allowing startups to tailor practices to their specific context while leveraging the strengths of Scrum and Kanban. Overall, this research contributes to a deeper understanding of how startups can navigate agile transformation and achieve success in a competitive market.

Keywords: Agile, Startup, Scrum, Kanban, Hybrid, Mind-Mapping, GTM, Grounded Theory Methodology

Acknowledgments

This work is dedicated to my wife **Azadeh** for her unwavering love, support, and understanding throughout this journey.

I acknowledge the crucial role of **Blekinge Tekniska Hogskola** and its affiliated departments in providing the necessary support and assistance for conducting this research. I sincerely thank all participants whose generous contributions enriched this research. Their willingness to share experiences and insights proved invaluable to the study.

I would also like to express my gratitude to **Dr. Mikael Svahnberg** and **Dr. Daniel Mendez Fernandez** for their invaluable support and contributions to this research.

Furthermore, I deeply appreciate the valuable contributions of colleagues and friends who provided constructive feedback, suggestions, and encouragement throughout the research process. Additionally, I am grateful for the meticulous review and insightful feedback provided by the reviewers, which greatly enhanced the quality and rigor of this study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Software startups are new temporary organizations that create high-tech innovative products and have no prior operating history [65], [12]. They are significant game players in the software industry landscape and differentiated and characterized by their novelty, employing rapid cutting-edge technological advancements [31], [60], [70] and reliance on external investments in uncertain markets [17], [77]. They have a significant role in driving innovation, economic growth, and job creation [6]. Despite the crucial role of innovative products and the necessity for external financing, the startup context suffers from a notable absence of comprehensive support for established software engineering practices [39], [51], [84]. The existing gap is compounded by the absence of development methods addressing the distinctive features of startups [37], including their flexible [74] and dynamic nature, emphasis on speed [60], and reliance on customer feedback [75]. Software startups often exhibit a pragmatic approach to methodology adoption, tailoring practices to their specific needs in the early stages of their rising period [37], [75]. However, the factors influencing the selection or adaption of agile methods in this context remain poorly studied. Challenges such as lack of competency in business administration, uncertainty in choosing technologies, and inadequate startup support in development methods further hinder the seamless integration of agile practices [35].

Existing software engineering methods, not attuned to the business intricacies of startups [37], [58], may fall short in addressing their unique requirements even though some of these methods (e.g. [31], [49], [69], and [85]) claim they are the best chance to achieve the company's goals and missions. The lack of literature on the application and benefits of agile development practices specifically within software startups impedes a comprehensive understanding of the dynamics at play [51]. Hence, this research aimed to investigate the utilization of Scrum and Kanban methods in software startups. They are the most popular [2] and powerful [46] agile methods that are introduced and used in different contexts and problems and many startups experienced them in their growth journey and product market fit stages. By identifying primary studies that delineate the application of Scrum and Kanban methods in startup settings, the study seeks to contribute insights into optimizing workflow and resource allocation, and ultimately understanding the factors influencing the success or failure of software startup companies.

Agile software development (ASD) has brought a variety of benefits to the software industry including enhanced "team capability", increased "customer involvement" [73],

and the delivery of high-quality products to customers [36], among others. While the entire software industry has embraced the agile development since the start of the 21st century, startups have encountered an array of obstacles. The Agile and Lean movement in software development emphasizes collaboration, customer-centricity, and adaptability [75]. Lean principles focus on eliminating waste and optimizing processes [8]. Agile principles values iterative development, flexibility and responding to change [9]. Research on agility and software processes is focused widely on frameworks, methods, and practices [43]. Popular Agile methods like Scrum and Kanban offer efficient frameworks for delivering high-quality software. Scrum, founded on empiricism and lean principles, emphasizes iterative and incremental development [68]. Kanban, focusing on flow system design and continuous improvement, promotes collaboration and transparency [8]. Both methods prioritize customer satisfaction and value delivery, making them suitable approaches for software startups.

According to [4], [37], and [75], software startups encounter unique hurdles when integrating agile development methodologies. Evaluating their effectiveness in meeting objectives is crucial, as each method offers distinct aspects and constraints essential for their successful utilization. Indeed, discrepancies arising from customized or absent features within these methodologies can hinder consistent implementation. Although software engineering methods attempt to respond to the business community's demand for lighter weight along with quicker and nimbler software development processes [3], Many of the existing software engineering methods and approaches are not intended for software startups [37] and poor support has been observed in the software startup context [35], [77]. As their time and resources are limited, and even one failure in achieving goals can put software startups out of the business mainstream, they need effective practices to face their unique challenges [32]. As a result, startups prefer to tailor methodologies that match their situation by dropping some of the features in the existing methodology and adding new practices from other methods [75].

While hybrid methods have been commonly generated and utilized (for instance see [36], [48], [16], and [66]), to avoid the consequences of customizing agile methods in software startups, understanding their aspects and capabilities is essential in different contexts and forming an extendable instance of mixing methods seems impossible.

This research is focused on transferable elements from Scrum and Kanban methods to a flexible dynamism across their concepts and practices to increase the success and sustainability of software startups. We propose a new hybrid on essential and adaptive components of these methods to form a journey for applying Scrum and Kanban approaches within software startups to drive them to success.

The remainder of this thesis is organized as follows. In the Background chapter, we delve into the context and background of previous studies relevant to Software startups, agile and lean methods, and especially Scrum and Kanban. The Related Work chapter presents a compilation of research by other scholars in the field. In the Research Method chapter, the utilized research methods are elaborated upon, spanning literature review, online surveys, and case study interviews. This section delves

into an analysis and reporting of survey results, covering challenges, techniques, advantages, and limitations. In the Results and Analysis chapter, we thoroughly examine the survey and interview findings in conjunction with our results, emphasizing any limitations encountered. Lastly, in Discussion and Conclusions and Future Work chapters, we discuss findings and outlined paths for future research.

1.1 Ethical, Societal and Sustainability Aspects

In this thesis research, addressing ethical considerations, societal impacts, and sustainability aspects is crucial for ensuring a comprehensive understanding of the research's broader implications. Ethical considerations include data privacy, confidentiality, and fairness in the decision-making process, and ensuring responsible research conduct. The societal impact of the study is evident in its potential to empower startups to develop more effective software solutions, indirectly benefiting end-users and society at large. Additionally, the proposed hybrid model for software startups holds promise in promoting sustainable software development practices, such as efficient resource allocation and reduced environmental impact. However, given the study's primary focus on methodological or technical aspects of agile methodologies in startups, ethical, societal, or sustainability considerations may not directly deal with the research objectives, warranting a clear rationale for their exclusion from this report.

Chapter 2

Background

This chapter introduces various aspects related to software development in the startup ecosystem, agile/lean movement in software development, Scrum concepts, elements, and background, as well as Kanban concepts, elements, and background. It also highlights the stages of progression typically observed in software startups, from the pre-startup phase to the post-startup phase.

2.1 Software Development in Startup Ecosystem

Software startups are often seen as distinct from other types of software organizations in several ways. According to Blank (2020) [12] and Blank & Dorf (2020) [13] studies, startups aim to grow incrementally while seeking a scalable and profitable business model. This distinction is significant, as it sets apart startups from small businesses, which may not have growth intentions and consequently lack a scalable business model. On the other hand, there is no consensus on the definition of software startup, even though many share an understanding that software startups deal with uncertain conditions, grow quickly, develop innovative products, and aim for scalability [77]. Hence, in characterizing them, we rely on indicators in our research for which there is an implicit agreement among researchers in the research literature.

According to [22], The evolution of companies from inception to maturity is split into three phases – Startup, Stabilization, and Growth. The startup phase is defined as the period between product conception and the first sale. The Stabilization begins when the first customer takes delivery of the product. It ends when the product is stable enough to be commissioned for a new customer without causing any overhead on product development. The growth phase begins when the product can be commissioned for a new customer without creating any overhead on the development team. It ends when market size, share, and growth rate have been established and all business processes necessary to support product development and sales are in place. Similarly, a newer series of studies conducted by Nguyen-Duc et al. [54–56], divides software startups typically progress through three distinct stages: the Pre-startup stage, the Startup stage, and the Post-startup stage. During the Pre-startup period, ideas are formulated and require validation, while startups actively seek financial and human resources. In the Startup stage, prototypes are developed and tested, and startups work to align problem-solving with solutions. The final phase, the Post-startup stage, signifies the attainment of product-market fit, where startups generate revenue, although they may not have reached the break-even point. Accord-

ing to [10], [33] and [50], a large group of startups apply market-driven requirement engineering or even agile practices to discover and validate ideas for innovative products but these applications are often rudimentary and lack alignment with other knowledge areas which causes unwanted technical debt, poor product quality, and wasted resources on building irrelevant features.

Although software startups typically aim to enhance product quality and optimize resource allocation [4] through customer feedback [70], there remains inadequate support for software engineering within the startup environment [39] and there is a growing demand for graduate students who possess the skill set to navigate trade-offs effectively, comprehend various facets of software development, and seamlessly integrate software management with development processes [25]. Another challenge faced by software startups is their company size. Securing a licensing deal with a large corporation can be especially challenging for small, early-stage startups. Owners of established businesses often hesitate to commit to such deals because the royalty income may take years to materialize and could be relatively modest due to the typical size of startups [78].

2.2 Agile/Lean Movement in Software Development

Nowadays, a variety of concepts and practices have had an effect on our workplaces and ways of working dramatically in the digital age and one of the most important reasons is the Agile and Lean approaches in software development. The word “Lean” generally refers to a set of knowledge called more specifically “Lean Manufacturing” or “Lean Production” as it uses less human effort, manufacturing space and equipment, investment, and time in comparison to the mass production [79]. It was developed in Japan in the 1950s and 1960s by Taiichi Ohno in Toyota company [8] and transformed many traditional concepts through product and service development. The seven Lean principles include eliminating waste, building quality, creating knowledge, deferring commitment, delivering fast, respecting people, and optimizing the whole [62] are fundamentals of lean thinking.

The Agile Manifesto was made in February 2001 by seventeen independent-minded software practitioners. It is a document that identifies 4 key core values and 12 principles [9] that its authors believe software developers should use to conduct their work. Agile values refer to the importance of collaboration, a customer-centric approach, responding to feedback and upcoming changes, and the quality of software. These methods fascinated many developers because they were explicitly created in opposition to classical models like waterfall [14] and helped developers address problems that traditional methods could not solve. Scrum and Kanban are two of the most popular agile development methods. The retention of these top positions follows extensive research conducted in the "HELENA" worldwide project [44] between 2016 and 2018. The results clearly show that these methods influence the market more than other methods and are used by different software development teams and companies globally. This fact suggests that software startups likely employ Scrum and Kanban concepts and artifacts, either informally or conceptually, based on their

own understanding and expectations of their way of working. In the 16th "State of Agile" Report, conducted in 2023, Scrum was reported as the way of working by more than three-quarters of respondents, with Kanban taking second place. The third place on the list belongs to Scrumban, a typical combination of Scrum and Kanban methods [2].

2.3 Scrum Concepts, Elements, and Background

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems [68]. The framework was officially first introduced to the public at the OOPSLA'95 conference by Jeff Sutherland and Ken Schwaber. They introduced Scrum to multiple companies like Motorola, Fidelity, and GE Medical, and after seeing great results and creating a community, they participated in the creation of the Agile Manifesto in February 2001 [47]. Figure 2.1 shows the framework at a glance.

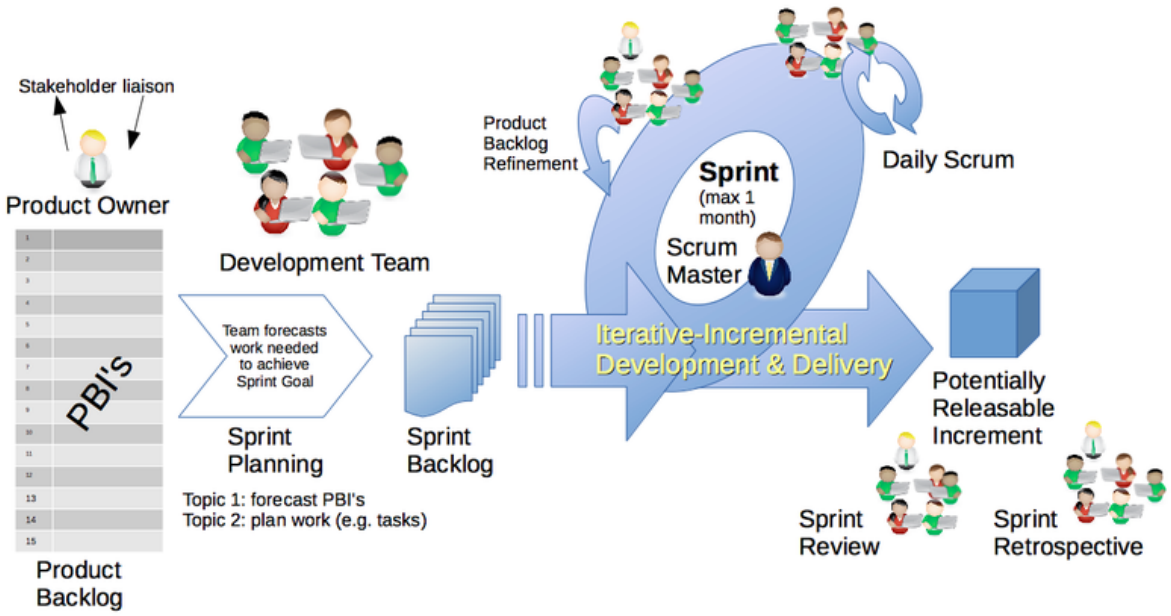

Figure 2.1: Scrum Framework at a glance. Figure produced by Ian Mitchell [CC BY (https://creativecommons.org/licenses/by-sa/4.0/deed.en)], via Wikimedia Commons.

2.3.1 Terminology, Philosophy, and Events

Scrum consists of Scrum teams, events, artifacts and rules. The rules are essential to bind teams, events and artifacts together during the project. They also provide an agreeable structure for resolving conflicts within a project [46]. Scrum is based on empiricism and lean thinking. Empiricism asserts that knowledge comes from experience and making decisions based on what is observed. Lean thinking reduces waste

and focuses on the essentials. Scrum employs an iterative, incremental approach to optimize predictability and control risk. Scrum engages groups of people who collectively have all the skills and expertise to do the work and share or acquire such skills as needed. Scrum combines four formal events (including Daily Scrum, Sprint Planning, Sprint Review, and Sprint Retrospective) for inspection and adaptation within a containing event, the Sprint. These events implement the empirical Scrum pillars of transparency, inspection, and adaptation. Scrum is designed around self-managing Scrum teams. Each Scrum team consists of a Scrum Master, a Product Owner, and the rest developers [68].

2.3.2 Artifacts, Values, and Principles

The Scrum framework relies on five fundamental values and six basic principles, guidelines that must be followed throughout every project. Must is the keyword, as Scrum followers are adamant that each principle remains intact and adhered to, lest the team lose focus or the project suffer any setbacks. The six principles include Control over the empirical process, Self-organization (Self-Management), Collaboration, Value-based prioritization, Time-boxing, and Iterative development [27]. Successful use of Scrum depends on people becoming more proficient in living five values including Commitment, Focus, Openness, Respect, and Courage [68]. Scrum's artifacts represent work or value. They are designed to maximize the transparency of key information. According to the Scrum Guide, the artifacts include Product Backlog, Sprint Backlog, and Increment. The Product Backlog is an ordered list of things that are needed to improve the product. It is the single source of work undertaken by the Scrum Team. The Sprint Backlog is a plan for developers to accomplish during the Sprint to achieve the Sprint Goal. Increment is a concrete stepping stone toward the Product Goal. Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together. To provide value, the Increment must be usable. Figure 2.2 shows Scrum Values and Scrum Pillars at a glance.

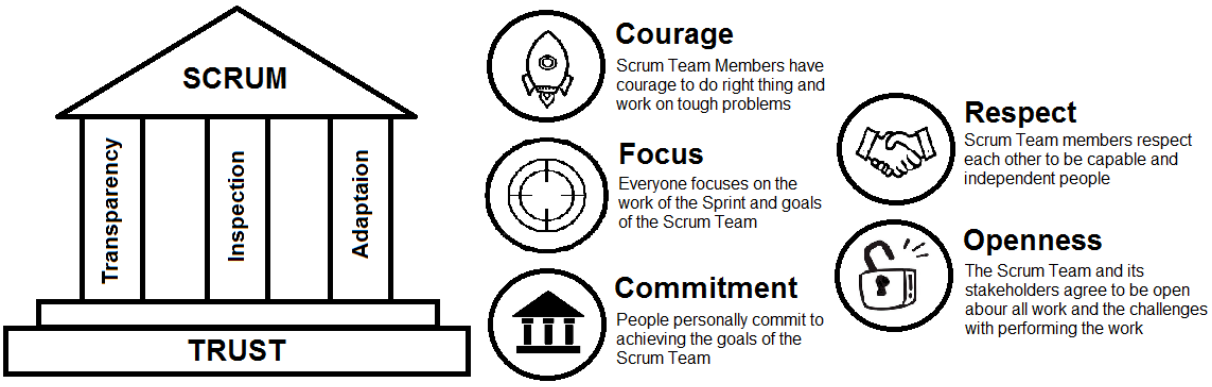

Figure 2.2: Scrum Values at a glance adapted from [1].

2.4 Kanban Concepts, Elements and Background

Kanban is a software development methodology that emphasizes "just-in-time" delivery. The main focus of Kanban is to accurately state what work needs to be done, and when it needs to be done. It does so by prioritizing tasks and defining workflow as well as lead time to delivery [7], [46]. This method is concerned with the design, management, and improvement of flow systems for knowledge work. These are systems in which intangible work items move through different stages, eventually resulting in value to their customers [8].

2.4.1 Kanban Principles and Practices

Kanban promotes continuous collaboration and encourages active, ongoing learning and improvement by defining the best possible team workflow. Customer satisfaction is accomplished through a disciplined approach to work, which emphasizes transparency, communication, collaboration, and clarity. This is where Kanban comes in. Kanban is a highly visual workflow management method that provides organizations with the visibility, metrics, and focus they need to better manage work and people [8]. The Kanban method consists of principles, practices, Kanban boards (physical or electronic), and Kanban cards. The basic principles of Kanban are limiting Work in Progress (WIP), pulling value through the development process, making the development process visible, increasing throughput, using a fixed backlog and embedding quality [46].

Kanban prioritizes the completion of tasks according to developers' skill sets, accommodating varying work speeds and expertise levels. It emphasizes adding value by implementing essential project components first, avoiding unnecessary features or excessive specifications beyond coding capacity, and maintaining balance in coding, testing, and deployment efforts. It eliminates waste at every stage and is well-suited for software engineering [7].

2.4.2 Cadences, Values, and Artifacts

Kanban uses physical or digital boards to represent a team's or organization's unique process. Work is represented by cards on the board, which move from left to right through the steps in the process [8]. A Kanban board (See Figure 2.3) is a tool to implement the method and usually is divided into several columns that represent a process's stages. Each Kanban card is a work item, moving through the columns on the Kanban board. Cards contain information about the work item.

The Kanban method is motivated by the attitude that all persons contributing to a common company must be respected. This mutual respect is essential if the company is to succeed and pay off. Without this mutual respect, the company itself is not promising [8]. Using Kanban, teams were able to communicate more effectively as this simple tool provided instance information on what needed to be worked on and when it is needed [46].

Figure 2.3: Kanban Board Example. Figure produced by Jennifer Falco [CC BY (https://creativecommons.org/licenses/by/4.0/deed.en)], via Wikimedia Commons.

Cadence is a concept in Kanban which determines the rhythm of a type of event [64]. They are essential for obtaining the customer's feedback and keeping them involved in the work of the team [24] and can be considered as the implementation of feedback loops in this method [8], [23]. As shown in Figure 2.4, Kanban method depicts a scheme of seven cadences, suggesting review frequencies in a typical enterprise or multi-service context. These include Strategy Review, which selects services and defines "fit for purpose" in response to changing external environments; Operations Review, which balances services and deploys resources to maximize value delivery aligned with customer expectations; Risk Review, which addresses risks to service delivery, such as through blocker clustering; and Service Delivery Review, which examines and improves the effectiveness of a single service. Additionally, the Replenishment Meeting moves items into the system and oversees future options; the Kanban Meeting, usually a daily stand-up, focuses on coordination, planning, and unblocking issues; and the Delivery Planning Meeting monitors and plans deliveries to customers. Implementing these cadences doesn't necessarily add seven new meetings but integrates their agendas into existing ones, adapting them to fulfill their goals. On a smaller scale, a single meeting may cover multiple cadences. The feedback loops in the cadence network diagram illustrate the information flow and change requests between reviews, facilitating decision-making at each level [8].

Anderson and Carmichael [8] stated nine core values for Kanban Method in Software engineering including Transparency, Balance, Collaboration, Customer Focus, Flow, Leadership, Understanding, Agreement and Respect. These values embody the driving forces behind Kanban's endeavors to enhance services delivered by collaborative teams. Transparency entails a willingness to share information that enriches the flow of business value. Balance underscores the various aspects and perspectives within the context. Collaboration involves working collectively in teams to refine the work

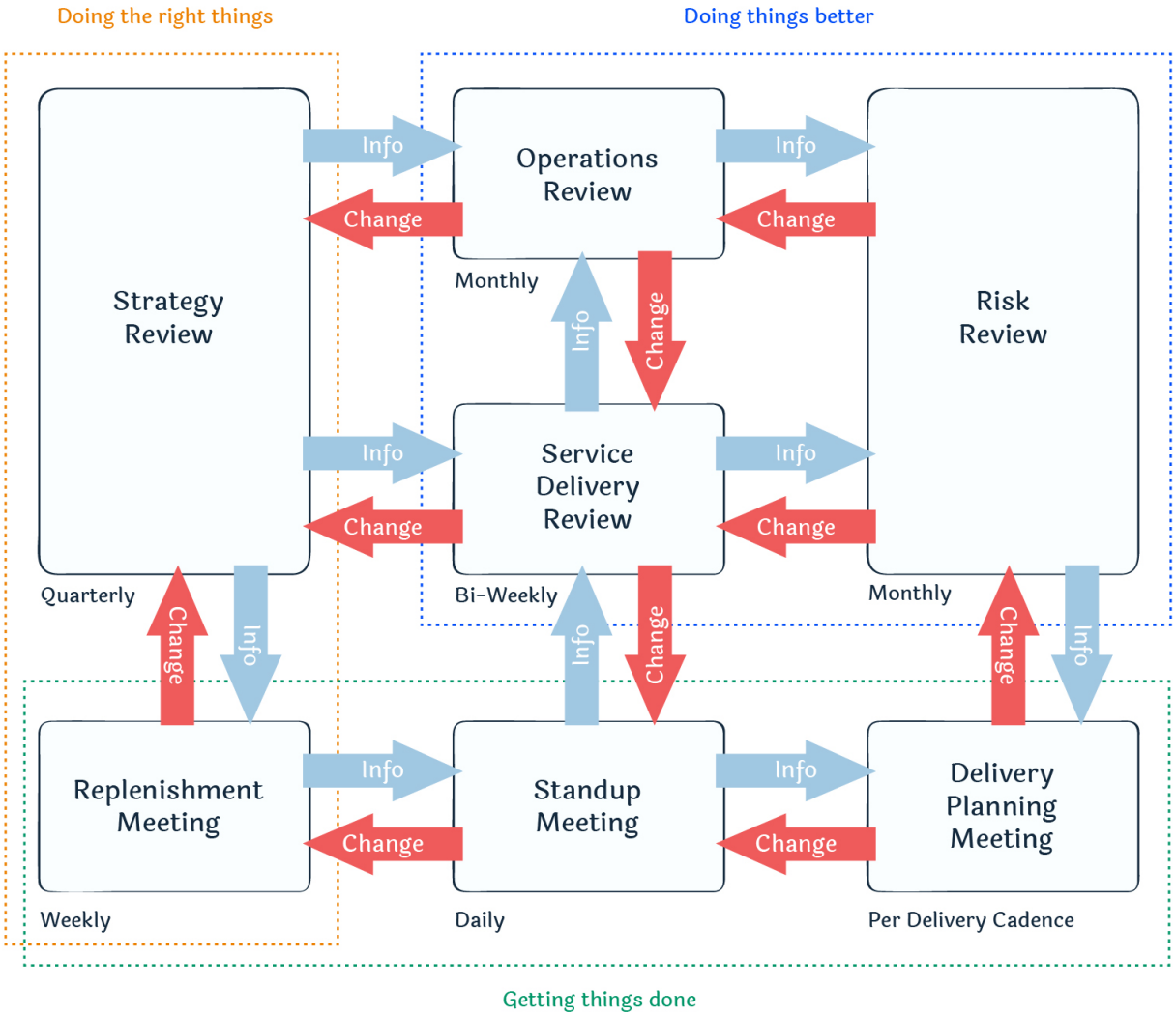

Figure 2.4: Kanban Cadences. Figure reproduced by Sonya Siderova [Source: https://getnave.com/blog/kanban-roadmap/ and adapted from [7].

process. Adopting a Kanban discipline necessitates customer focus, ensuring that both internal and external clients recognize value through this approach. Recognizing work as a continuous or episodic flow of value is crucial. Perceiving this flow is a fundamental starting point in utilizing Kanban. Leadership involves inspiring others to act within the given context and to reflect upon it. As Kanban is a method of improvement, self-awareness and understanding are foundational elements. Agreement entails a commitment to progress collectively toward objectives, while Respect involves valuing, comprehending, and demonstrating consideration for individuals [8].

2.5 Comparison between Scrum and Kanban

While Kanban and Scrum share some similarities in their operational mechanisms, they diverge significantly in their foundational concepts and methodological boundaries. This comparison is essential for our study since we we want to base our model

on the performance of agile methods. According to Kniberg and Skarin [40], Kanban and Scrum exhibit similarities in several aspects including embracing Lean and Agile principles, breaking down work into manageable units, fostering self-organized teams, prioritizing the early and frequent delivery of releasable software, swiftly adapting to changes, enforcing WIP limits, employing pull-scheduling, and emphasizing transparency. Moreover, beyond the elements delineated in [5] and [38], further similarities between Kanban and Scrum have surfaced, including their efficiency compared to the Waterfall method. Both methodologies incorporate feedback and improvement mechanisms, as well as provide clear project scope delineation. Additionally, both methodologies prioritize project value, the attainment of deliverables, and do not obligate specific technical practices.

Table 2.1: Important differences between Kanban and Scrum approaches

Criteria	Scrum	Kanban
Team Structure	A cross-functional team, including the product owner and Scrum master, with fewer than 10 members [5], [46], [68]	A flexible team, ranging from fewer than 5 to more than 11 members [5], [46] with the addition of a Service Request Manager and a Service Delivery Manager [8]
Delivery Frequency	Iterative, Incremental [68]	Continuous or Episodic [8]
Work In Progress	The sprints commence with the definition of all user stories to be addressed, aiming to minimize WIP [83]. Team should commit on delivering sprints on time [5]	optimizes the workflow by leveraging the capacity of each workstation, managing the flow of cards with the objective of keeping WIP as limited as feasible [5]
Requirements Prioritization	is done on the basis of the length of the sprint [5], [68]	is done continuously which can be daily/hourly [5]
Performance Metrics	Velocity [83], avoid cut the lead time [5]	reduces lead time by avoiding multitasking and limiting WIP [5]
Responding to the Change	is fixed during sprints [83], sprints can be canceled if their goals expired [68]	is flexible and can take many shapes according to different contexts [83]

Kniberg and Skarin outlined several key observational distinctions between Scrum and Kanban. These disparities encompass the necessity of time-boxed iterations, team commitment to predetermined workloads per product iteration, the prescription of cross-functional teams, limitations on work in progress (WIP), and the requirement to break down tasks for timely completion, among others [40]. In another study [83],identified nine distinctions between Scrum and Kanban methods including team structure and roles, performance metrics, flexibility to change, the amount of user participation, elasticity, events, work in process size, delivery frequency and task

board status. Moreover, in [5], authors mentioned extra items including different requirements prioritization mechanism, Feature sizing, cost saving and quality policies. Table 2.1 shows differences between Scrum and Kanban at a glance.

Chapter 3

Related Work

In a systematic mapping study done in 2014 [31], A total of 43 primary studies in the software startup context and 213 software engineering work practices were extracted, categorized, and analyzed. The research showed that there is no unique definition in the literature on what constitutes a startup and inconsistent use of this term constrained the creation of a coherent body of knowledge on software startups. Some of the features characterizing startups are innovative product development, market-driven development, small companies, short time-to-market, and web-based development. In a separate study analyzing the same dataset [60], the research team discovered that, among 213 software development practices referenced in 43 primary studies, only 16 were exclusively dedicated to software development methods in startups. Additionally, 19 studies focused on managerial and organizational factors. Notably, only 9 studies demonstrated high accuracy and scientific relevance. This underscores the significant need for further research to address the existing gaps and contribute uniquely to the field. Later this team conducted another research in 2015 [30], which performed semi-structured interviews with 13 startups and resulted in the Greenfield Startup Model (GSM) for software development in early-stage startups.

As ilustrated in Fogure 3.1, the model is formed in seven categories including speed-up development, evolutionary approach, product quality has low priority, team as the catalyst of development, accumulated technical debts, initial growth hinders performance, and severe lack of resources. The study stated that early-age startups prefer to build an initial prototype and iteratively refine it over time and they can focus on developing only parts of the system they want to validate instead of working on developing a whole new system. The results of this study indicate that the driving characteristics of startups were uncertainty, lack of resources, and time pressure. These factors influence software development to an extent that transforms every decision related to the development strategies into a difficult trade-off for the company. Additionally, They intend to discard formal project management, documentation, analysis, planning, testing, and other traditional process activities to satisfy the most urgent priority which is releasing the first version of the product as quickly as possible to verify the product/market fit.

Lately, based on this study, another research examined early-stage startups in Brazil [69] revealed that during the initial and validation phases, startups often selected practices and tools in an impromptu manner, leaning on the prior expertise of their

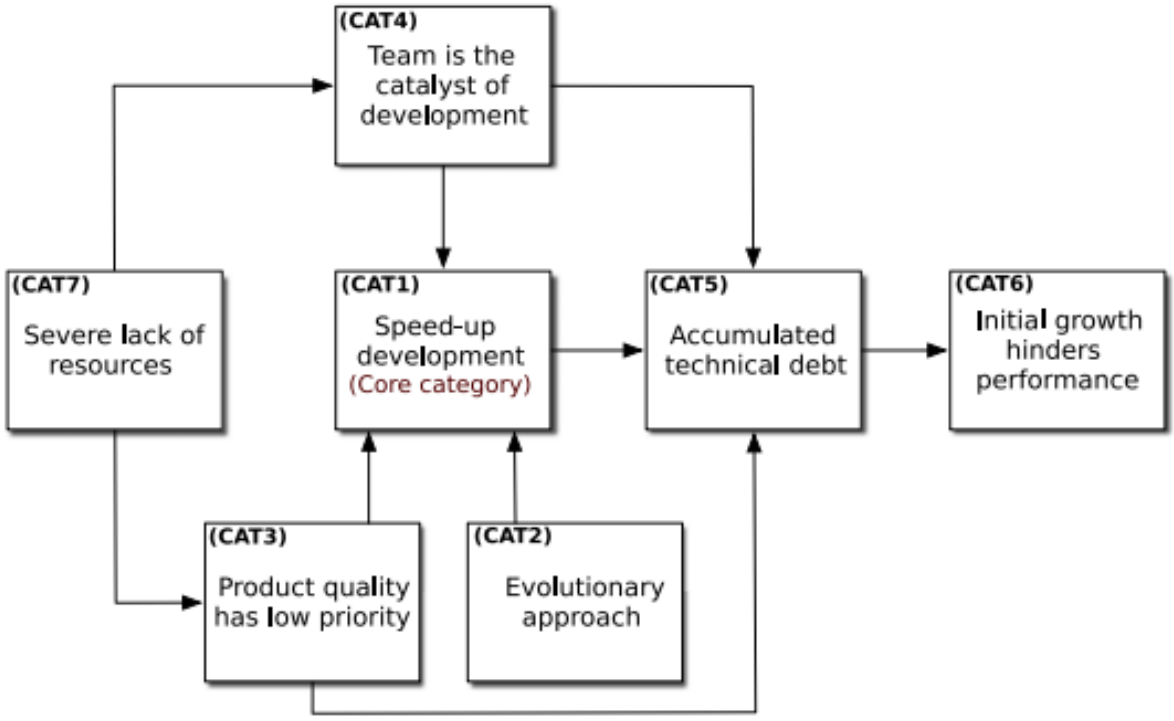

Figure 3.1: Greenfield Startup Model adapted from Giardino et. al [30].

development team. As they progressed into the growth phase, they recognized the significance of implementing more organized practices from the outset to enable product scaling with a matured team. Furthermore, the research findings emphasized that support tools were chosen based on their compatibility with existing tools and their ability to automate operational tasks. These studies employed the grounded theory research methodology to address their respective challenges, serving as a source of inspiration for our own investigation.

In 2015, a collaborative effort involving 30 researchers established a detailed research agenda for the software startup domain [77]. This agenda outlined six major clusters including engineering activities, evolution models and patterns, human aspects, startup ecosystems, applying startup concepts in non-startup contexts, and theories and methodologies for software startup research. This agenda serves as a valuable tool for gaining deeper insights into startup behavior and synthesizing valuable lessons and knowledge to guide future entrepreneurial ventures.

Afterward, in 2016, a team of 75 researchers started an worldwide study named "HELENA" which took around two years and collected extended data from around 1500 teams around the world about their software development methods, practices, and motivations [44]. Results show that Scrum and Kanban are two of the most methods used by teams globally. During the HELENA study, a survey [42] has been conducted on hybrid software development approaches and results show a variety of development approaches are used and combined in action. Most of these combinations follow a pattern in which a traditional process model serves as a framework

and agile practices are plugged in.

Based on a survey of 1,526 software startups [59], researchers discovered that speed-related agile practices, such as frequent releases and agile planning, are utilized more extensively than quality-related practices like regular refactoring and test-first development. This study specifically focused on the Lean Startup approach and did not address other lean practices, such as Kanban, highlighting a gap in research regarding Kanban aspects and hybrid models. In another independent study [46], again a survey has been used to collect numerical responses to questions about each of the project management factors in the six-pointed star model (including scope, budget, schedule, resources, quality, and risks) for both Scrum and Kanban based projects. Although the results show that Kanban performs better than Scrum in terms of managing project schedules, more respondents believe both Scrum and Kanban lead to successful software projects. Additionally, a similar study done in 2017 at Ericsson company shows neither of these methods is a perfect match for system testing and they need to be used together to complete each other [26]. Both of these recent studies underscore the significance of establishing a hybrid model that integrates elements of both Scrum and Kanban. Such an approach proves instrumental in steering companies towards success.

While these studies reveal the place of Scrum and Kanban among different software development methods, there is poor attention to their impacts and applications in the software startup context. Tegegne et al. [75] presents a systematic mapping study and identifies 37 relevant primary studies out of 1982 papers for validating the results, they conducted an empirical study based on the data collected from 14 real-life software start-ups located in Finland, Italy, and Norway. The results show that startups prefer to tailor methodology that matches their situation by dropping some aspects of the existing methodology and adding new practices from other methods.

Simultaneously in another research [51], An online survey was conducted for software startups in the Kingdom of Saudi Arabia to elicit their responses on their motivations for adopting agile methodologies. The outcomes reveal that the top five motivations for agile adoption are the need for accelerated product delivery, enhanced ability to manage changing priorities, increased software maintainability, simplified development process, and need for enhanced delivery predictability.

Although different hybrids of Scrum and Kanban methods have been developed and employed significantly (e.g. [36], [48], and [66]), there is a high research gap in identifying a suitable development method for software development at startups. For instance in 2012, A controlled case study [57] has been employed to contribute on hybrid of Scrum and XP model and later in another study there was a focus on combining Waterfall and Scrum methods [42]. A variety of studies (see [19] and [20]) intended to mix Stage-gate model with agile concepts or focused on large scale software development (e.g. [11]).

In an Empirical Study in 2018, authors defined a Scrumban as the combination of Kanban and Scrum practices to manage software development based on different

project situations. They proposed their model through the review of the previous work and semi-structured interviews with seven Agile experts. They highlighted different factors including method prescription, roles and responsibilities, adoption time, team size, batch size, requirements prioritization, feature size, lead time, technical practices, cost, and quality by combining appropriate Kanban and Scrum practices. The study results indicate that Scrumban was found to be more appropriate than Scrum or Kanban in saving time, improving quality, and minimizing waste [63]. The lack of reliable data about conducted development methods and their results in software startups has caused doubt on the outcome of having a structured development approach in both researchers and industrial practitioners. As a result, Understanding the capabilities and limitations (See [4] and [75] for more details) of agile and lean methods like Kanban and Scrum in different scales like teams, projects, and organizations is essential to determining suitable methods for startups.

Chapter 4

Research Method

4.1 Aims and Objectives

This study aims to explore which components of Kanban and Scrum are best suited for navigating uncertainty and ambiguity in a fast-paced environment. To achieve this objective, our first research question delves into the productive attributes of both Scrum and Kanban methods, laying the groundwork for our investigation.

Additionally, we sought to identify the key factors driving adaptations in the application of these methods within software startups, as well as pinpoint the highlighted concepts and elements that can be effectively utilized. It forms the basis of our second research question, with the ultimate aim of developing a hybrid model that integrates the most beneficial aspects of both methodologies, thus addressing the challenges posed by uncertainty and rapid change in the initial phase of the study.

4.2 Research Questions

As described in previous section, this research's progress is based on the following research questions:

Research Question 1 (RQ1): What specific aspects of Kanban and Scrum effectively address uncertainty and adaptability within the dynamic and fast-paced environment characteristic of software startups?

Research Question 2 (RQ2): Which elements of the Kanban and Scrum approaches can be included in implementing a hybrid way of working in software startups?

We aim to comprehensively address research questions by seeking the impact of both Scrum and Kanban on development approaches within software startups.

4.3 Research Methodology

Despite an abundance of literature on software startups, there exists a research gap in identifying a development method suitable for a specific startup context. To address this gap and since most previous research in this field is so much theoretical,

We need to avoid concentrating on a wide range of methods and approaches. As a result, we choose Scrum and Kanban methods as the foundation and develop a case study by following the guidelines and recommendations provided by Yin et al. [82] and Brereton et al. [15]. The case study is the complementary part of the research. Before going through it we need theoretical knowledge of methods and the practical expertise of software industry experts to achieve a reasonable outcome. As a result, we design a survey to collect industry-level experiences on both methods. These three angles form the foundation of this research. Figure 4.1 shows conducted research methodology at a glance.

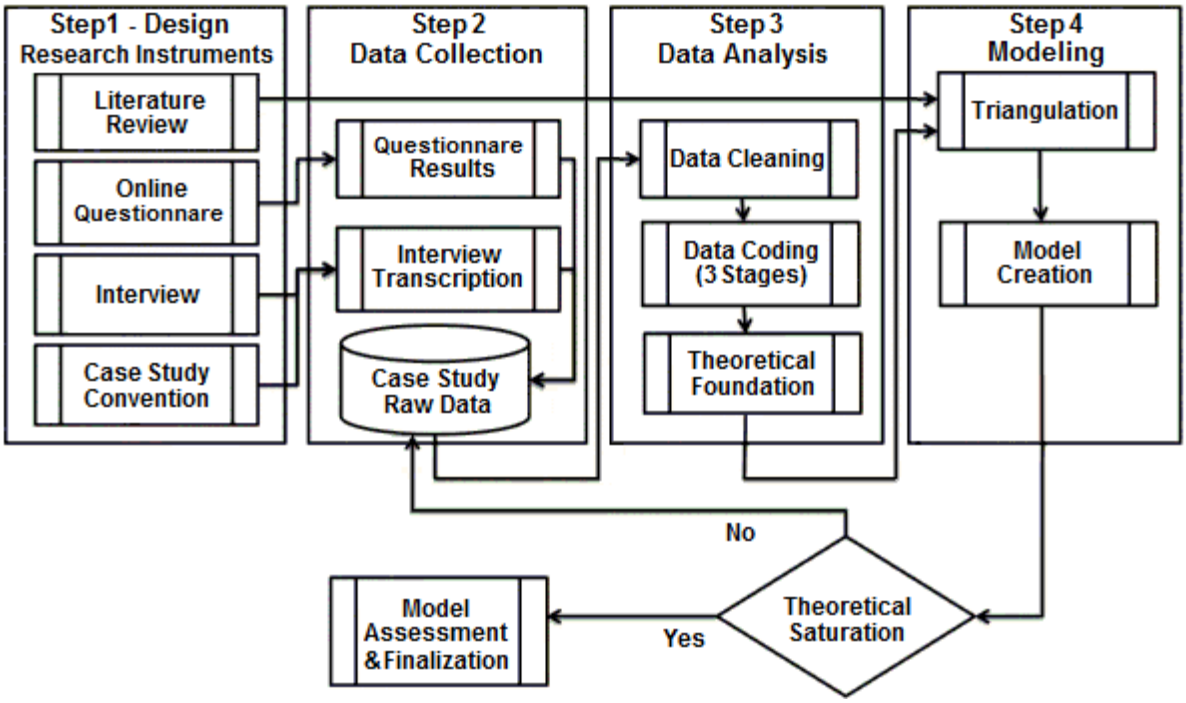

Figure 4.1: Research Methodology adapted from [30] and [69].

4.4 A Mind Map of Scrum and Kanban

To organize the gathered insights from both the survey and interviews, we have opted to create a comparative mind map delineating various facets of Scrum and Kanban within the startup context. According to [18] and [65], mind mapping serves as an effective tool for employing divergent thinking, fostering creative exploration, and making informed decisions. It enables researchers to visualize and investigate connections between various concepts, thereby facilitating the exploration of problem-solving approaches and theoretical analyses.

We utilize our understanding gleaned from established literature on Scrum and Kanban to incorporate their principles and components into crafting survey questions and conducting interviews. Following a structured approach, we can systematically identify cohesive concepts, patterns, segmentations, and mapping elements for both

methods. In this segmentation, we consider the aspects of each method that can deal with uncertainty and adaptability to get suitable clues for designing questions in other research instruments including the survey and interviews. By meticulously comparing the concepts and elements of both methodologies, each piece of data with independent significance is examined. This scrutiny resulted in the identification of six distinct sub-categories including values, principles/practices, essence/agenda, team structure, events/cadences, and artifacts. These boundaries guides the formulation of our survey and interview questions.

4.5 Online Survey

In software-oriented enterprises, the widespread adoption of agile and lean methods necessitates a systematic approach to elucidate similarities and differences in implementing Scrum and Kanban. According to the Startup Research Agenda [77], these criteria are placed among engineering activities, human aspects, and startup ecosystems. However, theoretical knowledge alone isn't sufficient for enterprise-level applications in software startups. Hence, we also designed an online survey (available as a form on Google [1]) to collect adequate data on practical experiences. Based on our research domain across software startups it is a great approach for illuminating commonalities and discrepancies among various implementations of Scrum and Kanban.

4.5.1 Questionnaire Objectives and Structure

The survey aims to gain insights into the motivations and evolving practices surrounding the adoption of Kanban, Scrum, or hybrid approaches. Accompanied by a mind map, this tool facilitates a deeper exploration of each method's nuances, which may not be immediately apparent in various software startup implementations. With a foundation in the existing literature on Scrum and Kanban, we ensure the comprehensiveness of our survey, capturing robust concepts related to each method.

The first part of the questionnaire is introductory, providing a definition and motivation on the research. The second part includes screening questions, focusing on demographics, preceded the main questionnaire to ensure respondent eligibility, maintaining data quality and relevance. The third part contains seventeen core questions in the form of multiple choices.

4.5.2 Target Population and Sampling

To properly execute the questionnaire, we need to create a repository of individuals considered software startup practitioners. According to various studies in the software startup context (such as [10], [54], [60], [70] and [76]), software engineers with varying levels of experience and skills are essential to startups of all types and sizes. Additionally, research on agile transformation in software startups (such

[1] `https://forms.gle/HVA4j2sqiRN4mbTM7`

as [4], [28], [72] and [76]) highlights agile practitioners as key players in this fast-paced, innovative environment. Consequently, we consider these two main groups to ensure our sample accurately represents any software startup. These groups are chosen because they best represent the perspectives of those working in startups at different stages of their life cycles. We further divide each group into different strata to cover all roles responsible for the Software Development Life Cycle (SDLC) in startups. These subgroups form our pool for "Quota Sampling" (See [53]and [34] for more information). To reach a diverse audience for the survey, we use email lists, social media platforms (such as LinkedIn, Instagram, Skype, Telegram, and Twitter), and online forums and communities. Quotas are set for different demographic groups within the target population to ensure a balanced sample, minimizing bias and accurately reflecting population diversity.According to [81], quota sampling cannot be considered as an acceptable alternative to probability sampling methods. Therefore, although we employ this strategy, we also obtain random samples from various channels simultaneously to mitigate any bias stemming from the influence of the survey topic and content on the respondents.

4.5.3 Pilot the Survey and Evaluation

The first designed survey had 86 questions and Likert scales with five scales was employed in capturing answers. A notable issue emerged concerning the alignment between participants' comprehension of Scrum and Kanban methods and the requested metrics. In many instances, individuals relied on their knowledge and judgment to assess the validity of statements. It presented a challenge in understanding the genuine experiences of participants with Scrum and Kanban methods and their associated aspects. Furthermore, the use of the Likert scale introduces complexities. According to [41], [45] and [67], The measurement through Likert scales appears susceptible to acquiescence bias and social desirability bias among participants, leading to a notable decrease in the accuracy of the responses. In light of these challenges and findings, a decision was made to embark on a fresh approach, prompting the design of a new questionnaire.

The revised strategy focused on the underlying concepts of implementing Scrum and Kanban methods,The second version of the questionnaire featured a controlled set of 35 questions, which included six demographic inquiries. These questions were crafted to explore the adaptation of Scrum and Kanban methods within startup environments. Notably, the questionnaire employed a Semantic Differential scale, allowing participants to provide responses across seven scales ranging from "Strongly Disagree" to "Strongly Agree". This modified questionnaire aimed to address the limitations encountered in the initial assessment and gather more nuanced insights from participants.

To reassess the questionnaire, we conducted corridor tests with a random person serving as the first candidate to evaluate the used language and timing. Moreover, we repeated the test with an experienced agile practitioner as the second. The first candidate efficiently completed the questionnaire in less than five minutes, swiftly reading and randomly selecting answers. The second candidate, an experienced ag-

ile practitioner, concluded the survey in 20 minutes. These test results instilled confidence in our ability to address previous issues and refine the questionnaire accordingly.

During the pilot phase of the initial round of the survey, after gathering over 40 samples, we noticed a strange pattern in responses to at least half of the questions. In twelve of the questions, just under one-tenth of respondents selected "Strongly Disagree" or "Disagree" answers. The wording of the questions seemed to have created uncertainty among respondents, resulting in less confident responses. Even if they had personal experiences contradicting the questions, respondents appeared to base their answers on their perception of correctness rather than their actual experiences. This behavior is attributed to an unusual frequency of responses leaning towards "Strongly Agree" on the 6 and 7 scales. As a result, we decided to consider this pilot as the third testing phase, make the questionnaire smaller by decreasing questions to 23, including six demographic questions, and eliminate the questions where a bias pattern has been detected. Appendix A shows the final version of questions used in this survey.

4.5.4 Survey Data Cleaning Policy

To keep the quality and validity of the survey data, we undergo thorough data cleaning and validation procedures. Initial attention directed toward respondents' answers to demographic questions, is crucial for filtering out inadequate samples. These questions are designed to ascertain participant eligibility, focusing on their firsthand experiences within software startup environments. Participants who lack such experience as software engineers within this ecosystem are excluded from the study. Furthermore, we established a minimum threshold of 15 answered questions (out of approximately 20 original survey questions) for each data entry. Entries failing to meet this criterion are removed from the dataset. Additionally, duplicate columns introduced during the data exporting process are eliminated. Lastly, we address various evident errors attributable to the survey design or data exporting process to ensure data integrity and accuracy.

4.6 Case Study

To achieve a thorough comprehension of the results and translate them into actionable development patterns for software startups, an explanatory case study is utilized in this research. The next section provides essential information on selecting the case study and managing its context within this research.

4.6.1 Case Study Selection and Justification

As suggested by Yin [82], investigating a contemporary phenomenon within its real-life context proves invaluable, particularly when the boundaries between the phenomenon and its context are unclear. This perfectly encapsulates our research context. Despite the abundance of theoretical or simulated models, it's imperative

to validate them through real-world experiences in software startups. According to [82], [71], [21] and [29], we recognized the importance of selecting a case study that is not only feasible and accessible but also directly relevant to our research topic. Moreover, it should possess unique attributes conducive to in-depth analysis and offer insights or data that directly contribute to addressing our research objectives. By adhering to mentioned criteria in the previous section, we chose to gather essential data from the firsthand experiences of ten people from five teams within an Iranian software startup that performed a successful nine-month platform transition project in the Summer of 2023.

4.6.2 Case Company and Context

The company under study is an online shopping platform established in 2020, specializing in various categories including personal appliances, books, apparel, home appliances, and digital products from reputable vendors, and offering fast delivery service. With over 200 employees, the company boasts an international, multicultural work environment. Its products are available online with a promise of delivery in less than three hours. Approximately ten thousand users across the country utilize its products daily. In 2022, facing technical and business challenges during the stabilization stage, the company decided to launch a new platform and move all customer data via a "big-bang" operation to it in the middle of 2023. This transition project, titled "Re-Platforming," tasked the technical staff with a nine-month project that concluded on July 1, 2023.

4.6.3 Data Collection and Analysis

In this study, semi-structured interviews are used as the primary data collection method and interviewees includes two Product managers and Senior developers from two development teams and six people selected from the other three supportive teams. Initially, we conducted a pilot study consisting of a first round of interviews. The preliminary findings from this pilot study confirmed that this startup is a viable candidate for further investigation. The interviews ranged from 30 minutes to 1 hour in duration, and we recorded and transcribed them verbatim for analysis purposes.

4.6.4 Target Population, Sampling and Unit of Analysis

Initially, our focus was solely on two development teams to gather their data. However, after conducting the first round of interviews and emphasizing the UAT[2] phase midway through the interviews, we modified some questions and adjusted our sampling strategy. As mentioned in section 4.6.2, we decided to target key senior people from the project, including those in the two development teams and three supportive teams: marketing, operations, and commercial departments. These individuals constitute the unit of analysis for the case study.

[2]User Acceptance Testing

4.6.5 Pilot Interviews

As mentioned in section 4.6.3, we conducted ten semi-structured interview sessions with with open-ended questions (See Appendix B for more detail) and went into more detail with interviewees to identify their perceptions of different aspects of transition and the development methodologies and their motivations toward the project goal. The majority of the second round interviews lasted around 45 minutes, and they were recorded and transcribed to be used in forming the research interview instrument. Appendix C contains the transcription of all interviews.

4.6.6 Data Analysis Approach

The determination of theoretical saturation in our project is guided by a combination of established principles and specific methodologies tailored to our research context. Here is how we achieved it:

Firstly, we established a clear scope for our study, defining our research questions, target population, and data collection methods. This clarity ensures that our efforts in data collection and analysis are focused and purposeful.

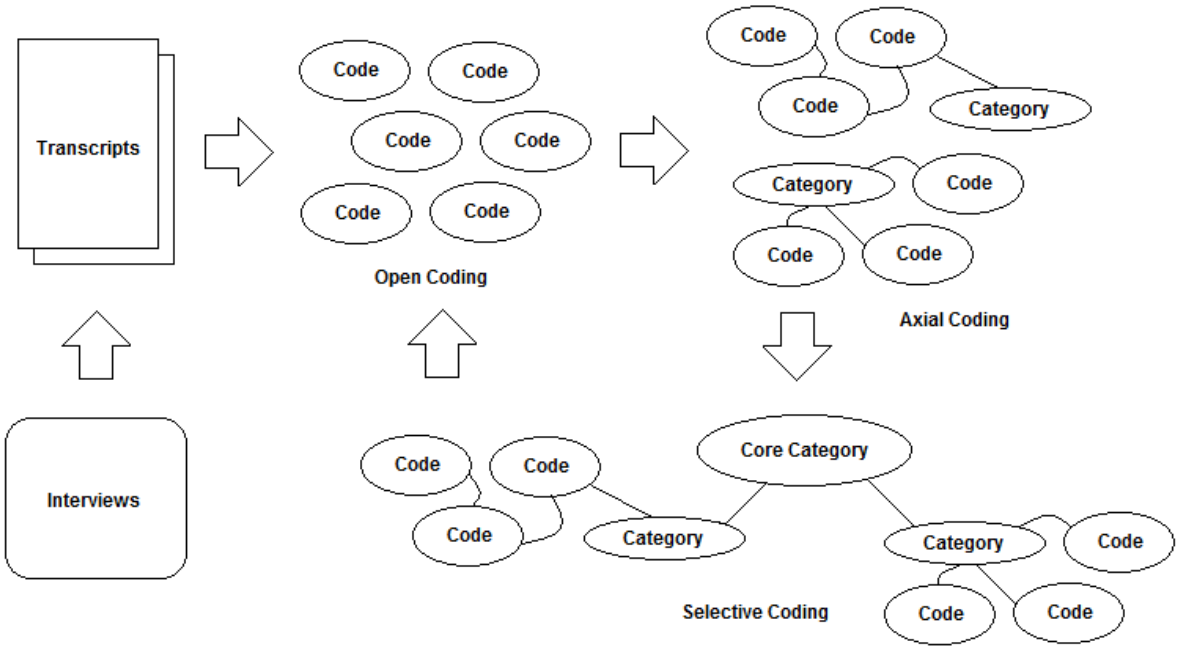

Figure 4.2: Qualitative Research via GTM adapted from Williams and Moser [80].

Secondly, we utilized multiple data collection methods, including interviews, surveys, and literature reviews. This multiplicity of approaches not only enriches our dataset but also allows us to cross-validate our findings, enhancing the credibility and reliability of our results. Conducting data analysis concurrently with data collection is another key strategy we employed. By analyzing emerging themes and patterns in real time, we are able to adapt our data collection methods accordingly, ensuring

that we gathered sufficient data to address our research questions comprehensively. Additionally, we employed Grounded Theory Methodology (GTM) for within-case analysis, following the technique of theoretical saturation as outlined by Morse [52]. It involved continuously sampling and analyzing data until no new insights emerged and all theoretical concepts were thoroughly developed.

Our data analysis process, which includes Straussian coding methods such as open, axial, and selective coding, further facilitates the identification and exploration of relevant themes and concepts (Figure 4.2). Drawing inspiration from the works of Giardino et al. [30] and Souza et al. [69], we ground a preliminary theory that informed our research methodology and shaped our understanding of the hybrid model. Lastly, we employ triangulation, integrating insights from both interviews and surveys to refine our final model. This process enhances the validity of our findings by corroborating them across multiple data sources. By combining a clear research scope, multiple data collection methods, concurrent data analysis, adherence to GTM principles, and triangulation, we confidently determine theoretical saturation in our qualitative research project, ensuring a robust and comprehensive exploration of the transition project and its associated phenomena.

4.7 Validity Threats

In the following section, we discuss the potential validity threats to our research. By identifying and addressing these factors, we aim to ensure the credibility and reliability of our findings.

4.7.1 Construct Validity

As discussed in section 4.5.3, Throughout this research, we conducted three iterations of questionnaire design and piloting. Initially, our questionnaire encountered several challenges including an overwhelming volume of questions, participant comprehension and interpretation issues regarding Scrum/Kanban methods, as well as biases such as acquiescence and social desirability. To address these issues, we iteratively redesigned the questionnaire, reducing the number of questions from 86 to 23. Additionally, we employed a corridor test to assess the relevance, seeking feedback from both a skilled agile practitioner and a randomly selected bilingual individual proficient in English and Persian to gauge the questionnaire's clarity and completion time. In the case study phase, we encountered integrity issues with questions during the pilot interviews, necessitating two rounds of interviews and modifications to better address concerns.

4.7.2 Internal Validity

Another potential threat lies in the selection of practitioners for sampling. We acknowledge that the obtained responses may not fully capture the diversity within the population of startup software professionals, posing a risk to internal validity. Nevertheless, by deliberately selecting software professionals and agile practitioners from

software startups across various domains and employing a quota sampling strategy to ensure a balanced representation of specialized areas, age, and experience levels, we aim to mitigate this concern and provide a dataset that reflects the broader population. Furthermore, as discussed in section 4.6.4, in order to normalize the list of interviewees, we adjusted our sampling strategy after conducting three interviews. This decision was prompted by feedback received from participants regarding significant achievements that were overlooked during the case study project.

4.7.3 External Validity

Although the respondents in our survey may not encompass the entirety of startup software practitioners, potentially limiting the statistical relevance of our findings, we believe that the analysis of the 134 responses provides a valuable qualitative data source for extracting insights. Additionally, in the case study component of our research, we engaged a critical chain of participants from a successful challenging project, which yield valuable insights into the effective application of Scrum and Kanban within software startups.

Chapter 5

Research Execution

This chapter delineates the timeline of the research execution along with notable occurrences throughout the process. The study comprises two primary activities: the survey and interviews. Subsequent sections of this chapter provide detailed plans and highlights of these activities.

5.1 Survey Execution

As discussed in section 4.5.3, After undergoing two rounds of design and evaluation in December 2023, the survey instrument was prepared. Initially, the questionnaire comprised an extensive 86 questions, which was deemed excessive and overly reliant on explicit methodological concepts. However, after encountering challenges during the corridor test, as described in Section 4.5.3, significant revisions were made. This led to a reduction in the number of questions to 35 for the first pilot. Despite these adjustments, issues emerged during execution, with half of the responses demonstrating an unexpected level of agreement between respondents' experiences and expected outcomes. Consequently, further revisions were undertaken, resulting in a streamlined questionnaire comprising 23 questions, encompassing six demographic inquiries and 17 core questions.

In January 2024, the revised questionnaire underwent another pilot phase, albeit with a one-month delay, which caused disruptions to the planned interviews scheduled for the first two weeks of the new year. The final iteration of the questionnaire was completed in the initial week of May 2024, with subsequent data collection and cleaning conducted under the data-cleaning policy outlined in Section 4.5.4.The survey results are analyzed and assessed in Section 6.1.

In the final round, As described in section 4.5.2,we utilized email lists, social media platforms, online forums and communities to dispatch the questionnaire among participants. We sent 70 direct emails to a list of candidates that we chose them as the quota samples including software engineers and agile practitioners. Moreover, we shared the questionnaire for 90 quota and 80 random candidates from different channels such as social medias and online forums. By employing this approach we expected to achieve a mix of quota and random answers. Figure 5.1 illustrates the distribution of survey channels that employed in this research. The total number of answers collected from audiences was 134 which shows that more than half (55.8%) of survey audiences responded to the questionnaire.

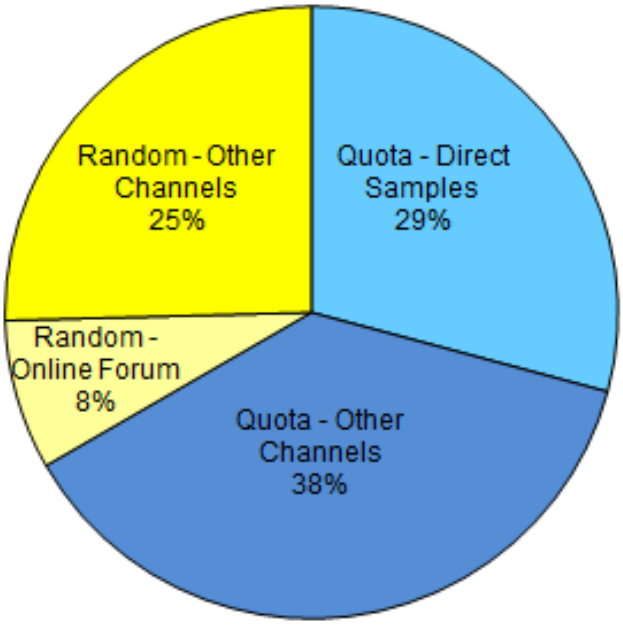

Figure 5.1: Survey samples distribution and channels - Yellow spectrum: Random Samples; Blue spectrum: Quota Samples.

5.2 Interviews Execution

As discussed in section 4.6.2, The initial round of interviews took place in December 2023, aiming to assess the selected case. During this phase, we engaged in discussions with interviewees regarding their experiences with a transition project conducted six months prior. Following the selection of cases and finalization of interview questions, meetings were scheduled with interviewees to commence in the second week of January 2024. However, as detailed in the preceding section, an unforeseen delay resulted in a postponement of the interview start date.

Ultimately, the first series of interviews was conducted and recorded in February 2024, with a delay of approximately one month. During these sessions, after completing three interviews, we encountered a topic not initially addressed in our interview questions: the significant experiences of interviewees with the User Acceptance Testing (UAT) of the new platform. In response, we incorporated a question regarding the tools utilized during the UAT and expanded our sample target to encompass a broader range of individuals involved in the project.

As outlined in Section 4.6.4, additional relevant participants were invited, and the second round of interviews was conducted in March 2024. However, one of these sessions was interrupted by an unforeseen incident, prompting its resumption two days later. The results of these interviews are analyzed and evaluated in Section 6.2.

Chapter 6

Results and Analysis

6.1 Survey Results

In this section, we report the results of our survey study. As discussed in the Survey Data Cleaning Policy section, we define a quality criteria on accepting the responses. Finally 134 respondent answered the questionnaire. However, 121 responses met our criteria and picked for data analysis phase. As outlined in section 4.5.4, we set a threshold value, considering only samples with fifteen or more answered questions as a credible source of experience. Consequently, we excluded two samples that fell below this threshold. Additionally, nine respondents indicated no software startup experience in their demographic responses. To ensure a consistent research population, we omitted their responses as well.

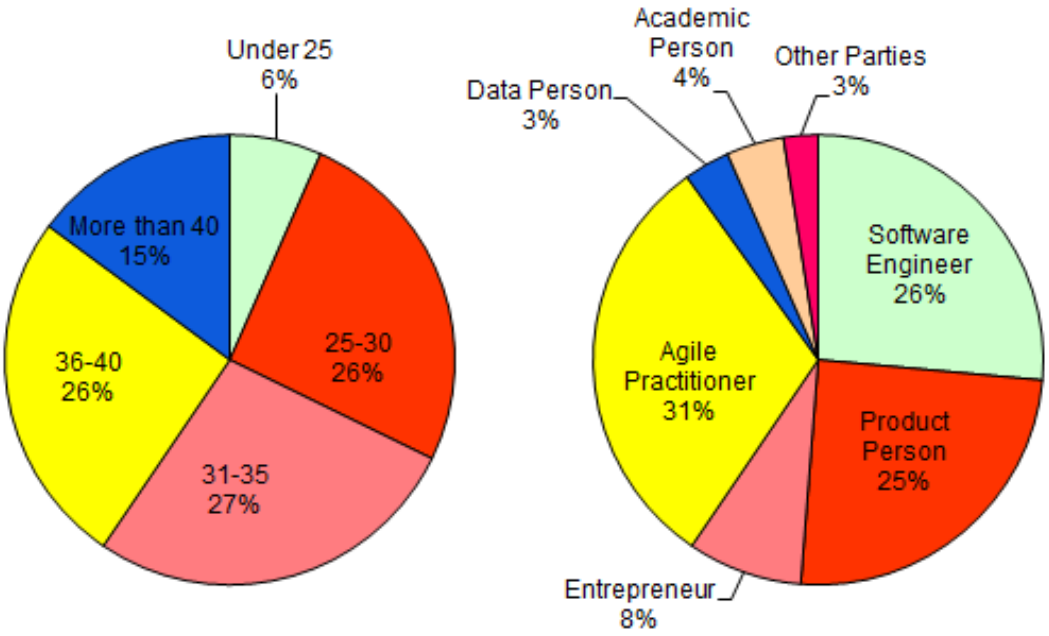

Figure 6.1: Demographic questions - Participants age distributions (left); Participants professions (Right).

6.1.1 Demographics

To establish the observed sample profile, we assess the participants' ages, professions, the size of the startups they collaborated with, the phase of company formation, and the duration of their collaboration. Responding to these inquiries is mandatory as one served as a filter to identify individuals lacking startup experience, and we consequently can exclude their data. Concerning participants' ages, the data distribution follows a normal pattern, with each main category representing a quarter of the total participants. Similarly, regarding skills and professions, adhering to the quota sampling technique, we identified three primary categories: Product People, Software Engineers, and agile practitioners (Figure 6.1).

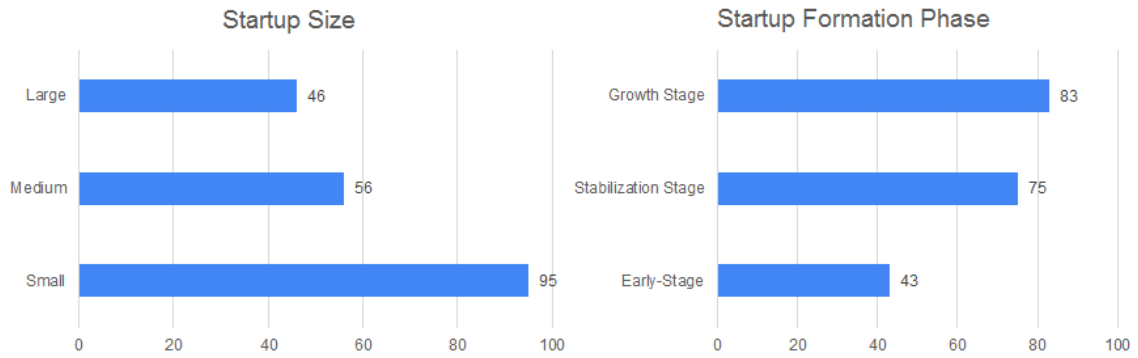

Figure 6.2: Demographic questions - Startup size that the participants worked for (left); The company formation phase during the collaboration with the survey audiences (Right).

Moreover, Participants were given the option to select more than one choice regarding the size of the startup(s) and the phase of company formation questions. The results indicate that the majority of participants were employed by small startups while simultaneously being in the growth stage (Figure 6.2). The final question focuses on the duration of collaboration between survey respondents and startup companies. As illustrated in Figure 6.3, approximately half of the participants collaborated for a period ranging from one to three years, which significantly enhances the quality of the published data in the questionnaire.

6.1.2 Core Questions

As illustrated in Appendix A, we utilize seventeen questions to gather insights into the experiences of the target population regarding the implementation of Kanban and Scrum methodologies in software startups. The results indicates that a majority of survey respondents endorsed the applicability of Scrum and Kanban as operational frameworks within software startup environments.

Following data collection and subsequent cleaning procedures (which involved eliminating ten samples during the analysis phase), a total of 2035 data points are cap-

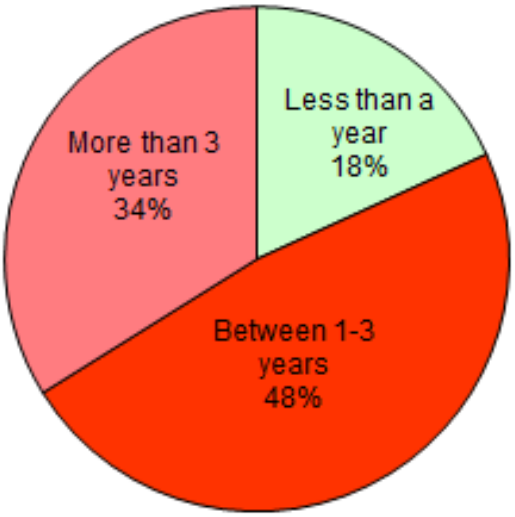

Figure 6.3: Demographic questions - The duration of collaboration.

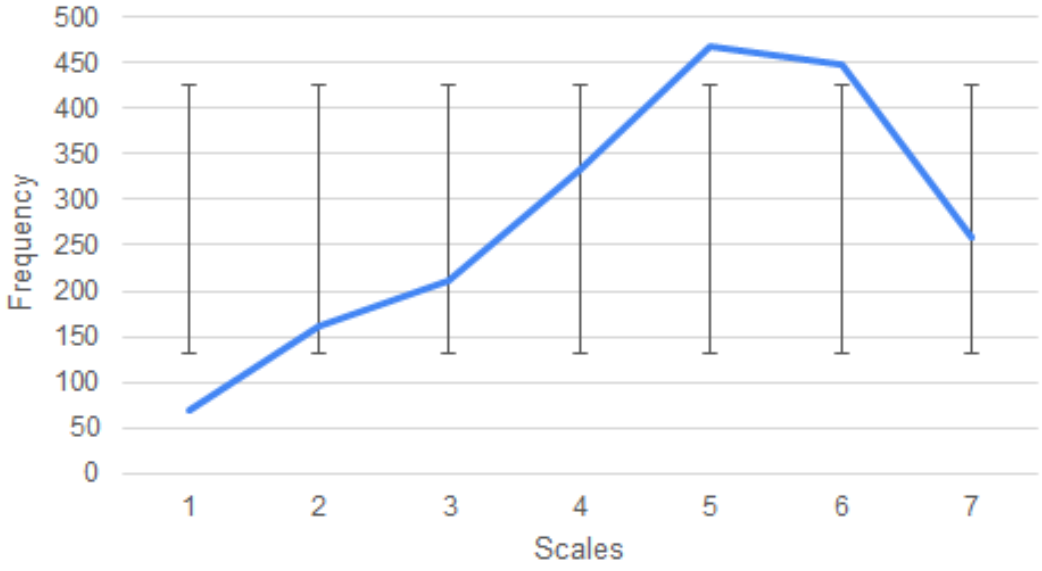

Figure 6.4: The Frequency Distribution of answers in scales from 1 to 7.

tured through these questions. It was observed that most participants rated their responses on a scale of 5, 6, or 7, indicating alignment with the items presented in the questionnaire (Figure 6.4). The coherence of the data is deemed satisfactory, with a total standard deviation of 1.606. Furthermore, a significant portion of the data points clusters around the mean value of 5, indicating a consistent distribution of responses. Figure 6.5, shows the distribution of data around central tendency, variability, and the presence of outliers of them.

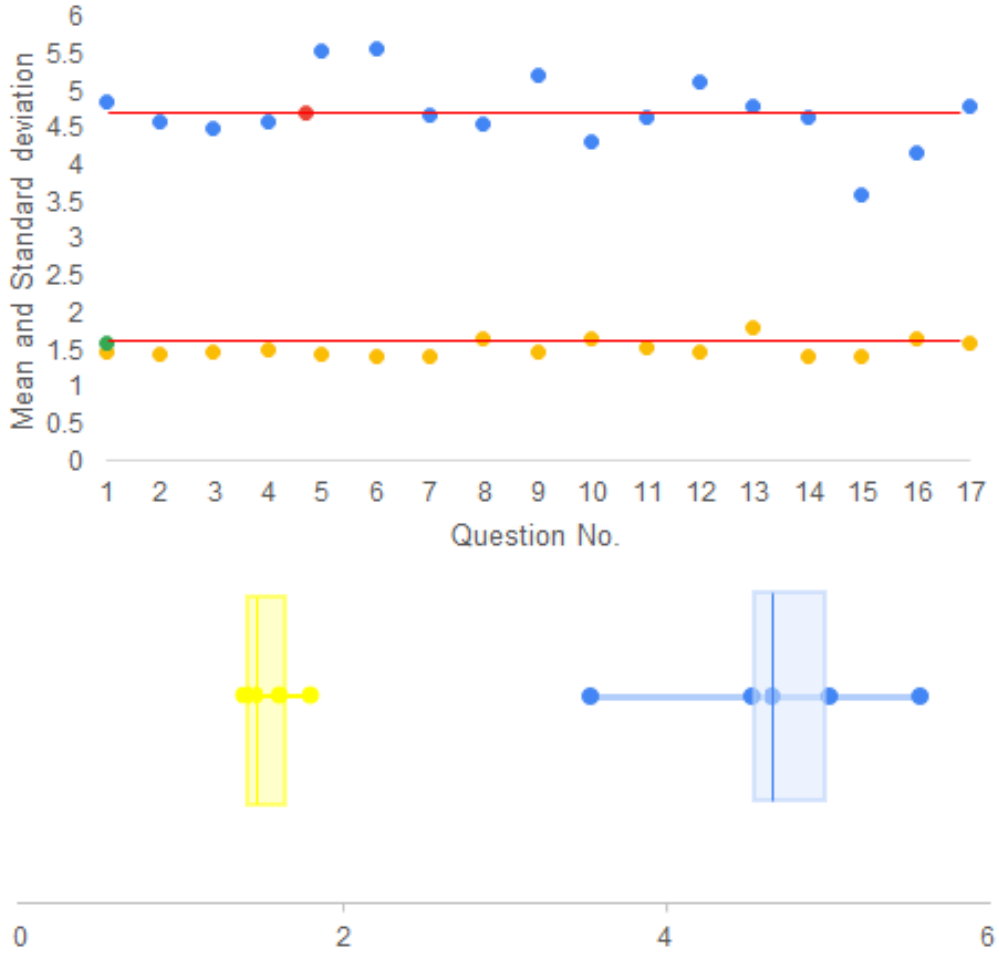

Figure 6.5: The distribution of data, central tendency, and variability: Mean value (blue) and Standard deviation (yellow).

6.1.3 Centric Responses

As highlighted in the preceding section, the survey participants exhibited commendable engagement, yielding a robust dataset for our analysis. As depicted in Figure 6.5, a significant proportion of responses are distributed around key central tendency metrics such as mean, median, mode, and standard deviation. The standard deviation serves as a measure of the dispersion or spread of data points around the mean. A higher standard deviation indicates greater variability in responses to a

particular question, whereas a lower standard deviation suggests that responses are more closely clustered around the mean. Our analysis reveals that, for the majority of questions, the disparity between the mean and standard deviation of samples in relation to their total values is minimal. Consequently, we interpret these responses as centric. Table 6.1 provides a concise overview of these questions.

Table 6.1: Questions with centric responses

Topics/ Principles/ Practices	Questions
Iterative development/ Requirement Engineering/ Self-Management/ Continuous Delivery	1- The company allowed the team to iteratively refine and develop their product/project requirements and deliverables 4- Implementing shorter iterations reduced the delay in the team's delivery
Sustainable Pace in Workflow/ Managed Flow/ Limited WIP	2- The team achieved an enhanced sustainable pace in releasing work during the iterations 3- The team effectively managed workloads and established a sustainable pace of workflow
Estimation/ Prioritization/ Self-Management	8- Estimating tasks aided the team in identifying better prioritization 9- An ordered backlog of tasks led the team to choose the next task promptly after completing the current one
Testing/ Cross Functional Team (Team Structure)/ Sustainable Pace in Workflow	10- There was an opportunity for the team to test during the development 11- Testing first caused a tangible decrease in the time needed to complete tasks
Pair Working/ Self-Management	12- Pair working on tasks helped the team to resolve them more accurately and quickly
Pair Working/ Self-Management	13- Short Daily Meetings reduced the need for other unplanned ad hoc meetings during workdays
Daily Work between Business and Tech/ Team Structure	16- Business and technical teams collaborated daily toward the goals of the product/project
Hybrid Way of Working/ Sustainable Pace in Workflow	17- Eliminating or modifying certain elements or concepts of Agile methods (like Scrum or Kanban) helped us to make the team's way of working more efficiently and better respond to project/product requirements

6.1.4 Varied None-Centric Responses

As depicted in Figure 6.5, for questions 5 and 6, which have the highest mean values, it suggests that participants, on average, strongly agree with the statements

or premises presented in these questions. Conversely, for question 15, which has a notably lower mean value compared to the others, it indicates that participants, on average, are less in agreement with the statement or premise presented in that question. However, the mean value alone cannot represent the distribution of opinions or sentiments among participants. As the result, we consider standard deviations as the complementary measure as well. Our analysis reveals that the standard deviations for six questions—namely Q5, Q6, Q7, Q14, and Q15—are lower than the total standard deviation value. This indicates a fewer degree of variability in responses to these questions when compared to the average variability across all questions. Conversely, for Q13, the situation is reversed. Its standard deviation surpasses the total standard deviation value, suggesting a heightened variability in responses to this particular question compared to the average variability across all questions. This doesn't imply that participants either agree or disagree with the underlying premise of the questions. Rather, it signifies a lower (for Q5, Q6, Q7, Q14, and Q15) or higher (for Q13) degree of diversity or inconsistency in participants' responses to these specific questions. This could be due to various factors such as the ambiguity of the question, differing interpretations among participants, or varying levels of agreement or disagreement on the topic addressed by the question. The data shows these questions have higher variability in responses compared to the other questions in the dataset. This variability could warrant further investigation to understand the reasons behind the diverse range of responses. Table 6.2 shows questions without convergent responses with their criteria.

Table 6.2: Questions with None-centric responses

Topics/ Principles/ Practices	Questions
Visualizing Flow/ Transparency, Inspection and Adaptation/ Prioritization	5- Visual task boards clarified work and facilitated a quicker completion 6- Visual task boards facilitated prioritization tasks
Limited WIP/ Managed Flow/ Self Management	7- There was an opportunity for the team to manage and limit work in progress
Daily Meetings/ Daily Work between Business and Tech/ Time Boxing	13- Short Daily Meetings reduced the need for other unplanned ad hoc meetings during workdays
Goal Setting/ Sprint Backlog/ Planning Events	14- The team formulated actionable plans for delivering their goal during planning event
Risk Management/ Daily Work between Business and Tech/ Team Structure	15- Managers or business owners adeptly recognize and assess business risks and threats inherent to the project/product, effectively communicating them to the software development team

6.1.5 Side Patterns

By posing three distinct demographic inquiries regarding the size of startups participants were engaged with, the phase of company formation during collaboration, and the duration of collaboration within startups, we are able to uncover similar trends in the responses provided. As illustrated in Figure 6.6, There appears to be a broad alignment between individuals with experience in all sizes and stages and duration of collaboration within startup environments and the wider statistical population regarding the adoption of Kanban and Scrum methods.

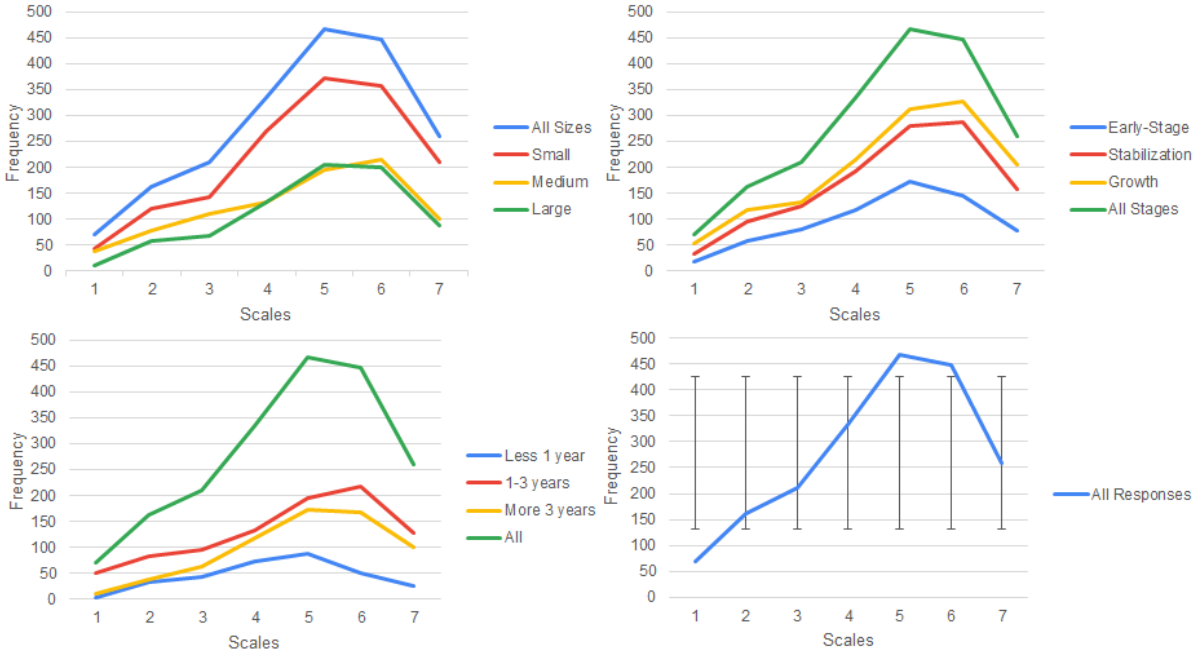

Figure 6.6: Side Patterns in comparison all samples.

6.1.6 Survey Summarizing

In summary, a significant portion, representing three-fifths of participants, reported having the opportunity to iteratively refine and develop requirements and deliverables, while also enhancing the sustainable pace of work during iterations. Approximately half of the respondents effectively managed workloads, established a sustainable workflow pace, and utilized shorter iterations to mitigate delays in delivery.

Addressing outlier responses, a substantial majority, comprising four-fifths of the audience, acknowledged that visual task boards facilitated clarity in work tasks and expedited completion, aiding in task prioritization. Moreover, three-fifths of Kanban teams successfully managed and limited work in progress, while employing task estimation proved beneficial for all teams in identifying better prioritization strategies.

Over two-thirds expressed satisfaction with pair working and programming, affirming that an ordered backlog enabled prompt task selection post-completion. Approximately half of the respondents conducted tests during the development cycle, with

over sixty percent endorsing testing-first approaches, citing reduced task completion times. Additionally, respondents attested that short daily meetings effectively curbed the need for unplanned gatherings.

Notably, discrepancies arose concerning participant experiences in risk management collaboration with business counterparts, as well as in daily collaborative efforts between technical and business teams toward shared objectives. The final inquiry seeks insights into the use of customized methodologies within startups, yielding generally positive feedback on the integration or removal of method components to develop hybrid models.

6.2 Interviews Results

In this section, we present the outcomes of our case study. As outlined in the Case Company and Context section, we conducted two rounds of interviews involving ten individuals associated with an Iranian startup that underwent a transition project in 2023. This section shows the result of this instrument of the research.

6.2.1 Interviews Transcription and Coding

To gather qualitative data for constructing our hybrid model of Scrum and Kanban, we meticulously recorded and transcribed all interview sessions. Prior to commencing the data analysis, we organized the transcripts into thematic categories based on various topic cards utilized during the interviews. Subsequently, the data analysis was initially carried out individually for each interviewee.

Our process comprised four steps. Initially, we tagged and associated highlighted topics and concepts with raw data (Open Coding). Subsequently, the derived concepts were clustered into theoretical categories and subcategories, with interconnections established based on cause-and-effect relationships (Axial Coding). These categories underwent refinement to achieve varying levels of abstraction and conceptual adjustment. In the third step, the coherence among categories was validated through Selective Coding, involving the exploration and analysis of links among subcategories. Finally, the core category is identified by analyzing the causal relationships between high-level categories. As shown in Tables 6.3 and 6.4 we recognize five different categories regarding topics/challenges and codes.

6.2.2 Interviews Summarizing

The central role of all interviewees in their respective projects was to contribute to the MVP[1] delivery within their teams, particularly within the context of the "Replatforming" project. Kanban and Scrum teams attributed enhances task efficiency to their team dynamics, whether through weekly iterative deliveries or continuous

[1]Minimum Viable Product

Table 6.3: Interviews Coding results

Categories	Codes	Related Codes/Actions/Results
Category 1: Tangibles	Main Responsibility	Delivery of the MVP with the Team
	Making tasks transparent	used Task boards
	Product Goal	caused synergy and more focus regarding the tasks and project
	Service desk Kanban board	helped in UAT and recognizing issues, following up and resolving them successfully
Category 2: Structures	Team Structure	was inappropriate and interference from the outside the lack of human resources caused overworking
	Joining and leaving people	have not affected teams as long as they had enough space to breathe recover and perform again
Category 3: Policies	Deliver Continuously in Ongoing Mode	Caused a better speed in doing tasks
	In working in Iteration Mode	Inspection/Adaptation went well Responding to change wasn't good
	In working in Ongoing Mode	Responding to Ad hoc went well Scope Management wasn't good and caused technical debts
	Keeping balance between work volume and allocation time	Picked the shorter Scope of the work and postpone technical debts for future
	To limit Work in Progress in team	controlled Work load by Setting WIP=1,2 accepted parallel working within a week
	Prioritizing the work within team	included estimations in prioritization
	Approaching the test cycle	conducted UAT with subject-matter experts inside company
	Autonomy to decide on tasks details and how to develop them	Teams were Self-managed

Table 6.4: Interviews Coding results (continue)

Categories	Codes	Related Codes/Actions/Results
Category 4: Attitudes	Teams had enough courage to do the right work regarding their tasks	they were motivated and committed to the project goal and showed transparent staying power to assessments
		happened in iteration-based teams with facilitation by the PM
	Daily work among Business and Technical people	crafted in ongoing teams by employing Pair Programming/working approach
Category 5: Mechanisms	Weekly Delivery in Iteration Mode	Caused a better speed in doing tasks
	Refine the project backlog	held backlog refinement meetings
		used Focus Groups
	Approaching the test cycle	wrote unit and acceptance tests and approach them before deploy
	conducted Daily Meetings in a time box of 15 min while working in Iteration Mode	was helpful and reduced the demand for other extra meetings
	conducted Daily Stand-ups in a time box of 15 min while working in Ongoing Mode	was helpful but several extra meetings were needed
	The majority of meetings were held in their time boxes	but some were not and it caused distractions and delays in deliver features
	recognizing risks before starting the tasks	not went well because the majority of risks were unpredictable or unknown
		has been treated by matching context and predefined data and visualizing in task boards

ongoing deliveries. The consensus among participants is that the overarching product goal of "Re-platforming" fostered synergy and sharpened focus on their tasks.

However, nuances emerged in the approach to handling changes within iteration-based work. While inspection and adaptation processes are generally effective, some teams struggled with responding to unforeseen alterations. Conversely, teams operating in ongoing mode reported better adaptability to ad hoc challenges but found scope management more challenging. Balancing workload and time constraints often necessitated overtime, with many interviewees opting for shorter scope selections to maintain equilibrium.

In the realm of requirement engineering, various techniques and practices are employed, with backlog refinement meetings and stakeholder focus groups being prevalent. Task boards depict a flow system in which work items flow through various stages of a process, usually ordered from left to right [8]. In this investigation, they emerge as a favored tool for enhancing workflow transparency, with Kanban teams implementing workload controls through WIP (Work in Progress) limits, while Scrum teams embraced parallel work within weekly sprints.

Estimations plays an important role in task prioritization for most interviewees, while testing approaches varied between Kanban and Scrum teams. Scrum teams typically conducted unit and acceptance tests pre-deployment, whereas Kanban teams emphasized UAT[2] in collaboration with subject-matter experts within the organization.

Interviewees expressed satisfaction with their teams' autonomy in task decision-making and highlighted the effectiveness of daily meetings in reducing the need for additional gatherings. However, structural issues within teams, such as external interference and resource shortages, are commonly cited concerns, impacting productivity and leading to overwork.

The effects of team turnover on productivity were disputed among interviewees, with some noting a decline in productivity following resource losses. Daily collaboration between Business and Technical teams are more pronounced in Scrum setups, whereas Kanban teams leveraged practices like pair programming to compensate.

Although most meetings adhered to designated time slots, disruptions and delays were occasionally experienced due to scheduling deviations. Similarly, the majority of interviewees encountered challenges in the risk management process, mirroring findings from the questionnaire.

Lastly, discussions surrounding ticketing tools used in UAT indicates widespread agreement among participants regarding their utility in issue recognition, tracking, and resolution during the testing phase.

[2]User Acceptance Testing

Chapter 7

Discussion

7.1 Summarizing on RQ1

This research aims to develop a nuanced understanding of the impacts of Scrum and Kanban methods on software development approaches within startup environments. The survey aims to uncover the predominant strengths of Scrum and Kanban. Through an examination of the questionnaire data, we uncover a notable insight regarding methodologies. Our findings reveals a shared consensus across diverse contexts concerning certain aspects of Scrum and other methods. For instance, the principle of iterative and incremental work, a cornerstone of Scrum, emerge as widely adopted among respondents, leading to tangible achievements. Moreover, our analysis unveils a prevailing belief among participants that shorter iterations contribute to reduced delivery delays, aligning with the concept of shorter sprints inherent in Scrum events.

Another noteworthy observation is the effective management of workload, a serious challenge, particularly in startup environments characterized by high uncertainty. Despite these challenges, a significant portion of respondents reported successful strategies for addressing workload concerns. When juxtaposed with the revelation that many respondents utilized visual task boards to elucidate their work processes, it becomes apparent that elements of Kanban, such as visualizing workflow and managing flow, play a pivotal role.

As discussed in section 4.4, we deliberate on these methodologies' fundamental elements and concepts, using targeted questions to glean insights and experiences from respondents. For instance, we focused on probing the influence of various Kanban method values such as Transparency, Balance, Collaboration, Customer Focus, Workflow, Leadership, Understanding, Agreement, and Respect. The survey findings shed light on the respondents' capability to effectively manage workloads and establish a sustainable pace of workflow which is a manifestation of commitment and leadership values inherent in the Kanban method.

We meticulously deconstructed both methodologies through our systematic approach into concise atomic concepts and elements, facilitating a deeper understanding of their fundamental effectiveness factors. By meticulously comparing these methodologies' concepts and elements, we rigorously examined each piece of data, identifying distinct sub-categories: values, principles/practices, essence/agenda, team discipline,

events, and artifacts. These categories were derived based on their alignment with the theoretical framework and components of each methodology. Additionally, we engaged in a comprehensive comparison of the conceptual frameworks and elements of both Scrum and Kanban methodologies. This comparison aimed to abstractly identify shared ideas and practices inherent in each method's body of knowledge. Our analytical approach systematically organized these elements, recognizing that each method encompasses distinct components. Finally, we constructed a comparative mind map delineating the six shared dimensions between Scrum and Kanban with different components of each method (Figure 7.1).

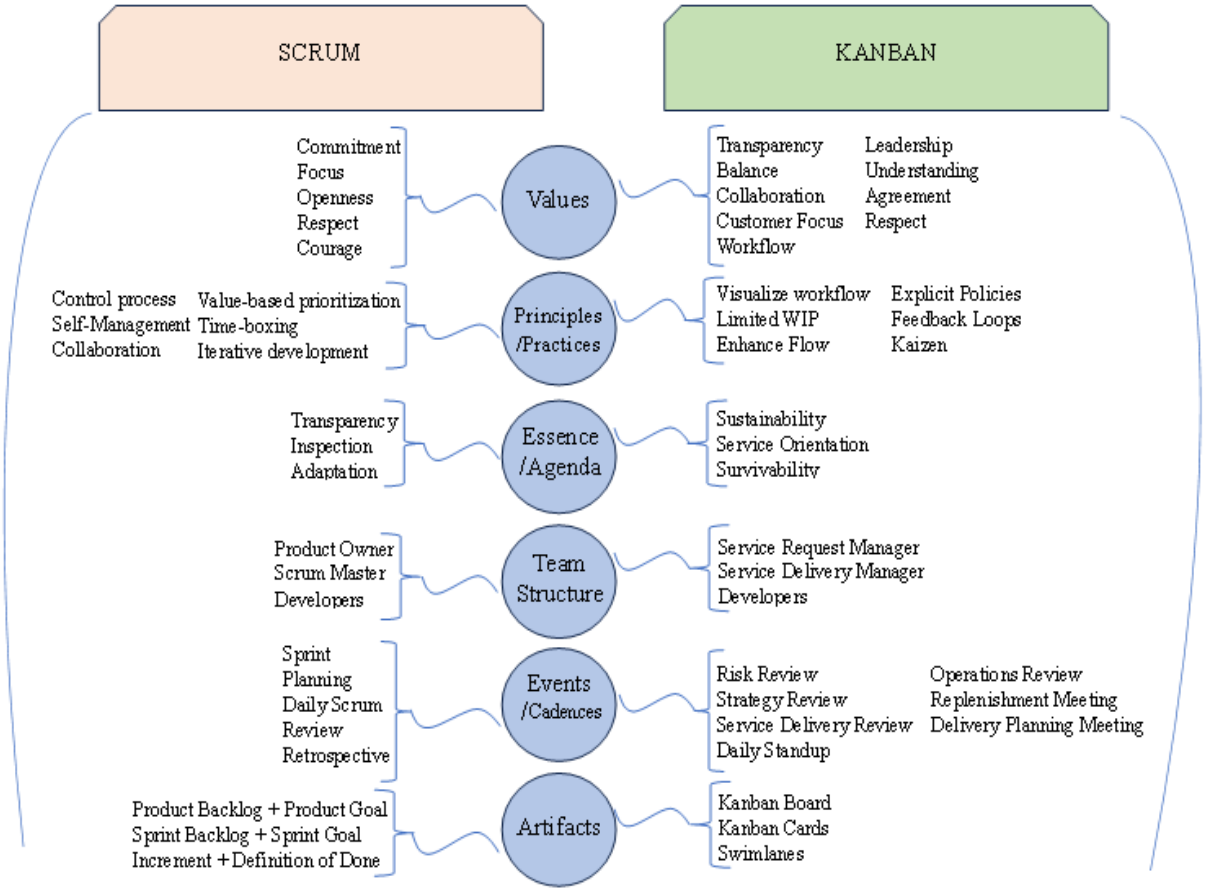

Figure 7.1: a Mind-map of Kanban (right) and Scrum (left) aspects in six different dimensions including Artifacts, Events/Cadences, Team Structure, Essence/Agenda, Values, and Principles/Practices.

The validation of this mind map with the survey results across all six criteria empowered us to address the first research question effectively (Table 7.1). In the dynamic landscape of software development, particularly within startups, careful consideration of six key dimensions is imperative to navigate uncertainty and devise adaptive strategies to overcome challenges. The choice of methodology—whether Scrum, Kanban or a hybrid approach—depends on the specific context and needs of the team. Software startups must evaluate their operational environment using metrics that elucidate the implementation of these dimensions clearly.

Table 7.1: Mind Map Validation with the survey results

Questions No.	Covered Aspect(s)	Survey Result
1,2,3	Principles/Practices, Essence/Agenda, Events/Cadences	Intended to Agreement
15	Principles/Practices, Events/Cadences	Intended to Agreement
4,13	Events/Cadences	Intended to Agreement
5,6,7,8,9	Principles/Practices, Essence/Agenda, Artifacts	Intended to Agreement
10,11,12	Principles/Practices, Team Structure	Intended to Agreement
15,16	Team Structure	Intended to Disagreement
14	Artifacts	Intended to Agreement
4,12	Values	Intended to Agreement
17	All	Intended to Agreement

7.2 Summarizing on RQ2

The second research question is intended to identify the elements of the Kanban and Scrum approaches that can be included in a hybrid model for software startups. As discussed in section 6.2.2, in the open coding phase, we delineated five categories of meanings and concepts about the coding of raw interview data. We labeled them as Category 1 through 5, each demonstrating an interconnected relationship between codes. Perry and Kaiser's model components [61], were utilized to categorize three of these groups as "policies", "mechanisms", and "structures". Additionally, two independent categories, related to deliverables and interpersonal interactions, were labeled as "tangibles" and "attitudes" respectively These categories are theoretical foundation of this research suggested model and will discussed and formulated in the next sections.

7.2.1 Category 1: Tangibles

Category 1 predominantly encompasses tangible expectation items, such as MVP[1], Responsibilities (Roles and Occupations), Task Boards[2], and Product Goal[3]. There exists an interconnected relationship among these elements and the dimensions of Mind Map Artifacts. This correlation indicates that by implementing Kanban, Scrum, or a hybrid of both, we can anticipate meeting these requirements.

To elucidate this correlation, it is imperative to triangulate the interview outcomes with survey results and mind map items. The mind map incorporates Scrum artifacts and commitments, including product goals, product backlogs, sprint goals, sprint backlogs, increments, and definitions of done, as well as Kanban artifacts such as Kanban boards, Kanban cards, and Swimlanes. In the survey section, particular emphasis was placed on Visual Task Boards, Ordered Product Backlog, action plans

[1]Minimum Viable Product
[2]Visual representations of ongoing duties and tasks during normal working or UAT phases
[3]An objective or purpose guiding explicit or implicit tasks

for goal achievement, and limited WIP[4]. Triangulating these aspects has provided us with a tangible list of components for the hybrid model, including Task Boards, a backlog of tasks to achieve specific goals and objectives, and action plans for their delivery.

7.2.2 Category 2: Structures

Category 2 comprises codes related to Team Structure, external interference, human resource constraints, and joining and leaving ceremonies. These items largely pertain to the team structure dimension depicted in the mind map. The interview results emphasized aspects such as Testing during development, Pair Working, risk management, and daily collaboration between technical and business teams.

To streamline daily operations and mitigate external interference, incorporating the Scrum-based concept of the Product Owner into the hybrid model proves beneficial. The Product Owner should be committed to collaborating with technical teams, empowered to resist external disruptions, and actively involved in risk management. Additionally, the model should establish a team with designated individuals responsible for conducting testing during development, whether following test-first principles or alternative approaches.

Whether forming a Scrum Team comprising developers with testing proficiency or a Kanban team comprising specialized personnel, it is imperative to appoint a robust Product person to lead the team effectively.

7.2.3 Category 3: Policies

This category encompasses codes related to rules and policies, including continuous delivery, Inspection/Adaptation, Transparency, Scope Management, Technical Debt, Workload-Time Balance, Limited WIP, Prioritization, Testing, UAT, and self-management. These concerns are aligned closely with the Principles/Practices dimension of the mind map and were highlighted by interviewees with similar titles.

Interpreting continuous delivery can be challenging and may lead us towards more Scrum-oriented aspects or alternative Kanban practices. Based on interview results indicating success with both Scrum and Kanban approaches, an initial prudent suggestion is to pick one discipline and adapt policies in response to unforeseen changes.

Additionally, interviewees expressed their contentment with the practice of frequent user acceptance testing as a strategy to address the issue of limited resources, identified as the primary driver of significant irregularities in work functionality and increment quality during the testing phase. Certain interviewees voiced apprehensions regarding the accumulation of technical debts due to tight deadlines, underscoring the effectiveness of implementing a service desk Kanban board in user acceptance

[4]Work in Progress

testing. This approach facilitated enhanced project oversight and yielded reductions in both defects and inconsistencies.

7.2.4 Category 4: Attitudes

This category encompasses codes related to behaviors and attitudes in fast-paced environments, including courage, motivation, commitment, transparency, facilitation, collaboration, Pair-Working, and daily work. A primary focus of our interviews was to gauge the team's resilience in the face of unexpected changes and uncertainties. This aspect is addressed within the Scrum and Kanban Values dimensions of the mind map.

The interview findings revealed respondents' adeptness in navigating challenges with resilience and steadfastness, reflecting the courage value inherent in Scrum. Triangulating these results with the survey findings provided a clearer understanding of these values. The majority of respondents emphasized the effectiveness of goal setting, self-management, transparency, and flow management. Consequently, our hybrid model must be aligned with the values of both methodologies, with a particular emphasis on Transparency, Collaboration, Respect, and Courage.

7.2.5 Category 5: Mechanisms

This category encompasses codes such as Weekly delivery, Refinement, Testing, Daily meetings, Timeboxing in meetings, and Risk management. Some of these aspects align with the events/cadences dimension in the mind map, while others may be more closely related to structures, as seen in category 2.

The interviewees emphasized iterative development, requirement engineering, continuous delivery, daily meetings, daily work, team structure, time boxing, and risk management. Given the success observed with both Scrum and Kanban approaches in interview results, an initial prudent suggestion is to pick one discipline and adapt policies in response to unforeseen changes.

7.2.6 Wrap-up and building the Theoretical foundation

The exploration into a hybrid approach for software startups reveals key insights across five categories. Firstly, in Category 1, tangible components such as MVP, Task Boards, and Product Goals form the foundation, suggesting that a hybrid model can address these through implementations of Kanban, Scrum, or a blend of both. Category 2 delves into structural aspects like Team Structure and mitigating external interference, proposing the inclusion of Scrum's Product Owner for effective management and risk mitigation. Category 3 highlights policy considerations like Continuous Delivery and Limited WIP, recommending an adaptive approach informed by both methods. Category 4 explores attitudes and behaviors essential in fast-paced environments, advocating for alignment with values from both methodologies, especially Transparency and Collaboration. Finally, Category 5 focuses on

mechanisms and events like Refinement and Daily Meetings, suggesting iterative development aligned with either Scrum or Kanban initially, with room for adaptation based on observed outcomes. These insights collectively provide a comprehensive hybrid framework that is tailored to the needs of software startups (Table 7.2).

Table 7.2: Matching the extracted category of codes with the Mind map dimensions

Questions	Categories	Mindmap Dimension	Related Item	Method
Q1,Q3	Tangibles	Artifacts	Increment, Product Goal	Scrum
Q7	Tangibles	Artifacts	Kanban Board	Kanban
Q19	Tangibles	Essence/Agenda	Kanban Board	Kanban
Q14,Q15	Structures	Team Structure	Scrum Team Structure, Kanban Roles	Scrum, Kanban
Q2 (Ongoing)	Policies	Practices/Principles	Kaizen	Kanban
Q4 (Iterative)	Policies	Essence/Agenda, Practices/Principles	Inspection/Adaptation, Value based prioritizing	Scrum
Q4 (Ongoing)	Policies	Practices/Principles	Visualize Workflow, Explicit Policies	Kanban
Q5,Q9	Policies	Practices/Principles	Value based prioritizing	Scrum
Q8	Policies	Practices/Principles	Limited WIP	Kanban
Q10	Policies	Practices/Principles	Feedback Loops	Kanban
Q11	Policies	Practices/Principles	Self Management	Scrum
Q13	Attitudes	Values	Courage, Respect	Scrum
Q16 (Iterative)	Attitudes	Practices/Principles	Collaboration, Self Management,	Scrum
Q16 (Ongoing)	Attitudes	Values	Collaboration, Transparency	Kanban
Q2 (Iterative)	Mechanisms	Events/Cadences	Sprints	Scrum
Q6	Mechanisms	Events/Cadences	Backlog Refinement	Scrum
Q10	Mechanisms	Events/Cadences	Service delivery review	Kanban
Q12 (Iterative)	Mechanisms	Events/Cadences	Daily Scrum	Scrum
Q12 (Ongoing)	Mechanisms	Events/Cadences	Daily Stand-up	Kanban
Q17	Mechanisms	Practices/Principles	Time Boxing	Scrum
Q18	Mechanisms	Events/Cadences	Risk Review, Strategy review	Kanban

7.3 Model Creation

To formulate boundaries of the conceptual model, we have organized the principal ideas embodying the core essence into overarching categories Figure 7.2 illustrates the array of causal relationships depicted by arrows, among the categories, represented

by blocks. In the forthcoming explanation of the model, we utilized the names of the identified categories as depicted in Figure 7.2. The model revolves around the central category, namely "mechanisms", serving as the most interconnected node in the theory, thereby reflecting the essence and holding the paramount explanatory position. As shown in Table 7.2, The events/cadences and principles/practices of Scrum and Kanban offer diverse insights for shaping a team's working mechanism. These insights are applicable in both ongoing and iterative modes and can be effectively utilized in the software startup context.

One prevalent contextual condition defining nearly every software startup is its team/organization structure. With limited access to human resources, proficient administration, time, and intellectual assets, startups face constraints in supporting their engineering activities. The team/organization structure notably influences other aspects, including people's attitudes and working methods, prompting a heightened emphasis on implementing a crucial set of functionalities reflected in tangible items such as MVP scope management and product goals. As depicted in Table 7.2, both Scrum and Kanban advocate for an appropriate team configuration to address challenges arising from limited human resources. Both methodologies propose the inclusion of a product person and a delivery leader, alongside a team of developers. The descriptions of these roles are interchangeable within their respective contexts, allowing for a flexible approach to tailoring an effective team structure.

Consistently echoed by interviewees, a prevalent concept within startup teams emerges from individuals' behaviors in response to the company's rapid-paced environment and demanding daily operations. We identified this category as "attitudes" influenced by team structure, encompassing its effects on team working mechanisms, applied practices, and policies. This concept is mirrored in the core Values and Principles/Practices of Scrum and Kanban, as elucidated and reinforced by the developed mind map. As indicated in Table 7.2, our research revealed that the most frequently mentioned aspects were respect, courage, collaboration, transparency, and self-management, with half originating from Scrum and the remainder from Kanban. Consequently, the hybrid model can incorporate prominent elements from both methodologies.

In contrast to Attitudes, which are intangible and must be implicitly considered in model creation within a startup context, interviews clearly emphasized two other categories. The first, titled "policies" aligns with equivalent concepts in Scrum and Kanban methods, referred to as "Principles/Practices" and "Essence/Agenda", encompassing ideas such as Continuous Integration or Kaizen, Pillars of Scrum, visualizing workflow, value-based prioritization, limited WIP, and Feedback Loops. This research primarily concentrated on assessing various requirements essential for software startups, refraining from detailed proposals. Nonetheless, certain interconnected concepts impact other dimensions, as illustrated in the conceptual model depicted in Figure 7.2

Another visible category is "Tangibles", representing the tangible outcomes teams

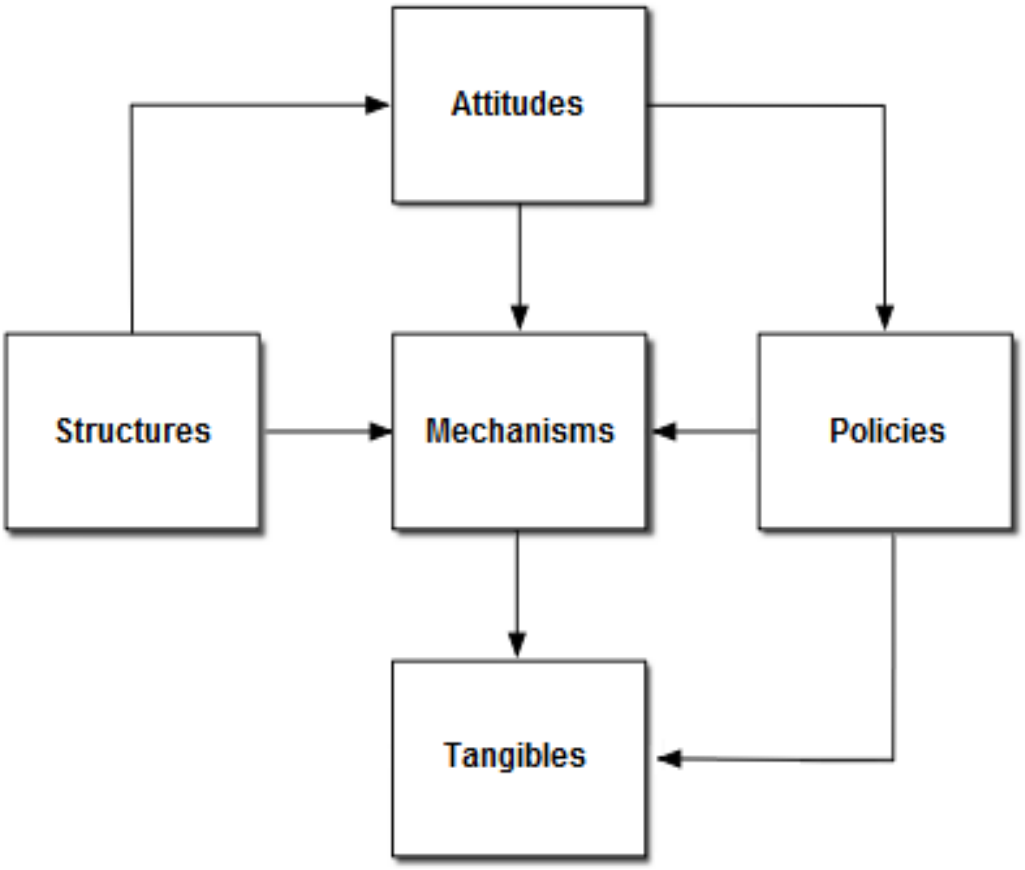

Figure 7.2: Key categories and relationships in the hybrid Scrum-Kanban Startup Model.

aim to deliver. In the developed mind map, these are labeled as "Artifacts", As evidenced by the interview findings from the conducted case study, the essential set of artifacts necessary for a software startup to operate effectively is encompassed and recommended by both Scrum and Kanban methodologies. Consequently, they can also be integrated into a hybrid model. These tangibles outcome are influenced by Mechanisms and Policies as shown in Figure 7.2.

Chapter 8

Conclusions and Future Work

The research provided insights into the impacts of Scrum and Kanban methodologies on software development within startup environments and the potential for integrating elements of both approaches into a hybrid model. The findings suggest that Scrum and Kanban offer valuable strengths that can be leveraged in startup contexts. Scrum's emphasis on iterative and incremental work and shorter iterations aligns well with the need for adaptable and efficient processes in startup environments. On the other hand, Kanban's focus on visualizing workflow and managing flow addresses challenges such as workload management common in startups.

By systematically deconstructing both methodologies and comparing their fundamental elements, the research identified key dimensions and components that can inform the development of a hybrid approach. These dimensions include tangibles such as product goals, MVP's scope management and task boards, structures such as team composition and external interference management, policies like value-based prioritization, feedback loops, continuous integration, and workload-time balance, attitudes emphasizing courage, respect, self-management, transparency and collaboration, and mechanisms such as daily work and risk management.

The proposed hybrid model seeks to capitalize on the strengths of both Scrum and Kanban while adapting to the specific needs and challenges of startup environments. It emphasizes flexibility and adaptability, allowing startups to initially adopt one methodology and adjust policies and practices as needed based on observed outcomes.

Overall, the research provides a comprehensive framework for implementing a hybrid approach tailored to the unique circumstances of software startups. By considering the insights gleaned from both the survey data and interview results, startups can make informed decisions about their software development processes, ultimately enhancing their agility and effectiveness in a dynamic industry landscape.

8.1 Future Works

As explored in Section 6.1.4, our findings reveal a considerable degree of variability in the data, suggesting promising avenues for future research in this domain. Specifically, investigating the effects of visualizing flow within the Kanban method and embracing empiricism principles inherent in Scrum, such as transparency, inspec-

tion, and adaptation, within the unique contexts of startups, offers fertile ground for further exploration.

Although the questionnaire results indicated that the selected aspects of both methods are applicable to software startups of various sizes and stages, examining the scalability of these aspects was beyond the scope of this study. Investigating how startups grow and evolve using agile methods, particularly the model proposed in this research, would be a promising direction for further study.

Moreover, our research underscores the significance of risk management in startup environments, particularly the detrimental effects of inadequate daily collaboration between business stakeholders and technical teams. Addressing this issue comprehensively could serve as a focal point for future investigations in the realm of software startup management.

Additionally, our study sheds light on the interplay between the testing process in startups and the mitigation of technical debts in the fast-paced landscape of engineering activities. While our research has begun to elucidate this relationship, there remains a need for more in-depth exploration in future studies to fully comprehend and address these intertwined concepts effectively.

Lately, Our study was designed with a specific focus on understanding the successful elements of the hybrid approach in startups. Including a detailed discussion on challenges and barriers may divert attention from the primary objectives and dilute the focus of the research. Keeping the scope narrow allows for a more in-depth analysis of the main research questions. Highlighting challenges and barriers along with strategies to overcome them could be framed as an area for future research. This approach acknowledges the importance of the topic while maintaining the integrity and focus of the current study. Suggesting this as a future research direction also opens up opportunities for further studies to build upon the current findings.

References

[1] "What is scrum?" [Accessed May 2., 2024]. [Online]. Available: https://www.scrum.org/learning-series/what-is-scrum/

[2] "State of agile report," November 2023, [Accessed May 2., 2024]. [Online]. Available: https://digital.ai/resource-center/analyst-reports/state-of-agile-report/

[3] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: Review and analysis," *arXiv preprint arXiv:1709.08439*, 2017.

[4] P. C. Abrantes and A. P. Furtado, "Agile development practices applied to software startups: A systematic mapping review," in *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2021, pp. 1–6.

[5] M. Alqudah and R. Razali, "A comparison of scrum and kanban for identifying their selection factors," in *2017 6th International Conference on Electrical Engineering and Informatics (ICEEI)*. IEEE, 2017, pp. 1–6.

[6] C. Alves, J. Cunha, and J. Araújo, "On the pragmatics of requirements engineering practices in a startup ecosystem," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*. IEEE, 2020, pp. 311–321.

[7] D. J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*. United States, Washington, Seattle: Blue Hole Press, 2010.

[8] D. J. Anderson and A. Carmichael, *Essential Kanban Condensed*. United States, Washington, Seattle: Blue Hole Press, 2016.

[9] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries *et al.* (2001, February) Manifesto for agile software development. [Accessed May 2., 2024]. [Online]. Available: https://agilemanifesto.org/

[10] V. Berg, J. Birkeland, A. Nguyen-Duc, I. O. Pappas, and L. Jaccheri, "Software startup engineering: A systematic mapping study," *Journal of Systems and Software*, vol. 144, pp. 255–274, 2018.

[11] S. Bick, K. Spohrer, R. Hoda, A. Scheerer, and A. Heinzl, "Coordination challenges in large-scale software development: a case study of planning misalignment in hybrid settings," *IEEE Transactions on Software Engineering*, vol. 44, no. 10, pp. 932–950, 2017.

[12] S. Blank, *The four steps to the epiphany: successful strategies for products that win*. John Wiley & Sons, 2020.

[13] S. Blank and B. Dorf, *The startup owner's manual: The step-by-step guide for building a great company.* John Wiley & Sons, 2020.

[14] A. Boes and T. Kämpf, "Agile methods, lean development and the change of work in software development," in *Future Business Software: Current Trends in Business Software Development.* Springer, 2014, pp. 83–92.

[15] P. Brereton, B. Kitchenham, D. Budgen, and Z. Li, "Using a protocol template for case study planning," in *12th International conference on evaluation and assessment in software engineering (EASE).* BCS Learning & Development, 2008.

[16] D. Brunner, J. Münch, and M. Kuhrmann, "Entrepreneurial software engineering: towards a hybrid development method for early-stage startups," *WI-MAW-Rundbrief*, vol. 27, no. 1, pp. 5–15, 2021.

[17] B. H. Cavalcante, G. C. L. Leal, R. Balancieri, and I. de Farias Junior, "Technical aspects of software development in startups: A systematic mapping," in *2018 XLIV Latin American Computer Conference (CLEI).* IEEE, 2018, pp. 100–109.

[18] J. Chen, "The using of mind map in concept design," in *2008 9th International Conference on Computer-Aided Industrial Design and Conceptual Design.* IEEE, 2008, pp. 1034–1037.

[19] E. C. Conforto and D. C. Amaral, "Agile project management and stage-gate model—a hybrid framework for technology-based companies," *Journal of Engineering and Technology Management*, vol. 40, pp. 1–14, 2016.

[20] R. G. Cooper and A. F. Sommer, "Agile-stage-gate: New idea-to-launch method for manufactured new products is faster, more responsive," *Industrial Marketing Management*, vol. 59, pp. 167–180, 2016.

[21] J. W. Creswell and C. N. Poth, *Qualitative inquiry and research design: Choosing among five approaches.* Sage publications, 2016.

[22] M. Crowne, "Why software product startups fail and what to do about it. evolution of software product development in startup companies," in *IEEE International Engineering Management Conference*, vol. 1. IEEE, 2002, pp. 338–343.

[23] E. F. C. d. Cruz, F. E. Fernandes Junior, and E. D. Sardinha, "An experience in the use of scrum and kanban for project development in a waterfall environment," in *Proceedings of the XX Brazilian Symposium on Software Quality*, 2021, pp. 1–7.

[24] N. Damij and T. Damij, "An approach to optimizing kanban board workflow and shortening the project management plan," *IEEE Transactions on Engineering Management*, 2021.

[25] N. M. Devadiga, "Software engineering education: Converging with the startup industry," in *2017 IEEE 30th conference on software engineering education and training (CSEE&T).* IEEE, 2017, pp. 192–196.

[26] K. Erhardsson and O. Ljungberg, "Team setup and ways of working at Ericsson: A study on system test teams," Chalmers Univerity of Technology, Gothenburg, Sweden, 2017.

[27] T. S. et. al., *A guide to the scrum body of knowledge (SBOK guide)*. SCRUMstudy, A brand of VMEdu, Inc., 2016.

[28] T. J. Gandomani, H. Zulzalil, A. Javanmardi, A. A. A. Ghani, A. B. M. Sultan, and M. Z. Nafchi, "How pre-start up assessment helps software companies in agile transition," *Science International*, vol. 25, pp. 1125–1130, 2013.

[29] J. Gerring, "What is a case study and what is it good for?" *American political science review*, vol. 98, no. 2, pp. 341–354, 2004.

[30] C. Giardino, N. Paternoster, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, "Software development in startup companies: the greenfield startup model," *IEEE Transactions on Software Engineering*, vol. 42, no. 6, pp. 585–604, 2015.

[31] C. Giardino, M. Unterkalmsteiner, N. Paternoster, T. Gorschek, and P. Abrahamsson, "What do we know about software development in startups?" *IEEE software*, vol. 31, no. 5, pp. 28–32, 2014.

[32] C. Giardino, X. Wang, and P. Abrahamsson, "Why early-stage software startups fail: a behavioral framework," in *Software Business. Towards Continuous Value Delivery: 5th International Conference, ICSOB 2014, Paphos, Cyprus, June 16-18, 2014. Proceedings 5*. Springer, 2014, pp. 27–41.

[33] V. Gupta, J. M. Fernandez-Crehuet, T. Hanne, and R. Telesko, "Requirements engineering in software startups: A systematic mapping study," *Applied Sciences*, vol. 10, no. 17, p. 6125, 2020.

[34] R. Iliyasu and I. Etikan, "Comparison of quota sampling and stratified random sampling," *Biom. Biostat. Int. J. Rev*, vol. 10, no. 1, pp. 24–27, 2021.

[35] P. Iversen, P. B. Johansen, D. Leebeek, E. Morch, E. F. Olsen, and S. B. Skarås, "Supporting the management of software development processes in startups," Norwegian University of Science and Technology, Trondheim, Norway, 2018.

[36] P. Jain, A. Sharma, and L. Ahuja, "The impact of agile software development process on the quality of software product," in *2018 7th international conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO)*. IEEE, 2018, pp. 812–815.

[37] K.-K. Kemell, A. Nguyen-Duc, X. Wang, J. Risku, and P. Abrahamsson, "Software startup essence: How should software startups work?" in *Fundamentals of Software Startups: Essential Engineering and Business Aspects*. Springer, 2020, pp. 97–109.

[38] L. Keogh, "Scrum and kanban both the same, only different," November 2011, [Accessed May 2., 2024]. [Online]. Available: https://lizkeogh.com/2011/11/20/scrum-and-kanban-both-the-same-only-different/

[39] E. Klotins, M. Unterkalmsteiner, P. Chatzipetrou, T. Gorschek, R. Prikladnicki, N. Tripathi, and L. B. Pompermaier, "A progression model of software engineering goals, challenges, and practices in start-ups," *IEEE Transactions on Software Engineering*, vol. 47, no. 3, pp. 498–521, 2019.

[40] H. Kniberg and M. Skarin, *Kanban and Scrum-making the most of both.* Research Triangle, North Carolina, United States: C4Media Incorporated, 2010.

[41] R. S. Kreitchmann, F. J. Abad, V. Ponsoda, M. D. Nieto, and D. Morillo, "Controlling for response biases in self-report scales: Forced-choice vs. psychometric modeling of likert items," *Frontiers in psychology*, vol. 10, p. 2309, 2019.

[42] M. Kuhrmann, P. Diebold, J. Münch, P. Tell, V. Garousi, M. Felderer, K. Trektere, F. McCaffery, O. Linssen, E. Hanser *et al.*, "Hybrid software and system development in practice: waterfall, scrum, and beyond," in *Proceedings of the 2017 international conference on software and system process*, 2017, pp. 30–39.

[43] M. Kuhrmann, P. Tell, R. Hebig, J. Klünder, J. Münch, O. Linssen, D. Pfahl, M. Felderer, C. R. Prause, S. G. MacDonell *et al.*, "What makes agile software development agile?" *IEEE transactions on software engineering*, vol. 48, no. 9, pp. 3523–3539, 2021.

[44] M. Kuhrmann, P. Tell, J. Klünder, R. Hebig, S. Licorish, and S. MacDonell. (2018) Helena stage 2 results. [Accessed May 2., 2024]. [Online]. Available: http://dx.doi.org/10.13140/RG.2.2.14807.52649

[45] O. Kuru and J. Pasek, "Improving social media measurement in surveys: Avoiding acquiescence bias in facebook research," *Computers in Human Behavior*, vol. 57, pp. 82–92, 2016.

[46] H. Lei, F. Ganjeizadeh, P. K. Jayachandran, and P. Ozcan, "A statistical analysis of the effects of scrum and kanban on software development projects," *Robotics and Computer-Integrated Manufacturing*, vol. 43, pp. 59–67, 2017.

[47] V. Lobellova, "The history of scrum: How, when and why'," 2020. [Online]. Available: https://www.scrumdesk.com/the-history-of-scrum-how-when-and-why/

[48] C. Matthies, "Scrum2kanban: integrating kanban and scrum in a university software engineering capstone course," in *Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials*, 2018, pp. 48–55.

[49] J. Melegati, "What influences software startups to use lean startup?" in *Proceedings of the 19th International Conference on Agile Software Development: Companion*, 2018, pp. 1–3.

[50] J. Melegati, A. Goldman, F. Kon, and X. Wang, "A model of requirements engineering in software startups," *Information and software technology*, vol. 109, pp. 92–107, 2019.

[51] E. Mkpojiogu, N. Hashim, A. Al-Sakkaf, and A. Hussain, "Software startups: Motivations for agile adoption," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 8S, pp. 454–459, 2019.

[52] J. M. Morse, "Theoretical saturation," *Encyclopedia of social science research methods*, vol. 3, pp. 1122–3, 2004.

[53] C. A. Moser and A. Stuart, "An experimental study of quota sampling," *Journal of the Royal Statistical Society. Series A (General)*, vol. 116, no. 4, pp. 349–405, 1953.

[54] A. Nguyen-Duc, K.-K. Kemell, and P. Abrahamsson, "The entrepreneurial logic of startup software development: A study of 40 software startups," *Empirical Software Engineering*, vol. 26, pp. 1–55, 2021.

[55] A. Nguyen-Duc, S. M. A. Shah, and P. Ambrahamsson, "Towards an early stage software startups evolution model," in *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2016, pp. 120–127.

[56] A. Nguyen-Duc, X. Wang, and P. Abrahamsson, "What influences the speed of prototyping? an empirical investigation of twenty software startups," in *Agile Processes in Software Engineering and Extreme Programming: 18th International Conference, XP 2017, Cologne, Germany, May 22-26, 2017, Proceedings 18*. Springer International Publishing, 2017, pp. 20–36.

[57] S. U. Nisa and M. R. J. Qureshi, "Empirical estimation of hybrid model: A controlled case study," *IJ Information Technology and Computer Science*, vol. 4, no. 8, pp. 43–50, 2012.

[58] F. L. Oliva and M. Kotabe, "Barriers, practices, methods and knowledge management tools in startups," *Journal of knowledge management*, vol. 23, no. 9, pp. 1838–1856, 2019.

[59] J. Pantiuchina, M. Mondini, D. Khanna, X. Wang, and P. Abrahamsson, "Are software startups applying agile practices? the state of the practice from a large survey," in *Agile Processes in Software Engineering and Extreme Programming: 18th International Conference, XP 2017, Cologne, Germany, May 22-26, 2017, Proceedings 18*. Springer International Publishing, 2017, pp. 167–183.

[60] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, "Software development in startup companies: A systematic mapping study," *Information and Software Technology*, vol. 56, no. 10, pp. 1200–1218, 2014.

[61] D. E. Perry and G. E. Kaiser. (1989) Models of software development environments. [Accessed May 2., 2024]. [Online]. Available: https://academiccommons.columbia.edu/doi/10.7916/D81261G0

[62] M. Poppendieck, "Lean software development," in *29th International Conference on Software Engineering (ICSE'07 Companion)*. IEEE, 2007, pp. 165–166.

[63] R. Razali and M. Alqudah, "An empirical study of scrumban formation based on the selection of scrum and kanban practices," *International Journal on Advance Science Engingeering Information Technology*, vol. 8, no. 6, pp. 2315–2322, 2018.

[64] F. J. Rego Lopes and F. Petrillo, "Simkan: Training kanban practices through stochastic simulation," in *Agile Methods: 7th Brazilian Workshop, WBMA 2016, Curitiba, Brazil, November 7-9, 2016, Revised Selected Papers 7*. Springer, 2017, pp. 110–121.

[65] E. Ries, *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Currency, 2011.

[66] M. Sakikhales, "Nonlinear project management: Agile, scrum and kanban for the construction industry," in *Industry 4.0 for the Built Environment: Methodologies, Technologies and Skills*. Springer, 2021, pp. 227–246.

[67] W. E. Saris, M. Revilla, J. A. Krosnick, and E. M. Shaeffer, "Comparing questions with agree/disagree response options to questions with construct-specific response options," *Survey Research Methods. 2010; 4 (1): 61-79*, 2010.

[68] K. Schwaber and J. Sutherland. (2020, November) The 2020 scrum guide. [Accessed May 2., 2024]. [Online]. Available: https://scrumguides.org/scrum-guide.html

[69] R. Souza, K. Malta, and E. S. De Almeida, "Software engineering in startups: a single embedded case study," in *2017 IEEE/ACM 1st International Workshop on Software Engineering for Startups (SoftStart)*. IEEE, 2017, pp. 17–23.

[70] R. Souza, K. Malta, R. Silva, P. Masiero, E. Almeida, and I. Machado, "A case study about startups' software development practices: A preliminary result," in *Proceedings of the XVIII Brazilian Symposium on Software Quality*, 2019, pp. 198–203.

[71] R. E. Stake, *The art of case study research*. United States , California 91320, Thousand Oaks: SAGE Publications, Inc., 1995.

[72] N. C. Surendra and S. Nazir, "Agile development: Exploring what practitioners want to know," *Journal of Software Engineering and Applications*, vol. 11, no. 1, pp. 1–11, 2018.

[73] C. Tam, E. J. da Costa Moura, T. Oliveira, and J. Varajão, "The factors influencing the success of on-going agile software development projects," *International Journal of Project Management*, vol. 38, no. 3, pp. 165–176, 2020.

[74] E. Tegegne, "Software development methodologies and practices in startups: systematic literature review," Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland, 2018.

[75] E. W. Tegegne, P. Seppanen, and M. O. Ahmad, "Software development methodologies and practices in start-ups," *IET Software*, vol. 13, no. 6, pp. 497–509, 2019.

[76] C. Tolfo, R. S. Wazlawick, M. G. G. Ferreira, and F. A. Forcellini, "Agile practices and the promotion of entrepreneurial skills in software development," *Journal of Software: Evolution and Process*, vol. 30, no. 9, p. e1945, 2018.

[77] M. Unterkalmsteiner, P. Abrahamsson, W. XiaoFeng, A. Nguyen-Duc, S. Shah, S. Shahid Bajwa, G. H. Baltes, K. Conboy, E. Cullina, D. Dennehy, H. Edison, C. Fernandez-Sanchez, J. Garbajosa, T. Gorschek, E. Klotins, L. Hokkanen, F. Kon, I. Lunesu, M. Marchesi, L. Morgan, M. Oivo, C. Selig, P. Seppänen, R. Sweetman, P. Tyrväinen, C. Ungerer, and A. Yagüe, "Software startups - a research agenda," *e-Informatica Vol. X*, pp. 2016; ISSN 1897–7979, 2016. [Online]. Available: http://www.e-informatyka.pl/attach/e-Informatica_-_Volume_10/eInformatica2016Art5.pdf

[78] M. Usman and W. Vanhaverbeke, "How start-ups successfully organize and manage open innovation with large companies," *European Journal of Innovation Management*, vol. 20, no. 1, pp. 171–186, 2017.

[79] A. N. A. Wahab, M. Mukhtar, and R. Sulaiman, "A conceptual model of lean manufacturing dimensions," *Procedia Technology*, vol. 11, pp. 1292–1298, 2013.

[80] M. Williams and T. Moser, "The art of coding and thematic exploration in qualitative research," *International management review*, vol. 15, no. 1, pp. 45–55, 2019.

[81] K. Yang and A. Banamah, "Quota sampling as an alternative to probability sampling? an experimental study," *Sociological research online*, vol. 19, no. 1, pp. 56–66, 2014.

[82] R. K. Yin, "Discovering the future of the case study. method in evaluation research," *Evaluation practice*, vol. 15, no. 3, pp. 283–290, 1994.

[83] S. Yordanova and K. Toshkov, "An agile methodology for managing business processes in an it company," *Business Management*, vol. 3, pp. 27–90, 2019.

[84] T. Zavazava, "Project management methodologies for software development in startups," Faculty of Social Sciences, Department of Computer and Systems Sciences, Stockholm University, Sweden, 2022.

[85] M. Zorzetti, I. Signoretti, L. Salerno, S. Marczak, and R. Bastos, "Improving agile software development using user-centered design and lean startup," *Information and Software Technology*, vol. 141, p. 106718, 2022.

Appendix A

Survey Questions

Topics/ Principles/ Practices	Questions
Demographic Questions	- Participant Age/ Profession/ Skills/ Job - Software Startup Experience (Yes/No) - Software Startup Size Worked for - The phase of Startup Company Formation - Duration of Collaboration
Iterative RE and Software Development	- The company allowed the team to iteratively refine and develop their product/project requirements and deliverables.
Sustainable Pace in SDLC	- The team achieved an enhanced sustainable pace in releasing work during the iterations - The team(s) effectively managed workloads and established a sustainable pace of workflow
Iteration Size Effects	- Implementing shorter iterations reduced the delay in team's delivery
Visualization of Flow	- Visual task boards clarified work for the team and facilitated a quicker completion - Visualizing the work facilitated prioritization for our team regarding their tasks

Topics/ Principles/ Practices	Questions
Limited WIP	- There was an opportunity for the team to manage and limit work in progress
Estimation and Prioritization	- Estimating tasks aided the team in identifying better prioritization - An ordered backlog of tasks let the team to choose the next task promptly after completing the current one
Testing first or during the development	- There was an opportunity for team to test during the development - Testing first caused a tangible decrease in the time needed to complete tasks
Pair Working / Programming	- Pair working on tasks helped team to resolve them more accurate and quickly
Daily Events	- Short Daily Meetings reduced the need for other unplanned ad hoc meetings during workdays - Business and technical teams collaborated daily toward the goals of the product/project
Goal Setting and Risk Management	- Business owners recognize and assess risks and effectively communicating them to dev team - The team formulated actionable plans for delivering their goal during planning events
Customizing Method	- Removed certain elements from Scrum/Kanban implementation

Appendix B

Interview Questions

Row	Questions
1	What was your main responsibility regarding the "Re-Platforming" Project?
2	Has weekly delivery caused a better speed in doing tasks? (If applicable)
3	"Re-Platforming" as the product goal was helpful to have more focus regarding the tasks and project.
4	Did you work with your team in iteration mode or ongoing? What went well with your rhythms? What was not good?
5	How did you manage to keep a balance between the work volume and the time to allocate?
6	How did you refine the project backlog? (If applicable)
7	How the tasks were transparent? Were "JIRA" task boards helpful in managing the team?
8	Did you limit work in progress toward deadlines and tasks? How did it go into action during the project?
9	Did you have a chance to prioritize the work with your teammates? Did it cause a better focus on delivering them?
10	How did you approach the test cycle? At first? In the middle? At the end?
11	Did you have enough autonomy to decide on details regarding tasks and how to develop them?
12	Were daily Meetings helpful for the team?
13	Did the team have enough courage to do the right work regarding their tasks?
14	Did the team structure and the number of team members was good for the project and tasks?
15	How do joining and leaving people effect on the project and teamwork?
16	Did you have daily work with Business/Technical people and how to evaluate it?
17	Did Meetings happen in their time boxes? What was the effects?
18	Did you recognize risks before starting the tasks?
19	How does the Shop-Ops Kanban board help the team? Did you use it? How evaluate it?
	Note: Shop-Ops *was a service desk Kanban board in the User acceptance testing phase in which internal company employees asked to use different aspects of the new platform and register tickets if they found a bug or inconsistencies.*

Appendix C

Interview Transcription

Respondent Number	Question 1: What was your main responsibility regarding the "Re-Platforming" Project?
P1	Backend Lead Developer - I was responsible to resolve external parties concerns
P2	Frontend Lead Developer - I was responsible for delivering seller center
P3	QA/QC Leader - Check Backend/Frontend codes sanity
P4	Data Specialist/Data Migration - Regenerate Data dashboards frequently
P5	Product Manager - Build Seller Panels and Deliver Features to Stakeholders
P6	Product Leader in Operation tribe - Satisfying Customer Care, Delivery, and Fulfilment of technical and business requirements
P7	Scrum Master - Team's dynamism, productivity and effectiveness
P8	Marketing - Recognizing new Features and website display be suitable for users codes sanity
P9	Commercial - Evaluate Seller Panels, UAT, all sellers and supply processes
P10	Operation - Preparing call center team for employing new platform and Extracting new workflow and processes and teaching people about them

Respondent Number	Question 2: Has weekly delivery caused a better speed in doing tasks? (If applicable)
P1	It was N/A for our team. We worked in the Kanban method with an Ongoing approach.
P2	Yes, weekly delivery worked but we considered them as weeks, not sprints. It facilitated to achievement of smaller tasks, better outcomes, met deadlines better, quick decisions, and better feelings about the work.
P3	No, the delivery rate improved during the project's evolution. Team Motivation and primary energy was more helpful
P4	Yes, we had a weekly release plan on data dashboards and other tasks.
P5	Yes, weekly delivery worked for us. It helped us to achieve smaller tasks, better estimation, no last-seconds delivery, and a sorted backlog.
P6	Not exactly, Weekly delivery needs to have a good understanding of the scope. We worked in Ongoing mode and faced a complex scope and unwanted changes
P7	Generally Yes, weekly delivery worked for us but not for all teams. It facilitated quick wrap-up and roll-out of agile principles and practices. It was close to people's desires while not fitted for all teams and based on our experience it was better than 2 weeks sprints.
P8	It was N/A for us. We worked continuously on our huge backlog. We tried to see the whole continuously and used different guidelines like less chatting and having more focus on daily work.
P9	It was N/A for us. We worked continuously on our huge backlog. We tried to discover issues continuously and used different guidelines like documenting and escalating issues and On-demand panel testing.
P10	It was N/A for us. We worked continuously on our huge backlog. We used different practices like quick response and working shifts.

Respondent Number	Question 3: "Re-Platforming" as the product goal was helpful to have more focus regarding the tasks and project.
P1	Yes, as a shared target and intention, it gave us a synergy about the purpose of the work and tasks.
P2	Yes, as a clear final goal, it gave us a better feeling about the purpose of the work, amplified the ownership and built synergy in teams.
P3	Yes, as a SMART goal, it was helpful to have more focus on tasks and dealing with work overloads.
P4	Yes, as the product goal, it was essential and directed us to a better outcome. It helped us on regenerating data dashboards, important reports and adding absent data dashboards.
P5	Yes, as a clear product goal, it was fine. We focused more on the part of the scope that needed more effort. It generally decreased unknowns.
P6	Yes, it was a fantastic goal and motivated me to participate in the project. We are always faced with unpredictability but it was a relevant goal.
P7	Yes, as a tangible target, it motivated people and made unity across technical and other people.
P8	Yes, as the product goal, it was fine and managed the work journey. It gave us a direction in our daily works.
P9	No, it was not a product goal for us and we adapted our work to it. We considered it as a tool and our specific goal was conducting UAT and negotiating with sellers as usual.
P10	No, it was not a product goal for us and we adapted our work to it. We considered it as a tool and our specific goal was the automation and Integration of data of clients.
	SMART: *Specific, Measurable, Achievable, Relevant, and Time-Bound.*

Respondent Number	Question 4: Did you work with your team in iteration mode or ongoing? What went well with your rhythms? What was not good?
P1	No, we worked in Kanban and ad hoc modes due to unpredictable requirements and issues. We faced a wide scope and strange changes. To deal with the situation We worked in pair mode.
P2	Yes, we worked in Sprints, but in the last weeks, most teams worked ad hoc. generally, it caused a better outcome but lots of work injections happened in iterations.
P3	Yes, we worked in Sprints, but in the last weeks, most teams worked ad hoc. We could not keep the boundaries of iteration-based working.
P4	No, it was N/A for us. We worked continuously toward our Gantt chart backlog items. We had a very heavy first 3 weeks but dashboards were delivered to stakeholders on time.
P5	Yes, we worked in Sprints. Inspection and Adaptation of the work were done properly by the team toward sprint goals and the whole project.
P6	No, we worked in the Kanban approach, continuously toward project milestones. But we did Inspection/Adaptation every day via daily meetings. We faced unknownness frequently regarding the project's progress.
P7	I served more than 1 team, some teams had Sprints and some were Kanban. The Inspection and Adaptation of the work happened on a daily basis. Teams swarmed on targets and tried to make work Transparent.
P8	No, it was N/A for us. We worked continuously on our backlog tasks.
P9	No, it was N/A for us. We worked continuously on our backlog tasks. Working in a pair mode went well. However, we dealt with an inconsistent work scope.
P10	No, it was N/A for us. We worked continuously on our backlog tasks.

Respondent Number	Question 5: How did you manage to keep a balance between the work volume and the time to allocate?
P1	We managed the workload and focused on the most important parts of the work. We employed a light code review approach and we focused on a shorter scope, refactoring, and accepting technical debt to achieve time to market.
P2	There was no balance between the volume of the work and the remaining time. We worked on a shorter scope to pass the situation and it caused technical debts for the future.
P3	Although we picked a partial scope of the work, we had to work overtime.
P4	There was no balance between time and deadlines, so we worked overtime because data is an atomic scope and usually cannot broken into smaller units of work and there was a business push frequently.
P5	Although we picked a partial scope of work, we had to work overtime. We used the JIRA dashboard for more transparency and to control the flow of the work.
P6	Although we picked a partial scope of work, we had to work overtime. There was a continuous business push on targets continuously. To keep the project timeline, we embraced working in a pressure situation and tried to limit work in progress.
P7	Although we picked a partial scope of work, we had to work overtime. We visualized the work by employing Task boards and controlled the flow of the work by limiting work in progress. We combated against obstacles by observing team performance metrics and conducted a process improvement operation to keep the project timeline.
P8	We implemented a trade-off SLA, allocating 30% of resources to the legacy platform tasks and 70% to the new platform tasks. To ensure greater focus on the new platform, we paused all legacy platform activities one month before the launch date.
P9	We implemented a trade-off SLA, allocating 70% of resources to the legacy platform tasks and 30% to the new platform tasks. Additionally, we conducted webinars to raise seller awareness and manage data during the transition period, enhancing product data within the company.
P10	Although we picked a partial scope of work, we had to work overtime.

Respondent Number	Question 6: How did you refine the project backlog? (If applicable for your scope)
P1	We refined the product backlog via several meetings inside the team. Some of this meetings took between 2-3 hours.
P2	We conducted daily work with both business and technical team members, involving 3-4 key participants in partial grooming disciplines. The rest of the team was informed of the results and decisions. Typically, 1-2 hours were sufficient to summarize these events.
P3	We used frequent refinement sessions to identify the tasks' definitions.
P4	We used a focus group to collect stakeholders' fundamental data requirements. Sometimes, we worked asynchronously and to assess the needs we collected descriptions and dates of issue.
P5	We conducted daily work with both business and technical team members. Moreover, we used PRDs to collect enough concepts and facts regarding each refinement process.
P6	We refined the product backlog via several meetings inside the team. These meetings helped us to be on the same page at teal-level of tasks and targets.
P7	We refined the backlog frequently during the project process in refinement meetings. PMs prepared PRD to start, We modified tasks based on them during the sessions and mostly used Planning Pizza and Story Mapping techniques to make the work feasible and achievable.
P8	We gathered ideas once and extracted tasks in several Spike periods. We used brainstorming with stakeholders to manage their requirements and follow up on the work progress with PMs directly or via a Focus Group.
P9	It was not applicable for us. We defined the scope and requirements at the beginning and completed the refinement process at that time.
P10	It did not apply to us. We continuously worked on our plans and tasks with dedicated ownership, operating 24/7 across three working shifts, with a focus on detail-oriented execution.

Respondent Number	Question 7: How the tasks were transparent? Were "JIRA" task boards helpful in managing the team?
P1	Generally, task boards were useful for tracking the project's progress, though the JIRA boards became crowded. I prefer an alternative approach to tools that enhances the transparency of status, progress, and seeds.
P2	Task boards, particularly JIRA, were instrumental in advancing the project, providing transparency in status updates, fostering a passion for resolving tasks, and delivering good outcomes. They also instilled a sense of discipline in our workflow.
P3	Task boards, especially when using JIRA, significantly contributed to the project's progress by enhancing transparency and providing visibility into the team's work. However, the team struggled with the physical board approach, finding it ineffective.
P4	We didn't utilize online or physical task boards; instead, we relied on a single-sheet Excel file. Tracking status wasn't necessary as being informed when tasks were completed sufficed for transparency.
P5	Task boards significantly contributed to the project's progress. We observed them during Daily Scrum meetings and followed up on TIA. They facilitated task ordering and helped identify next steps.
P6	ask boards were invaluable for advancing the project, especially within the fulfillment team. They enhanced transparency, prioritized tasks, and improved follow-up and visualization, enabling quick decision-making. Physical boards were not suitable.
P7	While task boards were beneficial, a third of ad hoc issues were missed, hindering transparency. Nevertheless, they effectively visualized work, improved follow-up opportunities, and highlighted product and sprint goals, fostering commitment. Despite these challenges, half of the teams successfully delivered iteratively.
P8	We used JIRA task boards just for reporting issues to the tech teams, not for supporting clients. To manage our tasks, we relied on a time-bounded single-sheet Excel system.
P9	Task boards were instrumental in advancing the project's progress. They enhanced transparency and visibility of work, fueled our passion for resolving tasks, and reduced unnecessary chatting.
P10	We used JIRA solely for reporting issues to the tech team, not for client support. Our processes were documented in an Excel sheet. Although the physical board was OK, it was not our primary tool.

TIA: *Transparency, Inspection and Adaptation*

Respondent Number	Question 8: Did you limit work in progress toward deadlines and tasks? How did it go into action during the project?
P1	Yes, we had this agreement within the team, but we couldn't limit it in practice and delved into tasks with too much detail. To fix this, we used mob programming several times.
P2	We tried to limit work in progress but in many cases Parallel works were inevitable
P3	Not 100 percent but it went well generally toward the project's progress
P4	We couldn't limit WIP as it often didn't apply to queries and work breakdowns. We adjusted our approach by changing the order, addressing queries first, then designing, to improve prioritization
P5	We couldn't limit WIP because most of the time pressure caused parallel tasks, except during the final week toward the deadline.
P6	Yes, we limited WIP, which helped us reduce context switching among tasks, leading to better follow-up, task management, and focus.
P7	In Kanban teams, we effectively limited WIP, while other teams made their adjustments. This led to increased focus, defect prevention, and a sustainable pace in delivery. Although pressure caused some parallel, two teams were fully prepared for the transition day.
P8	It was Not applicable. We deferred several features from our backlog repository, focusing on landing pages and limiting the scope to approximately 10% of the features.
P9	It was not applicable to us. We utilized pair working on each task, ensuring ongoing progress and continuous improvement with an atomic scope.
P10	While not 100 percent, the project generally progressed well. We agreed to limit WIP to 2 tasks, although leaders occasionally picked up more. The main issue was the lack of resources.

Respondent Number	Question 9: Did you have a chance to prioritize the work with your teammates? Did it cause a better focus on delivering them?
P1	No, we faced lots of changes in priorities and impediments from business and product. We had no successful experience in estimating the work volume.
P2	Not 100 percent but it went well generally toward the project's progress. Estimates always were wrong but a bit helpful to eliminate some aspects.
P3	Not actually, during the project's progress priorities changed over and over. We did not use estimating for the test and it was the right action.
P4	Yes, in general, we knew the priorities and focused on them continuously.
P5	Not 100 percent, the project generally progressed well. The team was capable of estimating issues, but uncertainty arose regarding the scope.
P6	Prioritizing work was simple because we did not launch increments frequently. Estimating was OK to us and except for 1-2 cases, generally happened in action.
P7	In half of the teams, we did gracefully but in others, uncertainty caused the mismatch. most teams performed with estimation and 1 successful team with No estimate approach.
P8	Yes, we prioritized the work based on our success criteria, user journey and release plan.
P9	N/A for us. We had specific repeatable targets specially in UAT.
P10	Yes, we prioritized work based on our success criteria, which increased focus. We estimated call durations with clients to evaluate peak times, with a threshold value for evaluation.

Respondent Number	Question 10: How did you approach the test cycle? At first? In the middle? At the end?
P1	We had no chance to be a test-first team and did it after coding mostly. We needed to do scenario-based tests and load tests more than the amount that did.
P2	We were not a test-first team and mostly tested after coding. Working with Senior developers made me confident in the results in most cases.
P3	We formed an independent test team for integration and functional testing. However, due to limitation on resources, we could not bring an automation test even though started implementing it. We tested manually more after coding except for unit tests that were done by developers and UAT that were done with external parties in a parallel mode.
P4	Yes, we used the first test approach to be sure of the first series of migrated data.
P5	We were unable to adopt a test-first approach during implementation due to integration issues caused by time constraints, uncertainty, and a shortage of testing resources.
P6	We could not be test-first due to time constraints, unknown situations, and a shortage of testing resources. but The team employed a unit tests approach.
P7	We were unable to implement a test-first approach and instead started with unit tests as part of TDD. Integration issues, time and budget constraints, and uncertainty arose, impacting our ability to conduct comprehensive testing.
P8	N/A for us. We used released increments to do our job.
P9	We conducted manual testing after every release as part of our quality assurance process. This included pair working, integration tests, and additional manual testing, culminating in UAT.
P10	N/A for us.

Respondent Number	Question 11: Did you have enough autonomy to decide on details regarding tasks and how to develop them?
P1	Yes, we were an independent team, from the outside there were some impediments.
P2	Inside team, we were autonomous, from the outside there were impediments.
P3	Yes, we had 100% autonomy to decide on details during the project
P4	Yes, we had enough autonomy to decide on details of work and assign them.
P5	Yes, we had enough autonomy to decide on details during the project.
P6	Yes, we had enough autonomy to decide on details during the project. Impediments happened from out.
P7	At the team level things were OK, In Strategic and vision we had interference from outside.
P8	Yes, we were a self-manage team and had enough autonomy to decide on details during the project.
P9	Yes, we had end-to-end ownership and enough autonomy to decide on details during the project.
P10	Yes, we encouraged delegating issues as long as the process remained effective, fostering a self-organizing environment. This approach was supported by diligent follow-up on results and a culture of openness within the team.

Respondent Number	Question 12: Were daily Meetings helpful for the team?
P1	Our Daily meetings was Ok but not reduced the need to other extra meetings.
P2	Yes, Recurring meetings like Daily Scrum reduced the need to extra meetings definitely.
P3	No, it did not worked because the team did not approach them well.
P4	N/A for us. We worked in a pair mode on a daily basis and not needed meeting.
P5	Yes, it worked for us and caused fewer challenges but ad hoc issues always happened and we needed extra meetings.
P6	Yes, It worked for us. it reduced the need for some ad hoc meetings during the workday and we on-boarded delivery PM in this ceremony.
P7	In half of the teams, they worked perfectly and reduced further sessions and in others not.
P8	No, we did not have daily meetings for new platform project.
P9	No, we did not have daily meetings for the new platform but worked in daily mode with product managers.
P10	Operation - Preparing call center team for employing new platform and Extracting new workflow and processes and teaching people about them

Respondent Number	Question 13: Did the team have enough courage to do the right work regarding their tasks?
P1	We strived to make the right decisions despite the absence of ownership from other parties, relying on 90% courage and 25% power to persist. We exerted external pressure to encourage ownership, supported by technically oriented leadership.
P2	Yes, my teammates demonstrated enough courage to undertake the right tasks and take ownership of them, with a 40% power to persist.
P3	No, while I possessed the courage to do what was right, my teammates lacked the necessary courage to do the same.
P4	In some cases, yes; in others, no. We compromised at times to meet deadlines.
P5	Yes, my teammates demonstrated enough courage to undertake the right tasks, resulting in fewer technical debts.
P6	Yes, we had sufficient autonomy to make detailed decisions throughout the project, fostering collaboration. However, impediments arose from external factors multiple times.
P7	In half of the teams, they performed the courage, and in others, conflicts happened.
P8	Yes, we assessed the current situation and modified our plan for it.
P9	Yes, transparency and not compromising were the priority for us.
P10	Yes, my teammates demonstrated enough courage to perform the necessary work and take ownership of it. We addressed any doubts within the team, reflecting our commitment to the project.

Respondent Number	Question 14: Did the team structure and the number of team members was good for the project and tasks?
P1	The team structure remained stable throughout the project, but conflicts arose due to external involvement. We operated as a cross-functional team, although we faced challenges due to a lack of development resources. Despite uncertainties, mid-level performers excelled. Improved clarity from the business side could have enhanced productivity.
P2	No, the scope exceeded our capacity, necessitating its reduction in the interim. Despite working overtime, as a cross-functional team, we faced challenges due to a lack of development resources. We bypassed formal structures and collaborated on targets and tasks, fostering ownership and motivation across both back and front ends.
P3	No, the majority of teams were organized by functions rather than by product and features, resulting in a sub-optimal setup characterized by poor interactions and a fragmented approach to components.
P4	No, the Team was small in comparison to the allocated work and caused overtime working
P5	No, the scope exceeded our capacity, necessitating a reduction in the interim. We worked overtime as a cross-functional team to address the challenge.
P6	Although the scope was greater than us, the team structure was OK.
P7	While the majority of teams were initially formed based on features, informal component teams emerged in practice, and surprisingly, this approach proved effective for some products.
P8	No, the scope was greater than us, specially in Campaign Management.
P9	We paired experienced and fresh individuals per topic, leading to successful collaboration, ongoing progress, and continuous learning.
P10	The team structure was fixed and the change happened only on people arrangement.

Respondent Number	Question 15: How do joining and leaving people effect on the project and teamwork?
P1	We just lost 1 developer in this period but it was affected us to cover the vacant place
P2	Yes, it was affected us to replace the vacant position after unplanned leave
P3	We were a stable team and did not affected by leaving people
P4	Our team remained stable and unaffected by the departure of team members.
P5	We were a stable team and did not affected seriously by leaving people.
P6	At the project's onset, our team arrangement was sub-optimal, requiring several months to rectify and improve team dynamics by addressing unproductive members.
P7	Significant disruptions occurred due to departures, with one team missing deadlines and experiencing delayed publishing as a result.
P8	The exit of team members had no impact on our team's stability, and we continued to work seamlessly.
P9	Our flat team structure allowed us to navigate efficiently through the challenges. However, after the launch a layoff happened, and some unnecessary individuals were released from the team.
P10	The call center is not a stable team in terms of people coming and leaving.

Respondent Number	Question 16: Did you have daily work with Business/Technical people and how to evaluate it?
P1	No, there were not a great communication and collaboration between team and business side.
P2	While our daily interactions with the project manager were productive, we encountered challenges with other business stakeholders. Conflicts and procrastination were common, compounded by high business pressure. However, the PM's facilitation skills and strong leadership helped navigate these difficulties.
P3	No, we lacked consistent daily engagement with business and technical stakeholders due to the absence of a clear Product Owner within the company. This lack of interaction posed significant challenges for the project.
P4	We did not had exactly daily work but we used a focus group to collect their demands.
P5	Indeed, we engaged in daily collaboration with both business and technical teams, fostering productivity and ensuring timely achievement of deadlines through pair working.
P6	Yes, we collaborated daily with business and technical teams, leveraging work to enhance productivity. Additionally, serendipitous conversations played a role in averting potential failures.
P7	Yes, we maintained daily collaboration between business and product teams, fostering productivity through pair working and daily navigation of challenges. Shorter feedback loops and a focus on time-to-market facilitated synergy within the project. Overall, I evaluated the process positively, noting successful performance from the majority of parties involved.
P8	Indeed, we were engaged in daily collaboration with both product and technical teams, facilitating face-to-face interactions and conducting daily reviews with content and project managers to ensure alignment and progress.
P9	Absolutely, our daily collaboration with both the tech and product teams was exceptional. This detail-oriented approach significantly improved productivity by 60%.
P10	Yes, we maintained daily collaboration with both product and technical teams, emphasizing transparency through face-to-face interactions and direct involvement with project managers.

Respondent Number	Question 17: Did Meetings happen in their time boxes? What was the effects?
P1	In most cases, we lost the time boxes except for daily meetings which we kept in 15 min.
P2	In most cases, we kept the time boxes, but some sessions took longer
P3	No, In most cases we broke the time boxes and faced lots of distractions
P4	Mostly, we adhered to the time boxes, though occasional lapses occurred.
P5	Generally, we managed to adhere to the time boxes, although there were instances where deadlines were missed.
P6	For the most part, we maintained adherence to the time boxes, but there were occasions where frustrated individuals led to prolonged sessions.
P7	We generally adhered to the time boxes, but certain sessions occasionally extended beyond the allotted time. This was influenced by factors such as commitment, collaboration, training for some individuals, uncertainty in scope, facilitation, ad hoc discussions, and delays or overlaps due to attendees' presence or absence.
P8	Typically, we stuck to the time boxes, but occasionally, sessions ran longer than anticipated, stretching from 30 minutes to up to an hour. We employed agendas to structure our meetings efficiently.
P9	Generally, we adhered to the time constraints, yet there were occasions where certain sessions exceeded their allocated duration.
P10	No, we frequently exceeded the time limits, contending with numerous interruptions. The abundance of details, the dynamic of new joiners, and a learning curve created an environment where the team required additional training to address these challenges.

Respondent Number	Question 18: Did you recognize risks before starting the tasks?
P1	We made some incorrect assumptions about third-party toolkits and platforms, leading to an impact on our work. This stemmed from a lack of understanding of potential risks, oversimplification of the situation, and unpredictability in the project environment.
P2	At times, both the tech and product teams accurately predicted certain issues while overlooking others. This discrepancy arose from factors such as uncertainty regarding risks, limited input from the business side, the need for comprehensive product requirement documentation (PRD), and guidance from experienced individuals.
P3	We encountered challenges in effectively managing risks across both technical and business sides.
P4	In the majority of instances, our ability to predict outcomes fell short, with only 40% accurately anticipated. We relied on predefined data, but encountered challenges such as indexing issues. To address this, we implemented two types of dashboards.
P5	In many instances, we effectively predicted outcomes through the use of predefined data.
P6	Our primary risk factor was the unpredictability of human resources, which impacted our ability to forecast timelines and budget constraints. Despite these challenges, we successfully anticipated risks and took appropriate actions to mitigate them.
P7	To be honest, our handling of business risks across all teams was inadequate. Complexity, miscommunication, weak business leadership, extensive scope, weak project management, ad hoc approaches, and insufficient human resources at all levels compounded the challenge.
P8	At times, we were successful in predicting risks accurately, while other times we fell short. The assistance of PMs played a crucial role in this process.
P9	In the majority of cases, we were able to predict risks effectively. It was facilitated by pair working, high motivation experienced people, and leveraging previous knowledge.
P10	In many instances, we accurately anticipated risks, although there were occasions where our predictions fell short. Our proactive approach, ownership mentality, and action plan development enabled us to identify approximately 80% of risks.

Respondent Number	Question 19: How does the Shop-Ops Kanban board help the team? Did you use it? How evaluate it?
P1	We failed to utilize the Shop-Ops Kanban board effectively due to the work overload on our team. We resorted to direct follow-ups on issues, as we faced challenges stemming from the lack of testing and development resources.
P2	The UAT sessions and SHOP-OPS Kanban board showcased transparency and defect prevention, despite resource limitations. Consolidating problems into a single channel fostered learning and continuous improvement, boosting team morale.
P3	The UAT with external parties and SHOP-OPS Kanban board demonstrated impressive transparency and waste reduction. I led by example and diligently followed up on issues.
P4	The UAT with technical teams showcased impressive transparency and collaboration, despite challenges like the lack of testing resources. Many issues were recognized, indicating effective training using the tool. Additionally, the ability to address data inconsistency and perform on-demand data analysis added further value to the process.
P5	The daily UAT with external parties, coupled with the SHOP-OPS board, were commendable, particularly due to the upstream support from management.
P6	We didn't use it. We followed up issues directly.
P7	UAT with SHOP-OPS happened successfully Transparency. People trained on it and lots of issues were recognized. It had enough support from the upper management and performed perfectly. Some teams failed in UAT due to their internal situations.
P8	UAT with tech and product parties and SHOP-OPS board was OK.
P9	The UAT process with technical teams and the SHOP-OPS showcased impressive transparency. Yet, challenges persisted, including the absence of an estimated time for fixing or the occurrence of duplicate tickets. Nevertheless, it emerged as a strong platform.
P10	The UAT with both technical and product parties, along with the SHOP-OPS board, proved beneficial. They facilitated transparency, prioritization, and the reporting of critical items, encouraging a sense of responsibility and contributing to the stability of the product.

Faculty of Engineering, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden