PGCOMP - Programa de Pós-Graduação em Ciência da Computação
Universidade Federal da Bahia (UFBA)
Av. Milton Santos, s/n - Ondina
Salvador, BA, Brasil, 40170-110

https://pgcomp.ufba.br
pgcomp@ufba.br

Early-stage startups, typically composed of small teams working with innovative yet unvalidated ideas, face the critical challenge of demonstrating the viability of their concepts in the market. These ideas are often shaped within a flexible business model that evolves until a repeatable and scalable solution emerges. Under continuous pressure to rapidly deliver a minimum viable product (MVP), software startups tend to focus heavily on technological solutions. Operating in highly competitive environments alongside companies of varying sizes, these startups face distinct challenges. Recent studies have examined the broader context, objectives, and operational difficulties of software startups. However, a significant gap remains in understanding how these companies select and apply specific software engineering practices during software development. Gaining insight into their methods and decision-making processes is essential, as these directly affect their ability to address the inherent uncertainties and constraints of the startup environment. Key questions arise: What are the defining characteristics of software development in startups? How do these organizations prioritize quality attributes in their products? Which engineering practices support their progress, and which prove most effective in their early stages? A limited understanding of how startups choose and implement development practices can result in inefficient processes, delays, higher costs, and reduced product quality. More critically, this lack of insight may compromise their competitive advantage and market sustainability. This thesis examines the software development dynamics of early-stage startups, focusing on their priorities, workflows, and tools. The study employs a qualitative methodology, using a multiple-case study design involving 14 organizations. Data were analyzed using Grounded Theory techniques, including open, axial, and selective coding. The findings underscore the importance of human factors, development processes, engineering practices, and external pressures in shaping startups' software development strategies. By identifying and understanding these elements, the study offers practical recommendations to support more effective development practices in the early stages of startup growth. Ultimately, this research identifies key factors influencing software development in startups and proposes actionable guidelines to enhance their technical and strategic capabilities during their formative phase.

Keywords: Early-stage startups, Software development, Innovation, Qualitative research, Grounded Theory.

DSC | 46 | 2023

A Multi-Case Study of Software Engineering Practices in Early-Stage Startups

Renata Maria de Souza

UFBA

A Multi-Case Study of Software Engineering Practices in Early-Stage Startups

Renata Maria de Souza

Tese de Doutorado

Universidade Federal da Bahia

Programa de Pós-Graduação em
Ciência da Computação

November | 2023

Universidade Federal da Bahia
Instituto de Computação

Programa de Pós-Graduação em Ciência da Computação

A MULTI-CASE STUDY OF SOFTWARE ENGINEERING PRACTICES IN EARLY-STAGE STARTUPS

Renata Maria de Souza

TESE DE DOUTORADO

Salvador
14 de Novembro de 2023

RENATA MARIA DE SOUZA

A MULTI-CASE STUDY OF SOFTWARE ENGINEERING PRACTICES IN EARLY-STAGE STARTUPS

Esta Tese de Doutorado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Ivan do Carmo Machado

Salvador
14 de Novembro de 2023

Ficha catalográfica elaborada pela Biblioteca Universitária de Ciências e Tecnologias Prof. Omar Catunda, SIBI – UFBA.

S729 Souza, Renata Maria de

A multi-case study of software engineering practices in early-stage startups / Renata Maria de Souza. – Salvador, 2023.

218 f.

Orientador: Prof. Dr. Ivan do Carmo Machado.

Tese (Doutorado) – Universidade Federal da Bahia, Instituto de Computação, 2023.

1. Engenharia de software. 2. Engenharia de software empírica. 3. Startup de software. 4. Desenvolvimento de software. I. Machado, Ivan do Carmo. II. Universidade Federal da Bahia. III. Título.

CDU:004.42

Ata da Sessão Pública de Defesa de Doutorado nº. 46

Colegiado do Programa de Pós-Graduação em Ciência da Computação

Ata da sessão pública do Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, realizada em 14 de novembro de 2023 para procedimento de defesa da Tese de Doutorado em Ciência da Computação nº. 46, linha de pesquisa Engenharia de Software, da candidata **Renata Maria de Souza Santos**, matrícula 216121800, intitulada "*A Multi-Case Study of Software Engineering Practices in Early-Stage Startups*". Às Catorze horas do citado dia, via web conferência foi aberta a sessão pelo presidente da banca examinadora, Prof. Dr. Ivan do Carmo Machado (Orientador -UFBA) que apresentou os outros membros da banca: o Prof. Dr. Gleison dos Santos Souza (UNIRIO), o Prof. Dr. Kiev Santos da Gama (CIn/UFPE), a Profª. Drª. Rita Suzana Pitangueira Maciel (UFBA), a Profª. Drª. Christina von Flach Garcia Chavez (UFBA). Em seguida foram esclarecidos os procedimentos pelo presidente que passou a palavra ao examinado para apresentação do trabalho de Doutorado. Ao final da apresentação, passou-se à arguição por parte da banca, a qual, em seguida, reuniu-se para a elaboração do parecer. No seu retorno, foi lido o parecer final a respeito do trabalho apresentado pela candidata, tendo a banca examinadora **aprovado** o trabalho apresentado, sendo esta aprovação um requisito parcial para a obtenção do grau de Doutor. Em seguida, nada mais havendo a tratar, foi encerrada a sessão pela presidente da banca, tendo sido, logo a seguir, lavrada a presente ata, abaixo assinada por todos os membros da banca.

Salvador, 14 de novembro de 2023

gov.br Documento assinado digitalmente
IVAN DO CARMO MACHADO
Data: 14/11/2023 18:38:53-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Ivan do Carmo Machado
(Orientador -UFBA)

gov.br Documento assinado digitalmente
Gleison dos Santos Souza
Data: 14/11/2023 21:29:03-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Gleison dos Santos Souza
(UNIRIO)

gov.br Documento assinado digitalmente
KIEV SANTOS DA GAMA
Data: 16/11/2023 08:17:05-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Kiev Santos da Gama
(CIn/UFPE)

gov.br Documento assinado digitalmente
RITA SUZANA PITANGUEIRA MACIEL
Data: 16/11/2023 16:07:40-0300
Verifique em https://validar.iti.gov.br

Profª. Drª. Rita Suzana Pitangueira Maciel
(UFBA)

gov.br Documento assinado digitalmente
CHRISTINA VON FLACH GARCIA CHAVEZ
Data: 16/11/2023 22:04:20-0300
Verifique em https://validar.iti.gov.br

Profª. Drª. Christina von Flach Garcia
Chavez (UFBA)

To my beloved sons, Gabriel and Rafael.

ACKNOWLEDGEMENTS

Dear friends and collaborators,

It is with immense gratitude and joy that I share this special moment marking the completion of my academic journey and the research that culminated in this doctoral thesis.

First and foremost, I offer my deepest thanks to God. His guidance and unwavering support sustained me throughout this path. In moments of difficulty, His grace gave me the strength and determination to persevere. His presence was my constant light, and to Him I dedicate all honor and glory.

To my psychologist, Dr. Isabela Arteiro, I extend my profound gratitude. Your consistent support, patience, and insight were vital in navigating the challenges of this journey. Your professionalism and empathy helped make this process more manageable and meaningful. Words fall short in expressing how essential your presence was to this achievement. I am deeply thankful for your dedication and care.

To my advisor, Prof. Dr. Ivan do Carmo Machado, I offer my heartfelt thanks. Your guidance and support were instrumental in reaching this milestone. I am also deeply grateful to the distinguished members of my examining committee—Prof. Dr. Christina Von Flach (UFBA), Prof. Dr. Gleison Santos (PUCRIO), Prof. Dr. Kiev Gama (UFPE), and Prof. Dr. Rita Pitangueira (UFBA)—for their insightful contributions, generous guidance, and thoughtful feedback, which greatly enriched this work.

I extend my sincere appreciation to the outstanding researchers with whom I had the privilege of collaborating. Your contributions expanded my knowledge and shaped the global perspective of this thesis. I thank: Karla Malta, Roselane Silva, Eduardo Almeida, Franklin Silva, and Larissa Rocha (UFBA); Paulo Masieiro (USP); Anh Nguyen-Duc, Letizia Jaccheri, and Orges Cico (NTNU); Jan Bosch (Chalmers University of Technology); and John McGregor (Clemson University).

To my lab colleagues—Alcemir Santos, Denivan Campos, Franklin Silva, Karla Malta, Larissa Rocha, Leila Kaitá, Luana Martins, Nildo Silva, Sávio Freire, Railana Santana, Roselane Silva and Sara Lima—thank you for your support, camaraderie, and intellectual generosity. Each of you has contributed uniquely to this journey, and I am truly grateful.

To my dear work colleagues —Marcos Santos, Igo Luz, Carla Simone, Raul Marques and Flávio Pereira —thank you for your encouragement, advice, and presence.

To my beloved family—Ayres de Souza, Jamile de Souza, Tatiana Souza, and Jefferson Nunes—your love, strength, and unwavering support were the foundation of my resilience. To my children: your love and understanding, patience, and quiet strength were a constant source of motivation. This accomplishment is as much yours as it is mine.

Finally, I extend my sincere gratitude to everyone who, in some way, contributed to this research and to my academic journey. This work is the result of a shared effort, built upon the care, encouragement, and collaboration of an incredible support network.

With all my affection and appreciation, Renata Souza

"If you keep silent, keep silent by love: if you speak, speak by love; if you correct, correct by love; if you pardon, pardon by love; let love be rooted in you, and from the root, nothing but good can grow.
Love and do what you will.
Love endures in adversity, is moderate in prosperity, brave under harsh sufferings, cheerful in good works, utterly reliable in temptation, utterly open-handed in hospitality, as happy as can be among true brothers and sisters, as patient as you can get among the false ones. The soul of the scriptures, the force of prophecy, the saving power of the sacraments, the fruit of faith, the wealth of the poor, the life of the dying.
Love is all."

—SAINT AUGUSTINE (Homilies on the First Epistle of John)

RESUMO

Startups em estágio inicial, geralmente compostas por pequenas equipes que trabalham com ideias inovadoras, porém não validadas, enfrentam o desafio crucial de demonstrar a viabilidade de seus conceitos no mercado. Essas ideias são frequentemente moldadas por um modelo de negócios flexível que evolui até que surja uma solução repetível e escalável. Sob pressão contínua para entregar rapidamente um produto mínimo viável (MVP), elas se concentram fortemente em soluções tecnológicas. Operando em ambientes altamente competitivos ao lado de empresas de tamanhos variados, essas startups enfrentam um conjunto distinto de desafios que exigem um exame mais específico do contexto mais amplo, dos objetivos e das dificuldades operacionais das startups de software. No entanto, ainda há uma lacuna significativa na compreensão de como essas empresas selecionam e aplicam práticas específicas de engenharia de software durante o desenvolvimento de sistemas. Obter insights sobre seus métodos e processos de tomada de decisão é essencial, pois estes afetam diretamente sua capacidade de lidar com as incertezas e restrições inerentes ao ambiente de startups. Surgem questões-chave: Quais são as características definidoras do desenvolvimento de software em startups? Como essas organizações priorizam os atributos de qualidade em seus produtos? Quais práticas de engenharia apoiam seu progresso e quais são mais eficazes em seus estágios iniciais? Uma compreensão limitada de como as startups escolhem e implementam práticas de desenvolvimento pode resultar em processos ineficientes, atrasos, custos mais altos e qualidade do produto reduzida. Mais criticamente, essa falta de conhecimento pode comprometer sua vantagem competitiva e sustentabilidade no mercado. Esta tese examina a dinâmica de desenvolvimento de software de startups em estágio inicial, com foco em suas prioridades, fluxos de trabalho e ferramentas. O estudo emprega ferramentas de design de estudo de casos múltiplos, envolvendo 14 técnicas de organização, incluindo codificação aberta, axial e seletiva. As descobertas ressaltam a relevância dos fatores humanos, dos processos de desenvolvimento, das práticas de engenharia e das pressões externas na formação das estratégias de desenvolvimento de software das startups. Recomendações práticas para apoiar práticas de desenvolvimento mais eficazes nos estágios iniciais de crescimento de startups. Em última análise, esta pesquisa identifica fatores-chave que influenciam o desenvolvimento de software em startups e propõe diretrizes acionáveis para aprimorar suas capacidades técnicas e estratégicas durante sua fase formativa.

Palavras-chave: Startups em estágios iniciais, desenvolvimento de software, inovação, pesquisa qualitativa, teoria fundamentada em dados.

ABSTRACT

Early-stage startups, often formed by small teams with innovative yet invalidated ideas, face the crucial challenge of proving their concepts in the market. These ideas are shaped by a flexible business model that adjusts until a repeatable, scalable approach emerges. Under constant pressure to deliver a minimum viable product quickly, software startups focus on technological solutions. Operating in a highly competitive environment alongside companies of various sizes, these startups face specific challenges. Recent research has explored these companies' context, objectives, challenges, and practices in depth. Although extensive studies have examined the involvement of software startups in systems development, a gap remains in understanding how they select and implement specific practices in this process. Understanding their methods and approaches is essential, as they directly affect their ability to overcome the industry's inherent challenges. Key questions include: What characteristics define software development in startups? How do these companies prioritize product quality attributes? Which software engineering practices underpin their efforts, and which ones drive their progress? A lack of understanding about selecting and implementing specific practices can lead to inadequate or inefficient processes, resulting in delays, increased costs, and reduced quality of the final product. Furthermore, this gap can compromise startups' market competitiveness. This thesis offers insights into the development of early-stage software startups, highlighting their priorities, processes, and tools. Our research employs a qualitative approach, based on a multiple-case study involving 14 organizations. Data were analyzed using Grounded Theory techniques, including open, axial, and selective coding. The results highlight the importance of human factors, development processes, software engineering practices, and external influences in the trajectory of these startups. Understanding these elements enabled practical recommendations to strengthen software development in the early stages. **Conclusions:** This study identifies critical factors that influence software development in startups and presents guidelines to improve their practices in this early stage.

Keywords: Early-stage startups, Software development, Innovation, Qualitative research, Grounded Theory.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Chapter 1

INTRODUCTION

With their unique capacity for innovation and adaptability, startups play a crucial role in addressing contemporary economic, social, and environmental challenges. They survive and thrive, even in the face of economic recessions, due to their ability to cater to changing consumer preferences and market conditions. This chapter underscores the distinct role of startups in promoting innovation, catalyzing change across sectors, and contributing to resolving social problems.

In this context, startups face significant obstacles when offering innovative solutions to global problems. The United Nations Sustainable Development Goals, formulated in 2012, provide a comprehensive framework for tackling pressing issues such as poverty, climate change, and environmental degradation. Entities such as the World Economic Forum and Singularity University are dedicated to driving advancements in these areas, using exponential technologies to impact the achievement of these goals significantly.

This chapter presents the motivation for this study, outlining the research problem, objectives, and methodology. We explore the multifaceted challenges and opportunities that startups encounter in their mission to provide innovative solutions to complex, global problems. The research examines startups' strategies, technologies, and practices to assess their effectiveness in achieving significant social and environmental impact.

1.1 CONTEXT AND MOTIVATION

In the contemporary business scenario, the role of software in innovations is increasingly crucial and comprehensive, abandoning varied sectors ranging from finance to production and services. Companies employ software as an essential tool in their innovative activities and develop software as a central component of their innovation strategies. Furthermore, we create innovations that are deeply integrated or heavily dependent on software, thus evolving traditional paradigms of product and service development (BOTELHO; FEHDER; HOCHBERG, 2021).

The strategic importance of software was highlighted by Matt, Hess e Benlian (2015), who proposed a comprehensive framework for digital transformation encompassing four critical dimensions: use of technologies, changes in value creation, structural changes, and financial aspects, one of which is the use of technology strategy. Recent studies highlight that economic growth, mainly related to entrepreneurship, is predominantly driven by a select group of high-growth startups strongly oriented towards innovation rather than simply by the volume of new ventures in the market (BOTELHO; FEHDER; HOCHBERG, 2021).

During the COVID-19 crisis, startups have proven resilient and specific to the global economy, with the Startup Genome report finding that these companies have developed approximately $2.8 trillion in global economic value (GAUTHIER; MORELIX, 2020). This contribution highlights the ability of startups to use software to innovate, providing companies with a sustainable competitive advantage that reduces costs, increases long-term productivity, and underpins the creation of innovative solutions that meet emerging customer needs and distinguish them in highly competitive markets.

Software startups are uniquely capable of leading innovation to solve hitherto unaddressed problems. They develop minimum viable solutions within constraints such as limited resources and tight deadlines, allowing them to survive and thrive in a challenging market environment. These initiatives strengthen their market position, increase their competitive capacity, reduce product costs, and expand their dynamic capabilities and ability to absorb new knowledge and technologies (HYYTINEN; PAJARINEN; ROUVINEN, 2015).

A notable example is Uber Technologies Inc. located in the San Francisco Bay Area, since its founding in 2009, has revolutionized the transportation industry with competitive pricing and superior mobility services, challenging traditional taxi services (FRIESENDORF; UEDELHOVEN, 2021). Similarly, Nubank emerged to transform the banking experience and, through aggressive growth strategies, reached a customer base of 48 million, consolidating itself as the largest fully digital bank in the world (ANALYTICA, 2021). These examples illustrate how software startups are crucial for driving innovation across diverse industries, tackling unique challenges and market opportunities with an approach fundamentally different from that of established companies.

In the digital transformation era, software companies, especially startups, are in a privileged position to explore great opportunities (CUSUMANO, 2008; VIAL, 2019). Digital transformation refers to integrating digital technology into all business areas, fundamentally changing how companies operate and deliver customer value. As software has become a fundamental component of almost all products and services (TANVEER, 2021), the success of startups often depends on the ability to develop innovative software solutions (CROWNE, 2002). Traditional development methodologies, both traditional and agile, may need to be revised to meet the specific needs of these companies (PATERNOSTER et al., 2014; KLOTINS; UNTERKALMSTEINER; GORSCHEK, 2015; BERG et al., 2018). The contemporary scenario requires new practices to face startups' challenges, which include market demands, regulatory restrictions, legal issues, and the need to monetize their customer base (MELEGATI; GUERRA; WANG, 2022). Even though they are inherently innovative, startups need guidance to thrive in a competitive and innovative market during their early years. It involves driving the software development process through shorter cycles, dealing with human resource constraints, and quickly adapting to technological changes.

Software engineering is fundamental in software startups. The context of software development in startups encompasses several elements, such as the product or service under development, the practices adopted, the tools and processes used, the team members, the startup's identity, and the market in which it operates (PETERSEN; WOHLIN, 2009). Collaboration between software engineers, designers, product managers, and stakeholders is crucial. Effective collaboration ensures that software is aligned with the company's vision and goals. Engineering in software startups focuses on creating innovative solutions to meet specific market needs. It requires a deep understanding of customers, their problems, and how the software can solve them effectively.

Software engineers at startups must be highly adaptable and thrive in a fast-paced environment where priorities and requirements can change quickly. Success in this challenging environment depends on technical expertise, creative thinking, collaboration, and adaptability.

Despite their potential for innovation and economic growth, startups face high failure rates, with only a tiny proportion achieving success (CROWNE, 2002; CANTAMESSA et al., 2018). Cantamessa et al. (2018) proposed a model to classify failures in startups, highlighting the importance of adopting Software Engineering practices that speed up software development and delivery. The unique context of development in startups presents challenges, with the main reasons for failure often related to the product: lack of focus on development, unviable product, low product quality, and lack of market interest. Challenges also include a need for knowledge about proper Software Engineering practices (CROWNE, 2002; PATERNOSTER et al., 2014; GIARDINO; WANG; ABRAHAMSSON, 2014; KLOTINS; UNTERKALMSTEINER; GORSCHEK, 2015; UNTERKALMSTEINER et al., 2016). Therefore, software startups involve high risks despite their significant contributions to innovation and economic growth.

Therefore, we consider software startups essential for driving innovation in different sectors, but they face unique challenges and risks. Adopting effective Software Engineering practices and promoting a culture of adaptability, innovation, and collaboration are crucial to success in the competitive market.

The role of software engineering practices in startups and their impact has yet to be comprehensively explored, leaving a significant gap in knowledge in this domain (UNTERKALMSTEINER et al., 2016). Recent studies on software startups have begun to highlight the critical software engineering challenges they face. A network of researchers dedicated to studying startups identified 71 research questions categorized into six clusters, addressing different aspects of software startups:

1. Support for startup engineering activities
2. Models and patterns of startup evolution
3. Human aspects in software startups
4. Applying startup concepts to non-bootable environments
5. Startup ecosystems and innovation centers

6. Methodologies and theories for startup research

In this changing context, startups must strongly emphasize architecture, testing, and deployment practices that facilitate their software products' development and continuous evolution. Building scalable and maintainable systems capable of supporting rapid growth is equally vital. Existing literature has introduced several models to elucidate how startups employ software engineering practices, such as the Greenfield Startup model, Progression model, PRESS model, and HyMap. However, while informative, these models need more empirical evidence for in-depth qualitative analysis through Grounded Theory.

Despite these valuable contributions to understanding startup software development practices, previous studies have limitations. Empirical research on these practices is notably scarce, as gaining access to startup environments for in-depth investigation can be challenging. Despite existing knowledge, additional empirical evidence is still needed to provide a more comprehensive and detailed qualitative analysis. Developing a model encapsulating the context of startup software development and the associated practices and toolset is essential for offering valuable insights to startups in their early stages. Understanding this context, including the quality attributes of the product and the tools used, is essential.

However, despite the availability in the current literature of theories and models that can help startups in their software development efforts, there is a conspicuous lack of empirical studies. These studies are essential for demonstrating how early-stage startups make decisions about software engineering practices, accelerating the development process while navigating the unique constraints of this lifecycle phase. Its knowledge is fundamental to ensuring that startups can sustainably transition to subsequent stages.

1.2 RESEARCH GOAL AND QUESTIONS

This thesis endeavors to examine the software engineering practices employed within software startups. At its core, this research seeks to answer a specific and fundamental question: How do early-stage startups develop software? These practices are uncovered through empirical research, and the central research question guiding this thesis is as follows:

RQ1: ***How do early-stage startups develop software?***

- ***Relevance:*** This question explores the general landscape of software development in startups, providing insight into the processes, practices, and challenges faced.

- ***Contribution:*** The answers to this question help identify gaps in the startup software development process. Understanding this can provide technical and process recommendations, such as improving software architecture, coding policies, and agile development procedures for startups that need to be fast and efficient, as exemplified in T-REC.01 and T-REC.03.

Given the breadth of this central question, we have structured a set of more specific research sub-questions that delve into the nuances of software engineering within startup environments. These sub-questions are tightly interconnected with exploring software engineering practices in startups and their distinctive contributions. The following sub-questions (RQ) will guide our investigation:

RQ1.1: What is the contextual framework for startup software development?

- ***Rationale:*** Startups exhibit unique characteristics, including resource constraints and dynamic technological environments. While informative, existing characterizations of startups in research may not suffice for making informed comparisons of software engineering contexts. A deeper understanding of the software engineering context within startups is crucial for guiding or recommending engineering practices that align with these unique contexts.
- ***Relevance:*** This question delves into the structure and characteristics of the startup development environment, such as limited resources, small teams, and rapid market changes.
- ***Contribution:*** Understanding this context leads to more appropriate recommendations to address the specific challenges of this scenario. For example, by understanding that resources are limited, recommendations such as HA-REC.02 (learning while doing) and HA-REC.03 (embracing failures and learning from them) are suggested to create a culture of learning and rapid adaptation, fundamental characteristics for startups.

RQ1.2: Which software engineering practices are most pertinent to early-stage startups?

- ***Rationale:*** This research question delves into identifying the specific software engineering practices relevant to early-stage startups, providing insights into their practical application.
- ***Relevance:*** This question investigates which practices startups adopt and why some work better in specific contexts.
- ***Contribution:*** Identifying the most effective practices allows for formulating technical recommendations, such as the need to adopt minimum viable processes, automation, and continuous integration to gain speed and efficiency (example: T-REC.05, T-REC.07, T-REC.08). Thus, the thesis not only analyzes which practices are in use, but also which practices should be used to increase efficiency.

RQ1.3: What quality attributes do startups prioritize in software development?

- ***Rationale:*** Identifying the software quality attributes that startups emphasize is essential, as these attributes significantly influence the distinctive characteristics of software products developed by startups.
- ***Relevance:*** This question explores startups' software quality priorities, such as performance, scalability, or ease of use.
- ***Contribution:*** Identifying critical quality attributes leads to recommendations that help startups focus on the exemplary aspects early in development, such as T-REC.06 (keep it simple) and T-REC.02 (spend minimal time defining the architecture), so that startups can deliver high-quality products without losing agility.

RQ1.4: Which software development startups commonly use supporting tools?

- ***Rationale:*** Due to constraints such as limited time, resources, or specialized knowledge, startups often struggle to identify the most suitable tools for their needs and may not fully harness their potential. This research question aims to furnish empirical evidence regarding the tools commonly employed to support software development in startup environments.
- ***Relevance:*** This question identifies startups' technology tools to facilitate software development.
- ***Contribution:*** By identifying the most valuable tools that can provide practical technical recommendations, such as T-REC.09 (find the toolset that helps accelerate software development), ensuring that startups are using the best options on the market to gain efficiency and avoid technical overhead.

RQ1.5: How can practical recommendations support early-stage software startups to optimize their development processes and overcome typical challenges of limited resources and high uncertainty?

- ***Rationale:*** This question seeks to understand how a set of practical recommendations can be directly applied to early-stage software startups, helping them deal with their unique challenges, such as resource scarcity, market uncertainty, and the need for agile and iterative development. By formulating recommendations, the goal is to provide clear and actionable guidance so that startups can optimize their software engineering processes without compromising innovation or delivery speed.
- ***Relevance:*** Software startups face a highly dynamic and competitive landscape, where time to market and rapid adaptation are

crucial for success. However, many studies focus on generic software engineering practices without considering these startups' operational peculiarities and constraints. Understanding how targeted recommendations can solve typical problems – such as the lack of formal processes, the absence of automated testing, or the accumulation of technical debt – is therefore essential for the success of startups. This issue is highly relevant to provide practical and scalable guidance that can be adopted without overloading teams or compromising development speed.

- ***Contribution:*** The research will contribute by mapping recommendations to the needs of early-stage software startups. These include integrating agile minimum viable processes, using automation tools efficiently, and creating an organizational culture that favors innovation and resilience. It is essential to highlight that these recommendations are practical and scalable, which means that they have the potential to be adopted without overloading teams or compromising development speed, helping them improve their results and grow sustainably.

The overarching objective of this research was to develop a comprehensive understanding of the contextual aspects, software engineering practices, quality attributes, and supportive tools that underpin software development within startups. This knowledge contributes to formulating practical recommendations, aiding startups in their early stages, and fostering sustainable growth and development.

1.3 METHODOLOGICAL APPROACH

The primary goal of this study was to delve into the software engineering practices employed in software startups, particularly their impact on the early stages of company development. This work aimed to describe and analyze current practices and identify significant gaps in the literature on software engineering in the startup environment. Doing so underscored the crucial need for a comprehensive understanding of this field.

Through rigorous empirical investigation, this study sought to elucidate how software startups adopt and adapt engineering practices to overcome specific challenges, such as resource constraints, rapid product delivery requirements, and agile technology adaptation. The results provide valuable insights for developing a theoretical-practical model that guides startups in implementing more effective and efficient software engineering practices.

Our research adopts a qualitative approach based on a multiple case study of 14 organizations. This approach was motivated by understanding early-stage software startups' processes and challenges.

This study builds on previous research by the authors, which explored different aspects of startup software development. Works such as "Software Engineering in Startups: A Single Embedded Case Study" (SOUSA et al., 2018) and "A Case Study about Startups'

Software Development Practices: A Preliminary Result" (SOUZA et al., 2019a) provided preliminary insights into practices used in these companies. In addition, "Investigating Agile Practices in Software Startups" (SOUZA et al., 2019b) and "A Survey on the Software Engineering Practices in Brazilian Software Startups" (SOUZA; CICO; MACHADO, 2021) expand the understanding of agile methodologies and approaches adopted in the Brazilian context.

Our research seeks to deepen knowledge about selecting and implementing specific software engineering practices in startups building on these investigations. Data were collected through semi-structured interviews, direct observation, and document analysis. The analysis was conducted according to the principles of Grounded Theory, using open, axial, and selective analysis to identify patterns and relationships between the practices adopted by startups.

This approach allowed us not only to map the methodologies employed but also to understand how startups select and implement specific practices to deal with the challenges of software development in its early stages.

Furthermore, this study's findings could influence startups' approaches to software development. By formulating practical recommendations that startups can adopt to improve their software development processes' quality, efficiency, and sustainability, this study can impact future practices in software startups, academic courses, as well as professional development programs.

Finally, by highlighting the practices that correlate with success in software development in dynamic and highly innovative environments, this study contributed to a better understanding of the critical success factors in software startups. It provided a solid foundation for future research and strategic decision-making by entrepreneurs and investors in the startup ecosystem.

Our research was developed using a structured methodological approach consisting of three main phases: (1) research planning, (2) research execution, and (3) synthesis and documentation of the results obtained.

Figure 1.1 shows the schematic overview of the thesis structure.

Previous studies have been instrumental in shaping the methodology of this doctoral thesis. They have provided a robust foundation for defining data collection and analysis methods, and have offered crucial insights for formulating the research questions. Below, we detail how these previous studies have influenced our methodology:

1. **Identifying Research Gaps** Previous studies, such as those by Paternoster et al. (2014) and Unterkalmsteiner et al. (2016), have provided in-depth insights into software engineering challenges and practices in startups, identifying gaps in the literature. By identifying these gaps, they could define research questions that directly address these underexplored areas, ensuring their research would provide original and relevant contributions. These gaps were essential in shaping the rationale for their research and directing their focus to topics such as:

    - The adoption of startup-specific software engineering practices.
    - How startups prioritize quality attributes and adopt software development tools.

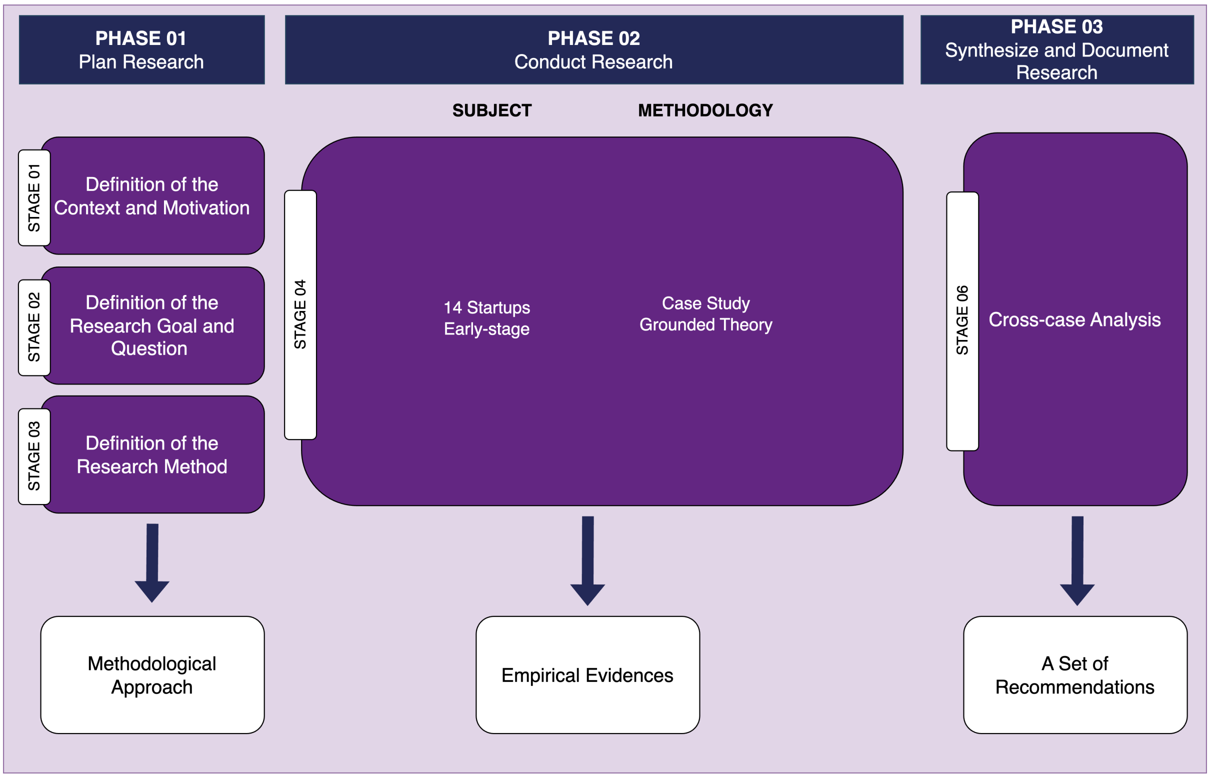


Figure 1.1 Schematic Overview of the Thesis Structure.

- There is a need for robust empirical studies exploring the particularities of early-stage software startups.

2. **2. Empirical Basis for Method Choice** The choice of multiple case studies and Grounded Theory was heavily influenced by previous studies also used qualitative approaches to investigate software engineering practices. Studies such as those by Giardino et al. (2016) and Crowne (2002) demonstrated that the case study method effectively deeply understood the context and challenges faced by startups.

   - Giardino et al. (2016), in particular, used case studies to investigate how startups manage development processes and evolve their business models.
   - Unterkalmsteiner et al. (2016) use of Grounded Theory to categorize practices and behaviors in startups was also a direct inspiration for their choice of qualitative analysis method.

3. **3. Designing Data Collection Protocols** The previous articles provided examples of how to conduct interviews and observations in startups, helping to shape the data collection protocol used in research. The experience from these studies helped structure a robust data collection strategy, combining interviews, observations, and document analysis, ensuring a more comprehensive understanding of software engineering processes in startups. These include:

- Semi-structured interviews: Inspired by studies such as those by Paternoster et al. (2014), you developed an interview protocol with open-ended questions, allowing you to explore specific practices, challenges, and contextual factors without limiting participants' responses.
- Non-participatory analysis: Previous studies, such as those by Unterkalmsteiner et al. (2016), guided the choice of non-intrusive methods to observe how startups execute their daily practices, complementing the information obtained in interviews.

4. **4. Validation of the Grounded Theory Approach** Grounded Theory was adopted based on its ability to build emergent theories directly from data. It was essential to deal with the unique context of startups, where processes do not always follow traditional software development models. The study by Melegati et al. (2019a), which also used Grounded Theory to analyze startups, validated the choice of this approach in research on innovation and agile development in startups. The successful use of Grounded Theory in previous studies served as a basis for applying it in this research. This allowed the construction of categories and relationships from the observed practices, leading to the formulation of practical recommendations and emergent theories on how startups can optimize their processes.

Previous studies provided the theoretical and practical basis for developing the methodology adopted in this research. By identifying gaps in the literature, validating qualitative approaches, and formulating data collection protocols, it was possible to build a robust and appropriate methodological framework for investigating software development in startups. Based on these insights, the research was organized into three main phases, as described below, detailing the planning, execution, and synthesis of results.

This methodological approach allowed for an in-depth analysis of software engineering practices in startups, ensuring that the final recommendations were grounded in solid empirical evidence and aligned with these companies' dynamic and innovative contexts.

Phase 1: Planning the Research

Stage 1 - Definition of the Context and Motivation: We extensively explored existing literature reviews to uncover established knowledge and identify prevalent challenges in the field.

Stage 2 - Definition of Research Goals and Questions: Building upon insights derived from our literature review, we meticulously defined the overarching research objectives and research questions that will steer our investigation.

Stage 3 - Definition and Selection of Research Methods: We methodically designed a set of empirical studies to serve as the foundation for our research.

Phase 2: Conducting the Multiple-case Study

Stage 4 - Case Study: This stage constituted a multiple-case study that involved in-depth, semi-structured interviews with the Chief Executive Officer (CEO) and Chief Technical Officer (CTO) of a diverse set of early-stage software startups. This approach also encompassed non-participatory observations and data analyses, offering profound insights into the software development practices within fourteen startups.

Phase 3: Synthesizing gathered Evidence

Stage 5 - Cross-Case Analysis: In this stage, we rigorously analyzed the empirical evidence gathered during the studies carried out in Phase 2. This comprehensive analysis formulated recommendations tailored to early-stage startups, providing invaluable guidance to these enterprises.

This methodological approach is meticulously structured to ensure the depth and breadth of our research, driving our investigation into software engineering practices within startups and the unique challenges and opportunities they present.

1.4 THESIS OUTLINE

Figure 1.2 illustrates the research design.

Figure 1.2 Research Approach.

- **Chapter 2 - Software Startups.** This chapter presents underlying concepts of software startups, including a brief discussion on existing empirical knowledge.
- **Chapter 3 - Grounded Theory.** This Chapter contextualizes the theoretical foundation of grounded theory.
- **Chapter 4 - Research Methodology.** This Chapter presents the specific procedures and techniques to identify, select, process, and analyze data from this thesis, evaluation, and the reliability of the study.
- **Chapter 5 - Startup Software Engineering Practices.** This Chapter presents the multiple-case study analysis, where we interviewed fourteen startups in their initial stage.
- **Chapter 6 - Discussion and Recommendations**. In this Chapter, we present a discussion of the results of this thesis and recommendations for early-stage startups.
- **Chapter 7 - Conclusions and Future Work.** This Chapter summarizes the thesis's key findings and sketches the direction for future work.

Chapter

2

SOFTWARE STARTUPS

The landscape of startups in Brazil and worldwide is experiencing an unprecedented growth surge. Even in the middle of constraints imposed by the global pandemic, the entrepreneurial spirit has remained unbroken, ushering in a wave of new startups. A report by the *Associação Brasileira de Startups* (ABASTARTUP, 2023) reveals a remarkable 207%

In the realm of innovation, Tim e Barry (2009) provides a crucial perspective, suggesting that for an innovative idea to attain success, it must adhere to three vital criteria: desirability (solving a genuine customer need), viability (leveraging existing operational strengths), and feasibility (rooted in a sustainable business model, c.f. Figure 2.1). Unfortunately, many startups place disproportionate emphasis on the latter two aspects, diving headfirst into technological inventions or novel business models while sidelining the user's perspective (MÜLLER; THORING, 2012). The consequence of this oversight is often the development of products that fail to address real user needs, rendering them undesirable, unneeded, and ultimately unmarketable.

Startups are more than small-scale versions of large corporations; their innovative business models distinguish them (OKANOVIĆ; JEVTIĆ; STEFANOVIĆ, 2020; BLANK, 2010). At the heart of every startup lies an inventive concept with the potential to drive social, environmental, or economic change (DASH, 2019). These agile entities nurture a culture of creativity, collaboration, and an unwavering commitment to problem-solving, fostering a distinct sense of purpose and agility. It is no surprise that a recent survey by Nielsen (2023), a global information and measurement company, found that 54% of Generation Z aspires to embark on their entrepreneurial journey, driven by the desire to be their boss. The team's competency is the backbone of a startup's ability to tackle the challenges associated with the problems it seeks to solve. Furthermore, startups operate with a forward-looking perspective, navigating the unpredictable terrain of the problem space (PATERNOSTER et al., 2014; KLOTINS; UNTERKALMSTEINER; GORSCHEK, 2015; BERG et al., 2018).

The startup market in Brazil and worldwide has been experiencing record-breaking growth. Even amid the restrictions imposed by the pandemic, many new startups have

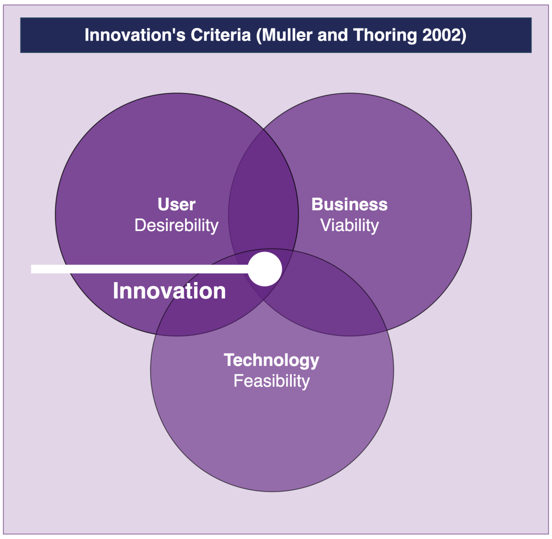


Figure 2.1 Innovation Criteria for Success Müller e Thoring (2012).

emerged. According to the Associação Brasileira de Startups (ABASTARTUP, 2023), the number of startups in Brazil grew by 207

As the market matures, new fundraising opportunities continue to emerge. Angel investment, loans, crowdfunding, and government funding are just a few of the many options available to startups seeking capital. Additionally, the increasing use of social networks enables anyone to launch a business and attract customers. Various platforms now allow entrepreneurs to connect with their target audience without the need for a physical location. Technological advancements remain the primary driving force behind the expansion of the startup market, creating numerous business opportunities.

Tim e Barry (2009) states that for an innovative idea to succeed, it must meet three key criteria: desirability, meaning it addresses a real customer need; viability, ensuring it leverages existing operational capabilities; and feasibility, meaning it operates within a sustainable business model (see Figure 2.1).

Many companies focus excessively on the last two criteria—starting with a technological invention or a business model—while neglecting the user perspective (MÜLLER; THORING, 2012). As a result, they often fail because their products do not solve a real user problem. When a product is undesirable, meaning it lacks demand or fails to address a meaningful need, customers will simply not buy it, leading to business failure.

Startups are not simply small businesses or scaled-down versions of large corporations (OKANOVIĆ; JEVTIĆ; STEFANOVIĆ, 2020; BLANK, 2010). One key way to identify a startup is by examining its business model. These companies develop innovative ideas with the potential for social, environmental, or economic impact (DASH, 2019). In general, startups cultivate a culture that fosters creativity, collaboration, agility, continuous learning, and incentives for solving complex problems. According to a recent survey by Nielsen (2023), a global information, data, and measurement company, 54% of Generation Z aspire to start their own businesses, primarily driven by the desire to be their own boss. A startup's ability to tackle challenges is strongly influenced by the competencies of its team. To achieve sustainable growth, startups must adopt a long-term perspective in their decision-making (PATERNOSTER et al., 2014; KLOTINS; UNTERKALMSTEINER; GORSCHEK, 2015; BERG et al., 2018).

This chapter delves into essential concepts and terminologies underpinning software startups' realm. We define and characterize software startups, explore their lifecycle, differentiate them from small and medium enterprises, and analyze the practices, theories, and models that guide their software development journey.

2.1 CONCEPTS AND TERMINOLOGIES

To ensure a shared understanding and prevent ambiguity, the following terminology is used throughout this thesis:

- **Software Development Strategy**: The overall approach a company adopts to guide product development.
- **Software Engineering Activities**: The essential tasks required to transform an idea into a market-ready product. These typically include, but are not limited to, requirements engineering, design, architecture, implementation, and testing.
- **Software Engineering Practice**: The established way of working within a startup's team, encompassing methodologies, workflows, and routines.
- **Software Engineering Elements**: Any practice, tool, or artifact that contributes to and supports software engineering activities.
- **Quality Attributes**: Factors influencing system runtime behavior, design, and user experience. These attributes affect multi-tiered applications and include concerns related to system architecture, performance, maintainability, usability, and scalability. Some are integral to overall system design, while others pertain to runtime, development, or user-specific aspects.
- **Software Product**: Any software-based product or service.
- **Software Process Improvement**: Any framework, methodology, or tool designed to enhance the efficiency and effectiveness of software development processes.

2.2 SOFTWARE STARTUPS: DEFINITIONS

Carmel (1994) was the first to introduce the term "software package startup" and the abbreviated term "software startup." In an exploratory study, the author investigated time-to-completion accelerators for software projects within twelve "bundled" startups. Time-to-completion, or time-to-market or rapid development, measures the duration from the beginning of software development to the release of a functional product. The study aimed to identify critical accelerators, including development methodologies, development teams, project management, resource allocation, risk analysis, and incremental innovation. The development team with cross-functional talents emerged as the most crucial accelerator. The study highlighted practices such as high communication within small teams, rapid product design, and non-formalized development methodologies.

Software startups have emerged as key drivers of the economy and innovation despite their relative inexperience (SUTTON, 2000). These startups face challenges related to limited resources, immaturity, multiple influences, high technology, and turbulent markets. According to Sutton (2000), startups share characteristics with small and medium-sized young companies but are typically less mature and need more resources. Hilmola, Helo e Ojala (2003) conducted a systems dynamics simulation to enhance understanding of product development in situations where a software startup may not fund its operations with cash flow. The study emphasized the critical role of improving product development lead time in the software startup environment.

Coleman e O'Connor (2008) characterized software startups as product-oriented with small development teams usually led by developers. They referred to such startups as "software product startups" and investigated the initial establishment of the software development process within these organizations. The study utilized a grounded theory approach and identified the experiences of small software organizations in developing processes to support their software development activities. According to Coleman e O'Connor (2008), startups heavily rely on a software development manager with the expertise and know-how to achieve their objectives.

A startup, as described by Blank (2010), is an organization seeking a repeatable and scalable business model that outlines how value is created, delivered, and captured. Startups formulate hypotheses about various aspects, including the customer, distribution channel, product, product value, operating costs, and users, which they need to validate in the market. The term "repeatable" refers to consistently creating and delivering value to customers, while "scalable" implies the ability to adapt the business model to a growing customer base. Ries (2011) defines a startup as a company dedicated to creating something new under extreme uncertainty with intense human initiative. The startup's product or service being innovative is also an essential part of the definition.

The term "software startup" encompasses several variables. Paternoster et al. (2014) conducted a systematic mapping study to structure and analyze research about software startups and identified 43 primary studies, with only four making substantial contributions to startup software engineering activities. Software startups are expected to achieve aggressive business expansion and growth in fast-growing markets. Giardino et al. (2016) and Unterkalmsteiner et al. (2016) use the term to refer to organizations focused on creating high-tech and innovative products with little or no operational history, aiming for aggressive growth in highly scalable markets. Software startups share characteristics with other startups, including limited resources, lack of operational history, high-tech challenges, cutting-edge tools, and innovation-driven development. Table 2.1 presents the timeline of different startup definitions. In this thesis, we adopt the definition of Unterkalmsteiner et al. (2016), which defines a software startup as an organization seeking a repeatable and scalable business model while developing software with little or no operational history.

Table 2.1 Software Startup Definition Timeline.

1994	Carmel (1994)
2000	Sutton (2000)
2003	Hilmola, Helo e Ojala (2003)
2008	Coleman e O'Connor (2008)
2010	Blank (2010)
2011	Ries (2011)
2014	Paternoster et al. (2014)
2016	Giardino et al. (2016)
2020	Blank e Dorf (2020)

2.3 KEY CHARACTERISTICS OF STARTUPS

Startups show specific organizational culture characteristics that enhance team performance and contribute to significant achievements. In Brazil alone, a 2019 survey revealed that the number of startups already exceeds 12,000 (ABASTARTUP, 2023). The primary characteristics observed in these organizations are described below:

- **Start Small:** Learning is essential to success. Even when there is a market opportunity, starting small, learning, and iterating is prudent. Developing a Minimum Viable Product (MVP) helps align user expectations by implementing the minimum functionality required to release a software version for user testing and feedback. Developing an MVP reduces risks and costs while capturing user needs.
- **Know their (potential) customers intimately:** Conduct preliminary research to create user personas and gain insights into target customers. These personas are informed by actual data, including thoughts, feelings, and motivations.
- **Have an Innovative Business Model:** Startups often emerge in complex environments with unknown problems and little information about solutions. An innovative business model differentiates the startup from other organizations in the same sector.
- **Impact the Market:** Startups aim to differentiate their products in the market and establish a competitive presence for their brand.
- **Be Disruptive:** Startups have the ability to redefine established practices and processes.
- **Have a Dynamic Team:** Startups require a multidisciplinary team that can adapt to rapid changes and uncertain conditions.
- **Develop a Plan:** Execution is key to turning a great idea into a successful business. A well-thought-out plan guides product launches, secures funding, and forms the basis for marketing and monetization strategies.

- **Seek High Growth:** Startups are driven to grow. While they may not experience high growth while discovering their business model, it remains a goal. Achieving high growth is a key element of their business model, distinguishing them from other businesses. Slowing growth indicates a transition to a more mature phase.
  - **Be Scalable:** High growth is achieved through the following steps:
    * **Use Minimal Resources:** Leveraging the business with as little funding, personnel, and space as possible.
    * **Use Technology:** Automation of operational processes, leading to efficiency and optimization.
    * **Focus on Customer Experience:** Prioritizing and enhancing the overall customer experience.
  - **Be Replicable:** The product or service should be easily replicable on a large scale.
- **Deal with High Risk:** Startups are adept at moving ideas from concept to execution, especially when dealing with new and uncertain aspects.
- **Stay Focused:** Maintaining focus and prioritization are essential. Business plans and canvases can be used as roadmaps to align the startup with its goals and customer needs.
- **Build a Satisfied Community:**
  - **Customers:** Creating a robust product or service that fosters an engaged community accelerates success.
  - **Company Culture:** A strong organizational culture based on values where team members feel comfortable and motivated.
- **Persevere:** Persistence is the ability to remain dedicated to goals and unwavering in the face of difficulties.

Additional Characteristics and Challenges: In addition to these strengths, startups also face specific limitations, difficulties, and challenges. Here are some of them:

- **Young and Immature:** Startups often have limited experience, immature processes, and organizational capabilities (BLANK, 2010; SUTTON, 2000).
- **Limited Resources:** Startup businesses deal with constraints, both financial and in terms of human resources (BLANK, 2010; RIES, 2011; SUTTON, 2000).
- **Multiple Influences:** In the early stages, startups are influenced by various sources, including investors, customers, partners, and market dynamics (BLANK, 2010; SUTTON, 2000).

- **Dynamic Technologies and Markets:** Startups often work on innovative products, demanding cutting-edge development tools and techniques (SUTTON, 2000).
- **Rapid Development:** Rapidly building and launching minimal, feature-intensive software products is a common practice in startups (UNTERKALMSTEINER et al., 2016).
- **Little or No Operational History:** Startups frequently develop innovative products with teams with limited knowledge about the product they are building.
- **Pressure Time-to-market:** Startups often enter new or highly competitive markets, which pressures them to launch and validate their products quickly (BERG et al., 2018).
- **Hypothesis-driven:** Unlike established companies that optimize an existing business model, startups aim to discover a scalable business model through various iterations and hypothesis testing in different markets (BERG et al., 2018).

2.4 STARTUPS LIFECYCLE MODELS

Crowne (2002) explored critical product development issues that can lead software startups to failure. The authors presented a startup evolution model from a software product development perspective. This perspective divided the startup lifecycle into three phases: *Startup*, referring to the period between product conception and the first scaling; *Stabilization*, which starts when the first customer receives the product; and *Growth*, beginning when the startup takes time to customize the product or service for a new customer without overwhelming the development team. According to the authors, the startup reaches *Maturity* when the company has all the necessary processes for product development. Figure 2.2 illustrates Crowne (2002)'s perspective.

Blank (2015, 2020) proposed a Startup Lifecycle with three steps: (i) search, (ii) build, and (iii) grow, as shown in Figure 2.3. In the search step, startups look for a repeatable and scalable business model, often requiring multiple iterations and pivots to find product/market fit. Startups exit the search step when they validate their customer acquisition and cost assumptions. The build step focuses on scaling the startup, achieving positive cash flow, and implementing organizational processes. The growth step involves scaling the startup further, with key performance indicators, processes, and procedures in place.

According to Blank (2020), focusing solely on the product development perspective can lead to disaster. A product-focused approach may lose sight of customer satisfaction, risking project success. Therefore, Blank (2020) proposed a customer-focused development process that complements the product-oriented software development model. The customer development process has four phases:

- **Customer Discovery:** In this phase, the startup declares and validates the customer's problem hypothesis. It moves to the next phase when there is evidence that customers are ready to pay.

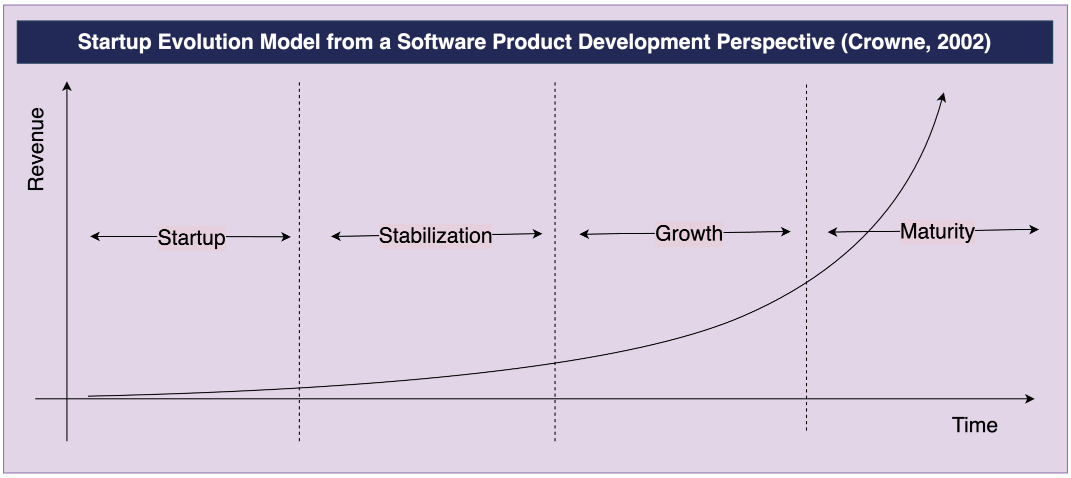


Figure 2.2 Startup Evolution Model from a Software Product Development Perspective (CROWNE, 2002)

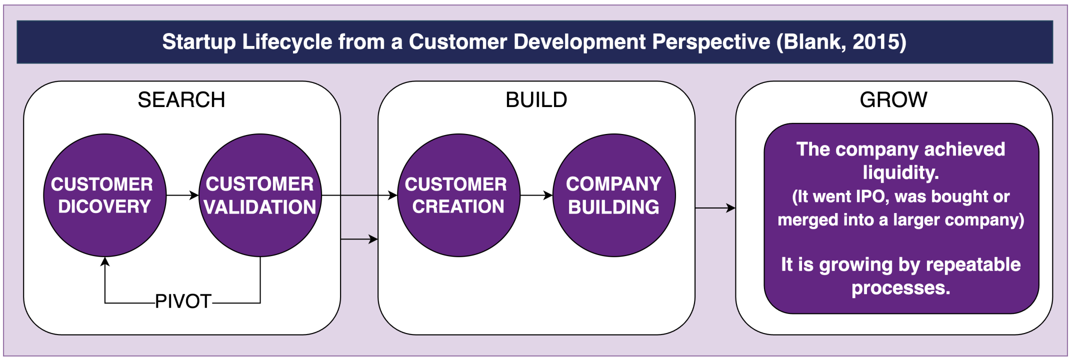


Figure 2.3 Startup Lifecycle from a Customer Development Perspective (BLANK, 2015, 2020)

- **Customer Validation:** This phase aims to build a repeatable sales process, ensuring a positive market response. Startups must develop and test a sales process. A pivot is necessary if the business model cannot be validated.

- **Customer Creation:** Building on initial sales success, startups need to increase their customer base. The goal is to create end-user demand and direct it toward the sales channel.

- **Customer Building:** This phase is transitioning from a startup focused on learning to a mature business. It includes departments and a scaled business development model.

Wang et al. (2016) proposed a two-dimensional startup lifecycle with learning and product development stages. The learning stages, based on the Lean Startup methodology (RIES, 2011), are defining/observing a problem, evaluating the problem, defining a solution, and evaluating the solution. These stages are not linear but involve multiple build-measure-learn loops to find a sustainable business model. The product development process comprises six stages: concept, under development, functional prototype, functional product with limited users, functional product with high growth, and product maturity. While these processes are presented as linear progress, they are iterated multiple times in practice.

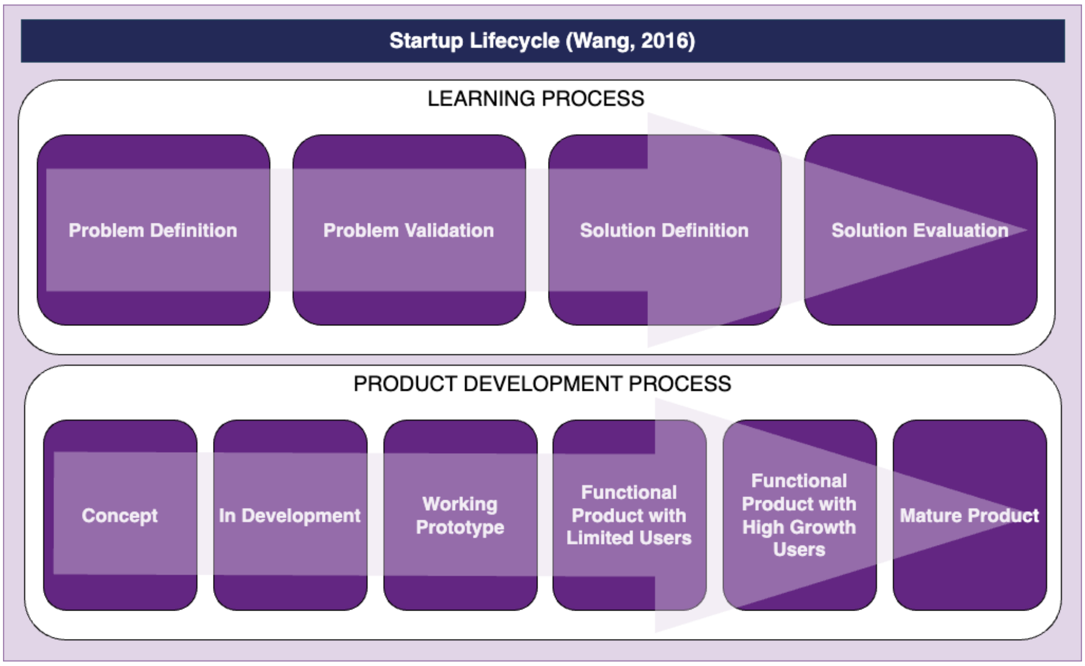


Figure 2.4 Startup Lifecycle (WANG et al., 2016)

2.5 STARTUP X SMALL AND MEDIUM ENTERPRISES

The Brazilian Institute of Geography and Statistics (IBGE)[1] is the country's leading provider of data and information, meeting the needs of Brazil's most diverse civil society and government agencies. One of the size classifications of companies by revenue that IBGE uses organizes companies as follows:

- micro-entrepreneur, less than or equal to R$81 thousand (≈$15k);
- micro-sized enterprise, less than or equal to R$360 thousand (≈$68k);
- small-sized enterprise, greater than R$360 thousand and less than or equal to R$4.8 million (≈$900k);
- medium-sized enterprise, greater than R$ 4.8 million and less than or equal to R$300 million (≈$56M);

[1]https://www.ibge.gov.br/estatisticas/economicas/comercio/9068-demografia-das-empresas.html

- large-sized enterprise, greater than R$300 million.

Table 2.2 Difference between SMEs and Startups adapted from Bajwa (2020)

Element	**SME**	**Startup**
Organizational Structure	Structured and stable group of employees	Varies from initially unstructured, agile team to more structured organization
Business	High Predictability	Low Predictability
Business Goals	Improvements	Transformation
	Incremental	Disruptive
	Vision	Purpose
	Mission	Manifesto
	Processes	Experiments
	Control	Autonomy
	Hierarchies	Networks
	Business Plan	Business Model
	Lean Six Sigma	Lean Startup
	Stable Business	High Growth
	Sales Plan	Growth Hacking
Business Risks	Low Uncertainty	High Uncertainty
Funding	Self-funded	Initially self-financed towards striving to secure large-scale funding from venture capitalists
	Bank Loan	Angel Investors
Product	Often known	Often unknown
		Often related to advanced technology
	Features	Experiences
	Quality	Innovation
Software Development Methodologies	Waterfall	Agile
	Agile	Lean Startup
		Design Thinking
Product Development	Validated Learning	Continuous Learning (Check Do Act Plan)
Team	Always Improve	Learn Fast
Customer	Customer Voice	Empathy
	Often known	Often unknown

We use the same classification criteria for the other variables as a range. Table 2.2 compares Small- and Medium-Sized Enterprises (SME) and Startups. Their primary goals are the main difference between these startups and Small- and Medium-Sized Enterprises

(SME). Small businesses are guided by images of profitability and stable long-term value, whereas startups are focused on funding revenue and growth potential. A startup will be designed to be temporary and achieve a repeatable and scalable business model. Managers are looking for market responses to the innovative product or service they hope to sell and will be very careful about delivering value to their customers. An SME is an independently owned and operated enterprise, heavily organized for profit, and will not be dominant in its field. Its activity is familiar to the market: the SME generally sells products or services known to the public with experienced local niches. On the contrary, a startup intends to grow into a much larger company, significantly impacting the existing market or creating new markets; the small business is often planned to stay small.

2.6 SOFTWARE STARTUP PRACTICES

Startups often encounter significant engineering challenges in their quest to employ effective software development practices that can efficiently structure and organize their activities while ensuring high-quality results in a timely manner (KLOTINS et al., 2019; RAVASKA, 2020; UNTERKALMSTEINER et al., 2016).

These challenges can be effectively addressed by implementing sound practices that focus on software development process improvement (SUTTON, 2000), requirements engineering (GRALHA et al., 2018), software development methodologies (PATERNOSTER et al., 2014; GIARDINO et al., 2016; CAVALCANTE et al., 2018; KLOTINS et al., 2019; POMPERMAIER, 2021), software testing (REN; DONG, 2017), development team management (MELEGATI; GOLDMAN, 2015; MELEGATI et al., 2019b), and technical debt management (BESKER et al., 2018; KLOTINS et al., 2018; CICO et al., 2021).

It's important to recognize that software engineering practices in the context of software development evolve from fundamental to more advanced levels as the team, organization, and product grow in size and complexity. Established organizations tend to have more stability and resources to support this evolutionary process. Consequently, startups can benefit greatly from adopting best practices commonly employed by established entities, especially small-sized and medium-sized companies (EDISON; WANG; ABRAHAMSSON, 2015; EDISON et al., 2018).

The dissemination of research findings regarding software startup practices and insights from small and medium-sized organizations can facilitate a mutually beneficial exchange of engineering best practices between startups and these more established entities. Despite the available resources, there remains a pressing need for practical guidelines tailored specifically to early-stage startups, particularly those dealing with the intricacies of managing software development activities.

In summary, Tolvanen (1998) defines information system development as a change process related to object systems using a method consisting of techniques and tools. Techniques are structured sequences of steps and rules for managing representations of information systems, while tools are computer-based applications that facilitate the application of these techniques.

The term "practice" is defined as the practical application or use of a specific method or technique. Jacobson et al. (2019) suggest that a collection of practices collectively forms a method, with each practice guiding a particular aspect of the team's work. A method may encompass a range of practices, extending beyond just software development to encompass organizational practices, such as those related to funding and customer relationships. This broader perspective is particularly relevant in the context of startups, where working with limited resources under high pressure demands an integrated approach to various aspects.

Ravaska (2020) posits that "practice" is best defined as the established way of working for a startup's team. In this thesis, we adopt Ravaska's definition of practice.

2.6.1 Software Engineering Practices

Startups confront unique challenges when implementing structured software development processes while preserving flexibility and agility (SUTTON, 2000; COLEMAN; O'CONNOR, 2008). They often turn to agile methodologies such as Lean Startup, Extreme Programming (XP), Scrum, and Scrumban due to their emphasis on iterative development, frequent feedback, and adaptability (ABRAHAMSSON; OZA; SIPONEN, 2019; RIES, 2011; GHEZZI; CAVALLO, 2020). These methodologies empower startups to swiftly adapt to market changes, evolving customer needs, and dynamic business goals.

As highlighted by Coleman e O'Connor (2008), the experiences of development leaders play a pivotal role in startups' software development processes. Therefore, a skilled and experienced development team is vital for navigating the dynamic and fast-paced software development environment that characterizes startups.

Time-to-market, which spans from the product's conception to its availability for sale, is a critical factor for startups (COLEMAN; O'CONNOR, 2008). Startups often require rapid and efficient software development processes to bring their products to market.

Agile and lean methodologies are favored among startups because of their flexibility and capacity to deliver swift results through an iterative and incremental approach (GIARDINO; WANG; ABRAHAMSSON, 2014). However, certain challenges can arise in software development, including informal requirement processes, deficient documentation, and post-launch product validation (GRALHA et al., 2018; KO, 2017; POMPERMAIER; PRIKLADNICKI, 2020; EDISON et al., 2018). The lean methodology addresses these challenges by emphasizing rapid learning, identifying critical aspects of the business, and developing a minimum viable product (MVP) that can be swiftly tested and modified based on feedback (GIARDINO; WANG; ABRAHAMSSON, 2014). Researchers like Nguyen-Duc e Abrahamsson (2016) have extensively investigated the role of MVP in startups.

Giardino, Wang e Abrahamsson (2014) proposed several strategies to expedite MVP development in startups, including utilizing established frameworks to adapt the product to meet market demands rapidly, employing evolutionary prototyping and experimentation with existing components, continuous validation with key user groups, focusing on delivering continuous value through key functionalities, empowering the team to enhance performance, learning from users, and using easily implementable tools to make product development more dynamic.

Despite existing research, the number of studies providing transferable results for the software development industry remains limited. Coleman e O'Connor (2008) "include and enable" management approach stands out as it provides development teams the freedom to work autonomously, aligning with the benefits identified by Giardino, Wang e Abrahamsson (2014).

Furthermore, the concept of ecosystems in the context of startups is gaining attention. Recent studies (CUKIER; KON; KRUEGER, 2015; KON et al., 2014) indicate that ecosystems foster interactions among startups and can attract investors and skilled talent. Team training remains an underexplored area.

Despite the existing body of literature, there is a need for additional empirical studies in software engineering practices for startups. Gaps and limitations identified in Giardino et al.'s 2014 work underscore the need for empirical evidence to validate and evaluate software engineering practices' effectiveness in the startup context. Conducting such studies will contribute to developing reliable guides and methods for software startups. Additionally, comparative studies evaluating the effectiveness of popular development frameworks in startup contexts can offer valuable insights to aid startups in choosing the right frameworks for their software development processes.

2.6.2 Startup Software Methodologies

The term "digital transformation" signifies that companies have harnessed digital technologies to enhance their operations and deliver added value to customers (MATT; HESS; BENLIAN, 2015; VIAL, 2019). This transformation is driven by various factors, including shifting customer expectations, technological advancements, heightened competition, remote work, and data-driven decision-making.

By striving for greater efficiency, an improved customer experience, a competitive edge, and the capability to collect and analyze data, digital transformation empowers companies to better respond to market dynamics and user needs. It also enables them to reach new markets and expand their client base. Furthermore, digital technologies, such as cloud computing, artificial intelligence, edge computing, 5G, and the Internet of Things, facilitate data collection and analysis, yielding valuable insights into customer behavior, market trends, business operations, and strategic decision-making.

Customers increasingly demand digital communication channels and transaction processes to streamline their experiences. The COVID-19 pandemic has further accelerated this digitization trend, with many companies adopting video conferencing and collaboration tools to support remote work. While digital transformation is not a software development methodology in itself, it significantly influences how companies approach software development. This transformation emphasizes agile development methods, prioritizing adaptability and responsiveness over rigid planning and extensive documentation. It often involves rapid prototyping and experimentation to validate and refine new concepts swiftly.

In response to the limitations of traditional software development methodologies, many startups have embraced alternative approaches. Software startups' most popular software development methodologies include prototyping, design thinking, agile methodologies, the lean startup approach, and DevOps.

2.6.2.1 Prototyping

Software startups use Prototyping to develop and test business ideas and to validate market viability (BJARNASON, 2021). *Prototyping* is an activity that commonly creates incomplete or minimalist software versions. Usually, to try out specific new features, request feedback on requirements or user interfaces, explore further requirements, software design, or other helpful information. The prototype becomes the final software product with extensive development, rework, or refactoring. We can prototype disposable code or paper products, an evolution of a working project, or an executable specification. Examples of prototyping targets include: *a requirements specification; an architectural design element or component; an algorithm; a human-machine user interface.* Software engineers or other stakeholders can evaluate a prototype in several ways. Prototypes can be assessed or tested against implemented software or a target requirements set. They could also function as a template for a future software development effort, such as in a user interface specification.

2.6.2.2 Design Thinking

According to Müller e Thoring (2012), Design Thinking is a versatile framework for comprehending the daily challenges faced by individuals and creating innovative solutions that are both practical and user-centric. It is an interdisciplinary approach that assembles teams to explore users' concerns, identify obstacles, and leverage the principles of Design Thinking to devise solutions that are not only novel but also widely beneficial.

Design Thinking teams aim to develop products or services that align with the trinity of feasibility, viability, and desirability, ensuring that their solutions are technically and economically sound while being genuinely appealing to end-users.

One hallmark of Design Thinking is its highly iterative and incremental nature. It relies heavily on extensive user research, constant feedback, and a series of iteration loops. This iterative process is meticulously organized.

Design thinkers begin by collecting comprehensive information about end-users and their specific domains. This data is subsequently validated, and design ideas are translated into tangible prototypes that substantiate their importance and value to end-users (PLATTNER; MEINEL; LEIFER, 2010).

In the context of software projects, the application of Design Thinking often involves the creation of Graphical User Interface (GUI) prototypes. However, it is important to note that GUI prototypes for complex multi-user software systems may primarily represent the system's view.

The transition from the Design Thinking phase to software development occurs when the problem and solution concepts have been adequately specified, transforming them into actionable development tasks. This transition blurs the lines of management responsibilities, as they overlap between the Design Thinking and software development phases (PLATTNER; MEINEL; LEIFER, 2010).

2.6.2.3 Agile Methodologies

Agile methods are designed to address the need to reduce the perceived overhead associated with traditional plan-based methods often used in large-scale software development projects. These agile methodologies, commonly referred to as lightweight methods, are characterized by several key features: short and iterative development cycles, more straightforward design approaches, code refactoring, test-driven development, self-managed teams, active customer involvement, and a strong emphasis on delivering a demonstrable functional product within each development cycle.

The literature presents a variety of agile methods, with some of the most popular approaches including Extreme Programming (XP), Scrum, and Feature-Driven Development (FDD) (BOURQUE; FAIRLEY, 2014). Let us briefly introduce each of them:

- **Extreme Programming (XP)**:

  It is an Agile methodology that employs stories or scenarios for requirements. It emphasizes developing tests first, involves the customer in the development team, and uses practices such as pair programming, refactoring, and continuous code integration. In XP, stories are grouped into tasks, prioritized, estimated, developed, and tested. Each software increment is tested with both automated and manual testing, so increments can be released frequently. XP adopts a strategy of constant monitoring and minor adjustments during software development. Its core values include communication, simplicity, feedback, courage, and respect. XP is based on principles such as rapid feedback, assuming simplicity, incremental change, embracing change, and working with quality.

- **Scrum**:

  Scrum is an Agile software development methodology that helps teams structure and manage their work through values, principles, and practices. It requires roles like a Scrum Master, Product Owner, and a Product Backlog. The Scrum Master is responsible for promoting an environment where the Product Owner requests the development of activities for complex problems through a Product Backlog. The team turns a selection of activities into a Value Increment during a Sprint. Scrum is intentionally incomplete, defining only the necessary parts for its implementation. Scrum's rules guide the relationships and interactions rather than providing detailed instructions. It is built by the group's collective intelligence using it.

- **Feature-oriented Development (FDD)**:

  It is an iterative, feature-driven approach to software development. It is divided into five phases: (i) develop a product model to cover the breadth of the domain, (ii) create the list of features, (iii) build the feature development plan, (iv) develop designs for feature-specific iterations, and (v) code, test, and integrate the features. FDD is an incremental software development approach similar to XP, but it assigns code ownership to individuals rather than the team. FDD emphasizes an architectural approach to software that promotes getting features right the first time rather than emphasizing continual refactoring.

Startups often prioritize delivering value to customers while avoiding unnecessary investments in features. This approach has led to the adoption of agile, iterative software development methods. These methodologies enable startups to quickly test assumptions, make necessary adjustments, and improve their chances of success. Learning from other startups' experiences, mistakes, and successes further contributes to their growth.

In addition to well-known agile methodologies such as Scrum, XP, and Lean, startups have explored other agile approaches that are gaining traction in the industry. Here are a few of these methods:

- **Lean Software Development (LSD)**: Emphasizes eliminating waste, building quality, and maximizing customer value.
- **Adaptive Software Development (ASD)**: Prioritizes human collaboration, team self-organization, and adaptation to external and internal changes.
- **Kanban**: Focuses on workflow optimization through visualizing work, limiting Work in Progress (WIP), and applying a "pull" principle.

2.6.2.4 Lean Startup

A startup serves as a catalyst that transforms initial ideas into viable business ventures. In the realm of software startups, the focus is on creating MVP intended for high-potential markets. At this stage, the startups often lack specific customers or possess limited knowledge about customer preferences. Consequently, the software development process in startups frequently entails building products with evolving requirements, huddles, minimal testing, and limited documentation.

Moreover, the landscape of software development has shifted its emphasis from traditional "hard" aspects such as system structure and process control to more "soft" elements, including the role of people, leadership, culture, and values within organizations (OKANOVIĆ; JEVTIĆ; STEFANOVIĆ, 2020). Startups navigate this transition by prioritizing product creation and customer development. It is widely acknowledged that during their initial years of operation, startups grapple with the challenge of establishing a repeatable and scalable business model that can ensure long-term sustainability.

The Lean Startup methodology, introduced by Ries (2011), provides a structured approach for startups to navigate this complexity through three key steps: build, measure, and learn. In this context, a startup's business model is considered a tentative hypothesis that must be rigorously tested to determine its ability to deliver value to customers. Ries' methodology empowers organizations to discover a repeatable and scalable business model, especially when the problem is ill-defined and no pre-existing solution can guide the creation of a new one.

When a startup proceeds to create and launch its product to validate the business model, customer feedback becomes an invaluable asset. This feedback loop generates essential data and insights that drive the startup's learning process.

2.6.2.5 DevOps

DevOps is a software development culture aimed at integrating development and operations teams to achieve the continuous delivery and operation of software. The core idea behind DevOps is to foster collaboration between development and operations teams, sharing responsibilities and resources to enhance the efficiency and quality of the software delivery process. It emphasizes automating various stages of software development, testing, and deployment to enable teams to work more quickly and efficiently.

DevOps is essentially an organizational transformation where cross-functional teams work cohesively to enable the continuous delivery of software, moving away from the conventional setup of disparate groups handling distinct functions. This shift in approach accelerates value delivery and minimizes miscommunication among team members, leading to faster resolution of the issue. The achievement of this goal relies on a set of tools and practices that standardize and automate processes, including code versioning, continuous integration, continuous delivery, continuous monitoring, and infrastructure as code.

At its core, DevOps represents a cultural transformation that strives to unite development and operations teams, facilitating the swift and efficient delivery of high-quality software. This cultural shift encourages collaboration among development, quality assurance, and operations teams. Organizations adopt continuous delivery practices, emphasizing the release of minor updates instead of adhering to rigid processes, leading to cycle times measured in minutes (EBERT et al., 2016).

One vital practice within the DevOps culture is Continuous Integration, where team members frequently integrate their work. Typically, individuals merge their contributions at least once daily, often doing multiple integrations throughout the day. Automated builds verify each integration, conducting tests to identify integration issues promptly (FOWLER; FOEMMEL, 2006).

2.7 RELATED WORK

Startups have good ideas, but only some can turn them into successful commercial products. These ventures face challenges in creating technologically innovative products and obtaining funding, whether through paying customers or investors. The challenges faced by startups are varied, and research on software startups can benefit the startup community and professionals interested in the topic.

Previous studies have extensively investigated the adoption of software engineering practices by startups. However, the contemporary business landscape has evolved rapidly due to several factors, including increasing consumer demands, climate change, and the redirection of business interests. Integrating software engineering practices into daily operations aligns with the inherent uncertainties entrepreneurs face. On the other hand, as companies mature, their development teams tend to adopt formal practices and industry standards. Startups, however, face the challenge of identifying the most appropriate software engineering practices, quality attributes, and tools during their early stages. Aligning these aspects with your limitations, such as time, resources, and knowledge, presents a great challenge.

The existing literature highlights the critical role of software engineering practices, essential quality attributes, and supporting tools in enabling startups to advance to the next stage of maturity and success. The main objective of this research is to provide empirical insights into the context of software development in startups, serving as a guide for entrepreneurs interested in creating software startups or in the early stages. To achieve this objective, case studies with early-stage software startups were conducted. This objective is aligned with the thesis, exploring how software engineering practices impact the early stages of development in startups and offering practical recommendations to improve the efficiency and quality of software development processes in these companies.

Another significant contribution comes from Cavalcante et al. (2018), who established specific guidelines for software development in startups. Their study follows a systematic approach, covering exploratory, development, and refinement phases, highlighting the growing body of evidence that addresses the specific needs of startups. The proposed guidelines provide valuable assistance for software development activities in startups, potentially improving their chances of success. These guidelines focus on two critical areas of software engineering specified in the SWEBOK: Software Construction and Software Quality. This targeted focus aims to maximize benefits for startups in a timely and practical manner.

Figure 2.5 visually illustrates the guidelines proposed by Cavalcante et al. (2018), which are structured around 11 software construction and 13 software quality guidelines for 24. Each guideline includes a title, description, expected outcomes, recommended tools, and the source.

Although some models in the literature attempt to understand how startups use these software engineering practices, such as the Greenfield Startup Model (GIARDINO et al., 2016), the Progression Model (KLOTINS et al., 2019), the PRESS Model (POMPERMAIER; PRIKLADNICKI, 2020). The HyMap (MELEGATI; GUERRA; WANG, 2022), each of these studies offers a different perspective on the context of software development in startups. However, they contribute to a broader and more detailed understanding of software engineering practices in these innovative and challenging environments.

Our research distinguishes from that of Cavalcante et al. (2018) in several aspects. Firstly, it is grounded in an empirical case study, whereas Cavalcante et al. (2018) initially conducted a systematic literature review and a subsequent survey with professionals working in startups. Additionally, the evaluation of the guidelines in this work has been carried out through the context of the case study, allowing a more in-depth exploration of the socio-technical aspects of software development in startups. Lastly, this research presents a more comprehensive approach, addressing both technical and social aspects of software engineering, acknowledging the socio-technical nature of this field, as proposed by Hoda (2021).

Our study sheds light on the software development practices of startups at different stages of their development, adding valuable insights to the field. Although there is limited research explicitly focused on this topic, we have identified several pain points that warrant further exploration:

Figure 2.5 Guidelines Maps from Cavalcante et al. (2018)

- Ongoing discussions surround best practices for various stages of the software development process.

- The academic community has shown significant interest in the subject of software engineering practices, as indicated by the partially achieved results of previous publications.

- Cavalcante et al. (2018) conducted a study on the context of software engineering in startups, offering recommendations related to software construction and quality aspects.

We address this knowledge gap by developing a model that provides startups with specific recommendations and practices tailored to their unique needs and challenges at different lifecycle stages. This model is a valuable resource for startups, offering guidance for selecting and adopting appropriate software engineering practices that align with their specific context and objectives.

2.8 CHAPTER CONCLUSION

This chapter established the foundation for this research by exploring fundamental concepts, definitions, and themes related to startups, with a particular focus on software startups. It provided a comprehensive characterization of startups, examined their life cycles, and analyzed software development within these organizations. Additionally, we clarified the term "practice" in the context of Software Engineering.

Although research on startups has been ongoing for some time, a recent surge in interest has led to a significant increase in published studies over the past decade. This growing attention highlights the relevance of startups in today's business landscape and the dynamic nature of their operations.

An important observation is that, while agile methodologies are often linked to startups, these organizations typically adopt only specific practices rather than the entire framework. They tailor these practices to fit their unique workflows, allowing them to leverage the benefits of agile methodologies while maintaining flexibility.

In summary, this chapter has provided a solid theoretical foundation for this research, offering an in-depth discussion of key concepts and setting the groundwork for further investigation. The primary contribution of this study will be the empirical insights gained from analyzing software development practices in startups. By addressing this critical knowledge gap, we aim to offer valuable guidance and recommendations to startups at various stages of their development journey.

Chapter

3

GROUNDED THEORY

Grounded Theory (GT) is a systematic research approach that offers a structured method for theory discovery through systematic data collection and analysis. It is commonly employed to generate theories about patterns of human behavior within social contexts (GLASER, 1992). This approach is particularly well-suited for addressing complex and deep questions related to the "how," "why," and "what" of the phenomena being studied (HODA, 2021). The unique characteristic of GT research is its focus on people's experiences, guiding the investigation to uncover patterns within these experiences (GLASER; STRAUSS; STRUTZEL, 1968; CORBIN; STRAUSS, 2014). As Glaser (2005) suggests, this approach can greatly enhance our understanding of various phenomena, especially from a social perspective.

In the context of this thesis, GT has been chosen as the research approach for several compelling reasons. Firstly, it allows researchers to maintain the richness of textual data, preserving the phenomenon as seen from the participants' perspectives, while considering their specific social and institutional contexts. Secondly, the utilization of GT has gained popularity as a research method within the field of Software Engineering (COLEMAN; O'CONNOR, 2007; WHITWORTH; BIDDLE, 2007; MARTIN; BIDDLE; NOBLE, 2009; DAGENAIS et al., 2010; HODA; NOBLE; MARSHALL, 2012; SOUSA et al., 2018; FARIAS et al., 2019). Lastly, numerous sources provide guidance on the concepts and practical application of GT (GLASER; STRAUSS; STRUTZEL, 1968; SEAMAN, 1999; CORBIN; STRAUSS, 1990; GLASER, 2005; SJØBERG et al., 2008).

Throughout this chapter, we will delve into the various aspects of Grounded Theory, starting with an introduction to Qualitative Research. We will then explore different Grounded Theory schools, the essential components of Grounded Theory, the method's limitations, and finally, its application in the context of Software Engineering.

3.1 QUALITATIVE RESEARCH

Research is a systematic process of data collection, analysis, and interpretation aimed at enhancing our understanding of a particular phenomenon or concern (LEEDY; ORMROD, 2005). This process usually begins with the identification of a problem or an unanswered question, which is subsequently articulated into research goals, further subdivided into smaller sub-problems. In this context, researchers may formulate hypotheses to guide their research efforts. A hypothesis represents a logical assumption, a well-informed conjecture, or an educated guess.

To address these research problems, researchers create a specific plan detailing how data will be collected, organized, and analyzed. Data analysis in research generally falls into two broad categories: quantitative and qualitative approaches. Quantitative research primarily focuses on measuring variables numerically, often employing well-established quantitative measures. In contrast, qualitative research is concerned with exploring the characteristics and qualities of a particular phenomenon, delving into nuances and complexities that may not be readily quantified.

It is important to note that these two research methods are not mutually exclusive, and they can be employed in a complementary fashion to provide a more comprehensive understanding of a given research topic. Table 3.1 provides an overview of the characteristics of quantitative and qualitative approaches.

Human behavior is a complex and multifaceted phenomenon that often eludes complete understanding through purely quantitative methods. Many aspects of human behavior cannot be adequately described and explained using statistical and other quantitative approaches. Quantifying textual data, for instance, may lead to the loss of valuable information that is essential for comprehending the phenomenon from the participant's perspective within their specific social and institutional context (SEAMAN, 1999). Qualitative research methods, such as action research, case studies, ethnography, and grounded theory (GT), play a crucial role in addressing these complexities. They have the capacity to generate well-founded hypotheses and findings that encapsulate the intricacies of the phenomenon under investigation, offering richer explanations and uncovering new avenues for further research (DYBÅ et al., 2011).

Following the identification of research problems and the formulation of research questions, qualitative researchers meticulously plan and execute their research projects. They collect detailed information directly from participants and then meticulously organize this data into categories or themes. These themes are subsequently developed into broader patterns, theories, or generalizations. These qualitative findings are then compared with personal experiences and existing literature on the topic, offering varied endpoints for qualitative studies, as depicted in Figure 3.1 (CRESWELL, 2009).

In the field of software engineering (Software Engineering (SE)), it has become increasingly evident that many of the prevailing issues are socio-technical in nature. Therefore, when conducting research in this domain, it is essential to consider a wide range of factors, including social, cultural, political, and organizational aspects (CUKIERMAN; TEIXEIRA; PRIKLADNICKI, 2007; HODA, 2021).

3.2 GROUNDED THEORY SCHOOLS

Researchers must maintain clarity regarding their choice of Grounded Theory (GT) type for their research and fully comprehend the distinctions between their chosen theory while remaining committed to the selected approach. Failing to do so may jeopardize the research's credibility. It is essential to acknowledge that Grounded Theory methodology

Table 3.1 Characteristics of quantitative x qualitative approaches. Adapted from Leedy e Ormrod (2005) and Abdullah e Raman (2001)

Questions	**Quantitative**	**Qualitative**
What is the purpose of the research?	To explain and predict	To describe and explain
	To confirm and validate	To explore and interpret
	To test theory	To build theory
What is the nature of the research process?	Focused	Holistic
	Known variables	Unknown variables
	Established guidelines	Flexible guidelines
	Pre-planned methods	Emergent methods
	Somewhat context-free	Context-bound
	Detached view	Personal view
What are the data like, and how are they collected?	Numerical data	Textual, image, and/or video-based data
	Representative, large sample	Informative, small sample
	Standardized instruments	Loosely structured or non-standardized observations and interviews
How are data analyzed to determine their meaning?	Statistical analysis	Search for themes and categories
	Stress on objectivity	Acknowledgment that analysis is subjective and potentially biased
	Primarily deductive reasoning	Primarily inductive reasoning
How are the findings communicated?	Numbers	Words
	Statistics, aggregated data	Narratives, individual quotes
	Formal voice, scientific style	Personal voice, literary style (in some disciplines)
How are the limitations?	Cannot determine how or why activity unfolds	Cannot determine effective practices
	Findings do not include all possible variants and outcomes	Findings do not generalize across settings
	Reliable findings may lack robust validity	Researcher worldview inform and limit findings
	Complicated for practitioner understanding and use	Complicated to lead to practitioners misunderstanding value with specific uses

is not designed as a rigid and unalterable framework.

GT originated from the work of Glaser, Strauss e Strutzel (1968) and underwent adaptation by Strauss e Corbin (1990), which introduced alternative terminology and a more intricate coding procedure to enhance GT's measurability. However, the changes in terminology and coding, as introduced by Strauss e Corbin (1990), have been critiqued by Glaser (1992), asserting that these modifications altered the core premise of GT.

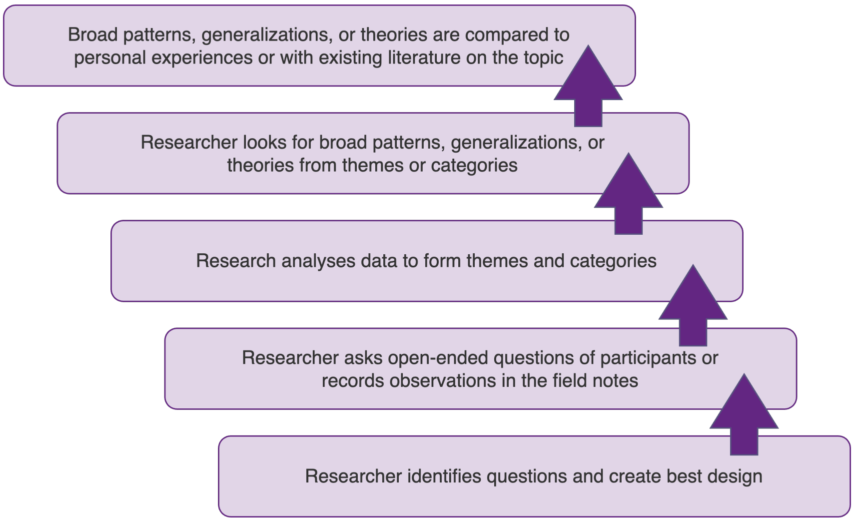


Figure 3.1 Qualitative research (CRESWELL, 2009).

This resulted in a distinct research methodology that seemingly prioritizes aligning data analysis with a predetermined coding process rather than allowing theories to organically emerge from the data, which is the overarching objective of GT.

Extensive discourse within the literature pertains to the defining characteristics of a GT study, addressing the primary distinctions between the original theory of Glaser, Strauss e Strutzel (1968), the adapted version presented by Strauss e Corbin (1990), and the constructivist version developed by Charmaz (2006). Table 3.2 offers a comparative overview of the various GT schools. Despite the versatility of GT, which extends to the handling of quantitative data, it is most commonly employed as a qualitative research method, relying on traditional data collection techniques like interviews and observations (HODA, 2021).

3.3 GROUNDED THEORY COMPONENTS

According to Glaser (1992), the key components of GT are data collection, coding, analysis, memo writing, and theoretical categorization. Here, we present a brief conceptualization of the terms and, in some cases, examples to demonstrate how we use the technique in this thesis to help the understanding.

Data collection. When conducting research, obtaining relevant and reliable information about the phenomenon under study is essential. Interviews and questionnaires are the most direct instruments, but they usually produce an incomplete picture of the phenomenon, like any technique. It is best to use various data collection methods to perform a triangulation that allows you to obtain a clearer and more reliable picture of the phenomenon. Lethbridge, Sim e Singer (2005) presents a taxonomy for data collection techniques: inquisitive, for example, brainstorming, focus groups, interviews, questionnaires, conceptual modeling; and observational, like

Table 3.2 Grounded Theory Schools adapted from Engward (2013), Sebastian (2019)

Component	**Glaser, Strauss e Strutzel (1968)**	**Strauss e Corbin (1990)**	**Charmaz (2006)**
Philosophical Influence	(Attempts to be) Free from influence	Interpretivism	Constructivism and Pragmatism
Theory	Grounded in data.	Interpreted by the observer.	Constructed by the researcher and research participant.
Role of the Researcher	Distant and detached.	Engaged and actively interpreted the data.	Constructs rather than discovers.
Starting Point	Starts with a general idea of where to begin.	Starts with a general idea of where to begin.	Starts with a general idea of where to begin.
Literature Review	Conducted following data analysis	Allowed before and during the data collection process. It can be used for data comparisons, enhance sensitivity, stimulate observation and confirm or explain results.	There is no prescribed location; it is up to the decision-making process of a given researcher. If written early, it should be revisited to critique and confirm it aligns with the researches conclusion.
Research Question(s)	There should be no pre-set or vaguely established question before data collection. Questions should become clear during data analysis.	Uses structured questions keeping partially vague for flexibility; it will become clear as data emerges. Each additional question can reference another topic of interest.	Influence how data is collected. Can and should be altered if more significant questions arise.
Data Coding and Analysis	Uses two coding phases to develop concepts that explain the phenomena: (1) substantive coding (breaking data down into small segments and grouping into similarities that begin to describe patterns in the data) and (2) Theoretical Coding (TC) (open or selective choosing of a core category and relating other categories to it to explore emergent patterns. Focuses on patterns or sheds within the data. Includes the Constant Comparative Method and use of core categories. Including TC to merge categories together into a substantive theory.	(1) Open coding (identifying, naming, categorizing, and describing phenomena), (2) axial coding (the process of relating codes to each other), and selective coding (choosing a core category and relating other categories to it). Single occurrences within the data can be coded and analyzed for significance. Includes the Constant Comparative Method and use of core categories. Have broken down and rebuilt codes to create more significant and descriptive categories that lend themselves to a substantive theory.	code everything, and (2) group all data around the most predominant codes (includes focused coding). Purports these as flexible guidelines rather than strict rules. Allowance of more than one core category.
Regard	‘True’ GT	A form of qualitative data analysis rather than GT	Twenty-first-century methodological approach

work diaries, thinking aloud protocols, shadowing, synchronized shadowing observation, participant observation, systems instrumentation, fly on the wall (participants recording their work), analysis of electronic databases of performed work, analysis of tool use, documentation analysis, static and dynamic analysis of a system.

Immediate Analysis. Immediate analysis refers to starting the analysis after coding and continuously comparing the data. This is important because it allows the researcher to identify similarities, differences, correlations, and insights based on the data. And then move on to new data collection.

Continue Comparison. Constant comparison is the process of constantly comparing derived code within the same source and between sources to identify key patterns in the data. The constant comparison concretely manifests an inductive approach, from specific instances to general patterns within the study context. Constant comparison is applied at increasing levels of abstraction to increase codes to the concept level, concepts to the subcategory (when applicable), and category levels (HODA; NOBLE; MARSHALL, 2012).

Coding. Coding is the root process in classic GT methodology. (HOLTON, 2007). Coding represents raw text data in condensed formats that best capture its essence and meaning. According to Seaman (1999), coding in this context means attaching codes, or labels, to pieces of text relevant to a particular theme or idea that interest the study. Coding means attaching labels to data segments that describe what each piece of text is about. Through coding, we raise analytic questions about our data. Coding distills data, sorts them, and gives us an analytical handle for comparisons with other data segments (CHARMAZ, 2006). Coding is the strategy that moves data from diffuse and messy text to organized ideas about what is happening. From a methodological perspective, codes hold the essence of the data. Codes serve as handles for specific occurrences in the data that simple text-based search techniques cannot find. Codes are used to classify devices at different levels of abstraction to create sets of related information units for comparison. From a technical perspective, codes are brief segments referencing another amount of text, graphical, audio, or video data. Their purpose is to classify data units. In information retrieval systems, the terms "tag," "keyword," or "tagging" are often used for "code" or "coding."

Open Coding. An inductive approach generates as many ideas as possible from initial data. Open coding refers to coding applied in the early stages of research, where the researcher remains open to any code that emerges, ensuring comprehensive coverage. Open coding might result in many codes due to its open nature. We provide an open coding example with the associate codes with quotations of interviewees' utterances below:

Raw Transcript. *"S1: Yeah, let's just say that we do not formally have a document, but for every screen or resource that we create, we list what we need... we used tools to register what we have to do, what we have done, and what needs to be evaluated... we register the tasks on Trello tool."*

Codes: Simple Design and Kanban Board

Code 1. Simple Design

Code 2. Kanban Board

Axial Coding. We related the codes through axial coding. The codes were merged and grouped into more abstract categories in this procedure, and the relation type was established. For instance, the previous codes were grouped into the following two categories:

- Category 1. Speed-up development.
- Category 2. Lack of resources.

Selective Coding. With continuous data collection and simultaneous analysis, coding becomes increasingly selective or focused as the researcher seeks a core set of recurrent codes prevalent throughout the data set. It requires decisions about which codes are most relevant or important and which contribute the most to the analysis. For each transcript, the codes, memos, and networks showing the categories and codes relations were peer-reviewed and only changed upon agreement between the researchers. Then, we used selective coding to identify core categories that best explain the phenomena.

Substantive coding. There are two coding types in a classic GT study: *substantive coding*, which includes both open and selective coding procedures, and *theoretical coding* (HOLTON, 2007). The process profits from the initial open coding of data to a core category, observed by data collection and analysis delimitation for selective coding to theoretically saturate the core category and the related categories. The researcher, in substantive coding, works with the data directly, breaking and analyzing it initially through open coding for the core category emergence and related concepts. Then, theoretically saturate the core and associated concepts through theoretical sampling and selective coding of data.

Memo Written. During the continuous coding and comparison process, the researcher should write memos to identify possible patterns within and between codes. The function of memo writing in GT organizes thinking about how data fit together and their correlations and helps articulate emerging patterns and links between codes (GLASER; STRAUSS; STRUTZEL, 1968). The writing of memos has four purposes:

1. Organize the thought easily;
2. Express ideas to elevate the data to a conceptual level;
3. Encourage the classification and reformulation of ideas;
4. Create a catalog of memos that can serve as a source for writing the theory.

The theoretical memos writing is the core stage in generating GT. If the researcher skips this stage by going directly to sorting or writing up, she needs to do GT (GLASER, 1978) after coding. Memos are theoretical notes about the data and the conceptual connections between categories.

Theoretical coding. It takes place in which the researcher refines the final theoretical concepts. Continuous comparison is a data analysis technique common to research projects. It is the simultaneous process of encoding and analysis, allowing data to be compared and contrasted.

Theoretical Sampling. Theoretical sampling refers to additional data collection guided by the findings of previous data analysis. In this phase, the researcher aims to collect more data to explore emerging patterns. The new data may confirm, refute, add, or challenge emerging practices and identify gaps in data analysis that require further exploration (GLASER; STRAUSS; STRUTZEL, 1968).

Theoretical Saturation. The researcher usually seeks to reach saturation in his studies. It is the moment when the researcher does not hear anything new from the data. In a GT study, theoretical saturation is one in which all the concepts of the theory developed are understood and substantiated from the data. Theoretical saturation occurs when the dominant emerging pattern becomes saturated (GLASER, 2005).

3.4 LIMITATION OF THE METHOD

GT can produce a substantial volume of data for analysis, which can pose challenges, especially for novice researchers. Proficiency in using GT methods is essential. Several key limitations to consider include:

Recruitment Difficulty: GT relies on an iterative recruitment process known as theoretical sampling, which involves continually recruiting and conducting new rounds of interviews with both new and previous participants while simultaneously analyzing the data. Recruitment criteria may evolve and change based on emerging insights. Since recruitment is not predefined, it can be challenging to consistently find suitable participants for the study or even gain access to past participants.

Delay in Data Collection: In GT, it is impossible to determine precisely how much data will be sufficient. Researchers continuously collect and analyze data until they reach theoretical saturation, where new data no longer contributes to novel insights for evolving theory. This often requires multiple rounds of data collection before the theory matures.

Challenges in Data Analysis: Data analysis in GT is an ongoing process involving constant comparisons between different pieces of data. Keeping up with these comparisons and findings as the research progresses can be demanding. Using qualitative data analysis software can help maintain organization during the analysis.

Trained Researchers: We recommend that more than one researcher participate in the classification, code selection, and theory development processes (GLASER, 1998). This collaborative approach allows for discussions and resolution of disagreements, helping to mitigate potential biases in the research. However, identifying researchers with a solid understanding of GT and its application can be time-consuming and require considerable effort.

3.5 GROUNDED THEORY IN SOFTWARE ENGINEERING

While originating in the social sciences, qualitative research has found applicability in various domains, including SE (STOL; RALPH; FITZGERALD, 2016; HODA, 2021).

GT, a qualitative research method, has gained increasing popularity in software engineering research, particularly for studies focusing on software development's human and social aspects.

GT has been employed to explore various phenomena in software engineering, such as software process improvement (COLEMAN; O'CONNOR, 2007), customer engagement (MARTIN; BIDDLE; NOBLE, 2009), self-organization (HODA; NOBLE; MARSHALL, 2012), agile architecture (WATERMAN; NOBLE; ALLAN, 2015), design issues (SOUSA et al., 2018), self-attribution (MASOOD; HODA; BLINCOE, 2020), and architecture design (FARIAS et al., 2019). In essence, GT offers a systematic and rigorous approach to studying complex social phenomena within software engineering, potentially yielding valuable insights that can inform both software engineering practice and theory.

For instance, Coleman e O'Connor (2007) utilized GT to investigate the use of SPI models within the Irish software industry. Dagenais et al. (2010) conducted a GT study to explore the dominant features of project landscapes, orientation challenges faced by new team members, and newcomer orientation in transitioning to a new project landscape. Whitworth e Biddle (2007) employed GT to delve into the social aspects of agile teams. Martin, Biddle e Noble (2009) used GT to investigate the role of on-premise clients in Extreme Programming (XP) projects. Hoda, Noble e Marshall (2012) carried out research spanning four years, involving 58 Agile practitioners from 23 software organizations in New Zealand and India. They identified informal, implicit, transient, and spontaneous roles contributing to the self-organizing nature of agile teams. Additionally, Sousa et al. (2018) conducted a multi-trial industrial experiment with professionals from five software companies to construct a GT explaining how developers identify design problems in practice. Lastly, Farias et al. (2019) developed an initial theory concerning the processes for designing the architecture of Smart City Mobile Applications (SCMA) systems.

3.5.1 Socio-technical Grounded Theory

Hoda (2021) emphasizes that SE is replete with socio-technical phenomena where social and technical aspects are deeply intertwined. Attempting to study one without considering the other leads to an incomplete investigation and understanding of the complex nature of SE phenomena. Socio-Thecnical Grounded Theory (STGT) is an iterative and incremental research methodology tailored to examine socio-technical aspects. It combines traditional and contemporary research techniques to generate innovative, pragmatic, concise, and adaptable theories. STGT involves interleaving rounds of *basic Data Collection and Analysis (DCA)* with emergent or structured modes of theory development. This process leverages advanced data collection, analysis, and theory development procedures, employing primarily inductive but also deductive and abductive reasoning. This holistic approach is known as a *full STGT study* (HODA; NOBLE; MARSHALL, 2012). The STGT method is an adaptation of three traditional Grounded Theory methods: *Glaserian*, *Strauss-Corbinian*, and *Constructivist*. Figure 3.2 presents the socio-technical framework proposed by Hoda (2021).

Basic stage. The basic stage consists in performing lean literature, study preparation and piloting, and basic Data Collection and Analysis (DCA) interactions. The basic

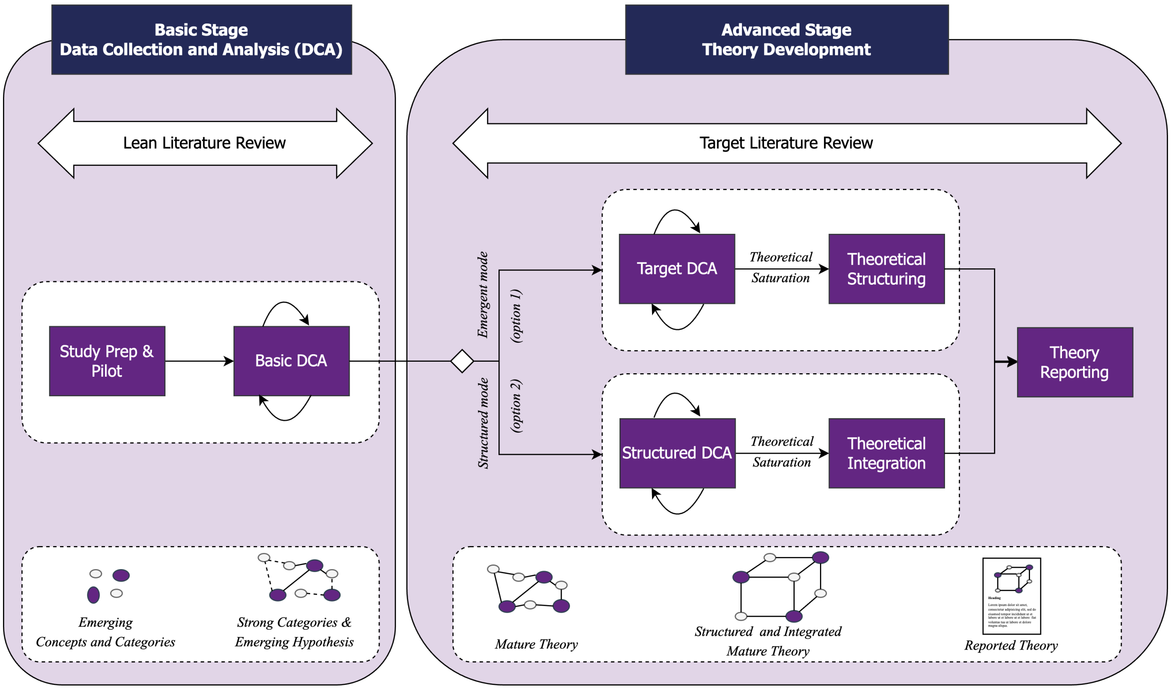


Figure 3.2 Socio-Thecnical Grounded Theory (HODA, 2021)

data analysis includes *open coding* and *constant comparison*. Initial concepts and categories are expected to emerge yet in the piloting. In addition, the strongest categories and preliminary hypotheses will emerge with each data collection and analysis cycle. This fact marks the end of the basic cycle.

Advanced stage. The advanced stage consists of the development (generation or creation) of a structured and integrated mature theory, which might involve a target literature review and a mode of theory development:

Emergent mode. Enables the emergence of theory through the iterative *target data collection and analysis*, which ends with *theoretical saturation* and results in a mature theory. The theory can be further structured using a theoretical presentation template during *theoretical structuring*.

Structured mode. Enables a structured development of theory through *structured data collection and analysis* ending with *theoretical saturation* and resulting in a mature theory that is structured and can be further integrated through *theoretical integration*.

3.6 CHAPTER CONCLUSION

Grounded theory is a qualitative research method crucial in understanding and explaining complex phenomena. Throughout this chapter, we have explored the application of grounded theory in software engineering research, providing insights into the skills and concepts necessary for its effective utilization.

We commenced our discussion by exploring qualitative data analysis, emphasizing that grounded theory is a robust approach for qualitatively analyzing data obtained through interviews, observations, or focus groups. To illustrate the method's practicality in SE, we presented several works that leveraged grounded theory to investigate various software-related phenomena.

Furthermore, we delved into the essential skills required for researchers or research teams to employ grounded theory successfully. These skills are foundational for conducting comprehensive qualitative analysis using the method. We also introduced fundamental concepts crucial for understanding the underlying principles of grounded theory, thus enhancing its applicability.

While grounded theory offers numerous advantages in research, we addressed its limitations and potential challenges that researchers may encounter during its application. Awareness of these constraints is vital for achieving meaningful and accurate results when using grounded theory.

In this thesis, we have employed grounded theory procedures to analyze our data, providing us with a systematic and rigorous framework to explore the complex world of software startups. This chapter lays the foundation for the empirical investigation conducted in subsequent chapters, shedding light on the methodological framework and tools used throughout this research.

Chapter 4

RESEARCH METHODOLOGY

The methodological considerations of this thesis rely on qualitative research methods. This chapter provides an overview and detailed explanation of the research methodology, the research questions addressed, and the data collection and analysis procedures.

4.1 RESEARCH CHARACTERIZATION

The research method adopted for this work encompasses the selected research strategies, aiming to achieve the study's general objective.

- **Nature:** Regarding its nature, this work is applied research, which aims to generate knowledge for practical application in solving specific problems.

- **Problem Approach:** This work is classified as qualitative research in approaching the problem. Qualitative research is appropriate for "discovering and understanding what is behind phenomena about which little is known or to obtain new perspectives on topics about which much is already known" (CORBIN; STRAUSS, 2014). In this type of research, data are analyzed inductively, with the process and its meaning being the primary focus.

- **Goal:** Considering the goal, this research is characterized as exploratory research as it aims to provide greater familiarity with the problem to make it explicit. It involves interviews with individuals with practical experience with the research problem and analyzing examples to encourage understanding. This approach aims to make the problem explicit and facilitate the formulation of hypotheses. It is also descriptive, which seeks to characterize the problem and identify potential relationships between variables.

- **Technical Procedures:** Finally, regarding technical procedures, a case study approach was chosen as it is suitable for answering questions like "how." Additionally, the researcher investigates a contemporary phenomenon and does not have control over the phenomena under investigation (YIN, 2017).

The methodological choice for this work aligns with the classification proposed by Silva e Menezes (2005), as depicted in Figure 4.1, which considers four aspects: (i) the nature of the research; (ii) the approach to the problem; (iii) the research goals; and (iv) the technical procedures.

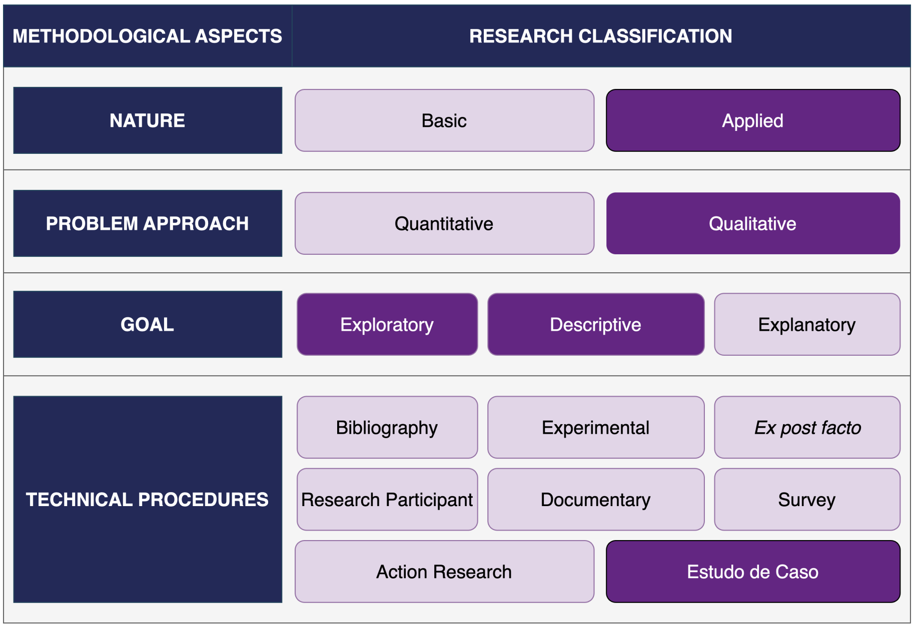


Figure 4.1 Methodological Representation adapted from Silva e Menezes (2005)

4.1.1 Research Method

In this study, we have adopted the Case Study research method, following the guidelines proposed by Yin (2017). The research is organized into three distinct phases, as illustrated in Figure 4.2:

1. The research planning phase encompasses the selection of the research topic and defining and delimiting the research problem.

2. The research conducting phase involves the execution of the research plan.

3. The research synthesizing and analyzing phase is focused on analyzing the data and information gathered during the research conducting phase.

The research planning phase played a critical role, as the way the researcher formulated it could have implications for the subsequent stages. The research planning phase encompassed the following stages:

1. Conducting a thorough literature review to establish the context and motivation of the study.

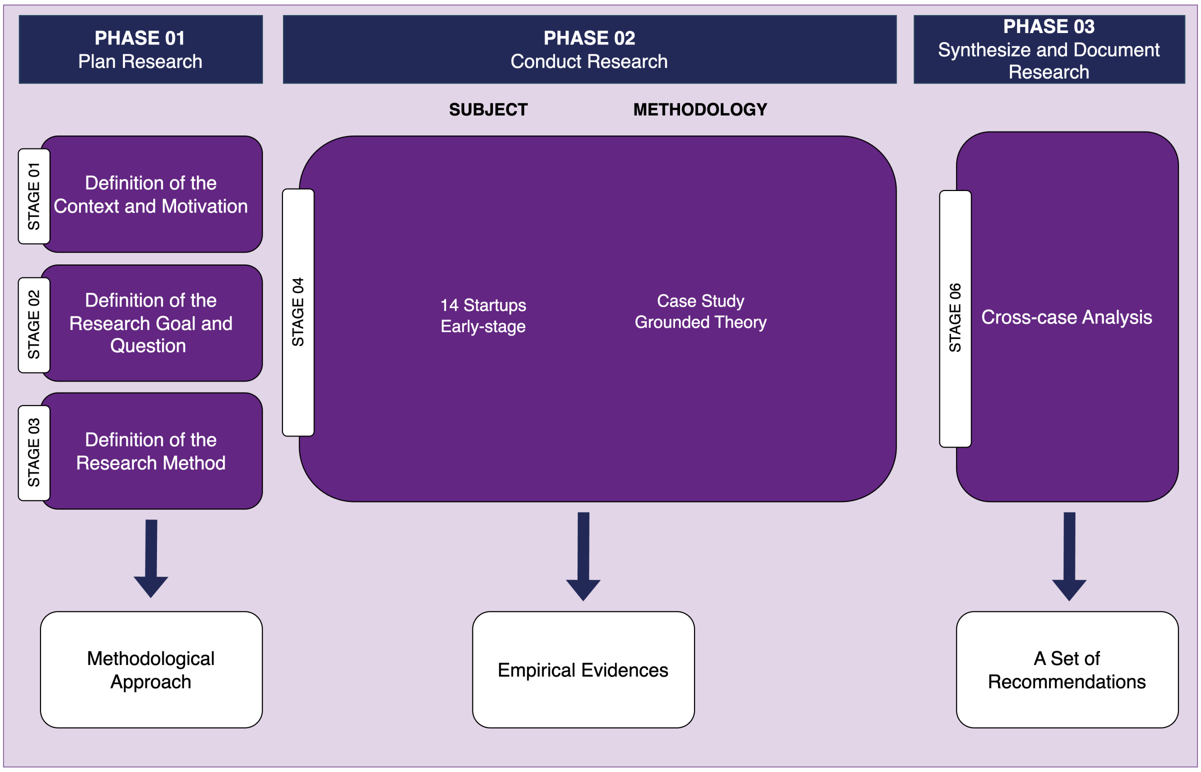


Figure 4.2 Research Approach.

2. Defining clear objectives and research questions to guide the investigation.

3. Selecting the most appropriate research method to address the research questions effectively.

4. Designing a robust data collection protocol to facilitate gathering relevant information.

5. Carefully selecting the cases examined in the study.

The subsequent phase is the research conduction phase, where the case study involves meticulous preparation, data collection, and preliminary analysis. It includes the execution of the case study itself, analyzing each case, and documenting the findings in separate case reports. Moreover, an interview with startups in the growth stage was conducted to validate the findings. As a result of this phase, we obtained a set of recommendations tailored explicitly for early-stage startups.

Finally, the last phase focuses on synthesizing the analysis and drawing conclusions from the field study, including arriving at cross-case conclusions and updating the recommendations based on the overall findings.

4.1.2 Research Objective and Questions

Some areas of software engineering still need to mature practices (JACOBSON et al., 2019). The Lack of a solid and widely accepted theoretical base can lead to methods and practices based more on fads than concrete evidence. Furthermore, the proliferation of methods and method variants can make it difficult for practitioners to choose the most suitable method for a given task. A lack of credible experimental evaluation and validation can also lead to ill-informed decisions about software engineering practices. The divide between industrial practice and academic research can make applying research results in production environments difficult. To meet these challenges, it is essential that software engineering practices are evidence-based and that there is close collaboration between academic researchers and industry professionals.

The general objective of this doctoral research was to advance the understanding of how software startups employ software engineering practices, which practices influence their behavior, what are the most relevant quality attributes at an early stage, and tools that support their development activities. The main research question of this work, also presented in Chapter 1, is:

RQ1: ***How do early-stage startups develop software?***

As this research question is vast, we structured more specific research subquestions, associating the exploration of software engineering in startups and their contributions. The following are the research questions (RQ) to conduct this investigation:

RQ1.1: What is the contextual framework for software development in startups?

- ***Rationale:*** Startups exhibit unique characteristics, including resource constraints and dynamic technological environments. Existing characterizations of startups in research, while informative, may not suffice for making informed comparisons of software engineering contexts. A deeper understanding of the software engineering context within startups is crucial for guiding or recommending engineering practices that align with these unique contexts.

RQ1.2: Which software engineering practices are most pertinent to early-stage startups?

- ***Rationale:*** This research question delves into identifying the specific software engineering practices that hold relevance for early-stage startups, providing insights into their practical application.

RQ1.3: What quality attributes do startups prioritize in software development?

- ***Rationale:*** Identifying the software quality attributes that startups emphasize is essential, as these attributes significantly influence the distinctive characteristics of software products developed by startups.

RQ1.4: Which software development supporting tools are commonly used by startups?

- ***Rationale:*** Due to constraints such as limited time, resources, or specialized knowledge, startups often struggle to identify the most suitable tools for their needs and may not fully harness their potential. This research question aims to furnish empirical evidence regarding the tools commonly employed to support software development in startup environments.

4.1.3 Propositions

Propositions are intended to guide the researcher in examining what is effective within the scope of the research (YIN, 2017). Typically, propositions are derived from literature, theories, or generalizations based on empirical data (BAXTER; JACK et al., 2008).

In this research, theoretical propositions were formulated based on the results of the previous research carried out in (SOUZA et al., 2019a), which were based on the most relevant categories that form the model of software startups in incubators: humans (the team is a catalyst of development, initial growth hinders performance, lack of resources), technical (accumulated technical debt, speed-up development, evolutionary approach, UX has high priority), and organizational (innovation-driven development, customer-driven development).

Here they are:

- Proposition P1: There are human factors that influence software development in startups.

- Proposition P2: There are technical factors that influence software development in startups.

- Proposition P3: There are organizational factors that influence software development in startups.

4.2 RESEARCH PROTOCOL

We conducted a multiple case study, following the guidelines and recommendations provided by Brereton et al. (2008). Fourteen startups were interviewed, and the units of analysis were divided into subgroups to highlight the business domain diversity of the interviewed startups, as shown in Figure 4.3. The primary audience for this case study was Software Engineering researchers and startup practitioners.

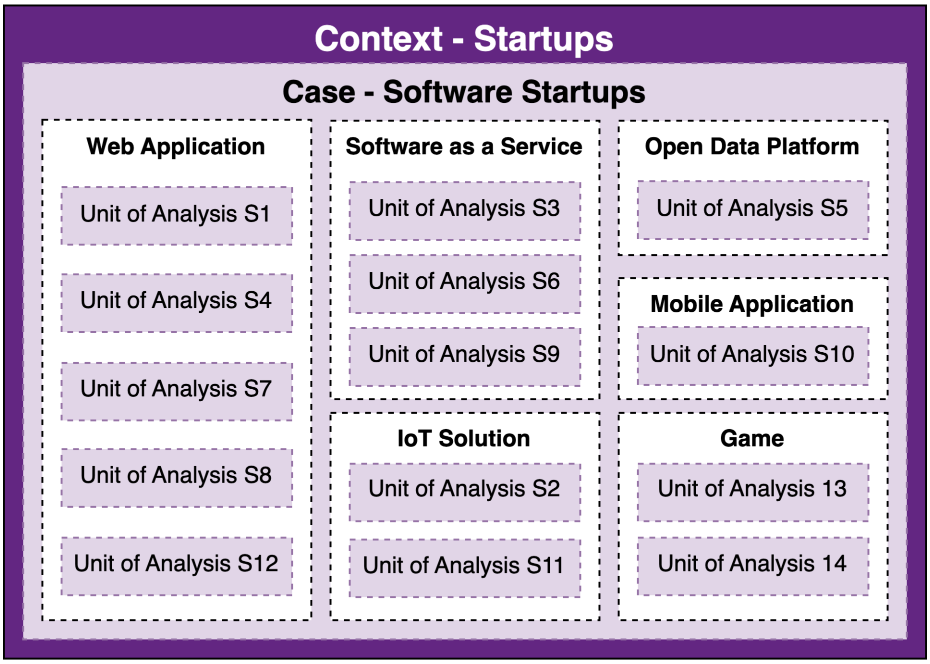


Figure 4.3 Case Study Design.

4.2.1 Operational Procedures

We incorporated a formal case study protocol while maintaining the critical characteristics of the software startup sample: early-stage, ecosystem-oriented, and with growth ambitions. Our case study protocol adheres to the guidelines and recommendations put forth by Brereton et al. (2008). Following this protocol, we diligently carried out the *data collection* and *data analysis* processes. To ensure comprehensive data gathering, we made necessary adaptations to the interview procedures (Appendix B) and questionnaire, creating various supporting documents, including templates, checklists, interview cards, and data collection forms. As part of our ethical considerations, we introduced a consent form signed by practitioners and researchers involved in the interviews. This form explicitly assured the confidentiality of company-specific information and any details unrelated to the study (Appendix A). To ensure accuracy and completeness, we recorded and transcribed all sections of the interviews.

The operational procedures of the research were defined and structured in three phases: research planning, conducting the research, data collection, and data analysis. In the research planning phase, we built the case study protocol, adapted the Giardino et al. (2016) questionnaire to our reality, and adjusted the interview process, which can be found in Appendix B. In the first phase of conducting the research, collecting data from the startups that responded positively to the interview invitation, we scheduled and carried out the interviews. Furthermore, in the second phase of conducting the research, we conducted the transcripts of the interviews, analyzed them using grounded theory, and applied open, axial, and selective coding to build the startup software development

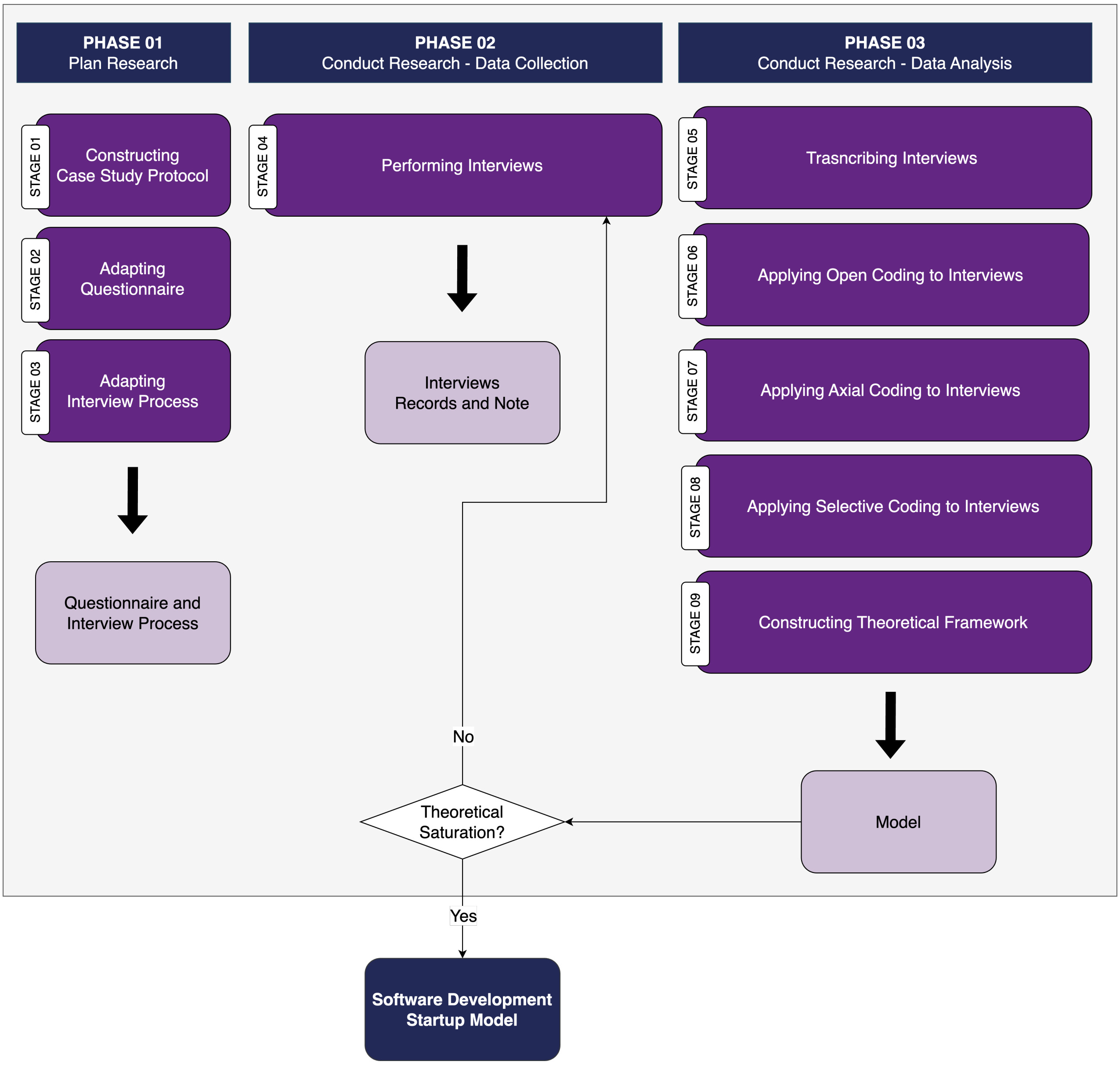


Figure 4.4 Research Operational Procedures.

model. We built a preliminary version with two interviews and followed the data collection and analysis flow until we reached saturation when we no longer found new codes. Upon completing the interview process, we conducted an individual analysis for each case study, which will serve as the foundation for the consolidated analysis of all cases to be presented in Chapter 5.

4.2.2 Case Selection

This section presents how we selected the startup cases. The criteria for selecting the cases were the following:

1. The startup must be a Software Startup.

2. We want to observe the software engineering practices.

3. We choose Academic Incubators' resident startups.

We looked for startups from incubators: Federal University of Bahia (UFBA), Salvador University (UNIFACS), and Technological Park of Bahia. From these websites, we extracted all the startups listed. Then, we created a list of startup candidates. We collected general information (product, size, location, description, founding year) and, whenever possible, the contact information of their founder's CTO.

The main criteria for selecting the startup were: it must be a software startup located in an academic incubator. Figure 4.5 shows the case selection process.

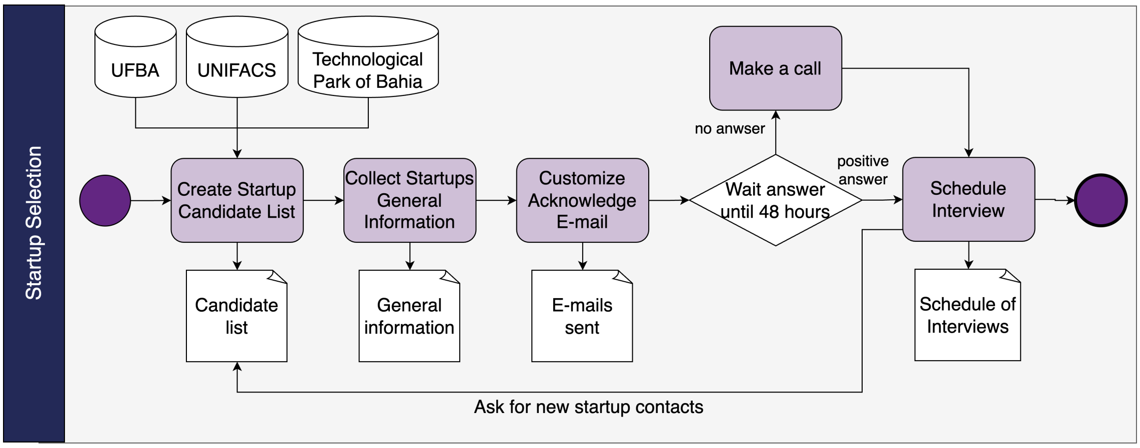


Figure 4.5 Case Selection adapted from Giardino et al. (2016)

Next, we packaged the data into a spreadsheet to keep track of the interviews' status throughout the process. We defined a template email to introduce our research project and show the startup our interest in interviewing them about their Software Engineering (SE) practices. After customizing the template email for each company, making it sound more regional and personal, we sent out the first batch of emails and waited 48 hours for answers. After that, we called those startups who still needed to answer our message, asking if they were interested in scheduling an interview. We also asked for new startup companies' contacts. In case further contact is provided, we updated the Candidates' List.

These choices were made because the researchers used the case study as a preliminary study to calibrate, align, and better understand the process, including timing, tasks to be executed, and documents to be delivered. We took these steps to help ensure the quality and compliance of the case study process. We follow the next steps to perform the case study:

1. *Planning case study* – The case study planning is done by preparing this case study protocol.

2. *Preparing case study* – The preparation phase consists of writing the protocol and all the artifacts needed for the interview and the data collection.

3. *Conducting case study* – The case study conduction consists of performing the data collection and analyses.

4. *Reporting case study* – In this phase, we report the study and its findings.

We interviewed 14 (fourteen) software startups. The units of analysis were divided into subgroups to highlight the diversity of business domains of the interviewed startups. The primary audience of this case study was software engineering researchers and professionals or teams interested in creating a software startup.

4.2.3 Units of Analysis

The concept of the unit of analysis is central to addressing the fundamental issue of defining a "case." In a classic case study, a “case” can refer to an individual, an event, or an entity that is less precisely defined than an individual. The determination of the unit of analysis, and consequently the case, is generally influenced by the specific formulation of the initial research questions (YIN, 2017). Considering the questions of this research, the following units of analysis were defined:

i. It is a startup that has a software solution as a product or service;

ii. It is an early-stage startup with the intention of growing.

Fourteen (14) organizations with different profiles participated in the research (Table 4.1) to observe the context of software development in startups. All startups were at an early stage. The interviews were carried out between August 2016 and December 2017. Regarding the business domain, two (2) companies are marketplaces - S1 and S4, which provide online platforms where buyers and sellers can connect and conduct business transactions. These marketplaces make it easy to sell products or services from different vendors in one place. One (1) company provides IoT Solutions, S2, which creates products and services that connect to smart devices and allow communication and data exchange between them. Three (3) companies are Software-as-a-Service (SaaS) - S3, S6, and S9; offer software-based services, where applications are hosted on servers and made available to customers over the internet, are turnkey software solutions, usually through a subscription or pay-as-you-go model. One (1) company is Open Data Platform - S5, which offers platforms for open data sharing and availability, allowing organizations or governments to publish and share data in an accessible and transparent way, encouraging collaboration and use of this information. Three (3) companies, S7, S8, and S12, are Web Applications. They develop web-based applications accessible through internet browsers for various entertainment purposes, with functionalities and resources running online. Only one (1) company, S10, is a company that implements mobile applications for devices such as smartphones and tablets for mobile operating systems such as Android and iOS, offering a variety of features and services to users. Only one (1) company, S11, works with

Table 4.1 Characteristics of the Startups Cases and Interviews.

Case	Business Domain	Segment	Size	NIP	IR	ITS
S1	Web Application	Agrotech	Small	2	CTO/Developer	00:58:58
S2	IoT Solution	Agrotech	Small	2	CTO/Developer	00:59:00
S3	SaaS	-	Small	2	CEO/CTO	01:12:39
S4	Web Application	Edutech	Micro	1	CEO	00:51:48
S5	Open Data Platform	Lawtech	Small	2	CEO/CTO	00:50:05
S6	SaaS	-	Small	2	CEO/CTO	01:10:10
S7	Web Application	Smart Cities	Micro	1	CEO/CTO	00:51:27
S8	Web Application	Govtech	Micro	1	CTO/Developer	00:55:23
S9	SaaS	-	Small	1	CEO/CTO	00:45:05
S10	Mobile Application	Healthtech	Small	2	CEO/CTO	00:44:25
S11	IoT Solution	HRtech	Micro	1	CTO	00:47:00
S12	Web Application	Funtech	Micro	1	CTO	00:45:39
S13	Game Development	Funtech	Micro	1	CEO/CTO	00:57:14
S14	Game Development	Funtech	Micro	1	CEO/CTO	00:36:28

NIP: Number of interviewed people
IR: Interviewees roles
ITS: Interview Time Spent

embedded applications that develop applications embedded in electronic devices, such as embedded systems in industrial equipment, medical devices, and automobiles. Two (2) companies, S13 and S14, are specialist game development companies that design, code, and create content for games ranging from console and PC games to mobile games.

Let us consider the company's segment, whether it is a business area or a specific sector. We have one (1) Agrotech, one (1) Edutech, three (3) Funtech, one (1) Govtech, one (1) Health, one (1) HRtech, one (1) Lawtech, one (1) Segment, and one (1) Smart Cities. Three companies work with SaaS, so the follow-up depends on the segment of their customers. Based on the classification provided by IBGE, as discussed in Section 2.5, we have identified that out of the total number of companies, four (4) fall under the category of micro companies, while seven (7) fall under the category of small companies. The invitation extended to startups specifically requested an interview with the technical overseeing the software development area or the Chief Technology Officer (CTO). Throughout the research process, we had the opportunity to conduct interviews with 19 people. In certain instances, both the CEO and the CTO were present, with at least one of them actively involved in software development activities in the startup. The column labeled "IST" provides the duration of each interview, showcasing the time allocated for each session, which, on average, amounted to 53 minutes. The script is organized into four parts:

i. characterization of the interviewee's organizational profile, professional experience, and academic background;

ii. interview about software engineering practices, quality attributes, and tools used by startups;

ii. non-participatory observation of the work environment; and,

iv. archival data.

The script questions supported the researcher during the semi-structured interview, acting as a checklist. Nevertheless, it is worth mentioning that although the semi-structured interview has predefined questions, it does not mean that the questions are immobilized. In this kind of interview, the researcher is free to add or adapt some questions according to the conversation that occurred during the field interviews and allowed greater flexibility to deepen or confirm certain information presented, as well as to identify new information and refute others. Ultimately, each interviewee was asked if there was anything to add. The primary source of data for data collection were interviews, non-participatory observation, and archival data. Data from observation, documents, and tool checks were essential sources for triangulation and a broad understanding of what each participant was trying to convey in the interviews to gain additional perspectives on critical issues. Field notes comprise records of the researcher's impression of the interviewee's behavior (motivation, lack of interest) or the organizational structure.

4.2.4 Data Collection

Case study evidence can come from various sources, such as documentation, archival records, interviews, direct observations, participant observation, and physical artifacts (YIN, 2017). We design and apply data collection in a way that allows data triangulation, integrating multiple data sources:

1. Questionnaires;
2. Semi-structured interviews;
3. Non-participant direct observation with field notes;
4. Written and electronic documentation;
5. Verification of software development support tools.

Table 4.1 presented, in addition to the characterization of the startups, the profile of the participants, and the time spent carrying out the interviews by the organization. There were 14 interviews with 19 people. An invitation email, as described in Section 4.2.2, was sent to the participants to carry out the interviews. The process is written in detail in this section. At the beginning of the interview, a consent letter was read and signed, and only by mutual agreement was the interview carried out explaining the research objectives and the confidentiality term signed by the researchers involved. We interviewed the CTOs, as this profile would represent the team members of a startup. The interviews were conducted with each participant individually and lasted an average of 40 to 60 minutes (approximately), with an average of 53 minutes. Before starting the interviews, the objective of the interview was again explained to each interviewee, and confidentiality was reinforced so that the participant felt at ease and understood that he was not being evaluated. All interviews were conducted in the startup's work environment. Data collection includes a questionnaire with the CTO; a semi-structured interview following the (RUNESON et al., 2012) funnel model, where we started with open questions that allowed the subjects to offer more information and then objective questions. We also audio-recorded the interview sessions, and each researcher played a different role in the interview. Two researchers participated in the interviews. One researcher was responsible for conducting the interview, and the other observed and recorded field notes. We conducted a part-time non-participatory observation at startups and verified respondents' information by checking artifacts and software development tools for startups.

Subsequently, the recordings were transcribed (about XXX single-spaced pages) and analyzed with the support of Atlas.ti tool. The script for the semi-structured interview is available in Appendix C.

4.2.5 Data Analysis

Yin (2017) guides researchers in defining the logic that connects the data to the study's propositions and the criteria for interpreting the results. In this research, we adopted the model proposed by Reinehr e Pessôa (2008), which defines Analysis Points (AP) as supporting concepts based on the literature review to assess whether a proposition is confirmed or not in order to answer the main research question. Albuquerque (2021) used the Reinehr e Pessôa (2008) method to investigate the abandonment and continuity of software process improvement programs based on implementing maturity models after official evaluation. Table 4.2 displays the defined research propositions and their corresponding analysis points. As previously explained, the propositions assist in fulfilling the research objective.

Table 4.2 List of Research Propositions (P) and Points of Analysis (PA).

#	Description
Proposition P1	There are human factors that influence software development in startups
PA.01	Team is the catalyst of development
PA.02	Initial growth hinders performance
PA.03	Lack of resource
Proposition P2	There are technical factors that influence software development in startups
PA.04	Accumulated technical debt
PA.05	Speed-up development
PA.06	Evolutionary approach
Proposition P3	There are organizational factors that influence software development in startups
PA.07	Innovation-driven development
PA.08	Customer-driven development
PA.09	Quality attributes

The analysis points serve as evidence sought in the interviews to refute or confirm a proposition. The theoretical foundation for constructing these research elements (propositions and analysis points) was the literature review presented in Chapter 2. The analyses were conducted after the completion of all interviews. For the qualitative analysis of the interview data, we employed the open, axial, and selective coding procedures of Grounded Theory (STRAUSS; CORBIN, 1990).

1. Labels assigned to raw data and carried out a first low-level conceptualization;
2. Concepts grouped into theoretical categories and subcategories and their relations linked together according to cause-effect relationships;
3. Categories refined to create different levels of abstraction and adjust concepts;
4. The consistency among categories were validated by selective coding, exploring, and analyzing links among subcategories; and,
5. The core categories were identified by analyzing causal relations between high-level categories.

This approach offers a systematic analysis that enhances academic rigor and provides validity in terms of traceability, from the initial data coding to the final analysis outcome (O'CONNOR, 2012). This research does not aim to create a theory using the interactive process of conducting interviews and then analyzing the data to guide subsequent interviews, as advised in Strauss and Corbin's book (STRAUSS; CORBIN, 1990). Therefore, theoretical saturation still needs to be pursued or achieved. The transcription was read multiple times and analyzed using Atlas.ti software. The author then conducted open coding, which involved a microanalysis of the interviews. Each transcription was analyzed line by line, and new codes were created and merged with existing codes as appropriate whenever new evidence emerged. We formalized the GT coding process in a theoretical model. We developed the model into two levels: a detailed level representing the network of subcategories; and a high-level representation of the main categories network. Memos were created to support the analysis considering field notes. Subsequently, the codes were grouped based on their properties, forming concepts representing categories. Finally, the categories and subcategories were interconnected in the axial coding stage.

4.2.5.1 Analysis Points

Yin (2017) highlights the importance of defining a set of questions to guide data collection and analysis. In this research, we defined these questions as "Analysis Points." The guidelines of (REINEHR; PESSÔA, 2008) were followed, representing critical points intending to incrementally cover the questions, the study propositions, and the analysis units providing the data analysis criteria. The objective of the points of analysis is to consolidate all the subjects that a particular point can contemplate so that an important subject is remembered during the interview, forcing a new intervention with the organization. In addition, analysis points help researchers analyze research propositions. The propositions were defined based on the categories that explain startup software development by (SOUZA et al., 2019a). They are Human, Technical, and Organizational. The factors related to each category were used to determine the analysis points, as shown below in the tables.

Table 4.3, Table 4.4, and Table 4.5 present the supporting concepts and bibliographical references for each point of analysis referring to proposition P1 on human factors.

Table 4.6, Table 4.7, and Table 4.8 present the supporting concepts and the bibliographical references for each point of analysis referring to the P2 proposition on Technology factors.

Table 4.9, Table 4.10, and Table 4.11 present the supporting concepts and the bibliographical references for each point of analysis referring to the P3 proposition on Organizational factors.

4.2.5.2 Analysis Procedures

The propositions help in fulfilling the objective of the research. The proven analysis points are sought in interviews to refute or confirm a proposition. A theoretical basis for constructing these research elements (propositions and points of analysis) were the most relevant related studies in the area presented in Chapter 2. These supporting concepts were further detailed in the research protocol (section 4.2.5.1).

Analyzes were performed after the completion of all interviews. For the qualitative analysis of the interview data, the Grounded Theory (STRAUSS; CORBIN, 1990) open and axial coding procedures were used, as it is a systematic analysis approach, which adds value in terms of academic rigor, providing validity in terms of traceability from the reduction of the initial data to the final result of the analysis (O'CONNOR, 2012). This research does not aim to create a theory using the interactive interview process and then analyze the data to guide the subsequent interviews, as oriented in (STRAUSS; CORBIN, 1990)'s book. Therefore, theoretical saturation was not sought or achieved.

The transcript was read (more than once by the author) and kept using the Atlas.ti software. So, the author carried out open coding, which is the micro-analysis of the interviews. Each transcript was observed line by line and merged codes created with existing codes, as appropriate, when new supporting data appeared. Memos were created to support the analysis (also considering the field notes generated in the observations). Then, the codes were grouped according to their properties, forming concepts representing categories. Finally, the categories and subcategories were related in the axial coding stage.

Figure 4.6 shows an example of analysis points identified in the interview excerpts. For example, when the researcher asked: "[Q2.1] How was the initial team composed? And now?" some responses from the participants were: "(...) first there were three, then five people" and "6 people with the CEO," referring to the number of team members. The generated code was "Small and co-location development team." The photos taken from the startups were encoded with the code "Co-location development team," which is proof of code "[CAT.08] Small and co-location development team," which, in turn, makes part of the point of analysis "[PA.01] The team is the catalyst for development."

Later in the reduction process, the previously mentioned analysis point was related

Table 4.3 Theoretical basis for the analysis point Team is the catalyst of development – PA.01.

Proposition P1 There are human factors that influence software development in startups	
PA.01 Team is catalyst of development	
Support Concepts	**Authors**
Access to external expertise (Mentorship)	(BOSCH et al., 2013) (MELEGATI et al., 2019a) (SOUZA et al., 2019a) (KEMELL et al., 2020) (POMPERMAIER, 2021)
High-impact of Founder/CEO background	(GIARDINO et al., 2016) (GRALHA et al., 2018) (SOUZA et al., 2019a) (MELEGATI; GUERRA; WANG, 2022)
Multi-role and full-stack engineers	(GIARDINO et al., 2016) (GRALHA et al., 2018) (SOUZA et al., 2019a)
Need a high-performance team	(GIARDINO et al., 2016) (SOUZA et al., 2019a) (KLOTINS et al., 2019) (KEMELL et al., 2020)
Productivity oriented	(SOUZA et al., 2019a) (KLOTINS et al., 2019)
Self-managed team	(GIARDINO et al., 2016) (SOUZA et al., 2019a) (KLOTINS et al., 2019)
Senior developers influence the development	(GIARDINO et al., 2016) (SOUZA et al., 2019a) (MELEGATI et al., 2019a)
Small and co-located development team	(GRALHA et al., 2018) (GIARDINO et al., 2016) (SOUZA et al., 2019a) (MELEGATI et al., 2019a) (KLOTINS et al., 2019) (KEMELL et al., 2020)
Tacit knowledge	(GIARDINO et al., 2016) (SOUZA et al., 2019a) (MELEGATI et al., 2019a) (KLOTINS et al., 2019)
Team under pressure	(GIARDINO et al., 2016) (SOUZA et al., 2019a) (GRALHA et al., 2018) (MELEGATI et al., 2019a)

to “Proposition P4. Processes.” Appendix C presents the remaining analysis networks. Figure 4.6 has colors and codes that differentiate range deviations. In open coding, in this example, the types of finding codes were identified in blue, varying depending on the code

Table 4.4 Theoretical basis for the analysis point Initial growth hinders performance – PA.02.

Proposition P1 There are human factors that influence software development in startups	
PA.02 Initial growth hinders performance	
Support Concepts	**Authors**
Focus change to business concerns	(GIARDINO et al., 2016)
Re-engineering/Refactoring code	(GIARDINO et al., 2016) (KLOTINS et al., 2019)
Payment of accumulated technical debt	(GIARDINO et al., 2016) (GRALHA et al., 2018) (KLOTINS et al., 2019)
Better quality product	(GIARDINO et al., 2016)

Table 4.5 Theoretical basis for the analysis point Lack of resources – PA.03.

Proposition P1 There are human factors that influence software development in startups	
PA.03 Lack of resource	
Support Concepts	**Authors**
Lack of expertise	(GIARDINO et al., 2016) (KLOTINS et al., 2019)
Lack of finance resources	(SOUZA et al., 2019a) (MELEGATI et al., 2019a)
Lack of human resources	(GIARDINO et al., 2016) (MELEGATI et al., 2019a)
Time shortage	(GIARDINO et al., 2016) (SOUZA et al., 2019a) (GRALHA et al., 2018) (MELEGATI et al., 2019a)

groupings. The axial coding grouped these codes into categories called [CAT.XX] and a number identified in lilac. Subsequently, these factors that influence startup software development were grouped in the category called points of analysis (PA), also in lilac color. The color pattern was used during the analysis to facilitate the visual analysis of the networks.

4.3 CHAPTER CONCLUSION

This chapter provides an overview of the research classification, the rationale behind selecting the research method, and the structure of the study, which follows a multiple-case study approach. It also delves into the specifics of defining and implementing each process step. Detailed information is provided regarding the data collection procedures and analysis conducted for each step. The subsequent chapter will present an in-depth analysis of the findings concerning the software development landscape in startups.

Table 4.6 Theoretical basis for the analysis point Accumulated technical debt – PA.04.

Proposition P2 There are technical factors that influence software development in startups	
PA.04 Accumulated technical debt	
Support Concepts	**Authors**
Documentation debt	(BOSCH et al., 2013) (GIARDINO et al., 2016) (GRALHA et al., 2018) (SOUZA et al., 2019a) (KLOTINS et al., 2019) (MELEGATI et al., 2019a)
Requirements debt	(GIARDINO et al., 2016)
Process debt	(GIARDINO et al., 2016) (MELEGATI et al., 2019a)
Architecture debt	(KLOTINS et al., 2019)
Test debt	(GIARDINO et al., 2016) (KLOTINS et al., 2019)
	(GRALHA et al., 2018)

Table 4.7 Theoretical basis for the analysis point Speed-up development – PA.05.

Proposition P2 There are technical factors that influence software development in startups	
PA.05 Speed-up development	
Support Concepts	**Authors**
Accelerate the product launch to reach the time-to-market	(GIARDINO et al., 2016)
Delegate complexity to third-party applications	(GIARDINO et al., 2016) (GRALHA et al., 2018) (SOUZA et al., 2019a) (KLOTINS et al., 2019)
Simple and informal Workflow	(GIARDINO et al., 2016) (SOUZA et al., 2019a) (GRALHA et al., 2018) (MELEGATI et al., 2019a)
Use of standard/known technology	(KEMELL et al., 2020) (KLOTINS et al., 2019)
Use of well-integrated and simple tools	(GIARDINO et al., 2016) (SOUZA et al., 2019a) (GRALHA et al., 2018) (MELEGATI et al., 2019a)
Work overtime to meet deadlines	(GIARDINO et al., 2016) (SOUZA et al., 2019a)

Table 4.8 Theoretical basis for the analysis point Evolutionary approach – PA.06.

Proposition P2 There are technical factors that influence software development in startups	
PA.06 Evolutionary approach	
Support Concepts	**Authors**
Find the product/market fit quickly	(GIARDINO et al., 2016) (KLOTINS et al., 2019) (GRALHA et al., 2018)
Product efficiency starts after launch	(KLOTINS et al., 2019) (MELEGATI; GUERRA; WANG, 2022)
Uncertain conditions make long-term planning non-viable	(KLOTINS et al., 2019) (GRALHA et al., 2018)

Table 4.9 Theoretical basis for the analysis point Innovation-driven development – PA.07.

Proposition P3 There are organizational factors that influence software development in startups	
PA.07 Innovation-driven development	
Support Concepts	**Authors**
Search for innovation opportunities	(SOUZA et al., 2019a) (GIARDINO et al., 2016) (GRALHA et al., 2018) (MELEGATI et al., 2019a)
Product innovation	(SOUZA et al., 2019a) (GRALHA et al., 2018) (MELEGATI et al., 2019a)
Business innovation	(BOSCH et al., 2013) (MELEGATI et al., 2019a)

Table 4.10 Theoretical basis for the analysis point Customer-driven development – PA.08.

Proposition P3 There are organizational factors that influence software development in startups	
PA.08 Customer-driven development	
Support Concepts	**Authors**
Customer inclusion in development since conception	(BOSCH et al., 2013) (GIARDINO et al., 2016) (GRALHA et al., 2018) (SOUZA et al., 2019a) (KLOTINS et al., 2019) (MELEGATI et al., 2019a) (KEMELL et al., 2020)
Customer requests are documented	(SOUZA et al., 2019a)
Meet customers needs	(BOSCH et al., 2013) (GIARDINO et al., 2016) (GRALHA et al., 2018) (SOUZA et al., 2019a) (KLOTINS et al., 2019) (MELEGATI et al., 2019a) (KEMELL et al., 2020)
Prototype validated by costumers	(BOSCH et al., 2013) (GIARDINO et al., 2016) (GRALHA et al., 2018) (SOUZA et al., 2019a) (KLOTINS et al., 2019) (KEMELL et al., 2020)

Table 4.11 Theoretical basis for the analysis point Quality attributes – PA.09.

Proposition P3 There are organizational factors that influence software development in startups	
PA.09 Quality attributes	
Support Concepts	**Authors**
User experience	(GIARDINO et al., 2016) (GRALHA et al., 2018) (SOUZA et al., 2019a) (KLOTINS et al., 2019)
Availability	(GRALHA et al., 2018) (SOUZA et al., 2019a) (SOUZA et al., 2019a)
Fault tolerance	(GRALHA et al., 2018) (SOUZA et al., 2019a)
Scalability	(GRALHA et al., 2018) (SOUZA et al., 2019a)

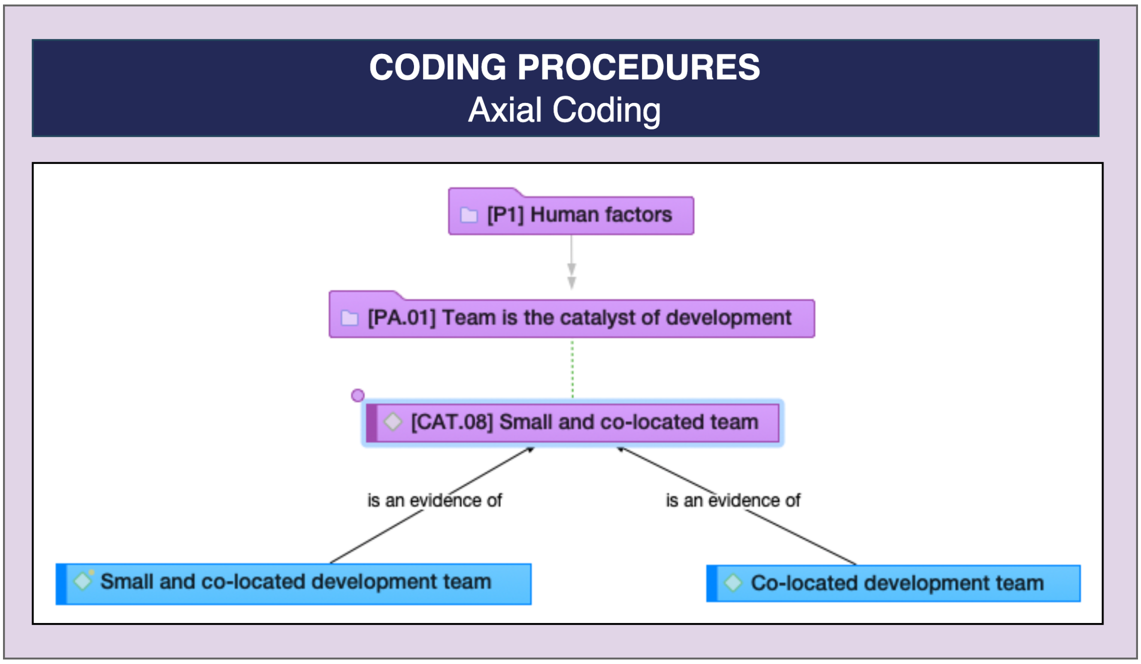


Figure 4.6 Example Analysis - Axial Coding.

Chapter 5

STARTUP SOFTWARE ENGINEERING PRACTICES

This chapter individually presents the results of each case, software startup presented in Chapter 4 in Section 4.2.3, and the cross-analysis of the cases studied from the research propositions supported by the concepts entered in the literature.

5.1 CASE STUDY

This chapter analyzes data from the fourteen software startups participating in this case study. Then, the next section will present the description of each case study unit of analysis related to the points of analysis of the Session 4.2.5.1. New findings were tagged with NPA code followed by numbering and participant quotes.

5.1.1 Case S1

Startup S1 was founded in 2013. It is a company classified as small, with ten employees. We interviewed the CTO and a software developer from S1 in October 2016, which was in the early stages. The interview lasted 58 minutes and 58 seconds. S1 develops a web application for the Agrotech segment. Currently, S1 has merged with another company entering the growth stage. Table 5.1 presents S1 characterization.

The team is a catalyst for development (PA.01). It is a team that has access to external expertise (CAT1.1) and is heavily influenced by the founders' knowledge (CAT1.2). The team has a senior developer (CAT1.7) who guides and guides the team despite already being a high-performance (CAT1.4), productivity-oriented (CAT1.5), and self-managed (CAT1.6) team. The team is small, with ten members (CAT1.8). Moreover, he works in a single room, on an extensive rectangular table, where the team places the computers facing each other (CAT1.8). This organization favors agility in the execution of tasks and the tacit transmission of knowledge (CAT1.9). The team has short delivery cycles and is constantly under pressure (CAT1.10).

> "Because, recently, it was about two weeks ago, we changed our development process. We wanted the process to be more precise for the developers. Why?

Table 5.1 S1 Characterization.

CASE S1	
Items	**Description**
Founded	2013
Stage when Interviewed	Early stage
Size	Small
Employees	10
Business Model	Web Application
Segment	Agrotech
Interviewee	CTO Developer
Interview date	October 2016
Interview Time Spent	00:58:58
Actual Status	Merged
Actual stage	Growth stage

> We noticed that sometimes, about three months of development, more or less, my project team ends up delivering their implementations, the new releases, with many errors, very discrepant from what we specified, so we looked for, we did benchmark, we tried to adapt the old process, in this case, today, we detail our features better, we have a bug report protocol, even for a lay person, the commercial staff can report a bug correct for us, and with this better description, we can attack the task cards." **Team is the catalyst of development (PA.01) - Interviewee 1**.

Startup S1, due to customer demand for better product quality (CAT2.4), reformulated the web application (CAT2.3) to improve the platform to make it more robust and user-friendly (CAT2.2).

> "The website was the first product. The site initially had a simple version. Then it was completely redesigned, making the platform much more robust." **Initial growth hinders performance (PA.02) - Interviewee 1**.

S1 initially started with resources from the three partners. There are no direct reports of scarce access to financial resources. We found records that there is pressure from the customer for delivery. The startup sustains itself and seeks to improve the product to satisfy the customer and increase its customer base (CAT3.2). Nevertheless, it was identified that the team members perform multiple functions and that testing activities are neglected or underestimated. Taking into account that it is a small team of 10 people, it is observed that there is a shortage of human resources (CAT3.3) and time limitation (CAT3.4). There are also reports of a shortage of specialists (CAT3.1), as we can see in the quote below:

> "We ask the person to try to unwind, but we leave it open if he wants to take his day off to go, for example, to JusBrasil, which is a startup that has around

> here, of life, that the staff is very more experienced, with more than ten years of a startup in the market to benchmark, implement, see the technology, we always leave this open, and we have the practice here of every Friday having what we call tech-talk, so sometimes one has more experience than the other in some tool or other technology, so people from other areas can know, say what they know, pass on the knowledge to us." **Lack of resource (PA.03) - Interviewee 1**.

The need to speed up development associated with deadlines, course, and pressure from the client makes the startup give up the formal processes of documenting requirements (CAT4.4), architecture (CAT4.1), process (CAT4.3), and tests (CAT4.5) and software documentation (CAT4.2).

> "When I arrived, they were launching a new platform version. It was a somewhat limited version when I arrived, and we redesigned the platform." **Accumulated technical debt (PA.04) - Interviewee 2**.

Software product development is accelerated through third-party applications (CAT5.1) and the definition of a simple and informal software development flow (CAT5.2). The team uses tools that allow integration (CAT5.4) to favor continuous integration and monitoring of activities. The technology choice is based on personal knowledge, mainly of the senior developer (CAT5.3). Even this practice has led them to reformulate the already built solution in a second moment to make the platform more robust. Work overtime is common, especially when the release date of a product is close (CAT5.5).

> "Nowadays, we use Scrum quite adapted because it is a framework. It is well-modified, and we adapt it to our reality. It is adapted because we cannot strictly follow it. To support Scrum, we use Trello." **Speed-up development (PA.05) - Interviewee 1**.

Checking whether the product is of interest to the market is one of the priorities of startups (CAT6.1). They contract a minimum viable product through prototyping and launch it on the market to verify the market's adherence to the product. Some pivots have already happened in the product in this sense. The mobile application was discontinued, while the web version underwent refactoring to improve the application's usability and performance. The efficiency of the product comes after the launch (CAT6.2) since the MVP is launched at agricultural fairs where the target audience can test it and provide feedback on how to improve the application. Uncertainty conditions (CAT6.3), such as whether the product adheres to the market, security issues, multiple platforms, and technologies, make long-term planning unfeasible.

> "Then there was a whole reformulation. It made the platform much more robust in 2014. So in 2014, we thought about launching the App, but today, we do not have the App anymore. We discontinued it because we tried to see if the market would accept our application integrated with the platform, but

> the market did not accept it. We only had a little movement from this application. We ended up killing him. However, we continue with the platform." **Evolutionary approach (PA.06) - Interviewee 1**.

The search for innovation (CAT7.1), mainly of the product (CAT7.2), a marketplace, can be observed through the reformulation in 2014 to make the platform more robust. At that time, they also planned to launch an app, but the market did not accept the idea well, and there was little adoption. As a result, the app has been discontinued. However, the platform itself remained active and underwent constant improvements. The company seeks to vary its business model and continually introduce new features to adapt and find ways to monetize. This approach is driven by both the market and the target audience's needs. One example mentioned is the application, which was discontinued due to the public's preference for a more detailed view of the products, which was only possible on the main platform.

> "The product remains. Nevertheless, what over the years is that we try to vary. It is the business model question. We want the platform all set up. We have improvements maintained at all times, but we always try to create new functionality trying to adapt its business model. That is what startups are. A business model to monetize and pay bills and debts. We keep changing in this aspect." **Innovation-driven development (PA.07) - Interviewee 1**.

Developers have a particular focus on security and meeting users' needs. Before the weekly meetings, they make plans and corrections, considering the client's specific demands (CAT8.2). A redesign in 2014 made the platform more robust, but the app launch was unsuccessful due to users' preference for a more detailed view of the core platform (CAT8.3). The company continues to pursue business model variations and develop new capabilities to adapt and monetize. Interaction with users is valued, and their demands and suggestions are considered for improvements to the platform (CAT8.1). At events, they use a touchscreen kiosk so people can interact with the platform and give feedback on their experiences (CAT8.4).

> "As a dev, we cannot think about other things. We can think about the security part related to the platform to meet users' needs." **Quality attributes (PA.09) - Interviewee 1**.

The startup S1 places a high value on user experience (UX) (CAT9.1) and prioritizes this issue from the design process. They recognize that their audience needs to become more familiar with the internet. That is why it is essential to constantly think about their experience so that they feel comfortable on the platform and want to use it. Usability is an essential concern. The platform must be user-friendly to be valid for users. They are used to doing business through word of mouth, so providing a secure and intuitive web experience is critical to making them feel confident using the platform. Table 5.2 presents the summary of the identified factors in S1.

Table 5.2 Summary of identified factors in S1's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.1 Access to external expertise (Mentorship) CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.5 Productivity oriented CAT1.6 Self-managed team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge CAT1.10 Team under pressure
PA.02	CAT2.2 Re-engineering/Refactoring code CAT2.3 Payment of accumulated technical debt CAT2.4 Better quality product
PA.03	CAT3.1 Lack of expertise CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.1 Delegate complexity to third-party applications CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.1 Find the product/market fit quickly CAT6.2 Product efficiency starts after launch CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation CAT7.3 Business innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.2 Customer requests are documented CAT8.3 Meet customer's needs CAT8.4 Prototype validated by customers
PA.09	CAT9.1 User experience

"We prioritize this a lot, so much so that he already thinks about this issue of User Experience (UX) from the designer's process. We always have to think about their experience with the internet so that they feel more comfortable on this platform and want to keep using it. We are cautious with these UX issues. We are constantly adapting and always undertaking..” **Quality attributes (PA.09) - Interviewee 1**.

5.1.2 Case S2

S2 is a startup in the early stage, founded in 2015, that dedicated itself to developing an IoT product for the agribusiness segment. Their solution encompasses both a physical component and software. It is in its initial stage and comprises a small team of 8 people. The interview was conducted with the CTO and a developer, lasting 59 minutes. However, currently, the company is no longer active. Table 5.3 presents S2 characterization.

Table 5.3 S2 Characterization.

CASE S2	
Items	**Description**
Founded	2015
Stage when Interviewed	Early stage
Size	Small
Employees	6
Business Model	IoT Solution
Segment	Agrotech
Interviewee	CTO Developer
Interview date	October 2016
Interview Time Spent	00:59:00
Actual Status	Not active
Actual stage	-

S2 values the opportunity to access external expertise. They recognize the benefits of learning from experienced professionals to enhance their skills and knowledge (CAT1.1). There is a significant influence and impact of the founder/CEO’s background on the company. Their past experiences and expertise play a crucial role in shaping the direction and success of the business (CAT1.2). The company values engineers who can take on multiple roles (CAT1.3), indicating their preference for versatile team members who can contribute across various areas of the development process. Recognizing the importance of a high-performance team (CAT1.4) emphasizes the need for talented individuals who consistently deliver exceptional results and meet challenging goals. They strongly emphasize productivity (CAT1.5) and strive to create an environment and culture that promotes efficient and effective work practices.

The company values a self-managed team that encourages autonomy and accountability among team members, allowing them to take ownership of their work and make independent decisions (CAT1.6). Senior developers have a significant influence (CAT1.7) in the development process. Their knowledge and experience are highly valued and are crucial in shaping the company’s projects. The company has a small, co-located development team (CAT1.8). They believe in close collaboration and face-to-face interactions

Table 5.4 Summary of identified factors in S2's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.1 Access to external expertise (Mentorship) CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.5 Productivity oriented CAT1.6 Self-managed team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge CAT1.10 Team under pressure
PA.02	CAT2.3 Payment of accumulated technical debt CAT2.4 Better quality product
PA.03	CAT3.1 Lack of expertise CAT3.2 Lack of finance resource CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.1 Delegate complexity to third-party applications CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.1 Find the product/market fit quickly CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.2 Customer requests are documented CAT8.3 Meet customer's needs
PA.09	CAT9.1 User experience

between team members for effective communication and coordination. This environment is conducive to tacit knowledge (CAT1.9), shared experiences, and insights and contributes to the team's overall expertise. However, it favors the non-documentation of the decisions taken. The team works under pressure (CAT1.10), hoping it brings out the best in their team members, driving them to achieve their goals.

> "At S2, I analyzed what the project itself was, what our needs were, and in this case, what was characteristic of the application. On top of that, I decided to see what technology we were going to use or we use GSS both in the API and in the web application, so it was an architectural decision, which we needed, in the case of choosing the technology." **Team is the catalyst of development (PA.01) - Interviewee 1**.

S2, due to customers' needs, seeks to present better product usability (CAT2.4) and improvement of the application (CAT2.2).

> "Usability is crucial to us. Customers will only use it if it is easy to use. For them, it has to be something easy, and as I told you, they are used to doing business face-to-face and then doing something on the Internet. If it is not easy for customers, we will lose customers." **Initial growth hinders performance (PA.02) - Interviewee 1**.

Despite having a motivated, creative team that is self-responsible for the activities, the team needs more experience (CAT3.1) or people specialized in IoT development (CAT3.3). Learning occurs either through mentoring with other startups or through the self-taught profile of team members. This fact indicates a possible associated financial constraint (CAT3.2). The reports indicate that the team works beyond the planned hours and workload, as reported below:

> "We developed a lot. Sometimes we are there at night, 10 p.m. People send new patches." **Lack of resource (PA.03) - Interviewee 1**.

Another sign of a need for more time and human resources is the accumulation of technical debt. The team gives up writing formal documents such as requirements documentation (CAT4.4), software architecture (CAT4.1), process (CAT4.3), tests (CAT4.5), and software documentation (CAT4.2).

> "We ended up not working with class diagrams here at S2. I analyzed what the project itself was, what our needs were, and what was characteristic of the application. On top of that, I decided to see what technology we would use or use GSS both in the API and in the web application, so it was an architectural decision, which we needed, in this case, to choose the technology." **Accumulated technical debt (PA.04) - Interviewee 1**.

Giving speed to development is essential for S2 since it develops the product and aims to validate the product-market fit quickly. In order to achieve this goal, they use third-party applications (CAT5.1) through a simple and informal flow of software development (CAT5.2) and tools that allow integration (CAT5.4). The choice of technologies was made after a brief study of the necessary technology (CAT5.3). Extra work is common, mainly because they are on a product launch date (CAT5.5). The quote below refers to the intention to sell the IoT product to the market:

> They want to sell fast, these things. **Speed-up development (PA.05) - Interviewee 2.**

S2 aims to launch the IoT solution as soon as possible to confirm product-market fit (CAT6.1). For this reason, we could not find evidence indicating that product efficiency comes after launch (CAT6.2). This fact further increases the uncertainty conditions of S2 (CAT6.3).

> "Always launch an MVP to validate your product because sometimes you have an idea and think that, wow, it will impact the market and such, but what do we recommend? You make an MVP. Something significantly simplified to verify the product-market fit" **Evolutionary approach (PA.06) - Interviewee 1.**

The quote below demonstrates S2's search for innovation (CAT7.1), mainly the product (CAT7.2), an IoT solution with functionalities supporting the agro-business to carry out preventive interventions.

> "And then, what happens? We are developing hardware that we will put on the farms. This hardware works like a weather station. We will collect these inputs, and using machine learning, we will start making predictions about pests and diseases in some crops, crops, and fields. **Innovation-driven development (PA.07) - Interviewee 1.**

The IoT solution was installed on a potential customer's farm, interaction with users is valued, and their demands and suggestions are considered (CAT8.3) for improvements to the platform (CAT8.1). These improvement suggestions are documented and forwarded to the development team.

> "Not just the market. But the need of our audience. Because we must find a way to avoid being stuck to a business model, we must vary it." **Customer-driven development (PA.08) - Interviewee 2.**

Startup S2 highly values the user experience (UX) (CAT9.1) and prioritizes this issue from the design process. They recognize that their audience needs to become more familiar with the internet, so it is essential to constantly think about their experience so that they feel comfortable on the platform and want to use it. Table 5.4 summarizes the factors identified in S2.

> "We prioritize this a lot, a lot, so much so that from the designer's process, he already thinks about this issue of UX (User Experience)."**Quality attributes (PA.09) - Interviewee 1.**

5.1.3 Case S3

Founded in 2015, the company was in its early stage at the time of the interview. It had a small team of 8 employees and operated under a Software as a Service business model. The interview was conducted with the CEO and the CTO, taking 1 hour, 12 minutes, and 39 seconds. As of the interview date in October 2016, the company was still active and remained in the early-stage phase. Table 5.5 presents S3 characterization.

Table 5.5 S3 Characterization.

CASE S3	
Items	**Description**
Founded	2015
Stage when Interviewed	Early stage
Size	Small
Employees	8
Business Model	Software as a Service
Segment	-
Interviewee	CEO CTO
Interview date	October 2016
Interview Time Spent	01:12:39
Actual Status	Active
Actual stage	Early-stage

Startup S3 comprises members with experience in application development and are even mentors in the Ecossistema Baiano de Startups (Abastartup). For this reason, they do not seek external specialists. For the same reason, your influence and impact are significant as a founder/CEO of the company. Members perform multiple roles (CAT1.2). The team has members who perform various functions, such as the Project Manager, CTO, and software developer, while the Business Manager is also a software developer (CAT1.3). It is, therefore, a small team and works in the same room (CAT1.8) but with high performance (CAT1.4). Focused on efficiency and productivity (CAT1.5). The self-managed team encourages autonomy and accountability among members, allowing them to take ownership of their work and make independent decisions (CAT1.6). Senior developers have a significant influence on the development process (CAT1.7). They only have a weekly face-to-face meeting, after which all progress notifications are made to the project manager via Slack (CAT1.9). They have minimal documentation done through annotations in GitLab and Slack. The team works with tight deadlines and is constantly under pressure from clients (CAT1.10).

> "Right. In the case of S3, we sell services. We sell software development services: the back-end. We also sell the maintenance, the back-end support for applications, websites, systems, whatever." **Team is the catalyst of development (PA.01) - Interviewee 1**.

Table 5.6 Summary of identified factors in S3's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.5 Productivity oriented CAT1.6 Self-managed team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge CAT1.10 Team under pressure
PA.02	CAT2.3 Payment of accumulated technical debt CAT2.4 Better quality product
PA.03	CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.1 Find the product/market fit quickly CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation CAT7.3 Business innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.3 Meet customer's needs CAT8.4 Prototype validated by costumers
PA.09	CAT9.1 User experience CAT9.4 Scalability NCAT1.1 Maintainability

The startup S3 mentioned only one moment that there were some changes and improvements (CAT2.3). Regarding product quality, they mentioned investing more time and effort to achieve a higher quality product (CAT2.4), especially regarding usability, scalability, and maintenance.

> "Over time, we spent more time on some of these things. For a while, it was scalability. Currently, we are spending much time on reliability and maintain-

> ability. We are spending much time looking for technologies that are easy to maintain." **Initial growth hinders performance (PA.02) - Interviewee 1**.

The team is small, and despite having high performance and focusing on productivity, there is more demand for work than human resources to absorb them (CAT3.3), which is why they mention a constant time shortage (CAT3.4).

> "In the beginning, there were two people, me and him (Lucas). And then the company grew, and we had six people." **Lack of resource (PA.03) - Interviewee 1**.

The cost of a high-performance team comes from the accumulation of technical debt. The team moves quickly to create a software solution. However, it gives up writing formal documents such as requirements documentation (CAT4.4), software architecture (CAT4.1), process (CAT4.3), tests (CAT4.5), and software documentation (CAT4.2) and uses only the documentation automatically generated by the built-in tools, Gitlab, Slack and Email. The most considerable debt is from software testing activities performed manually and undocumented.

> "This is a part that we are still working on a lot. We use minimal testing. More unit testing and continuous integration is being implemented. We will use pipelines to prevent any wrong commits from messing up the rest of the team." **Accumulated technical debt (PA.04) - Interviewee 1**.

S3 did not mention the use of third-party applications (CAT5.1). Nevertheless, he mentioned the choice of a simple and informal software development flow (CAT5.2) mediated by software tools that allow integration (CAT5.4) with communication tools. The choice of technologies was based on the knowledge of the senior developer, who is also the project manager, as he has more experience with software development projects (CAT5.3). The extra work beyond the hours of (CAT5.5).

> "Weekend too." **Speed-up development (PA.05) - Interviewee 1**.

S3 has members who already work as software development freelancers. For this reason, they already had experience in providing software development services. Thus, building an application up to product-market fit (CAT6.1) took around one month. There is little evidence fr that product effectiveness comes after launch (CAT6.2). One reason could be that S3 considers all activities after software deployment maintenance activities. There is also little evidence of uncertainties since the startup has many clients experienced in the software development activity (CAT6.3), and the team is self-responsible for the projects, thus increasing the feeling of security in the startup.

> "So we realized that we were spending much time on this and time for us was money. So we started investing in technology focused on that, to be able to reduce the time we spend on maintenance, deploying, and that kind of thing." **Evolutionary approach (PA.06) - Interviewee 1**.

S3 is a startup that seeks innovation (CAT7.1), whether through a product (CAT7.2) or business (CAT7.3).

> "Today we work at S3, a technology agency focused on innovation." **Innovation-driven development (PA.07) - Interviewee 1**.

Innovation is present in the tools they choose to support software development, the tools they use to integrate communication, and software development and maintenance.

> "A startup is a highly scalable technology company with some nature of innovation at an early stage." **Innovation-driven development (PA.07) - Interviewee 1**.

The customer is involved in software development (CAT8.1) from scope definition to assessment for system improvements (CAT8.4). In addition, S3 participates in events attended by potential customers, where they can publicize the product, count on the promotion of current customers and collect improvements (CAT8.3).

> "For the service part, we adopted the strategy of being together with the target public. We started going to events. We already knew many people. Most services are by appointment. We started to do an effective after-sales service so that the user who has already used the service can also make the referral." **Customer-driven development (PA.08) - Interviewee 1**.

The startup S3 values the user experience (UX) (CAT9.1) but also showed concern with scalability and maintainability quality attributes. Mentioning that for startups, as well as meeting customer needs is an important factor, scalability and maintainability characteristics are also essential so that the customer base can grow without demanding more resources from the company. Table 5.6 summarizes the factors identified in S3.

> "Over time, we spent more time on some of these things. For a while, it was scalability. In the first startup, the system had many accesses and little scalability, and we spent much time solving this. Today this is already easy to maintain. Currently, we are spending much time on reliability and maintainability. We are spending much time looking for technologies that are easy to maintain." **Quality attributes (PA.09) - Interviewee 1**.

5.1.4 Case S4

The startup was founded in 2016 and was in its early stage during the interview. It had a small team of 10 employees and operated with a business model focused on web applications. The company belonged to the Edtech segment, targeting the educational technology market. The interview with the CEO lasted for 51 minutes and 48 seconds. However, the company was no longer active as of the interview date in October 2016. Table 5.7 presents S4 characterization.

Table 5.7 S4 Characterization.

CASE S4	
Items	**Description**
Founded	2016
Stage when Interviewed	Early stage
Size	Small
Employees	10
Business Model	Web Application
Segment	Edtech
Interviewee	CEO
Interview date	October 2016
Interview Time Spent	00:51:48
Actual Status	Not active
Actual stage	-

S4 is a startup seeking access to external expertise (CAT1.1). All team members need to gain experience in software management and development. This lack of experience impacts the startup's business development (CAT1.2). The team has multiple responsibilities and does not have a senior developer (CAT1.3). Therefore, they need a high-performance team, which they still need to achieve this objective (CT1.4). The team members jointly develop the activities, and they are all founding partners, thus being able to say that it is self-managed (CAT1.6). The team is small. It started with five members, and throughout the construction of the product idea, five more members were involved, reaching ten members (CAT1.8). The team highly values the involvement of all members in the company's discussions and decisions but does not document these discussions, leaving all knowledge tacitly (CAT1.9). Table 5.8 summarizes the factors identified in S4.

> "I prefer to discuss it there on the day. It is good because it talks about everything. We will take it until we discuss everything usually." **Team is the catalyst of development (PA.01) - Interviewee 1**.

The startup S4 is in the MVP construction phase and has been putting effort into building a quality application for its client (CAT2.4).

> "Moreover, we also thought about usability. We know that it has to be something the public looks at quickly. It is not that they feel attracted but comfortable using the tool. He is comfortable, and he will enjoy being there." **Initial growth hinders performance (PA.02) - Interviewee 1**.

Startup S4 suffers from a massive shortage of resources: specialized professionals (CAT 3.1), financial (CAT3.2), human (CAT3.3), and time (CAT3.4). Nevertheless, the need for more senior software developers has affected its performance the most. According to the reports in the interviews, they spend much time discussing technical decisions regarding the programming language, technologies, and tools to be used.

Table 5.8 Summary of identified factors in S4's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.1 Access to external expertise (Mentorship) CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.6 Self-managed team CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge
PA.02	CAT2.4 Better quality product
PA.03	CAT3.1 Lack of expertise CAT3.2 Lack of finance resource CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.1 Delegate complexity to third-party applications CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.1 Find the product/market fit quickly
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.3 Meet customer's needs CAT8.4 Prototype validated by costumers
PA.09	CAT9.1 User experience

> "Regarding the software structure, we decided not to create a direct application for cell phones. It will be a progressive web app., something like. that we took to Vinícius. In this case, who would structure it, and how would it be at the beginning? If we were to create one iOS application and another Android application, we would have more work because we are testing it." **Lack of resource (PA.03) - Interviewee 1**.

S4 is not aware of the importance of documentation and, consequently, the generation of technical debt for architecture (CAT4.1), documentation in general (CAT4.2), process (CAT4.3), requirements (CAT4.4), and testing (CAT4.5).

> "Our first meeting did not have Meeting Minutes or anything. The second

> one already had. In the third will also have. Furthermore, from then on, we will do. I would not say I like that one very much. I don't know if it's that other document right at the beginning that people send to... the agenda." **Accumulated technical debt (PA.04) - Interviewee 1**.

S4 did not address the use of third-party applications to streamline the software development process (CAT5.1) can be attributed to the team members' lack of software development knowledge. In addition, they adopt a simple and informal workflow (CAT5.2). Although they mention the use of technologies and tools already known by the team (CAT5.3), they showed no interest in exploring new technologies and tools through studies. There was also a greater focus on simple tools (CAT5.4) without an evident concern with their integration. The team recognizes the need for more experienced professionals to reduce the additional workload and meet deadlines but currently needs more financial resources to hire (CAT5.5).

> "Regarding the software structure, we decided not to create a direct application for cell phones. It is going to be a progressive web app., something like this. If we were to create an app for ios and another for Android, we would have more work because we are testing it. We will show it to those interested, they will attest, they will see what is worth it and what is not, and then we will change it. Then a web application would be better because you can only make one change, and it solves everything, and the web application, everything that is a device, is acceptable." **Speed-up development (PA.05) - Interviewee 1**.

S3 demonstrates that he is focused on building the MVP to validate the product on the market (CAT6.1) but reports that he only wants to launch the product in January of the following year. It does not mention releasing any internal releases for testing, nor does it comment on the efficiency of the software. It also does not mention documents regarding risks and uncertainties to which the startup and the product are potentially subject.

> "Just the bare minimum. We will not even focus on the future or anything. It will only really be on what will be there in January. The whole idea is structured to offer." **Evolutionary approach (PA.06) - Interviewee 1**.

Startup S4 seeks innovation (CAT7.1) by offering an innovative product (CAT7.2) for education.

> "We want something similar to Uber. The person below four stars is out. In this case, the teacher would be out. It would have a credibility ranking and give that credibility that the person with that assessment is reliable, a teacher and student." **Innovation-driven development (PA.07) - Interviewee 1**.

S4 is intended to engage customers from the beginning of development (CAT8.1). However, it has yet to gain any paying customers. They believe that customers are the best evaluators in determining whether certain functionality should be added, changed, or removed (CAT8.3) in the application. To validate this approach, they use prototypes submitted for customer validation (CAT8.4).

> "We will show it to the stakeholders, they will test it, they will see what is worth it and what is not, and then we will give their opinion. A web application would be better because you can only make one change, and it solves everything, and the web application is acceptable on any device." **Customer-driven development (PA.08) - Interviewee 1**.

The startup S4 team has a primary concern with usability (CAT9.1). They recognize the importance of a user-friendly and easy-to-use interface for users. The application must be intuitive and provide a pleasant user experience, facilitating their interaction with the system.

> "It has to be something the audience looks at quickly, feels attracted to and comfortable using the tool, and will enjoy." **Quality attributes (PA.09) - Interviewee 1**.

5.1.5 Case S5

The startup was founded in 2014 and was in its early stage at the time of the interview. It had a small team of 10 employees and operated with a business model focused on an Open Data Platform. The company belonged to the Lawtech segment, targeting the legal technology market. The interview with the CEO and CTO lasted 50 minutes and 5 seconds. As of the interview date in October 2016, the company was still active and had progressed to the growth stage. Table 5.9 presents S5 characterization.

Table 5.9 S5 Characterization.

CASE S5	
Items	**Description**
Founded	2014
Stage when Interviewed	Early stage
Size	Small
Employees	10
Business Model	Open Data Platform
Segment	Lawtech
Interviewee	CEO CTO
Interview date	October 2016
Interview Time Spent	00:50:05
Status	Active
Actual stage	Growth stage

Startup S5 has experienced developers. For this reason, they do not mention the need to access external expertise (CAT1.1). The knowledge of the three founding partners who also participate in software development positively impacts (CAT1.2) software development activities. The team has full-stack developers (CAT1.7), knowledge is shared, and members accumulate multiple responsibilities (CAT1.3). As the business manager also develops, the project manager performs software testing. They are a high-performance team (CAT1.4) focused on productivity (CAT1.5). It is a self-managed team (CAT1.6), small, consisting of 8 members (CAT1.8), based on tacit knowledge (CAT1.9). And works under pressure (CAT1.10).

> "S5 has been in this format for about 2 or 3 years. We were in college, we came here, and we came to work together. We had some startup ideas, and the one that worked was S5, a platform that mines data." **Team is the catalyst of development (PA.01) - Interviewee 1**.

S5 mentioned having performed refactorings in the code, which implies changes in the application (CAT2.2). While they did not mention bug fixing or technical debt, they acknowledge that they constantly evolve. In the team's words, they are always looking for continuous improvements and enhancements, which indicates that the team is aware of the need for frequent corrections and improvements (CAT2.3). The team's main objective is to guarantee the application's usability and make the data more accessible to the end user (CAT2.4).

> "We made three bigger ones and changed the whole site." **Initial growth hinders performance (PA.02) - Interviewee 1**.

Although S5 does not mention a lack of specialists, they share decisions with the entire team until reaching a consensus (CAT3.1). They also do not mention the need for external resources, declaring themselves bootstrapping, but the company is located in the *Parque Tecnológico da Bahia* or BahiaTec. Its objective is to promote integration between companies, research institutions, universities, and the public sector, to boost innovation and economic growth in the region. It offers infrastructure and support services to companies and institutions in its space. Conducive environment for applied research, technology transfer, and promoting entrepreneurial and innovative culture in Bahia (CAT3.2). The team faces human resource limitations (CAT3.3), which means they have a reduced team regarding the number of members to meet the demands received. This restriction can affect the product's quality and the project's development. In addition, they also face time constraints (CAT3.4), which indicates they have tight deadlines or a significant workload. These constraints are challenges the team needs to face and overcome to ensure the project's continued success and progress.

> "Leaving here late is common. What is it that we do something when it is urgent? Usually, we stay here later." **Lack of resource (PA.03) - Interviewee 1**.

Table 5.10 Summary of identified factors in S5's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 igh-performance team CAT1.5 Productivity oriented CAT1.6 Self-managed team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge CAT1.10 Team under pressure
PA.02	CAT2.2 Re-engineering/Refactoring code CAT2.3 Payment of accumulated technical debt CAT2.4 Better quality product
PA.03	CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.1 Delegate complexity to third-party applications CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.1 Find the product/market fit quickly CAT6.2 Product efficiency starts after launch CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.2 Customer requests are documented CAT8.3 Meet customer's needs CAT8.4 Prototype validated by costumers
PA.09	CAT9.1 User experience

During the interview, participants expressed the existence of technical debts, although they did not use specific terms. They recognize that these debts are present and that they are working to resolve them over time. The areas affected by the technical debts mentioned are: architecture (CAT4.1), the participants recognize that there are issues related to the architecture of the system that need to be addressed and improved, suggesting the need for adjustments and optimizations in the general structure of the application;

documentation (CAT4.2), lack or deficiency in project documentation is mentioned as a technical debt, indicating the importance of properly recording and documenting decisions, functioning and changes made to the project; process (CAT4.3), the participants recognize that there are problems in the development processes that need to be corrected, implying the need to establish and follow best practices, efficient methodologies and more effective workflows; requirements (CAT4.4), technical debts are also present in the requirements area, suggesting that there are gaps or inconsistencies in the project requirements, since the screens are initially defined through a paper prototype that circulates around the team, emphasizing the importance an in-depth analysis and clear understanding of system requirements; testing (CAT4.5), the lack of adequate testing is recognized as a technical debt highlighting the need to implement comprehensive testing strategies, integration testing, system testing and acceptance testing in order to ensure the quality of the developed software. Only unit tests were mentioned.

> "We defined a standard; the team always gave many opinions. The fundamentals here are very pointed out, they give their opinions." **Accumulated technical debt (PA.04) - Interviewee 1**.

> "Initially, it was a bit crazy, but then we used prototyping." **Accumulated technical debt (PA.04) - Interviewee 2**.

During the interview with S5, it was evident that they adopted efficient strategies to optimize the project's development. S5 recognizes the importance of leveraging off-the-shelf solutions to streamline the development process. It demonstrates an intelligent and pragmatic approach, seeking to use external resources and third-party applications (CAT5.1), which can boost the efficiency and productivity of the team. S5 values the knowledge and expertise of the team when choosing technologies they are familiar with (CAT5.3), allowing them to maximize productivity and avoid unnecessary learning curves by focusing on solutions they have already mastered.

S5 prioritizes using tools that integrate (CAT5.4) well with each other, facilitating the team's workflow, avoiding information fragmentation, and improving collaboration. The effective integration of tools allows for greater efficiency and agility in problem-solving. S5 has the talent to transform complex tasks into simple and easily solved activities (CAT5.2). This simplification skill allows them to deal with complex problems agile and efficiently, optimizing the team's time and resources. Be willing to work overtime (CAT5.5) to meet established deadlines. This dedication demonstrates commitment and determination to deliver the project on time, even if it requires additional effort. The combination of these strategies demonstrates the effectiveness of S5 in dealing with challenges and maintaining an agile and productive workflow. Its ability to simplify complex tasks, utilize standard technologies, and integrate tools efficiently are critical success factors.

> "Because they had very long tasks that we had to break down, a repetitive task can take weeks." **Speed-up development (PA.05) - Interviewee 1**.

Startup S5 has a highly focused approach to validating the product with the market (CAT6.1). Throughout the interview, it was clear that they believed that product efficiency could only really be achieved after market launch and customer validation (CAT6.2). This market-driven mindset reflects the importance S5 places on continually getting honest customer feedback and insights to improve the product. Regarding planning, S5 adopts a more short-term focus, not having long-term planning (CAT6.3), indicating that they prefer to adapt and adjust their strategies based on market responses and customer demands rather than sticking to a rigid, inflexible plan. This more agile and flexible approach can allow them to be more responsive to changes and innovations in their industry. This constant validation and strategic adaptation mindset demonstrates S5's commitment to delivering high-quality products that meet customers' needs and expectations. By prioritizing direct interaction with the market and being open to continuous adjustments and improvements, they are positioned to stand out in a highly competitive and constantly evolving environment.

> "We prototype, and then we iterate until we close a product. We separate features. The process is more or less like this." **Evolutionary approach (PA.06) - Interviewee 1**.

The S5 startup team intensely focused on pursuing innovation opportunities (CAT7.1). During the interview, it was evident that they are constantly looking for ways to improve and innovate their product. In addition, S5 also showed a commitment to innovation in the product itself (CAT7.2). They recognize the importance of continually developing and improving products to meet ever-evolving customer demands. This innovation-driven approach means that S5 is willing to invest time and resources in developing new features and enhancements to ensure customer satisfaction and stay ahead of the competition. With its mindset to pursue opportunities for innovation and dedication to continuous product improvement, S5 is positioned to stand out as a leader in its field. Their proactive and open approach to new ideas and technologies allows them to explore the full potential of innovation and offer ever more effective solutions to their clients.

> "We innovate these frequencies when we have bigger versions, but we cannot say how long we worked on them because we stopped to do other things, services, and other products, so we do not know precisely how long it took." **Innovation-driven development (PA.07) - Interviewee 1**.

Startup S5's team is committed to engaging customers from the beginning of development (CAT8.1), ensuring their needs are met. They carefully document customer requests (CAT8.2) and strive to meet their needs (CAT8.3). In addition, they validate prototypes with customers to ensure that proposed solutions meet expectations (CAT8.4). With this customer-centric approach, S5 ensures that the final product is highly relevant and meets market demands.

> "It is because of everything like that. I would consider everything as an amendment. Because they have needs, right? The customer needs, right, even

> small changes, he needed them.” **Customer-driven development (PA.08) - Interviewee 1**.

The startup S5 values the user experience (CAT9.1), seeking to offer a friendly and easy-to-use interface in its products. They are concerned with providing an excellent user experience, ensuring that interaction with the product is intuitive and pleasant.

> “Because everything I consider an improvement to the customers’ needs, even small changes they need (to improve their experience).” **Quality attributes (PA.09) - Interviewee 1**.

5.1.6 Case S6

The startup company was founded in 2006 and was in its early stages during the interview. It had a small team of 10 employees and operated with a Software-as-a-service business model. The interview with the CEO and CTO lasted 1 hour, 10 minutes, and 10 seconds. As of October 2016, the company was still active and remained in the early stage of development. Table 5.11 presents S6 characterization.

Table 5.11 S6 Characterization.

CASE S6	
Items	**Description**
Founded	2006
Stage when Interviewed	Early stage
Size	Small
Employees	10
Business Model	Software as a Service
Segment	-
Interviewee	CEO CTO
Interview date	October 2016
Interview Time Spent	01:10:10
Actual Status	Active
Actual stage	Early stage

The CEO and CTO are business managers technically responsible for the project; their training (CAT1.2) and presence positively impact software development acceleration. We found a team with members who perform different responsibilities (CAT1.3). They have a team made up of experienced developers (CAT1.7) with high performance (CAT1.4) and productivity-oriented (CAT1.5) due to the integrated tools they use, not the feeling of belonging to the business. There is a boss-employee relationship, rather than a leader-team relationship, a posture that negatively affects communication. The team is small, consisting of 10 members (CAT1.8). Knowledge is tacit (CAT1.9); there is only the registration of tasks in the tools, and in addition, the environment and layout of the work table do not favor communication among members. The team is pressured to meet the

Table 5.12 Summary of identified factors in S6's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 igh-performance team CAT1.5 Productivity oriented CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge CAT1.10 Team under pressure
PA.02	CAT2.4 Better quality product
PA.03	CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.1 Delegate complexity to third-party applications CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.2 Product efficiency starts after launch CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.2 Customer requests are documented CAT8.3 Meet customer's needs CAT8.4 Prototype validated by customers
PA.09	CAT9.1 User experience

deadlines defined by the customer (CAT1.10). It is a company focused on productivity and intends to innovate. However, According to the quotation below, it is not open to new ideas, the possibility of change, and the team's self-management, which does not favor the construction of an innovation environment. A company that is closed to the possibility of failures in developing its software products but chronicles products that failed when they moved into the product-market fit.

> "It kills creativity; you cannot have someone in the process with a new idea and want to change the specification because they are wasting time, even if it is a good idea. Also, it will be a person you cannot control, a very creative person in a position like that who will have to do something pre-specified in a specified way. He will not collaborate with the company, which will be a

> problem. So this growth jumps here. We have permanently opted for growth in terms of more productivity. Use more productivity tools." **Team is the catalyst of development (PA.01) - Interviewee 1**.

S6 is a startup focused on business (CAT2.1) and agile software development (CAT2.2). They prioritize speed, quickly building products based on previous versions of software. However, they do not recognize the existence of technical debt and do not dedicate efforts to resolve it (CAT2.3). S6 repeatedly emphasizes the importance of the product quality offered to the customer (CAT2.4).

> "The user's company will be better, which makes much difference. I can make a system with the exact requirements of another system but make it more beautiful and easier to use. That is what a startup has to offer: a great user experience." **Initial growth hinders performance (PA.02) - Interviewee 1**.

The team is small (CAT3.3) and tries to overcome the time shortage (CAT3.4) through tools that support software development. One point that drew attention was the lack of interest in making the development team grow due to the loss of quality or performance. Another interesting point is when the interviewee narrates that the number of team members is the same. However, they are different people. There is a potential loss of speed in development when a team member leaves, and another with no experience arrives.

> "The amount of people is the same, but they are not the same person." **Lack of resource (PA.03) - Interviewee 1**.

No technical debt was identified concerning the software architecture (CAT4.1). This fact could occur due to the team relying on previous solutions. However, this indicates that startup S6 is potentially not solving new problems (innovating), just conquering new customers. Technical debts of documentation (CAT4.2), process (CAT4.3), requirements (CAT4.4), and software tests (CAT4.5) were identified.

> "Documents do not exist." **Accumulated technical debt (PA.04) - Interviewee 1**.

S6 strives to speed up software development, use third-party solutions (CAT5.1), have a simple workflow (CAT5.2), and use technology they know to speed up development (CAT5.3). They use proprietary Microsoft tools that are well-integrated (CAT5.4). Despite this, they work extra hours to reach the deadlines agreed upon with the clients (CAT5.5).

> "We pay to use it. However, the cost is worth it. We do not need to develop it because it is already ready, but it pays off in practice using it. The cost to use is lower than the cost we would have to develop." **Speed-up development (PA.05) - Interviewee 1**.

S6 does not test its product in the market (CAT6.1); it seeks to sell software development services to specific customers. S6 recognizes that it deploys the software and improves over time (CAT6.2). S6 is under constant uncertainty due to the challenging context in which software development is currently submitted with the diversification of technology and electronic devices (CAT6.3).

> "Implementation today is a significant problem; first, deadlines are decreasing because you have to implement quickly. You have to have the right tools to do the implementation. The implementation is diverse. You have this problem, for example." **Evolutionary approach (PA.06) - Interviewee 1**.

S6 repeatedly mentions an interest in innovation (CAT7.1), but their products are based on their other previous solutions (CAT7.2) and how companies have specific customers (CAT7.3). In addition, he is not open to new ideas and believes in controlling the team's activities to speed up software development. This stance indicates that the S6 needs help creating an environment open to change and innovation. There is even a mistaken use of the term innovation with current technologies, as shown below.

> "We train ourselves to be innovative and always use the latest technologies, talking to clients with the potential to innovate in their businesses. In addition, the company began to receive public notices from SEBRAE, sometimes being hired directly to carry out innovations for customers. Our story so far." **Innovation-driven development (PA.07) - Interviewee 1**.

There is a concern at startup S6 to involve the customer in developing the software product from conception (CAT8.1) to understand what he needs and align expectations. There is a pre-defined delivery schedule for the customer (CAT8.2) to meet their needs (CAT8.3). In these meetings, the client validates the prototype evolutions (CAT8.4).

> "It always is. Moreover, the first version is always a prototype. We leave the doors open for them to give their opinions." **Customer-driven development (PA.08) - Interviewee 1**.

S6 mentions concern with the user experience, especially presenting an elegant and easy-to-use application (CAT9.1).

> "Even in this new software, we do not work with the requirements specification. We work with prototypes. First, the tool allows a prototype to be made quickly. Then it's easier for the guy to look and say it is not that, it is not that color, not like that, this field is not here, seeing what and specifying." **Quality attributes (PA.09) - Interviewee 1**.

5.1.7 Case S7

The company was founded in 2012 and was in its early stages during the interview. It operated as a micro-sized business with only three employees. Their business model revolved around a web application. The company's focus was on the development of solutions for smart cities. The interview with the CEO/CTO lasted for 51 minutes and 27 seconds. As of February 2017, the company remained active and continued to be in its early stage of development. Table 5.13 presents S7 characterization.

Table 5.13 S7 Characterization.

CASE S7	
Items	**Description**
Founded	2012
Stage when Interviewed	Early stage
Size	Micro
Employees	3
Business Model	Web Application
Segment	Smart Cities
Interviewee	CEO/CTO
Interview date	February 2017
Interview Time Spent	00:51:27
Actual Status	Active
Actual stage	Early stage

S7 is a small startup formed by three partners (CAT1.8), then the company grew, and today it has five members. They accumulate responsibilities related to software development (CAT1.3). One of the members is a project manager and software developer. Another is a business manager and software developer. Therefore, the founders' knowledge significantly impacts startup decisions (CAT1.2, CAT1.7). In addition, the members were self-taught but sought external knowledge (CAT1.1). The team managed itself (CAT1.6), registering development demands through Trello (CAT1.5). It was a high-performance team because it was small (CAT1.4) and based on tacit knowledge (CAT1.9). It did not suffer external pressure (CAT1.10).

> "Look, we saw the internet. It was brute force. We researched, found a way to solve that problem, tested it, solved it or not, and consulted other people with more experienced colleagues. There was a colleague who helped a lot in this part, who had already developed other things." **Team is the catalyst of development (PA.01) - Interviewee 1**.

S7 faces challenges in its early phase, which has affected its performance. The team shifted its focus to business concerns (CAT2.1) to improve overall performance. They recognized the need to refactor their database (CAT2.2) to improve its efficiency and maintainability. S7 recognized the existence of accumulated technical debt and the importance of resolving it (CAT2.3). They understand that resolving these issues is crucial

Table 5.14 Summary of identified factors in S7's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.1 Access to external expertise (Mentorship) CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.6 Self-managed team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge
PA.02	CAT2.1 Focus change to business concerns CAT2.2 Re-engineering/Refactoring code CAT2.3 Payment of accumulated technical debt CAT2.4 Better quality product
PA.03	CAT3.1 Lack of expertise CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology
PA.06	CAT6.1 Find the product/market fit quickly CAT6.2 Product efficiency starts after launch
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.3 Meet customer's needs CAT8.4 Prototype validated by customers
PA.09	CAT9.1 User experience CAT9.2 Availability CAT9.3 Fault tolerance CAT9.4 Scalability

to the long-term success of their product. By investing efforts in eliminating technical debt, they aim to deliver their customers a better product (CAT2.4), ensuring a better user experience.

> "The biggest was usability. First, we launch only a web application version. A web version was created. After the apps appeared, we were initially concerned about usability in the web version." **Initial growth hinders performance**

> **(PA.02) - Interviewee 1**.

There needs to be more experience (CAT3.1). For this reason, they report searching the internet or with friends for the knowledge to develop the web application. The initial team is small. It starts with three components and grows to 5 members (CAT3.2). In addition, they report seeking specialized knowledge outside the team, reinforcing the need for more human resources to build the team (CAT3.3). The only time pressure they suffered was due to the notification of delivery of the software provided to the press that announced the application's release site (CAT3.4).

> "Look, I became a manager because we started developing with some tools I mastered. At the time, my partner did not call himself a programmer, so he was more in charge of layout, design, and front-end. So, I managed the development. This was at the beginning." **Lack of resource (PA.03) - Interviewee 1**.

S7 is a startup that faces several challenges related to the accumulation of technical debt: architectural debt (CAT4.1), which indicates the need to review and improve the structure of the system to ensure its scalability and efficiency; documentation debt (CAT4.2), which highlights the importance of documenting to facilitate future maintenance; process debt (CAT4.3), suggesting that the team needs to improve its workflows and practices to improve efficiency and product quality; requirements debt (CAT4.4), the impact is minimal, which indicates that the team generally has a clear understanding of the project requirements; and, test debt (CAT4.5), which indicates the need to improve and expand your testing processes to ensure software reliability.

> "To give us an idea of what we still must do." **Accumulated technical debt (PA.04) - Interviewee 1**.

S7, to speed up development, just invested in a simple workflow (CAT5.2) and use of well-known tools to develop the web application (CAT5.3): Trello for project management and Github for configuration and change management.

> "Let us do this here in a week because in four was our deadline. So we were into other things." **Speed-up development (PA.05) - Interviewee 1**.

The startup S7 seeks to quickly find the adequacy of the product to the market (CAT6.1). They recognize that product efficiency is only achieved after market launch (CAT6.2), demonstrating their results-oriented approach and the importance of validating the product with customers to ensure its success. S7 does not mention dealing with any uncertainty (CAT6.3), nor does it mention scarce resources. This fact may be due to those involved having other parallel businesses. There is not a 100% focus on the startup.

> "First, of course, we made the initial screen. And then you saw, for example, how we will make the user search for his city. There it developed. Let us put some information here in the beginning to instigate the user. There it developed." **Evolutionary approach (PA.06) - Interviewee 1**.

S7 looks for innovation. One of the pieces of evidence in this regard is the search for collaborating with a public security service through a crowdsourcing application (CAT7.1, CAT7.2).

> "We talked occasionally about social media, startups, and projects. Then we saw a public safety factor, that crime was evident to people. We watched newspapers and discussed the themes." **Innovation-driven development (PA.07) - Interviewee 1**.

Through the interview, we collected evidence that S7 included the system user from the beginning of development by providing a web page about the project and discussing it with potential users (CAT8.1, CAT8.3, CAT8.4). However, there is no record that customer requests were documented (CAT8.2).

> "The first version we deliver in the first three months. We just launched a page to get feedback from users. He shared it with friends to understand what it would be. Moreover, we worked for three more months on the first version. So it took six months of development to go live." **Customer-driven development (PA.08) - Interviewee 1**.

S7 mentioned concern with the following quality attributes: user experience (CAT9.1), availability (CAT9.2), fault tolerance (CAT9.3), and scalability (CAT9.4). These attributes are essential for a crowdsourcing application.

> "Yes. Scalability. Right away, it was one of the things we were most concerned about because it was a platform for crimes. We wanted anyone to be able to use it, class D, E. Although the process is not mass for these people, we were concerned that anyone could use it." **Quality attributes (PA.09) - Interviewee 1**.

5.1.8 Case S8

Founded in 2012, the startup was in the early stages when the interview took place. It operated as a micro-sized business with a team of 3 employees. Their business model revolved around a web application, focusing on providing solutions in the Govtech sector. The interview with the CTO/Developer lasted for 55 minutes and 23 seconds. As of March 2017, the company remained active and continued to be in its early stage of development. Table 5.15 presents S8 characterization.

S8 can access external specialists who answer questions and advise on software development issues (CAT1.1). The team is small (CAT1.8), so there is an accumulation of responsibilities (CAT1.3). They have a team that has high performance (CAT1.4) and is highly organized (CAT1.6). They use Microsoft's integrated software development tools (CAT1.7). Much knowledge is transmitted tacitly since they are close to each other, and the organization of the room and the layout of the table and computers favor communication (CAT1.9). The CEO of S8 is from a different area than the software development

Table 5.15 S8 Characterization.

CASE S8	
Items	**Description**
Founded	2012
Stage when Interviewed	Early stage
Size	Micro
Employees	3
Business Model	Web Application
Segment	Govtech
Interviewee	CTO/Developer
Interview date	March 2017
Interview Time Spent	00:55:23
Actual Status	Active
Actual stage	Early stage

area. For this reason, his knowledge does not impact software development (CAT1.2). He is more responsible for contacting customers and bringing their needs to the team. The team manages itself; everyone can see the tasks and their progress through the tool. Nevertheless, they do not mention concerns about productivity (CAT1.5) or pressure to deliver (CAT1.10).

> "He is a DBA, he is an analyst, he is a developer, if you are kidding, he is a designer too, but like here we have, despite talking about the prototype, the design, and everything, we meet with all those involved, in theory, then the designer he will prototype. We will talk pow. It was fantastic, I liked it, and will it catch on? We will throw ideas to reach a consensus and leave the product in a way that we believe will be accepted in the market. Something like that." **Team is the catalyst of development (PA.01) - Interviewee 1**.

S8 mentions that they have already made improvements to the application code (CAT2.1), (CAT2.2), and technical debt payment (CAT2.3) to improve the quality of the application for the end user (CAT2.4). Mainly, it is an application for mobile devices with a series of restrictions on memory and internet access, for example.

> "We wanted something of quality, to be fluid so that the application would not crash or consume too much memory. However, thinking about the risk, we thought yes, like. In undocumented conversation. A memory overflow can happen, and if the application crashes, it will close the application. What to do at this point?" **Initial growth hinders performance (PA.02) - Interviewee 1**.

S8 does not mention the scarcity of financial resources (CAT3.2). Nevertheless, it is located in the Technological Park of Bahia, which offers some benefits related to different rental prices, other services, and the possibility of networking since it has several

Table 5.16 Summary of identified factors in S8's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.1 Access to external expertise (Mentorship) CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.5 Productivity oriented CAT1.6 Self-managed team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge CAT1.10 Team under pressure
PA.02	CAT2.2 Re-engineering/Refactoring code CAT2.3 Payment of accumulated technical debt CAT2.4 Better quality product
PA.03	CAT3.1 Lack of expertise CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.1 Delegate complexity to third-party applications CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.1 Find the product/market fit quickly CAT6.2 Product efficiency starts after launch CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.3 Meet customer's needs CAT8.4 Prototype validated by customers
PA.09	CAT9.1 User experience CAT9.2 Availability CAT9.3 Fault tolerance CAT9.4 Scalability NCAT1.1 Maintainability

companies, institutes, and universities in the area. S8 lacks expertise in mobile development (CAT3.1). Its professionals are specialists in web development. However, since the startup project involved a mobile application, they studied and consulted external experts to meet this need. The team is small (CAT3.3), and since their last application was unsuccessful, it has shrunk. Therefore, they report a need for more time (CAT3.4) that they try to circumvent with planning and a simple workflow.

> "As it was mobile and we had never done it before, we knew what we wanted but needed to know how. This is the very truth. Ah, we want information on how to do it on mobile. Let us study. We knew what we wanted, but knowing how to do it was the crux." **Lack of resource (PA.03) - Interviewee 1**.

Through reports and observation, we identified that S8 had difficulty generating and maintaining documentation even when the team had five members in the development part (CAT4.2). When the team was reduced to two members, they felt that documentation needed to be revised since there was no third person to report on development issues. They already had a weekly meeting with the CEO. We were able to identify debt technical architecture (CAT4.1), process (CAT4.3), requirements (CAT4.4), and software testing (CAT4.5).

> "We had no documentation." **Accumulated technical debt (PA.04) - Interviewee 1**.

S8 works with Microsoft's built-in tools (CAT5.4), Visual Studio, TFS, Github, Visio, Astra, Erwin, Photoshop, and Android (CAT5.3). They have a simple workflow (CAT5.2), initially used the physical Scrum framework, and later moved to the tools. They also mentioned the use of Google/Android API (CAT5.1). They only reported working overtime when there was a problem with the system, but they usually worked 40 hours a week (CAT5.5).

> "Being from Microsoft, they are integrated. We use. However, we also use it outside as a Git. No problem about that; we use third parties." **Speed-up development (PA.05) - Interviewee 1**.

S8 has the practice of launching the product on the market to validate it (CAT6.1) and is aware that efficiency comes after the users use the application (CAT6.2). S8 is under several conditions of uncertainty, one of them is the background of the development team, which is a web application and will need to develop an application for a smartphone, the number of different technologies and devices, the number of different technologies that restrict memory and access the internet (CAT6.3).

> "Because we always work with the web, the reality is different when you go to develop for mobile. The time is different. You have other issues, such as resource limitations, battery limitations, and things you would bring to the web. You have to give a swipe for mobile. You cannot fit everything." **Evolutionary approach (PA.06) - Interviewee 1**.

S8 is an innovation initiative of a medium-sized private company (CAT7.1). The company created this independent team based in the Parque Tecnológico da Bahia to develop innovative products (CAT7.2).

> "Try to invest, innovate, and differentiate from what it had been doing before." **Innovation-driven development (PA.07) - Interviewee 1**.

The client is involved in the software development process through virtual meetings via Skype and face-to-face (CAT8.1). These meetings occur to align the product with the customer's needs (CAT8.3). Prototypes and intermediate versions of the software are validated at these meetings (CAT8.4).

> "It is via email or Skype meeting. There was face-to-face too. We had to go there several times for a little meeting to show how it was." **Customer-driven development (PA.08) - Interviewee 1**.

The application developed by S8 was a product with a web version and an app version. As it was an application with a large number of accesses, they mentioned that the critical quality attributes for their application were: user experience (CAT9.1), availability (CAT9.2), fault tolerance (CAT9.3), (CAT9.4), scalability and maintainability (NCAT1.1). The exception is the user experience that was achieved through S8 software development. Other quality attributes were achieved through third-party services such as Microsoft Azure.

> "There was absurd access, at least an initial one. If we think about the city of São Paulo, hundreds of millions of people use this section, so we immediately thought about the scalability of your servers. It certainly will not hold up. It is better to put it safely in the cloud to scale this. Initially, we thought about Google and Amazon, but then they saw a better package for them and suggested Microsoft Azure. However, not by our influence, ah, it will be this one or the other one." **Quality attributes (PA.09) - Interviewee 1**.

5.1.9 Case S9

The company was founded in 2009 and was in its early stages during the interview. It operated as a small business with a team of 11 employees. Their business model focused on Software as a Service (SaaS). The interview, conducted with the CEO/CTO, took place in March 2017 and lasted 45 minutes and 5 seconds. As of that date, the company remained active and continued to be in its early stage of development. Table 5.17 presents S9 characterization.

S9 is a startup located in a room at Parque Tecnológico da Bahia (CAT1.8). S9 is a company that diversifies in the range of software services. It offers cloud computing services, infrastructure services, server support (IBM, JBoss, IIS, Tomcat, and Apache), software development (Maker, Java, Delphi, Python, Ionic), and API development. It is formed by a small team (CAT1.8) composed of 7 members, two partners, and five

Table 5.17 S9 Characterization.

CASE S9	
Items	**Description**
Founded	2009
Stage when Interviewed	Early stage
Size	Small
Employees	11
Business Model	SaaS
Segment	-
Interviewee	CEO/CTO
Interview date	March 2017
Interview Time Spent	00:45:05
Actual Status	Active
Actual stage	Early stage

developers. All senior developers (CAT1.7). The most experienced person on the team is the CEO (CAT1.2); his knowledge positively impacts the team, speeding up development and avoiding pitfalls. Because they are experienced, the team performs well (CAT1.4) and is productivity-oriented (CAT1.5). All members developed multiple responsibilities in the startup (CAT1.3), and the partners, business, and project managers are also developers. Each developer also assumes technical responsibility for projects. Perform good teamwork and appear to have a professional relationship based on trust, as each is responsible for a project internally and must manage that project while also participating in developing software products. Generally, do not prioritize documents, so most knowledge is passed on tacitly (CAT1.9), including using XP to exchange knowledge between team members so that one member can always compensate for the absence of the other (CAT1.1). In addition, S9 is a company that provides services to software development companies, so they have specialized knowledge. S9 is also not under pressure (CAT1.10), possibly because they can plan the activities quickly and efficiently. They know how to price their time and have different plans for large, small, and medium-sized companies.

> "Most of our APIs are customer needs that he does not find native in a tool, or he does not find easy in the market. Then comes that turnstile that our client bought in China without documents, so we integrated it there and made it communicate with the system. So the government started launching new features like electronic and service invoices, and now e-social is coming. So this news will interact with innovation in our client's system, and we went to interact with the component. So we had a whole expertise." **Team is the catalyst of development (PA.01) - Interviewee 1**.

Through the semi-structured interview and observations, we found that S9 is a team focused on software development and with experience. For this reason, we have not found evidence that the focus has shifted to business (CAT2.1), refactoring/reengineering (CAT2.2), or technical debt repayment (CAT2.3). We only note that he is focused on

Table 5.18 Summary of identified factors in S9's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.5 Productivity oriented CAT1.6 Self-managed team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge
PA.02	CAT2.4 Better quality product
PA.03	CAT3.1 Lack of expertise CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools
PA.06	CAT6.2 Product efficiency starts after launch CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation CAT7.3 Business innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.2 Customer requests are documented CAT8.3 Meet customer's needs CAT8.4 Prototype validated by customers
PA.09	CAT9.1 User experience

offering a higher quality product to the customer (CAT2.4), which does not affect the development speed.

> "We like small deliveries, small developments for the customer to have something to see faster, and if it is not that, because sometimes the customer does not know what he wants, there is a lot here. So, like this, he says: Ah, that is what I wanted. Then we accelerate." **Initial growth hinders performance (PA.02) - Interviewee 1**.

We only found evidence that S9 lacks human resources (CAT3.3). The accumulation of responsibilities that the team accumulates is great. In addition, they have a high demand from customers. The team is composed of seven members, as mentioned earlier.

> "So it is a small team, and each manages a project." **Lack of resource (PA.03) - Interviewee 1**.

Startup S9 has technical debts in different areas: architecture debt (CAT4.1), documentation debt (CAT4.2), process debt (CAT4.3), requirements debt (CAT4.4), and testing debt (CAT4.5). These debts reflect areas where the company needs to dedicate efforts to improve its structure, documentation, processes, requirements, and tests. Attention to these areas is crucial to ensure the quality and efficiency of the products and services, especially when the startup enters the growth stage.

> "This is a problem on the S9 because, as it is a small team, the tester is usually addicted, right? We try to run it, but due to lack of time, another team is busy, so we cannot run the products. So, today we have a team developing a new component, a new grid, a more visual grid, so when a beta is released, it is passed on to everyone in the company to test. So some test, others don't because they are busy. However, it is what we try to minimize addiction. We pick up the products and switch teams to test them." **Accumulated technical debt (PA.04) - Interviewee 1**.

S9 works with well-known (CAT5.3) and well-integrated (CAT5.4) tools by the team. With a simple and informal workflow, productivity-oriented (CAT5.2). We have not identified that S9 uses third-party applications. S9 produces APIs for software development companies (CAT5.1). We also did not identify that S9 works overtime to meet deadlines (CAT5.5). S9 is a team of experienced developers, self-managed, with a simple and efficient workflow.

> "We have to have a person who is the father of that project so that everything is not overloaded in our hands and also." **Speed-up development (PA.05) - Interviewee 1**.

S9 does not need to validate the product because it offers various services and software to large, medium, and small customers (CAT6.1). This factor makes them generate revenue, not depending on external resources. However, we identified that S9 is quick to bring the customer to the development of this definition of scope and construction of the prototype until the implementation of the software (CAT6.2). Thus, the software is adjusted, meets the customer's real needs, and becomes more efficient. Despite S9's productivity and focus, she experiences conditions of uncertainty (CAT6.3). Recently, one of its products failed due to third-party technology change.

> "Yeah, usually, we do a test with our client. We do. Right now, we have a severe problem: We bet a lot on a new market niche for our APIs consummated in the client layer in all technologies. It was cool because the technology area

> would significantly increase the sales range of our product, but then Google Chrome blocked Java, and our APIs needed to use Java in the client layer. It was pretty close to release. Furthermore, Firefox also blocked it. It was only working in IE. That practically broke our legs." **Evolutionary approach (PA.06) - Interviewee 1**.

Startup S9 demonstrates a strong focus on innovation opportunities (CAT7.1). They constantly look for ways to innovate their products and services to meet customer needs (CAT7.2). We registered several pivots: customer pivot, product pivot, technology pivot, and business pivot (CAT7.3). Still, their commitment to innovation is evident and could drive their success.

> "Yes. We do not rule out using a technology the team has not already used. So, that is why it is multidisciplinary and self-taught. We already used technologies in projects that were the first time the team used them. Nevertheless, we are comfortable with it." **Innovation-driven development (PA.07) - Interviewee 1**.

Startup S9 takes a customer-driven development approach, involving customers early in the product design process (CAT8.1). They recognize the importance of documenting customer requests (CAT8.2) and strive to meet customer needs (CAT8.3). Additionally, they validate their prototypes through customer feedback (CAT8.4), ensuring the final product meets their expectations. This customer-centric approach is a crucial factor in S9's success.

> "It is more as we talk to the customer, divide it into small steps, small deliveries, and then start to document what we have in the work of the week or month." **Customer-driven development (PA.08) - Interviewee 1**.

When S9 was asked about the most relevant quality attributes (CAT9.1), she always replied that the strategy was to draw the customer into the development. Customer satisfaction is the most critical parameter. Therefore, we register that the user experience is an important quality attribute for S9.

> "The customer participates a lot in all phases, as a validator, as a customer, as a tester, and sometimes as a developer, because we do not have the end user as a customer. We have the software factory customer." **Quality attributes (PA.09) - Interviewee 1**.

5.1.10 Case S10

The early-stage startup was founded in 2014. With a small team of 8 employees, they operated in the Healthtech segment. Their business model revolved around developing a mobile application. The interview, conducted with the CEO and CTO, took place in July 2017 and lasted 44 minutes and 25 seconds. As of that date, the company remained

Table 5.19 S10 Characterization.

CASE S10	
Items	**Description**
Founded	2014
Stage when Interviewed	Early stage
Size	Small
Employees	8
Business Model	Mobile Application
Segment	Healthtech
Interviewee	CEO
	CTO
Interview date	July 2017
Interview Time Spent	00:44:25
Actual Status	Active
Actual stage	Early stage

active and continued to be in its early stage of development. Table 5.19 presents S10 characterization.

S10 develops a software solution for three platforms: mobile for Android, mobile for iOS, and web application. An initial vision was to use native development for mobile platforms. Initially, the solution began to be developed by a development team in São Paulo, a high-performance team capable of efficiently facing challenges (CAT1.4). Subsequently, the transition from the software to the team in Salvador was carried out. S10's first challenge was to carry out this development team transition. The second challenge was understanding the development and security technologies implemented. Startup S10's team needed access to external experts who provide mentoring (CAT1.1). Furthermore, the background and experience of the founder/CEO positively impact business rules (CAT1.2). The team needed to learn and take on multiple areas to play different roles and handle different technologies (CAT1.3). The developers have experience in the software development process, bringing their experiences and knowledge to the product (CAT1.7). However, they do not know the technologies used to build the first version of the application. The small development team works in one location, facilitating communication and collaboration (CAT1.8). Despite the pressure and demands of the work, the team remains resilient and committed to delivering high-quality results (CAT1.10). Team collaboration and talent are critical drivers of S10's success.

> "In the beginning, they were outsourced. In outsourcing, there were around 3 to 4 people in development." **Team is the catalyst of development (PA.01) - Interviewee 1**.

S10 received the first version of the first software development team, a simple version aimed at B2C. Moreover, initially had to deal with the challenge of learning the new technologies used (CAT2.1). However, they still needed to receive documentation, which made this process more onerous. In addition, they decided to rewrite all the documentation (CAT2.2), that is, paying off this technical debt (CAT2.3). Nor have time for

Table 5.20 Summary of identified factors in S10's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.1 Access to external expertise (Mentorship) CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.10 Team under pressure
PA.02	CAT2.2 Re-engineering/Refactoring code CAT3.2 Lack of finance resources CAT2.3 Payment of accumulated technical debt CAT2.4 Better quality product
PA.03	CAT3.1 Lack of expertise CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.1 Delegate complexity to third-party applications CAT5.2 Simple and informal Workflow CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.1 Find the product/market fit quickly CAT6.2 Product efficiency starts after launch CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.2 Customer requests are documented CAT8.3 Meet customer's needs CAT8.4 Prototype validated by customers
PA.09	CAT9.1 User experience

this initial version to implement improvements (CAT2.4). They dedicated themselves to making minor adjustments and moving on to the implementation phase with potential customers. They opted for a customer segment pivot and started a new development cycle with new features for the application.

> "What arrived was the software and some of the documentation. Today, we are going to play everything that did not come new. We will document the requirements. We will make a use case, a test case" **Initial growth hinders**

> **performance (PA.02) - Interviewee 1**.

S10 needs more resources: expertise (CAT3.1), financial (CAT3.2), human (CAT3.3), and time (CAT3.4). The developers who became part of the team receiving the S10 application did not have Web development and security expertise. In addition, they depended on external resources from FAPESB and FINEP. The team was small. Moreover, their time and financial resources were running out; they needed to deploy the software. To subsequently offer to pay customers.

> "So not with software development; the challenge was in the technologies that some people needed to learn. For example, the web part was unknown, the iOS part was known, the Android part was known, and the communication and security part was unknown to us; we had to learn. So the challenges were technological rather than procedural." **Lack of resource (PA.03) - Interviewee 1**.

Startup S10 faces challenges related to technical debt in different areas. The architecture debt (CAT4.1) is a critical point, requiring improvements and adjustments to ensure the sustainability and scalability of the system. In addition, the documentation debt (CAT4.2) needs to be addressed to ensure clarity and understanding of the software, a point that the team felt most when they received the software to continue. Process debt (CAT4.3) is one of the team's main challenges, indicating that current processes need to be optimized and refined to improve efficiency and quality of work. Regarding requirements debt (CAT4.4) and test debt (CAT4.5 Test debt 3), while there are challenges, they are less critical than the other areas. However, it is essential to address these debts to ensure that requirements are clearly defined and testing is comprehensive enough to ensure product quality. The S10 team must be aware of these technical debts and take steps to pay them off over time, thus ensuring a robust system and a more efficient development process. This is because they are performing a pivot regarding customer segmentation, and the characteristics tend to be more complex.

> "The first was the challenge of the transition. This transition process was a challenge. Make this transition as smooth as possible. This was a big challenge for it to happen and not impact those already using it." **Accumulated technical debt (PA.04) - Interviewee 1**.

The team at startup S10 takes some exciting approaches to the way they work to speed up development. They can offload the complexity to third-party applications (CAT5.1), which allows them to focus on other essential tasks. In addition, they have a simple and informal workflow (CAT5.2), which can promote agility and collaboration among team members and is more suitable given the resource constraints it faces. However, the team has to work beyond regular hours to meet deadlines (CAT5.5). While it is understandable that sometimes extra effort is needed, it is essential to ensure a healthy balance.

The S10 development team did not use available technologies to develop the software because they could not choose once they received the first ready version (CAT5.3). Moreover, the last team that started the development chose to program natively (CAT5.4).

> "Through this schedule, we make it available on an internal network here of ours, and through subversion, people know exactly when they will be participating in the project. Furthermore, they participate in an internal project planning meeting, thus disclosing and maturing the project plan. We read the schedule and everything for the people here and follow up weekly." **Speed-up development (PA.05) - Interviewee 1**.

Startup S10 aims to quickly find the fit between the product and the market (CAT6.1). They strive to identify customer needs and adapt their products to meet those demands to achieve product-market fit. Also, product efficiency started after release (CAT6.2). They recognize that adjusting and improving the product based on customer feedback is necessary to ensure a high-quality product with better efficiency. However, S10 suffers due to numerous uncertain market conditions and an ever-changing environment. S10 considers that long-term planning is not feasible (CAT6.3). They take a more flexible and adaptable approach, focusing on short-term initiatives. In summary, S10 seeks to quickly find the fit between the product and the market by implementing the first version on the client. Moreover, move on to the development of the B2B version beforehand.

> "Now, with technology, it was taking this differentiated structure that existed and absorbing it that was the biggest challenge. However, we are already 100% of the projects conceived internally today. The other challenge now is getting these new features and running this project." **Evolutionary approach (PA.06) - Interviewee 1**.

Startup S10 is very focused on innovation, constantly looking for opportunities to innovate in different aspects of the business. They constantly seek innovation opportunities (CAT7.1) and value innovation in their products (CAT7.2). However, the emphasis goes on the importance they give to innovation within the scope of the business itself. S10 recognizes the need to innovate in its business practices to remain competitive and drive growth. They value innovation at the business level (CAT7.3), which demonstrates a strategic and forward-looking approach. In summary, S10 is committed to seeking innovation opportunities in its products and business practices, evidencing a proactive and growth-oriented posture.

> "We always submit some innovation projects, FAPESB, FINEP, and Secretary of Culture, and then within these projects, sometimes they require us to have a research technology institute behind them. We see that it is crucial because there are things we do not know, and then we put that part that we do not know for the research institute. For example, SEBRAI-TEC and SENAI-CIMATEC already had a partnership with us. Let me remind you of another one here, the Recôncavo Institute, where we are submitting a project now. They are starting with developing specific hardware, and we are starting with the software." **Innovation-driven development (PA.07) - Interviewee 1**.

Startup S10 strongly focuses on customer inclusion from conception to product development (CAT8.1). They value the active participation of customers in the improvement and implementation process, seeking to understand their needs, preferences, and feedback throughout the development. To ensure efficient follow-up of customer requests, S10 documents all (CAT8.2), allowing for better organization and analysis of demands, facilitating communication, and meeting customer needs. S10's main objective is to meet customers' needs effectively (CAT8.3). They strive to meet expectations and solve customer problems, prioritizing their satisfaction. Prototype validation is done by the customers themselves (CAT8.4). They value direct feedback from end users, allowing for tweaks and enhancements to the product before the final release.

> "We do the internal tests and then put them [the customer] to validate. In this validation, too, before putting the application into production, he will be there making the necessary criticisms, and we will make the necessary adjustments. So, in that way, he participates in the three phases. Thus, the customer is participating in the process." **Customer-driven development (PA.08) - Interviewee 1**.

The S10 is significantly concerned with user experience (CAT9.1 User experience). They recognize the importance of offering a product that is intuitive, easy to use, and provides a positive experience for its users, taking into account the product's visual aspects, usability, and navigability. They understand that a good user experience is critical to product success and market acceptance. In addition, S10 is attentive to user feedback and needs, constantly seeking to improve the experience.

> "We have a requirements analysis. We are going to make a requirements document. We are going to ask him to approve it. So, he is already participating in the process. Then, we are going to move on to the prototype. We are going to develop the prototype. It is going to be approved by them." **Quality attributes (PA.09) - Interviewee 1**.

5.1.11 Case S11

The company was established in 2016 and was in its early stages when the interview occurred. With a small team of 8 employees, they focused on the HRtech segment. Their business model revolved around developing embedded applications. The interview with the CTO in November 2017 lasted for 47 minutes and 5 seconds. As of the interview date, the company remained active and transitioned into the growth stage of development. Table 5.21 presents S11 characterization.

S11 is a small company formed by five partners, with the natural characteristic of being self-managed (CAT1.6). All initially worked on side jobs while building the startup. The idea came from the need that one of the companies had. With the passage of time and a collaborative environment for the development of the application, the first version of the innovative product of S11 was launched. Over time, the technical software development team grew from one (CTO) to three people. The CTO is the senior developer on the team

Table 5.21 S11 Characterization.

CASE S11	
Items	**Description**
Founded	2016
Stage when Interviewed	Early stage
Size	Small
Employees	8
Business Model	Embedded Application
Segment	HRtech
Interviewee	CTO
Interview date	November 2017
Interview Time Spent	00:47:05
Actual Status	Active
Actual stage	Growth stage

(CAT1.7). The exchange of knowledge by the team is passed on tacitly (CAT1.9). With such a small team, members have multiple roles (CAT1.3). Therefore, the CEO and CTO had a high impact on the development of the application (CAT1.2). Despite being small, this impact was positive because it was a productivity-oriented team (CAT1.5) with high performance (CAT1.4). However, it was still too small for the demand (CAT1.8). The software solution proposed by S11 is an application with a web and mobile version. The first version of the mobile application S11 needed to consult external experts (CAT1.1), and in the end, it outsourced the software development for this platform, as the team was under pressure (CAT1.10) to launch the product quickly.

> "From the beginning, I used the tools for the web part, so when we sometimes had a bottleneck for development, it was because the tool did not have it, and I had to develop some specific API. Then, I stopped to do it in Java and integrated it into the IDE. By doing that, I had a significant gain in performance agility in development because to modulate, every time I needed it, it was already there. All I had to do was make the call and pass the parameters. There was no great difficulty." **Team is the catalyst of development (PA.01) - Interviewee 1**.

We found evidence that S11 performed code refactoring (CAT2.2) to improve software quality (CAT2.4). The concern may have turned to something other than business because S11 comprises a multidisciplinary team with several experts (CAT2.1). However, possibly for the slightest reason, there was no technical debt payment, as S11 comprises five partners, and only one was from the technical area of software development (CAT2.3). When these S11 members have free time, they dedicate it to building new features or fixing bugs.

> "We made this first version and delivered it to the customer. But then it arrived at the client, and there were all the client's questions, there were things that we thought about doing, and we only see in practice, and then

Table 5.22 Summary of identified factors in S11's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.1 Access to external expertise (Mentorship) CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.5 Productivity oriented CAT1.6 Self-managed team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge CAT1.10 Team under pressure
PA.02	CAT2.2 Re-engineering/Refactoring code CAT2.4 Better quality product
PA.03	CAT3.1 Lack of expertise CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.1 Delegate complexity to third-party applications CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.1 Find the product/market fit quickly CAT6.2 Product efficiency starts after launch CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.2 Customer requests are documented CAT8.3 Meet customer's needs CAT8.4 Prototype validated by customers
PA.09	CAT9.1 User experience CAT9.4 Scalability

there were things that changed the entire structure of the software because we had not planned right." **Initial growth hinders performance (PA.02) - Interviewee 1**.

S11 faces challenges related to a need for more resources. They need to gain specialized knowledge regarding mobile application development (CAT3.1). In addition, S11 also deals with financial constraints (CAT3.2), which may affect your ability to invest in additional resources or innovative technologies. Another challenge faced by S11 is the need for more human resources (CAT3.3), which results in an overload of work for the existing team and impacts the productivity and efficiency of product development. In addition, S11 also faces time constraints (CAT3.4).

> “We need to invest in a technical team, specifically in development.” **Lack of resource (PA.03) - Interviewee 1**.

With the resource constraints that S11 is under, there is a massive backlog of technical issues: architecture (CAT4.1), documentation (CAT4.2), process (CAT4.3), requirements (CAT4.4), and testing (CAT4.5). However, as the team is small, high-performing, knowledgeable, and motivated, S11 meets the most emerging needs and speeds up development.

> “Moreover, despite being a technology I mastered, I still had much difficulty in some situations I thought I would not have. When we think about innovating processes or technology, we always arrive at a point or another that we do not master. There were several things to do. I had to stop, study and do.” **Accumulated technical debt (PA.04) - Interviewee 1**.

S11 uses all resources to try to speed up development, uses third-party applications (CAT5.1), has a simple workflow (CAT5.2), uses known technology (CAT5.3), makes use of well-integrated tools (CAT5.4), and works overtime (CAT5.5). In addition to these factors, it resorted to outsourcing to develop the mobile part.

> “In the web part, I used this idea. I am still using it today. It was essential to speed up the process because you managed to centralize all the development in a more complex platform for a single person in a short time.” **Speed-up development (PA.05) - Interviewee 1**.

S11 released the first version as quickly as possible to the first customers and started to validate the product in the market (CAT6.1) and to evolve the prototype (CAT6.2). Despite all the conditions of uncertainty (CAT6.3), such as technology challenges, and the need to speed up software development with few human resources, S11 has tried to achieve its objectives and serve its customers by providing a quality product. This success was also due to the client being error-tolerant in beta versions.

> “So when we sometimes had a bottleneck for development, it was because the tool did not have it, and I had to develop some specific API. Then I stopped to do it in Java and integrated it into the IDE.” **Evolutionary approach (PA.06) - Interviewee 1**.

S11 constantly seeks innovation (CAT7.1) through products (CAT7.2). S11 seeks to pilot the application through sound-out and zoom-in, a change in customer segmentation.

> "The initial idea was a broader business, but like that, as I think in several startups, we tend to change a lot, right? We start with an idea, but it changes when we evaluate the market." **Innovation-driven development (PA.07) - Interviewee 1**.

One of the advantages of S11 was collaborating with customers from the beginning of the conception of the first MVP (CAT8.1). S11 had a culture of listening to the customer (CAT8.2), making adjustments to meet their needs (CAT8.3), and participating in validating the application (CAT8.4).

> "Two others are partners in another company, which was where the initial idea came from, which had an outsourcing company, so they needed help with working hours control. So the initial idea was for them. Many ideas come like that. The initial idea was far from what we have today. They both own this company and also have a cleaning franchise. Moreover, another partner is also a computer student." **Customer-driven development (PA.08) - Interviewee 1**.

For S11, two quality attributes were important, initially the user experience (CAT9.1). Moreover, when the product was more mature, they invested in selling the application to more customers and scalability (CAT9.4).

> "Then he feels the user's difficulty. I think we can do this here in any other way. Moreover, that is interesting sometimes because, from my point of view, I always try to burden the server where the application is as little as possible because we always think on a large scale. Today, we have, I do not know, 200 companies using it. Then, in February, we can have 500. Moreover, we try to reduce costs and ensure the application works well. However, there is a powerful dynamic with the issue of user experience. Sometimes, to promote an excellent user experience, you have to use some resources that slow down the server. Then we always have this type of evaluation." **Quality attributes (PA.09) - Interviewee 1**.

5.1.12 Case S12

The early-stage startup was founded in 2015 with a micro-size team of 5 employees. They operated in the Fintech segment. Their business model centered around a web application. The interview, conducted with the CTO in December 2017, lasted for 45 minutes and 39 seconds. As of the interview date, the company remained active and was still in the early stage of development. Table 5.23 presents S12 characterization.

S12 is a web and mobile application company. The team is made up of a small group of 5 partners. Founder/CEO experience has a significant impact on operations (CAT1.2). The team is a cross-functional, broad-based team of advertising professionals and senior software developers (CAT1.3). However, when it is necessary to seek access to knowledge, they do so through specialized external guidance (CAT1.1). The company's focus is on

Table 5.23 S12 Characterization.

CASE 12	
Items	**Description**
Founded	2015
Stage when Interviewed	Early stage
Size	Micro
Employees	5
Business Model	Web Application
Segment	Fintech
Interviewee	CTO
Interview date	December 2017
Interview Time Spent	00:45:39
Actual Status	Active
Actual stage	Early stage

forming a high-performance team (CAT1.4). The company's culture is geared towards productivity (CAT1.5) and values team autonomy (CAT1.6). The most experienced developers have a direct influence on the development of projects (CAT1.7). The small development team works in a physical environment (CAT1.8). S12 shares knowledge tacitly (CAT1.9) to streamline its activities. Because they value teamwork and productivity, they speed up development so much that they do not mention being pressured to deliver. They even defined that the demands have a maximum delivery limit of one month or 20 working days.

> "We send the approval, but the site is not online for the guy to see it online, so he does not have to go there to see it; we put a bar on something, then the person only has access to that there; he looks at the cell phone I liked it, or I did not like it, then we send it by e-mail, it was validated, ok ok." **Team is the catalyst of development (PA.01) - Interviewee 1**.

There was no evidence regarding shifting focus to business issues (CAT2.1), code reengineering or refactoring (CAT2.2), or paying off technical debt (CAT2.3). The focus of S12, judging by the evidence found, is essential to achieve customer satisfaction by offering a product with the best possible quality within the agreed delivery time (CAT2.4).

> "The last one, taking it as a reference, was generally it. We put it in the store, and the guy showed it to the people caring for it. They looked, and it worked. They said everything was okay, we moved forward, the application was released, and it is already on the market. You have to have, I know that at the same time, you have to have a whole delivery schedule, management of this and that, or a repository of that. If the business is not so agile, it does not work." **Initial growth hinders performance (PA.02) - Interviewee 1**.

S12 faces challenges related to the lack of expertise in the team (CAT3.1), but this does not impact the team's technical capabilities. When this occurs and is necessary,

Table 5.24 Summary of identified factors in S12's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.1 Access to external expertise (Mentorship) CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.5 Productivity oriented CAT1.6 Self-managed team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge
PA.02	CAT2.4 Better quality product
PA.03	CAT3.1 Lack of expertise CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.1 Delegate complexity to third-party applications CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.1 Find the product/market fit quickly CAT6.2 Product efficiency starts after launch CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.2 Customer requests are documented CAT8.3 Meet customer's needs CAT8.4 Prototype validated by customers
PA.09	CAT9.1 User experience

they use outsourcing. Although they did not mention suffering from a lack of resources (CAT3.1), we found evidence regarding participation in programs such as SEBRAE-TEC. In addition, the company also faces a shortage of human resources (CAT3.3); the team is overloaded because there is a large volume of projects and the team is small. Time restriction is also a present factor (CAT3.4); it occurs because a small team forms it, but we have not recorded any experience that this has affected the ability to deliver products within the stipulated deadlines. These challenges require creativity and practical strategies to better manage the time and scarce resources that S12 has available.

> "Look at the people who founded it and are still there today. Everyone is a partner. There are five partners." **Lack of resource (PA.03) - Interviewee 1**.

The agility of startup S12 comes with a cost: generating technical debt in different areas. Architecture debt (CAT4.1) indicates the need for documentation, revision, and system improvement. The documentation debt (CAT4.2) highlights records of the decisions taken, creating an essential history over time for the company and the understanding and collaboration of the team, especially when receiving new members to help mainly in the technical part. Process debt (CAT4.3) can impede having a clear overview of the company's activity flows, thus preventing awareness of optimization and improving efficiency and quality of work. Requirements debt (CAT4.4) masks requirements not implemented, poorly implemented, defined, or understood, which can only be perceived when the project is already underway, potentially harming the client and the startup. Finally, the testing debt (CAT4.5) points to improvements in test strategy and execution to ensure software quality. These areas of technical debt will require resolution efforts to improve software development's stability, maintainability, and efficiency.

> "As everyone is very close, everyone cannot freeze, generate a document, show that there was only one error, print a business to send an email for the guy to see, or register a business we never see. We already registered a client for a project that did not work out." **Accumulated technical debt (PA.04) - Interviewee 1**.

S12 demonstrates a strategic approach to managing technical complexity and optimizing workflow efficiency. The team has delegated the complexity to third-party applications (CAT5.1), allowing them to leverage existing solutions and focus on other activities. They adopt a simple and informal workflow (CAT5.2), and streamlining processes promotes effective collaboration between team members. Using standard and known technologies (CAT5.3) guarantees speed, stability, and compatibility, minimizing the risks of adopting unknown technologies. S12 uses integrated and simple tools (CAT5.4), allowing speed in application development. Only evidence indicates that they work overtime when necessary to meet deadlines (CAT5.5). Overall, S12 demonstrates confidence in making decisions, performing software development tasks, and delivering successful results.

> "As everyone is a partner, we work a little more, but that is that. It is 8 hours a day, and we get there at 9, so sometimes we get there early when it is more stuffy, and we leave late, or sometimes quieter. So it is 40 hours, but it could become 50 or 35 because we already had models, it was tested, the home office. So, sometimes on a Friday, we work from the home office, the division was each day of the week was one, ah it did not work, let us go one day of the week everyone." **Speed-up development (PA.05) - Interviewee 1**.

S12 takes an evolutionary approach, releases the MVP within the shortest possible time, tests in the development environment quickly, and releases the product for customer validation (CAT6.1). Once the product has been launched, it proceeds through a transparent and quick dialogue with the customer to generate new versions with adjustment requests (CAT6.2). S12 is aware that software development and technologies are constantly evolving, and taking too long can put them in the context of launching an obsolete product on the market (CAT6.3).

> "The last one, generally taking it as a reference, was it. We put it in the store, and the guy showed it to the people caring for it. They looked, and it worked. They said everything was okay, we moved forward, the application was released, and it is already on the market. You have to have, I know that at the same time, you have to have a whole delivery schedule, management of this and that, or a repository of that. If the business is not so agile, it does not work." **Evolutionary approach (PA.06) - Interviewee 1**.

S12 seeks innovation opportunities (CAT7.1) through software products (CAT7.2).

> "We participated in SEBRAE-Tec, a SEBRAE project that they ended up paying. I do not know if you know, but they paid 70% of the project, and the applicant/client paid 30%. However, Sebrae is a rigid company. It is almost a government thing. One of the managers said: "Look, I understand that a company like you is a very agile business, but not here. We have audits." So, this is very different, the startup market, their development, not that this means that there are standards, some strategies, or some ways to take care of that there, but many times we call for agility a little. Otherwise, it does not work." **Innovation-driven development (PA.07) - Interviewee 1**.

S12 strongly emphasizes customer-centric development practices, ensuring customers are actively involved in product development (CAT8.1). They value documenting customer requests (CAT8.2) and strive to meet their customers' needs (CAT8.3). By incorporating customer feedback and preferences, they adapt their product to serve their customer better. Furthermore, S12 understands the importance of validating its product through prototypes with real customers (CAT8.4), allowing it to collect valuable information, identify areas for improvement, and ensure that the final product meets customer expectations by actively involving customers in the development and validation process. Overall, S12's customer-centric approach reflects its dedication to understanding its customers, documenting their requirements, and delivering a validated prototype that aligns with customer expectations.

> "Back to our first chat. If this has been involved in a requirements survey and the guy wants to do something completely different or even not just the requirement to produce the business or even the layout, I do not know. I passed the layout to him, and he said, I approve, but I do not want that white color. I want everything yellow, that is a cost. This way, we do it at

> no cost and is also very malleable. So that is why it is interesting for you to carry out the survey and have the client's approval because anything he says or wants to imply a lot is man-hours." **Customer-driven development (PA.08) - Interviewee 1**.

User experience is the main quality attribute (CAT9.1) for S12.

> "The application is released and is already on the market. You have to have it. I know that at the same time, you have to have a whole delivery schedule, management of this and that, or a repository of that. If the business is not as agile and validated by the customer, it does not work." **Quality attributes (PA.09) - Interviewee 1**.

5.1.13 Case S13

Some points are interesting to be pointed out before the analysis of Startup S13. Some peculiarities differentiate game development and software development. Starting with the nature of the product. Game development is focused on creating engaging, interactive experiences for players, while software development focuses on functional and utilitarian solutions to meet users' needs. Game development involves a multidisciplinary team, including designers, artists, programmers, historians, psychologists, musicians, screenwriters, and writers, who create visuals, narrative, game mechanics, and the overall game experience. The team usually focuses on programming, interface design, and testing in software development. Another point is that game development follows a more iterative and prototype-oriented cycle, where developers create playable game versions and receive feedback from players to tweak and improve the design. The game goes through several refinements until it is ready to be released on the market. A more linear cycle in software development is generally followed, with defined phases, such as requirements analysis, design, implementation, and testing. However, documentation needs to be improved, as in the case of startups.

Regarding complexity and optimization, game development often deals with advanced 3D graphics, complex physics, and performance demands, requiring optimization and efficient use of hardware resources. 2D games, in turn, do not have this performance and resource optimization challenge. However, they have a less attractive graphic limitation, leaving artists' work more complex to make the game experience more attractive for players. Software development generally focuses more on functionality, efficiency, and usability. While software development provides users with an efficient and intuitive experience, game development seeks to create an emotional and engaging experience, considering factors such as immersion, challenge, and fun.

The company was founded in 2010 and was still in its early stages during the interview. With a micro-sized team of 6 employees, they specialized in game development within the Fintech segment. Their business model revolved around creating engaging games. The interview with the CEO and CTO in July 2017 lasted for 57 minutes and 14 seconds. As of the interview date, the company remained active and was still in the early stage of development. Table 5.25 presents S13 characterization.

Table 5.25 S13 Characterization.

CASE S13	
Items	**Description**
Founded	2010
Stage when Interviewed	Early stage
Size	Micro
Employees	6
Business Model	Game Development
Segment	Fintech
Interviewee	CEO/CTO
Interview date	July 2017
Interview Time Spent	00:57:14
Actual Status	Active
Actual stage	Early stage

S13 is a small gaming team focusing on developing games for human training. The multidisciplinary, high-performance S13 team (CAT1.4, CAT1.5), with professionals from various areas: software developers, designers, artists, screenwriters, historians, and psychologists. When game development involves expertise the team does not have, they access external experts (CAT1.1). The CEO is a very experienced person in the field of game development and human development (CAT1.2). As a small team, they accumulate functions such as the CEO, who is involved in the game's development and is the business and project manager (CAT1.3). The team is self-managed (CAT1.6). Formed by people with experience in game development (CAT1.7). The team works in a room in the Parque Tecnológico da Bahia (CAT1.8). Knowledge is transmitted tacitly and through a document called GSS, an acronym in English for the acronym Game Design Document, which is the game design document (CAT1.9). The development of a game is a complex activity that involves many details, so to achieve a good experience and user engagement, the team is under pressure to meet the agreed delivery deadlines (CAT1.10).

> "So, for me, the improvement is precisely this. Smaller, project-focused teams." **Team is the catalyst of development (PA.01) - Interviewee 1**.

S13 focuses on offering its customers an engaging and quality experience (CAT2.4).

> "In some aspects of game theory, in flow theory, in the psychological approach, an adaptation has already been developed: e-games flux, which is when people enter a flow within a game experience. And then, it helps both in evaluating the game and the development because we can try to reach those items that lead the person to immersion. They help even more than usability studies in the case of games." **Initial growth hinders performance (PA.02) - Interviewee 1**.

S13, we face significant challenges. One is to find software developers with expertise in games (CAT3.1). This lack of knowledge affects the ability to speed up game development.

Table 5.26 Summary of identified factors in S13's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.1 Access to external expertise (Mentorship) CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.5 Productivity oriented CAT1.6 Self-managed team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge CAT1.10 Team under pressure
PA.02	CAT2.4 Better quality product
PA.03	CAT3.1 Lack of expertise CAT3.2 Lack of finance resources CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.1 Delegate complexity to third-party applications CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.1 Find the product/market fit quickly CAT6.2 Product efficiency starts after launch CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.3 Meet customer's needs CAT8.4 Prototype validated by customers
PA.09	CAT9.1 User experience

The most critical challenge for S13 is more financial resources (CAT3.2). This restriction limits the ability to invest in game refinements, which increase the level of engagement and player experience. The financial constraint also affects the need for more human resources (CAT3.3), making it difficult to form a qualified and sufficient team to deal with the demands of developing a game. It is even more so for games that want to bring innovation with intrinsic characteristics. Subject to this scarcity, it is difficult for S13 to meet deadlines (CAT3.4) that require a fast pace of work, which negatively affects the game's quality.

> "The challenges in developing games are like this: you cannot take a person just because he knows how to develop, he knows how to program, not necessarily he will know how to make games." **Lack of resource (PA.03) - Interviewee 1**.

S13 accumulates technical debt in 5 areas: architecture, documentation, process, requirements, and game testing. In game development, software architecture (CAT4.1) refers to the structure and organization of code, and components that make up the game, such as images, characters, environments, and level design. Game development involves graphics creation, programming, level design, and game mechanics. Integrating these elements harmoniously and functionally is a challenge. Documenting a game includes creating the following documentation: concept document, which describes the overall vision of the game, including the core idea, key mechanics, target audience, and goals; game design document or Game Design Document (GDD) that details the game mechanics, rules, progression, levels, characters, story, user interface and other elements of the game; art document, contains information about the visual style, color palette, characters, backgrounds, animations and visual effects of the game; sound document, which specifies the sound effects, soundtrack, dialogues and any other audio present in the game; script document, describes the game's story, events, and progression, including dialogue, cut scenes, and key moments; requirements document that lists the functional and non-functional requirements of the game, such as supported platforms, hardware capabilities, multiplayer support, among others; timeline document, which details key project steps, including milestones, deadlines, tasks, and team responsibilities; testing document, which specifies the test cases, test scenarios, and quality requirements to be checked during the game testing process. S13 built GDD only (CAT4.2, CAT4.3, CAT4.4, CAT4.5).

> "It is complex. Because, like, this is natural in software development: making an initial prototype and from there to the final product will be testing, testing, testing, adjusting, testing to the public... the game is an interaction, so if you do not make it suitable for your target audience is useless. However, beyond that, we have to do too. Another test we have to do is whether the game achieves the results we want with our human development approach. So, it is like an educational game. It is entertainment. However, he has goals beyond entertainment, so he is considered innovative and financed by development agencies. Nevertheless, to be able to test this, I have to have a prototype that is what the agencies finance and test if they have objectives beyond entertainment, which are human development objectives." **Accumulated technical debt (PA.04) - Interviewee 1**.

S13 has a simple and informal workflow (CAT5.2) and tries to adapt Scrum by making records in the online tool Trello. Game development involves a team knowledge engine (CAT5.3) called Unity (CAT5.4) that generates game versions for mobile, PC, and Xbox platforms. Overtime is common to try to reach established deadlines (CAT5.5).

> "Yeah, we needed another six months. Because the project was for 18 months, I remember that we asked for an extension of the deadline for the development itself and also because of the delays in the installment from faces itself. It is always difficult to estimate. We had estimated a bigger game, but then it got smaller so I could deliver it in time." **Speed-up development (PA.05) - Interviewee 1**.

Due to scarce financial resources, S13 must validate the product-market fit (CAT6.1). The game's efficiency comes after successive iterations of tests and refinements with players that are part of the target audience we want to reach (CAT6.2). The conditions of uncertainty in the game's development already make long-term planning not feasible (CAT6.3).

> "We expect this, like, then the game will have a dialogue, an exploration, and a combat part, so we do the minimum of each one and then test it. Test the flow. Test each of the layers to see if that minimum will work. Furthermore, to add things that will only come out in the test." **Evolutionary approach (PA.06) - Interviewee 1**.

S13 is a highly focused team looking for innovation opportunities (CAT7.1). They are constantly exploring new ideas and approaches to improve their products. They value innovation in the product itself, constantly seeking improvements and new game applications (CAT7.2). The team strives to develop games that are unique and highly engaging in innovative ways.

> "Look, for example, now we want to finalize this phase that we have been doing since October, just perfecting. Moreover, we want to close it to move on to the other phases with some other funding. Publish as a playable demo on our website or Steam itself, which is about games. We are trying to negotiate with the Xbox people to put it there. So, it will take at least another six months." **Innovation-driven development (PA.07) - Interviewee 1**.

S13 values the inclusion of customers from the beginning of development (CAT8.1). The team is committed to satisfying and engaging players with an engaging gaming experience (CAT8.3). This includes actively listening to player feedback and refining to ensure player interest and engagement. The team also recognizes the importance of quickly validating their prototypes with players (CAT8.4).

> "We have a questionnaire that we apply right after, but also the whole team is watching. Because sometimes you play, there is a bug we did not see. Because then, the purpose of the test is different from functionality. For that, we get the maximum internally. We play all the time and remove the bugs that we have. Then, when people come to test it, they do unexpected things, then one bug or another appears, then we keep watching, taking notes, and also a functionality that needed to be clarified which they should run or where it

> should go. However, we use note that is difficult to learn, but after that, they answer a questionnaire that goes into the flow theory, which is what we are most interested in seeing, the flow and the interest of the game. Then, they respond and give suggestions for improvement. It is a form that we made in Google Docs." **Customer-driven development (PA.08) - Interviewee 1**.

S13 emphasizes the importance of user experience (CAT9.1) in game development. The game's genres and layers will have been defined during the concept phase, including dialogue, combat, and exploration. Then comes the game designer, responsible for building the GDD (Game Design Document). The GDD serves as a roadmap for development, where the game rules are created, and the roadmap is developed. The graphic assets and the didactic part of the game are also produced. Thus, S13 highlights the importance of planning and documenting, at least minimally, all these steps to ensure a satisfactory gaming experience for the player.

> "Look, the game. It has the concept phase, the genres, and the layers the game will have. In the dialogue phase, the combat layers, there is the exploration layer, which the person will explore. So, you define the concept of the game, what the game will have, what layers it will have, and what experience the user will have. After the concept, what we do is go to someone from the design, so in this phase of the game design, we build the GDD, we define how this is all going to happen, what mechanics are going to be developed, what arts are going to be necessary to start from the concept, which characters, how many objects, what you will need, then all this goes into the GDD, these specifications. And then, after the GDD, you move on to development, which we call rules development. In this development, we will develop the script, which will develop the assets, graphics, and the didactic part." **Quality attributes (PA.09) - Interviewee 1**.

5.1.14 Case S14

The early-stage startup was founded in 2015 with a small, nimble team of 6 employees. They specialize in game development within the Fintech segment. Their business model revolved around creating exciting and entertaining games. The interview, conducted with the CEO and CTO in July 2017, lasted for 57 minutes and 14 seconds. As of the interview date, the company remained active and was still in the early stage of development, striving to make a significant impact in the gaming industry. Table 5.27 presents S14 characterization.

S14 is a software development startup. Formed by a small team of 5 members. All are partners. This fact significantly impacts (CAT1.2), positively influencing the startup. When specific external expertise is required (CAT1.1), S14 outsources. S14 has a multidisciplinary team (CAT1.3) with high performance (CAT1.4). Productivity orientation (CAT1.5) and the team's self-management ability (CAT1.6) are also crucial to achieving the successful results that the team has been achieving. In addition, the presence of

Table 5.27 S14 Characterization.

CASE S14	
Items	**Description**
Founded	2015
Stage when Interviewed	Early stage
Size	Micro
Employees	6
Business Model	Game Development
Segment	Fintech
Interviewee	CEO/CTO
Interview date	July 2017
Interview Time Spent	00:57:14
Actual Status	Active
Actual stage	Early stage

senior professionals (CAT1.7), small and co-located (CAT1.8), based on tacit knowledge (CAT1.9), teamwork, and who can work under pressure (CAT1. 10) contribute to an environment conducive to successful game development.

> "Yes, we always work as a team. We seek to listen to the people here, also the ideas. Because the more ideas favorable to the project, the better. " **Team is the development catalyst (PA.01) - Interviewee 1**.

In the first product, S14's main challenge was the post-production stage, especially publicity and promotion. The startup needed to find effective ways to show players that the game existed and was available in the store. Now, S14 has taken a different approach. Since the pre-production phase, it has actively worked on building a community of players for the game so that by the time they reach post-production and release, the community is already engaged and ready to buy the game (CAT2.1). S14 released a prototype within three to four months, presented the game at important events, and received great acceptance thanks to the excellent quality and positive reviews from the public (CAT2.4).

> "We launched the prototype three to four months ago. We presented at a fair in São Paulo, the BGS-Brasil Games Show, one of Latin America's biggest games fairs. We performed there, at Comic-Con, also with the same prototype, in the same year. Despite being a prototype that was made quickly, it was well accepted, but it was of excellent quality, and the criticism was good out there." **Initial growth hinders performance (PA.02) - Interviewee 1**.

The most challenging feature of a game development company is having resources with expertise in game development, especially in Brazil (CAT3.1). When asked about the startup's biggest challenge, S14 said it was the lack of financial resources (CAT3.2). This lack of financial resources affects human resources (CAT3.3). The following quote is evidence of the shortage of human resources. The team's pressure is caused by itself, given the need to find financial resources (CAT3.4).

Table 5.28 Summary of identified factors in S14's Points of Analysis (PA).

#	Description
Proposition P1	Human factors
PA.01	CAT1.1 Access to external expertise (Mentorship) CAT1.2 High-impact of Founder/CEO background CAT1.3 Multi-role and full-stack engineers CAT1.4 High-performance team CAT1.5 Productivity oriented CAT1.6 Self-managed team CAT1.7 Senior developers influence the development CAT1.8 Small and co-located development team CAT1.9 Tacit knowledge CAT1.10 Team under pressure
PA.02	CAT2.1 Focus change to business concerns CAT2.4 Better quality product
PA.03	CAT3.1 Lack of expertise CAT3.2 Lack of finance resources CAT3.3 Lack of human resources CAT3.4 Time shortage
Proposition P2	Technical factors
PA.04	CAT4.1 Architecture debt CAT4.2 Documentation debt CAT4.3 Process debt CAT4.4 Requirements debt CAT4.5 Test debt
PA.05	CAT5.1 Delegate complexity to third-party applications CAT5.2 Simple and informal Workflow CAT5.3 Use of standard/known technology CAT5.4 Use of well-integrated and simple tools CAT5.5 Work overtime to meet deadlines 4
PA.06	CAT6.1 Find the product/market fit quickly CAT6.2 Product efficiency starts after launch CAT6.3 Uncertain conditions make long-term planning non-viable
Proposition P3	Organizational factors
PA.07	CAT7.1 Search for innovation opportunities CAT7.2 Product innovation
PA.08	CAT8.1 Customer inclusion in development since conception CAT8.3 Meet customer's needs CAT8.4 Prototype validated by customers
PA.09	CAT9.1 User experience

"We are still at a stage looking for investment for its development itself, and we do not have a date and a stipulated deadline, just in the head, but there is no way to put this within a program since we are still at a stage we are raising funds for final development. from that moment on, when we get the necessary resources, we will structure the team's products even better to be

> able to accelerate even more." **Lack of resource (PA.03) - Interviewee 1**.

The accumulation of technical debt in S14 is significant. There are architecture debts (CAT4.1), documents (CAT4.2), processes (CAT4.3), requirements (CAT4.4), and tests (CAT4.5). Everyone can impact development speed and product quality. However, S14 overcomes this situation through a small, high-performance team that shares the same workspace, where communication occurs quickly and clearly, below the technical split evidence of requirements documents.

> "No, we did not document the requirements. We thought about the requirement and already imagined it to be, as it was a simple game, it was more devices, GDD, it is just GDD, but since this was a straightforward game, we did not even do GDD in this game. I only had an idea in my head; we developed it thinking about simpler devices, from the simplest to the most advanced, and we launched it as quickly as possible." **Accumulated technical debt (PA.04) - Interviewee 1**.

S14 makes use of third-party applications to accelerate development (CAT5.1) and has simple workflow (CAT5.2) tools that the team is aware of (CAT5.3). Although complex, the tool makes S14's job less onerous (CAT5.4). S14 works overtime to have a quality product that the gaming community engages in (CAT5.5) and can acquire paying customers.

> "Yes, we always do. There is the engine's community that we work with, which is Onew. It is a robust community, and it has a virtual store that allows other professionals from outside to develop codes, access them, and place them in this store so that other studies can buy and accelerate development." **Speed-up development (PA.05) - Interviewee 1**.

In the development process, S14 seeks to quickly find the fit between the product and the market (CAT6.1). After launch, product efficiency starts to improve (CAT6.2). However, uncertain conditions make long-term planning impracticable (CAT6.3). S14's flexible and adaptable approach allows it to respond promptly to market changes and ensure constant evolution for the client.

Startup S14 seeks innovation (CAT7.1) through game development (CAT7.2).

> "Yes, we always work as a team. We seek to listen to the people here, also the ideas. Because the more ideas favorable to the project, the better." **Innovation-driven development (PA.07) - Interviewee 1**.

Another major challenge is to engage a community of customers in the game (CAT8.1). Only through this inclusion they will be able to test and refine the game prototype (CAT8.3, CAT8.4).

> "The great challenge of the first product was post-production. The publishing part. Not putting the product in the store itself but publicizing it. Our most significant difficulty was showing people that the product exists and is inside the store. Now, this product needs to be publicized." **Customer-driven development (PA.08) - Interviewee 1**.

The main focus of S14 is to provide an exceptional user experience (CAT9.1). Player satisfaction and engagement are critical to the success of our game. Investing time and effort in understanding the target audience for which the game is being created, their preferences, and expectations can bring excellence in the user experience, which is an important differential.

> "It was well accepted, despite being a prototype that was made quickly, but it was of excellent quality, and the criticism was good out there." **Quality attributes (PA.09) - Interviewee 1**.

5.2 CROSS-CASE ANALYSIS

This section presents the cross-analysis of data from fourteen organizations based on research propositions. Discoveries made in the field are also listed at the end of this section. We used three criteria to characterize the points of the analysis presented in the Tables (Table 5.29, Table 5.30, and Table 5.31):

- **N (Unidentified):** means that the analysis point was not identified at any level of the organization.
- **P (partially identified):** means that the analysis point was partially identified in the organization.
- **T (fully identified):** means that the startup thoroughly identified the analysis point.

To assess whether a proposition is not confirmed, we define that if all analysis points are identified in the startup, it means that the organization presents this factor that indicates the context of startup development.

Table 5.29 presents the cross-analysis for proposition P1 on human factors. Therefore, proposition P1 is confirmed, indicating that the team is a catalyst for software development.

5.3 CHAPTER CONCLUSION

Startups construct high-impact software products for the market, contributing to the local, regional, national, or global economy. Software development practices are the core of companies' daily activities, especially in early-stage startups. We presented a single embedded case study with four units of analysis in an academic context. We build the Academic Startup Model (ASM) based on the knowledge obtained through interviews

Table 5.29 Analysis of Proposition 1.

Proposition P1 There are human factors that influence software development in startups														
PA	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**
PA.01	T	T	P	P	P	P	P	P	P	P	T	P	T	T
PA.02	P	P	P	P	P	P	T	P	P	P	P	P	P	P
PA.03	T	T	P	T	P	P	P	P	P	T	T	P	T	T

PA.01: Team is the catalyst of development
PA.02: Initial growth hinders performance
PA.03: Lack of Resource

Table 5.30 Analysis of Proposition 2.

Proposition P2 There are technical factors that influence software development in startups														
PA	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**
PA.04	T	T	T	T	T	P	T	T	T	T	T	T	T	T
PA.05	T	T	P	P	T	T	P	T	P	P	T	T	P	T
PA.06	T	P	T	P	T	P	P	T	P	T	T	T	T	T

PA.04: Accumulated technical debt
PA.05: Speed-up development
PA.06: Evolutionary approach

Table 5.31 Analysis of Proposition 3.

Proposition P3 There are organizational factors that influence software development in startups														
PA	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**
PA.07	P	P	T	P	P	P	P	P	T	P	P	P	P	P
PA.08	T	P	P	P	T	T	P	P	T	T	T	T	P	P
PA.09	P	P	P	P	P	P	T	T	P	P	P	P	P	P

PA.07: Innovation-driven development
PA.08: Customer-driven development
PA.09: Quality attributes

with practitioners of Software Engineering (SE) practices in startups. We captured how software startups execute their SE activities in the ASM and the Greenfield Startup Model (GSM) (GIARDINO et al., 2016) and discussed the differences and similarities. The results confirm a similar conclusion obtained by Giardino et al. (2016).

Startups focus on increasing the development speed through prototyping and team capabilities once they are the development catalyst. The most urgent priority for startups is to validate the product as soon as possible to verify that the market is interested in the proposed business model, product, or software service. At this stage, startups relinquish formal project management, design, analysis, documentation, and testing. Professionals use the prototyping approach and well-integrated tools that help speed up software development. In addition, we identified that startup faces a challenging environment, given the customer pressure to release the product as soon as possible to the market. All units of analyses reported one month to deliver the first minimum viable product. In this context, even though the most lightweight, agile methodology did not meet their needs. Startups need to pay their accumulated technical debt to expand operations, which hinders their performance. Startups need to improve their state of practice, investing in Continuous Integration (CI) and Continuous Deployment (CD) practices and more automated and integrated tools that reduce the overhead of SE and communication activities.

Chapter 6

DISCUSSION AND RECOMMENDATIONS

The research sought to answer the question: "How do early-stage startups develop software?" Qualitative research was conducted to answer this question in a multiple-case study with fourteen early-stage software startups. In this chapter, we answer research questions and provide recommendations for early-stage startups.

6.1 CONTRIBUTION STATEMENT

Considering the objectives defined in this thesis described in Section 1.2, we obtained the following results and contributions.

Research Goal: exploring software engineering practices over the software development process in the context of startups.

6.1.1 Empirical Results

Therefore, we answer this thesis's research questions through the four sub-questions below.

RQ1: ***How do early-stage startups develop software?***

RQ1.1: *What is the contextual framework for software development in startups?* We aimed to investigate the context of the software development process in startups with research question 1. In this research, the factors that influence the development of software in startups (section 6.1.1).

Startup Software Engineering Practices Model. Producing and acquiring data from software engineering (SE) activities is nearly simple. The challenge is to increase knowledge and insight and obtain meaning from the data representing something real. Theories offer common conceptual

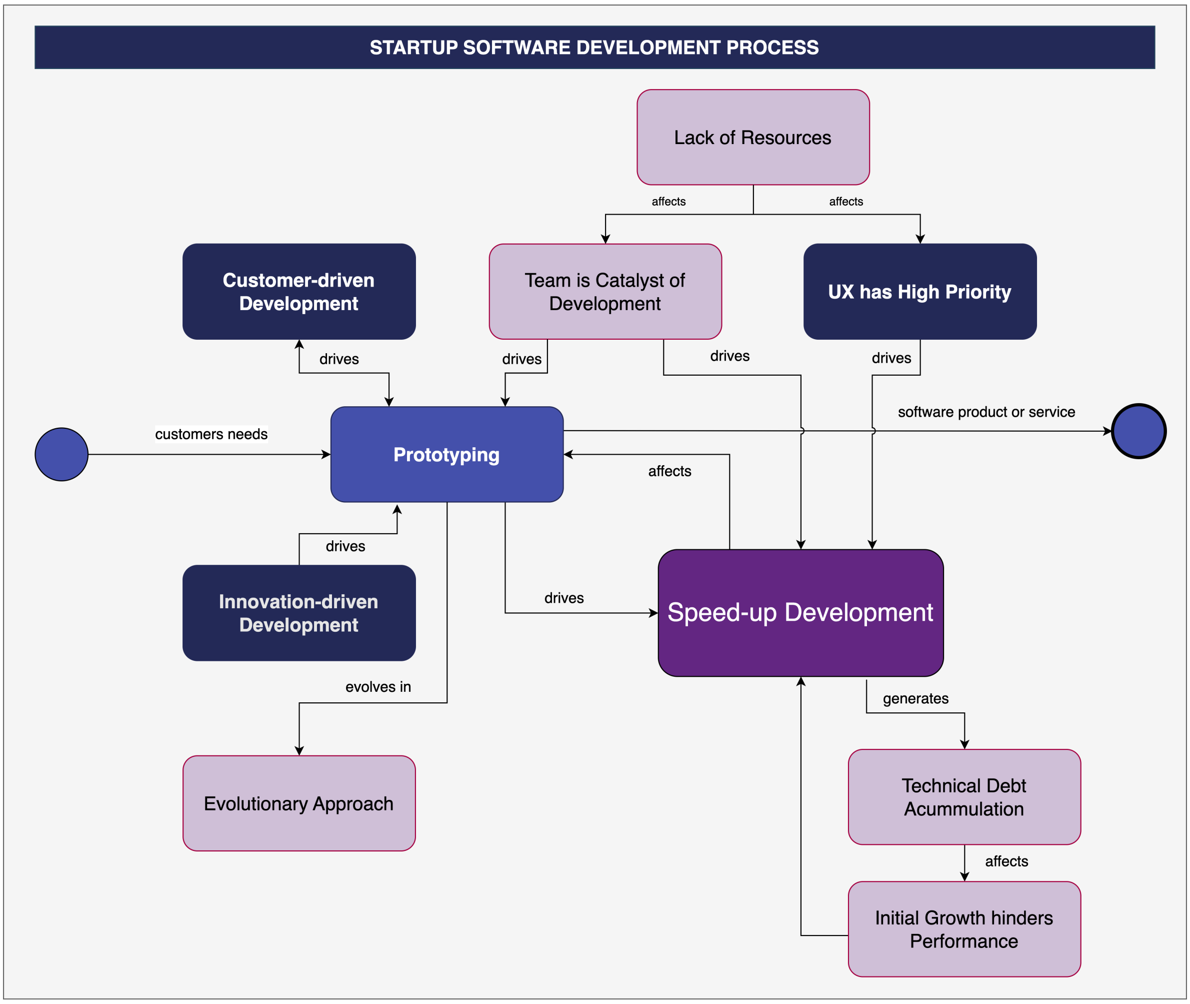


Figure 6.1 Software Engineering Practices in Startups.

frameworks that allow the organization and structuring of facts and knowledge concisely and precisely, thus facilitating the communication of ideas and knowledge. In this study, we will call the framework a model, as we cannot guarantee that we have reached theoretical saturation with the 14 interviews carried out with startups in their initial stages.

We choose Sjøberg's framework (SJØBERG et al., 2008) because it considers the Software Engineering aspects when creating the theory from scratch. They proposed a template with four archetype concepts: (i) actor, (ii) technology, (iii) activity, and (iv) software system. A typical Software Engineering situation is that an *actor* applies *technologies* to perform certain *activities* (practices) on an *software system* (existing or planned). They presented a set of steps to construct theories for the software engineering domain. The theory is divided into four parts: the constructs, which are the essential elements of the theory; propositions, which are how constructs

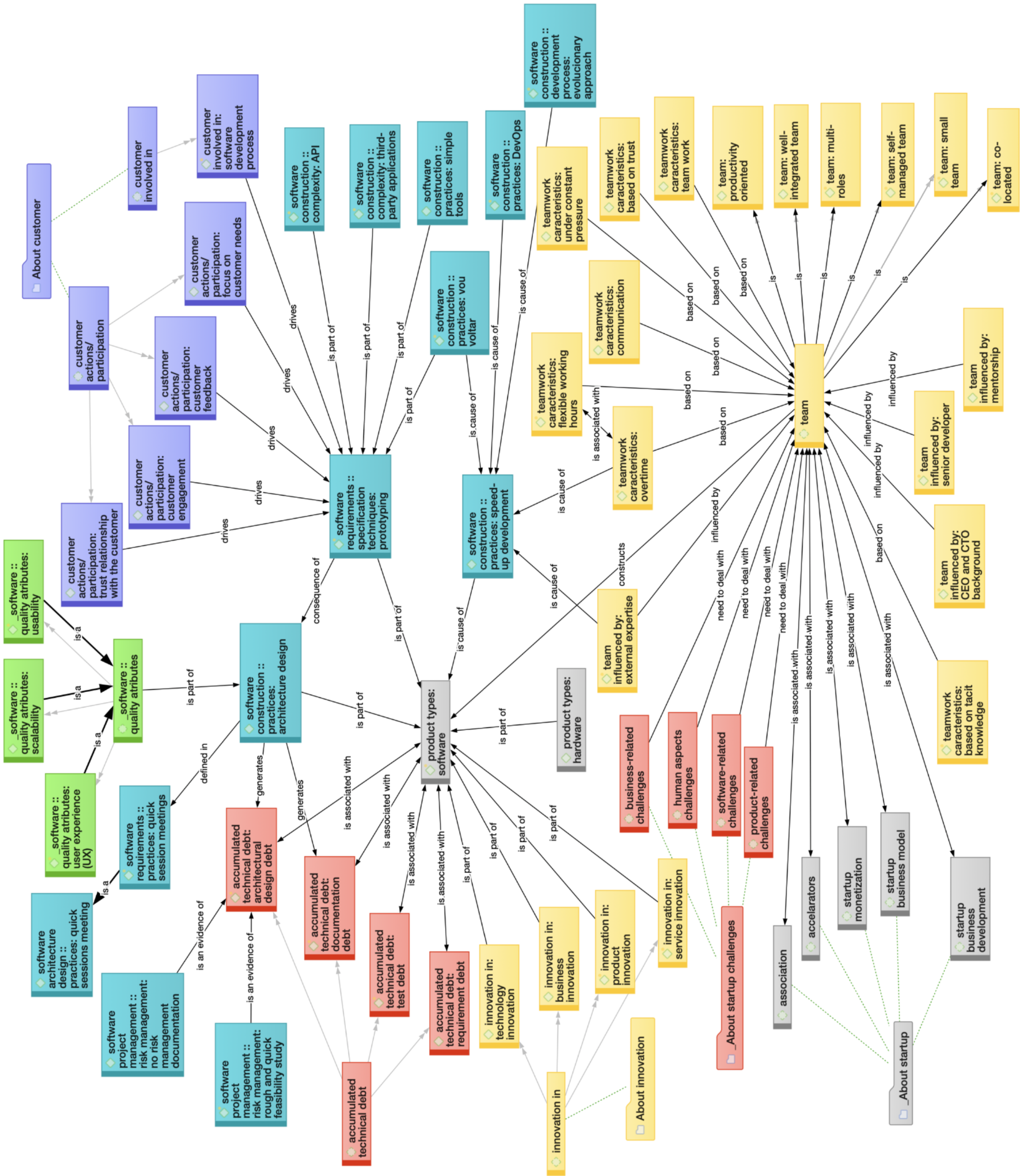


Figure 6.2 Software Engineering Practices in Startups Model in Atlas.ti View.

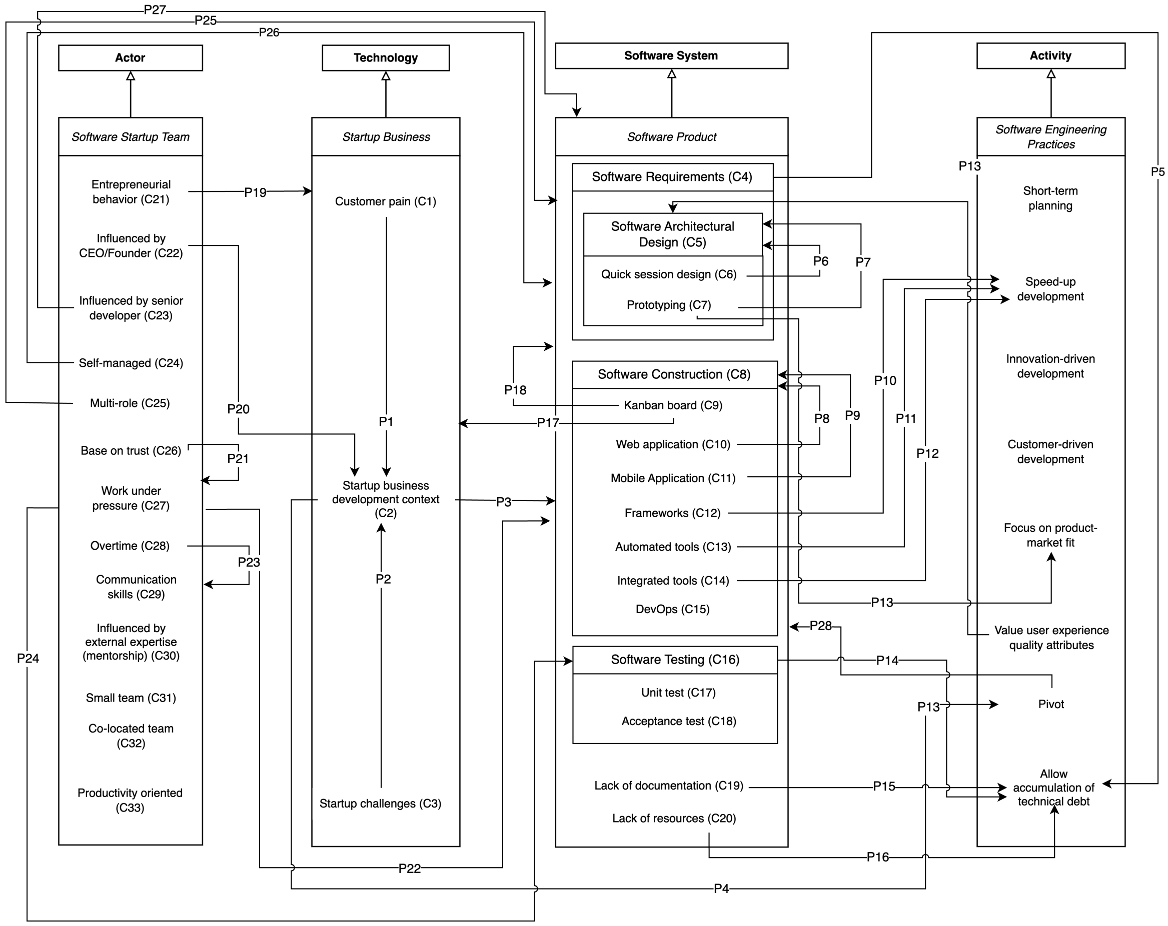


Figure 6.3 Software Engineering Practices in Startups Model using Sjorberg framework (SJØBERG et al., 2008).

are related; explanations, which narrate why the propositions are specified; and scope, which is the universe of discourse in which the theory is applicable. It provides the "why" of the theory, which is probably the most challenging. This core step is to provide explicit assumptions and logical justifications for the constructs and propositions of the theory. Sjøberg also describes criteria to evaluate theories. Testability is one criterion that indicates "the degree to which a theory is constructed such that empirical refutation is possible."

For readability reasons, the framework proposed by Sjoberg brings a framework that helps bring knowledge more comprehensively since it considers the context of software engineering. Some works already awarded are part of the literature, such as (SOUSA et al., 2018), bringing evidence that the model can help transfer knowledge more straightforwardly.

We use Sjøberg's structure to represent and describe the constructions, propositions, and explanations. The study replication is feasible given that the protocol, the interview process, and the questions are available in the Appendix E. We discuss the propositions and their constructs next. We also present explanations for propositions studies' findings and explanations that comprise findings.

Table 6.1 Elements

Arch-types Class	Sub-classes
Actor	Startup Software Team
Technology	Software Startup Business
Activity	Software Engineering Practices
Software Systems	Software Product

The list of practices found by (DANDE et al., 2014) is not comprehensive enough. Startups have a lot more working practices. The most important practices regarding the opportunity alpha are related to the customer relationship and product focus. Startups usually use advisors or mentors to guide their work in the early stages. This is one of the major stakeholder groups, along with the customers and investors. Software startups' most used practices regarding requirements are focused on prioritizing the product features that are the most needed now and focusing on the concept. These are both tested and validated. Flat and self-organizing teams are the most suitable structure for software startups. Startups tend to tailor common agile methods to form their working practices. Current alphas don't consider business aspects well enough to cover software startups' work.

The software development process is a complex activity that involves risks, and the search for best practices that can lead to functional and efficient software is constant. Software engineering is an ever-evolving discipline; new approaches and techniques are constantly being developed and improved. Lean and agile thinking is an example of an approach widely adopted by software development teams in recent years. These approaches emphasize collaboration, flexibility, continuous delivery of customer value, and adapting to changing requirements. They effectively reduce costs, increase software efficiency and quality, and improve customer satisfaction. Other innovations such as cloud computing, artificial intelligence, DevOps, and microservices architectures have transformed how software is developed, deployed, and maintained. As software engineering is constantly evolving, its professionals must be able to learn and are open to new ideas and practices to ensure that they can produce high-quality software and meet the constantly changing market needs.

Software development teams worldwide widely adopt Agile and lean methodologies because they balance fast software delivery needs with quality, cus-

Table 6.2 Constructs

#	Constructs
C1	Customer pain
C2	Startup business development context
C3	Startup challenges
C4	Software Requirements
C5	Software architecture design
C6	Quick session design
C7	Prototyping
C8	Software Construction
C9	Kanban board
C10	Web application
C11	Mobile application
C12	Frameworks
C13	Automated tools
C14	Integrated tools
C15	DevOps
C16	Software Testing
C17	Unit tests
C18	Acceptance test
C19	Lack of documentation
C20	Lack of resources
C21	Entrepreneurial behavior
C22	Influenced by CEO/Founder
C23	Influenced by senior developer
C24	Self-management team
C25	Multi-role
C26	Based on trust
C27	Work under pressure
C28	Overtime
C29	Communication skills
C30	Influenced by external expertise (mentorship)
C31	Small team
C32	Co-located team
C19	Productivity oriented

Table 6.3 Propositions

#	Propositions
P1	Business development is based on customer pain.
P2	The startup's business development is to solve the customer's pain, is influenced by the background of the Founder/CEO, and faces challenges.
P3	Business development involves the development of a software product.
P4	The development of the business model changes (pivot).
P5	Requirements elicitation through a quick meeting with the customer through a prototype generates technical debt.
P6	Software architecture is designed in a quick meeting.
P7	Software architecture is designed through a prototype.
P8	Building the software involves a web application.
P9	The construction of the software involves a mobile application.
P10	Software acceleration occurs through the use of frameworks and APIs.
P11	Software acceleration occurs through the use of tools that enable automation.
P12	Software acceleration occurs through the use of built-in tools.
P13	The startup seeks to launch the prototype as quickly as possible to validate the product on the market.
P14	Software testing is performed ad hoc, accumulating technical debt.
P15	Building software is prioritized over writing documentation.
P16	The scarcity of resources favors the accumulation of technical debt.
P17	Business development activities are managed through the Kanban board.
P18	Software development activities are managed through the Kanban board.
P19	It takes entrepreneurial behavior to run a startup.
P20	Startup software development is to solve customer pain, is influenced by Founder/CEO background, and faces challenges.
P21	Teamwork is based on trust.
P22	The team work under constant pressure.
P23	Staff work overtime.
P24	Software testing is performed by the software development team itself.
P25	Team members have multiple responsibilities.
P26	The team shares responsibility and manages software development activities.
P27	The senior developer influence software development.
P28	The software product undergoes several changes (pivot) throughout the development of the business.

tomer satisfaction, and risk reduction. These methodologies emphasize collaboration, communication, and continual development process improvement, allowing teams to adapt quickly to changes and deliver customer value in short development cycles.

Each practice is treated as a separate, autonomous unit that teams can adopt based on their specific needs and circumstances. This allows teams to leverage the most effective practices relevant to their needs without being constrained by a specific method. Furthermore, this approach encourages experimental evaluation and validation of practices to ensure they are effective and reliable.

Startups develop a business plan, assemble a team, develop the product, and scale up. Unfortunately, most startups fail in the early stage because of following a traditional product development path and lose focus. Problems related to adoption might happen on the way: a common phenomenon in software markets. The literature offers the context and use of Software Engineering (SE) practices, and the existing evidence on SE practices in startups suggests findings that show a set of goals, challenges, practices, and maturity models. However, each startup faces unique challenges when developing software in ecosystems with different maturity levels. All of these factors influence the startup's success. It is necessary to show new early-stage software startups the development context and challenges while adopting SE practices and propose guidelines that help them make better choices. So, early-stage startups can evaluate the best practices for their context to maximize the possibility of success. We conducted empirical studies and proposed a set of recommendations serving as a basis for early-stage software startups.

- Focus on delivering a Minimum Viable Product (MVP),
- quickly aimed at innovation, whether with an innovative product, service, or business model, as a way to enter the market and remain competitive,
- inclusion of the customer in the software development process, and
- from the requirements definition phase through prototyping, functional prototype approval, product/service validation, and delivery.

The main category is the acceleration of development dealing with financial and human resources scarcity affecting the development team and the quality of the product, which becomes a low priority compared to the delivery of a functional product. However, they unanimously prioritize the user experience. Other relevant quality aspects are scalability, security, availability, and fault tolerance. Although the team is the catalyst for development, it has little or no time for training, and it is common to have little experience in technologies, making them use technologies based on personal knowledge instead of choosing the most appropriate technologies to solve the problem. With core and flexible development processes, they rely on tacit knowledge rather than formal documentation to hit time-to-market and meet customer requests. While running to meet time-to-market and customer satisfaction, they accumulate technical debt, compounded by minimal development processes, non-formal specifications, and little or no automated testing. They specify requirements in a quick meeting with the client through mock-ups, wireframes, and prototypes. Thus, they adopt an evolutionary approach. This accumulation of technical debt impacts development performance when the startup begins the growth phase.

RQ1.2: *Which software engineering practices are most pertinent to early-stage startups?*The software development process begins with the customer's need, modeled together with the customer/user through paper prototypes, mock-ups, wireframes, or even the construction of application screens without functionality. This solution requires innovation so that the startup can differentiate itself in the market and increase its customer base and revenue. The development team takes an evolutionary approach to the product or service, defines a set of activities to be carried out to build the application, and cycles of quick meetings are scheduled for customer validation. The team uses its knowledge to choose development tools and technologies to accelerate development and validate the product with the customer. Therefore, the team decides the most appropriate methodologies for implementing the solution they are trying to solve. Furthermore, given this need for acceleration to validate the product, the development team uses software quality attributes since they deal with the extreme scarcity of resources, financial, human, and material. It manages to launch a minimum viable product. Still, it accumulates technical debt, and when it starts to grow, this acceleration decreases, either because it needs to pay the debt or to include more team members, and the lack of formal documentation compromises this activity. Figure 6.4 shows software engineering practices startups use to support the software development process.

RQ1.3: *What quality attributes do startups prioritize in software development?* The quality aspects most cited by startups were the user experience (UX), mainly the ease of use and the elegance of the user interface. All startups mentioned looking for a ready-to-scale business model (referring to the product). Other relevant quality attributes are scalability, security, availability, and fault tolerance. Figure 6.5 shows the quality attributes network.

RQ1.4: *Which software development supporting tools are commonly used by startups?*The tools most cited by startups are organized by area of SE. Using tools can bring benefits, from financial to management, for the startup. They are essential because startups need much technical support to speed

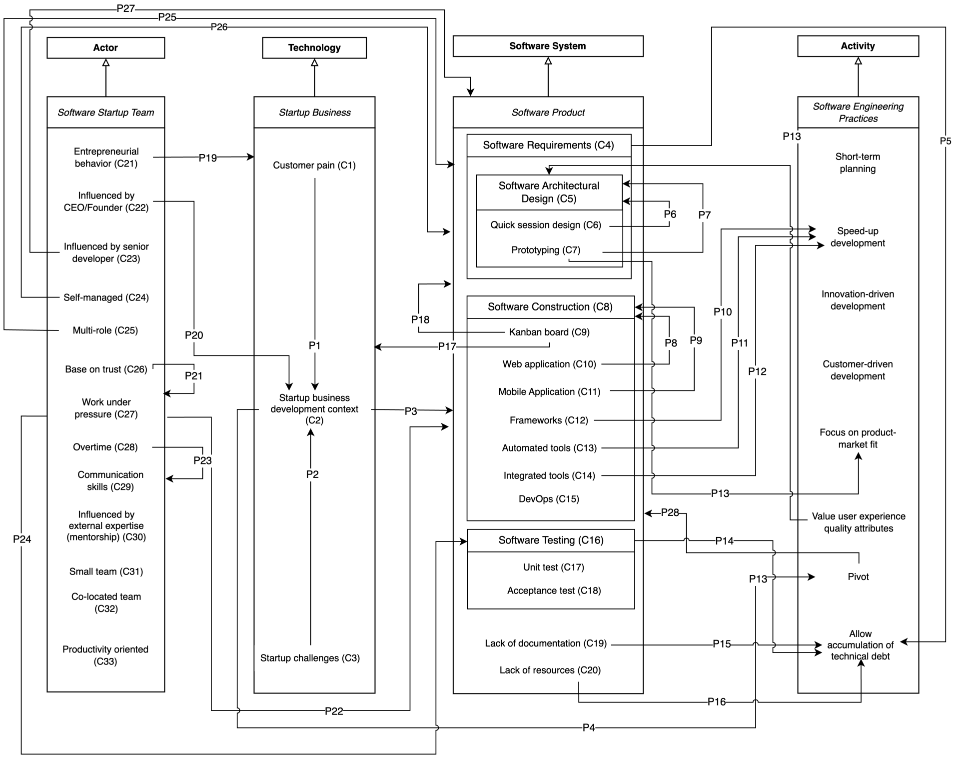


Figure 6.4 Software Engineering Practices.

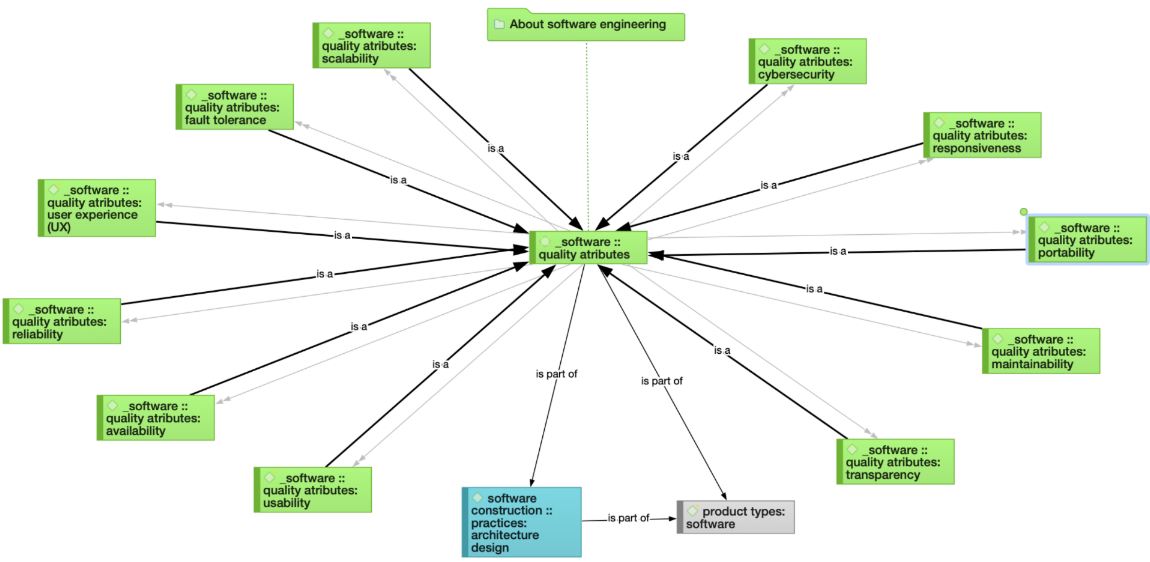


Figure 6.5 Quality Attributes.

up their daily tasks and reduce the effort to develop their products. In this way, your startup can guarantee an increase in overall productivity, accelerated growth, and low costs. Other benefits include automating essential but still manual tasks and saving on hiring and buying more tools.

6.1.2 Theoretical Implications

The implications related to theory in the context of software development in startups are diverse. Researchers should go deeper in understanding the unique context of startups, considering the specific characteristics of this environment, such as time pressure, the need for innovation, and the quest to accelerate development. Researchers can focus on investigating and understanding the challenges faced by startups in software development, such as managing technical debt, integrating tools, and dealing with the human aspects of the team. They can analyze and identify best practices that startups can adopt to overcome challenges and succeed in software development, including studying innovative approaches, agile methods, integrated tools, and effective customer engagement strategies. Existing theories about software development may need to be adapted or expanded to suit the specific context of startups. Researchers can explore how established theories apply or must be adjusted to address the unique challenges and opportunities startups face. There are several research gaps in the context of software development in startups. Researchers can focus on less explored areas, such as the intersection between innovation and software development, managing technical debt in startups, or the unique challenges of small, cross-functional teams. In summary, the implications for researchers in the theory of software development in startups involve understanding the context, identifying best practices, adapting existing theories, and contributing to practice. By exploring these areas, researchers can provide valuable insights to drive startup innovation and success in software development.

6.1.3 Practical Implications

The implications for startup software development professionals are significant and may influence their daily practices. First, professionals looking to find a startup need to understand the unique context in which startups are embedded, including time constraints, the quest for innovation, and the need to accelerate development. The accelerated pace of startups requires professionals to be agile and efficient in their activities, always looking for ways to optimize the software development process, reduce bureaucracy, and eliminate unnecessary activities. Automating repetitive tasks and adopting integrated and simple tools can help in this regard. Professionals need to be aware of the challenges related to technical debt and look for ways to manage it effectively. It primarily involves avoiding the generation of technical debt and, secondly, prioritizing its resolution and ensuring that the work is done sustainably to avoid the accumulation of problems in the future.

The importance of customer involvement in the development process, so practitioners should look for opportunities to collect customer feedback, understand their needs and expectations, and incorporate this information into development work will help create

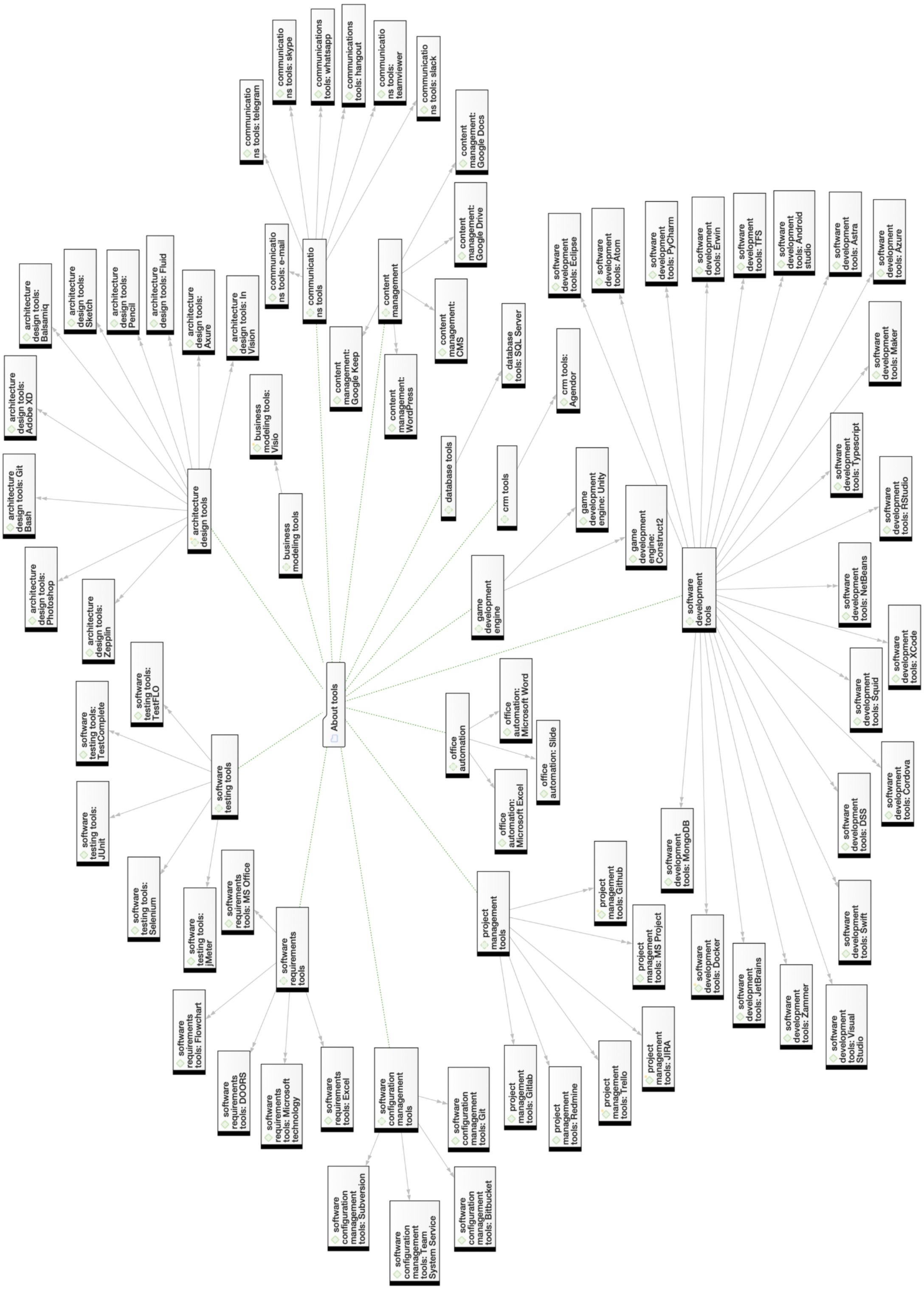


Figure 6.6 Startups Software Development Tools.

products that meet market demands and ensure customer satisfaction. They also need to be open to the search for innovation and the adoption of new approaches and technologies, be willing to explore creative solutions, think outside the box, and constantly seek ways to improve and differentiate their products and services. In startups, teamwork and collaboration are fundamental. The members involved must be willing to collaborate, share knowledge and experiences, and work together to achieve team goals through effective communication and building trusting relationships. Given the dynamic environment of startups, professionals must be open to continuous learning, seeking opportunities to improve their technical skills, market knowledge, and understanding of current trends.

6.2 RECOMMENDATIONS

This section presents how the recommendations were structured, and the recommendations are technical, organizational, and related to human aspects. The Appendix E presents the analysis networks for each recommendation.

The recommendations are based on the knowledge gained from the case study during the exploratory phase. The recommendations aim to offer practices related to technical, organizational, and human aspects to increase the chances of success of a startup. The Recommendations will be systematized through the following points:

- Recommendation: Provides a summary of the recommendation.
- Recommendations description: presents the recommendation description based on the recommendation analysis network.
- Observed behavior: refers to how people act or behave in a given situation, which can be directly observed through direct observation, records, or witness reports.

Based on the data analysis notes, 17 recommendations were proposed to support early-stage software startups Table 6.4. The recommendation means the intersection of behaviors identified in the startups investigated in this work. They are situations that must be considered to minimize the risks of failure in the stage of software startups. Therefore, these recommendations can be used by professionals and startups starting to define their software development strategies. Table 6.4 presents the description of the recommendations. The terms highlighted in the table refer to the factors identified as essential to speed up software development.

6.2.1 Human-Aspects Recommendations

6.2.1.1 HA-REC.01

Recommendation: Make a mindset shift.

Recommendations description: A change in mindset (HA-REC.01) is essential to drive the formation of a high-performance team (CAT1.4) and pursue innovation

Table 6.4 Recommendations.

#	Description
Human-Aspects Recommendations	
HA-REC.01	Make a mindset shift.
HA-REC.02	Learn while doing.
HA-REC.03	Accept failures, fail fast, and learn from them.
HA-REC.04	Promote diversity.
HA-REC.05	Make organizational culture change.
HA-REC.06	Have a team with new skill sets.
HA-REC.07	Trust each other.
Technical Recommendations	
T-REC.01	Define a minimum viable process.
T-REC.02	Spend minimal time and effort on defining the best possible software architecture.
T-REC.03	Define software development and coding policies and patterns with the team.
T-REC.04	Include customers as soon and frequently as possible in the software development.
T-REC.05	Define a minimum software testing process.
T-REC.06	Make [it] simple.
T-REC.07	Make [it] integrated.
T-REC.08	Make [it] automated.
T-REC.09	Find the toolset that helps your team to accelerate software development.
Organizational Recommendations	
T-ORG.01	Define software development and coding policies and patterns with the team.
T-ORG.02	Include customers as soon and frequently as possible in the software development.

opportunities. Adopting a proactive and open approach to new ideas and perspectives is necessary. This involves stepping out of your comfort zone, challenging established beliefs, and exploring new paths.

By adopting a high-performance mindset (CAT1.4), the team strives to overcome challenges, constantly strive for excellence, and achieve exceptional results, including setting ambitious goals, developing relevant skills and competencies, and working collaboratively.

In addition, looking for innovation opportunities (CAT7.1) is fundamental to boosting creativity and generating new ideas, exploring different perspectives, questioning the status quo, and being open to innovative solutions. The team must constantly look for improvement opportunities, identify market gaps, and develop unique and disruptive solutions.

By adopting this change-and-opportunity mindset (HA-REC.01), the team is in a favorable position to achieve high performance and constantly seek innovation. It is an ongoing learning, adapting, and growing process driven by the right mindset and the relentless pursuit of new possibilities.

Observed behavior: We observed that startups that promote a change in mentality build an environment conducive to developing a highly efficient and productive team. Team members are more willing to take on challenges, strive for excellence in their work, and collaborate effectively. In addition, being focused on innovation, the team needs to be more open to exploring new ideas and perspectives, which is only possible in a team authorized to think differently and whose ideas are considered. This behavior is fertile for identifying opportunities for improvement and generating innovative solutions to the challenges faced by the startup.

The change in mentality also allows the team to be better prepared to deal with challenges and adversity, an intrinsic characteristic of startups. This posture helps them to be more resilient, open to learning from mistakes instead of being blocked and without perspectives, and willing to seek creative solutions in the face of obstacles.

The mindset of change promotes a culture of continuous learning, encouraging the search for constant learning and development. Startups with this characteristic had teams that were more likely to seek new knowledge, acquire relevant skills and apply best practices, leading to individual and collective growth.

Whoever had this mindset changed culture, had a team more prone to seek opportunities, and was more prepared to identify gaps in the market and develop unique and disruptive solutions, resulting in a range of differentiated products or services, increasing competitiveness and the chance of scaling the startup's business.

6.2.1.2 HA-REC.02

Recommendation: Learn while doing.

Recommendations description: A high-performing team (CAT1.2) looking for innovation opportunities (CAT7.1) needs to adopt a "learn while doing" approach (HA-REC.02). This approach emphasizes the value of learning through experience and continuous improvement as integral parts of the team development process.

By encouraging a startup mindset (HA-REC.02), team members actively engage in hands-on experiences, allowing them to gain practical knowledge, hone their skills, and develop a deep understanding of industry and market dynamics. This active learning approach (HA-REC.02) fosters a high-performance culture within the team (CAT1.2) as individuals constantly seek to improve their skills, collaborate effectively, and achieve exceptional results.

In addition, the "learn while doing" approach (HA-REC.02) also drives the search for innovation opportunities (CAT7.1). Through continual experimentation, the team gains valuable insights, learns from successes and failures, and adapts its strategies accordingly. This iterative learning process allows the team to identify gaps, challenge existing assumptions, and explore innovative solutions that drive the company's growth and competitive advantage.

Startups combine the principles of continuous learning and hands-on experience, making the team more adaptable, resilient, and responsive to changes in the business environment. Usually, those who cannot make this change die. They become adept at seizing opportunities, generating creative ideas, and staying ahead of the market. However, when this dynamic mindset and learning approach is part of the team's culture, it contributes as much to building a high-performing team as it does to uncover opportunities for innovation within the organization.

Observed behavior: Startups that have in their culture the characteristic of "learning while doing" (HA-REC.02) achieve high performance (CAT1.2) and find it easy to search for innovation opportunities (CAT7.1). Team members are actively involved in the practice, enabling them to gain industry-relevant knowledge and understand market dynamics, strengthening their knowledge base and decision-making.

Team members honed their skills and competencies, becoming more proficient in their areas of expertise, contributing to exceptional performance and better quality results. Members strove to constantly improve their skills, collaborate effectively, and achieve better results. Team members felt a motivating work environment, making them more productive.

Startups with this ability were more adaptable and resilient to changes in the business environment, being open to experimenting with new approaches and learning from successes and failures rather than being paralyzed by them. Instead, these startups adapted to pivot the product or business quickly and remain competitive.

Startups that enabled this learning approach had members who questioned existing assumptions and explored creative solutions seeking improvement and differentiation. This posture is what drove innovation and strengthened its position in the market.

6.2.1.3 HA-REC.03

Recommendation: Accept failures, fail fast, and learn from them.

Recommendations description: By adopting the "accept failure and fail quickly" approach (HA-REC.03), the high-performing startup (CAT1.4) looking for innovation opportunities (CAT7.1) is willing to break barriers, try new ideas, and act quickly.

This approach emphasizes the importance of not being afraid to fail, as failure is seen as an opportunity for learning and growth (HA-REC.03). The team is willing to take calculated risks and make quick decisions, even if it means making mistakes.

By "breaking things" and "failing quickly", the team can quickly identify what works and what does not, allowing them to adapt and adjust their approaches based on the insights gained from the failures. A team with these experiences can quickly apply (CAT1.4) lessons learned to improve and move forward.

The "fail fast" and "learn fast" mindset drives innovation and the discovery of new solutions (CAT7.1). The team constantly explores new ideas, prototyping, testing, and validating their concepts. They move quickly in developing products and services (CAT1.4), looking to improve their offer and offer value to customers constantly.

Adopting this approach makes the startup agile, flexible, and adaptable. They embrace the continuous learning mindset and are always ready to adapt and evolve based on results. Feedback received strengthens the culture of innovation (CAT7.1) and drives the team towards its high-performance objective (CAT1.4).

Observed behavior: Startups that "accept failure and fail quickly" (HA-REC.03) have the potential to achieve high performance (CAT1.4) and do not tire in the pursuit of innovation opportunities (CAT7.1).

Startups that have embraced failure as a learning opportunity have become agile at identifying what works and what does not, allowing them to hone their approaches and strategies more efficiently, accelerating their learning, and developing their skills.

When startups that were allowed to fail learned from mistakes, and the team was even for failure for the simple fact of understanding that it is part of the learning process, they quickly adapted to market changes, customer demands, or emerging opportunities and pivoted their business models. Thus, they became agile and flexible, able to overcome the competition, stood out in the market, and, mainly, for finding a scalable business.

Failure is also part of the innovation process. Teams encouraged to think creatively and explore new ideas felt motivated to experiment with innovative solutions, increasing the chances of finding unique ways to solve problems and meet customer needs. , despite the possibility of failure.

These startups gain a significant competitive advantage when they can innovate quickly, launch better products and services, and adapt to market changes agilely, putting them ahead of the competition and gaining a prominent position in the market.

6.2.1.4 HA-REC.04

Recommendation: Have a team with new skill sets.

Recommendations description: A startup that wants a high-performing team (CAT1.4) must look for members with new skills (HA-REC.04), mainly openness to continuous learning, versatility, adaptability, creativity, and willingness to change and innovate. Encouraging the search for continuous learning is a fuel to develop new skills, expand knowledge, and keep up to date with trends and technologies, fueling the personal and professional growth of the team and the startup.

These new skill sets (HA-REC.04) strengthen the startup's ability to tackle various challenges and projects. Team members gain new perspectives, knowledge, and experiences, enriching the collective and driving innovation. A team with these skills (HA-REC.04) becomes universal and adaptable, able to assume different roles and responsibilities depending on the project's needs, allowing quick adjustments (CAT1.4) to market changes and customer demands.

With these skills (HA-REC.04), team members bring new ideas and approaches to combine their knowledge in unique ways, generating innovative and creative solutions, driving the startup's differentiation in the market and the ability to find opportunities for innovation.

Observed behavior: By seeking members with new skills, the startup strengthens its time, allowing them to face different challenges and projects with new perspectives, knowledge, and experiences; the team becomes more capable of dealing with complex demands and offering high-quality results. The new skills brought by team members make the startup more adaptable and assume different roles and responsibilities according to the project's needs, facilitating rapid adaptation to market changes and customer demands provides agility and flexibility to deal with new situations and challenges.

The new skills promote new ideas and approaches, allowing startups to combine their knowledge uniquely, followed by innovative and creative solutions. This capacity for innovation drives the differentiation of the startup in the market and makes it more able to identify and take advantage of innovation opportunities.

Encouraging the search for continuous learning strengthens the team's personal and professional development, allowing them to expand their knowledge, acquire new skills, and keep up with relevant trends and technologies, benefiting the team and contributing to the growth and success of the startup.

6.2.1.5 HA-REC.05

Recommendation: Trust each other.

Recommendations description: Establishing mutual trust (HA-REC.05) constructs a high-performing team (CAT1.4).

Trust (HA-REC.05) is the basis for effective teamwork. When teams trust each other, they feel safe sharing ideas, collaborating, and taking risks, resulting in greater cohesion and synergy and more efficient and productive work. People feel valued, respected, and appreciated when they know and trust them. This deposited trust conveys a sense of security and recognition of their abilities and capabilities.

Through trust (HA-REC.05), open and honest communication is achieved within the team, where everyone feels valued and heard, comfortable expressing their opinions, sharing constructive feedback, and appropriately resolving conflicts. Healthily facilitates the flow of information and safely for a positive work environment. Thus, they support each other and recognize and value individual contributions, creating a thriving climate of collaboration and support where everyone feels encouraged and motivated to reach out to the collective.

Trust (HA-REC.05) also brings greater responsibility, as it is through it that the team understands that its actions and decisions have an impact and are valued, allowing it to feel more secure in facing challenges and assuming new responsibilities.

Also, when someone knows that you trust them, they are more inclined to trust others and to establish positive and constructive relationships. Mutual trust creates a collaborative environment, facilitates communication, and makes people feel motivated and responsible. Emotional well-being and the ability to play a meaningful and effective role on the team.

In this context, a team makes effective decisions more quickly, streamlines the decision-making process, and helps to avoid delays or impasses (CAT1.4). The team is more prepared to face challenges and solve problems quickly to find creative and efficient solutions. They work together, sharing knowledge and experience, which leads to faster and more effective problem resolution. Mutual Trust (HA-REC.05) promotes coordination and collaboration among team members so that they work seamlessly, oversee responsibilities, and support each other to achieve common goals.

Observed behavior: In the startups we identified, there was mutual trust between team members; the more critical the trust, the more collaborative and encouraging the work environment. People in this environment felt free to share ideas, collaborate and work together more efficiently; communication was open and honest, promoting transparent and accurate exchange of information. Trusting and supportive, the team works together to achieve common goals, shares responsibilities, and is willing to help each other. This confidence encourages the free expression of ideas, encouraging creativity, and the search for innovative solutions. Team members feel comfortable approaching problems proactively and collaboratively when trust is present. They rely on each other's skills and knowledge, facilitating the search for practical solutions and the agile resolution of problems. Furthermore, trust creates a healthy work environment where everyone feels valued, respected, and supported.

On the other hand, we also observed a startup where there was a lack of trust. We do not observe information and knowledge sharing in this environment, which undermines collaboration and teamwork. Communication could have been better, creating difficulties in solving problems and a possible cause of the failure of its first innovative product. There was tension in the work environment was tense, and this factor affected the productivity and overall performance of the team. We identified that the CTO deposited the certainty of agility and meeting the client's needs through well-integrated tools that helped automate some tasks. This team also identified that the activities were not self-managed but distributed by the CTO, evidencing an inevitable centralization, naturally generating a slowdown in software development.

Conflict resolution suffers when people feel that they are not trusted because it inhibits the expression of ideas, killing creative and innovative thinking. Team members may feel insecure about sharing their perspectives and taking risks, limiting the search for creative solutions. People feel devalued and discouraged, resulting in low engagement, lack of motivation, and even looking for other job opportunities. Therefore, establishing and cultivating mutual trust among team members is critical to strengthening teamwork, promoting effective communication, generating mutual respect and support, facilitating decision-making, and enabling agile problem resolution. Trust contributes to a healthy, motivating work environment conducive to innovation and high performance.

6.2.2 Technical Recommendations

6.2.2.1 T-REC.01

Recommendation: Define a minimum viable process.

Recommendations description: Defining a Minimum Viable Process (T-REC.01) allows the team to address process technique debt issues (CAT4.3) and pursue a simple and informal workflow approach (CAT5.2). This approach involves applying minimal and efficient project management practices focused on delivering value and reducing excess bureaucracy.

Defining a minimum viable process (T-REC.01) aims to avoid the accumulation of process debts (CAT4.3), which are the negative results of inefficient or overly complex practices. These process debts (CAT4.3) can lead to low productivity, delays, and additional costs. By adopting a minimum viable process (T-REC.01), the team simplifies and streamlines its activities, eliminating unnecessary processes and focusing on essential tasks.

A simple and informal workflow approach (CAT4.3) allows the team to work more agile and flexibly by adopting lean project management practices, prioritizing effective communication, collaboration, and continuous feedback. Thus, the team quickly adapts to changes, makes more informed decisions, and maintains a steady pace of work.

Applying Minimum Project Management Practices (T-REC.01) helps the team focus on essential practices such as a clear definition of objectives, clear assignment of responsibilities, regular communication between team members, and evaluation of the constant progress of the project. The team benefits from a streamlined workflow and more efficient project management, boosting productivity and the quality of work done.

Observed behavior: The main observed advantage of having a simple and informal workflow was increased productivity. The team is small, based on tacit knowledge, works in the same place, and with a structure that facilitates the exchange of communication, simplifies the process, and adopts more agile and efficient project management. The team improves productivity and quality of work. The quality of work can come from the will to solve the client's pain rather than from a simple process. Despite delivering value to customers, they cannot avoid rework and technical debt.

A simple and informal workflow approach (CAT5.2) allows the team to work more agile and flexibly by adopting lean project management practices. The team prioritizes effective communication, collaboration, and continuous feedback allowing a quick adaptation to changes, more informed decision-making, and an accelerated and constant pace of work.

Applying minimum project management practices (T-REC.01) directs the team to clearly define objectives, assign responsibilities, maintain regular communication between team members, and evaluate project progress. With a sharp focus, the team streamlines its workflow and improves the efficiency and quality of work. By avoiding inefficient and complex practices, the team avoids accumulating process debt, which can cause problems such as low productivity, delays, and additional costs, by reducing the risks associated with process debt.

6.2.2.2 T-REC.02

Recommendation: Spend minimal time and effort on defining the best possible software architecture.

Recommendations description: The startup needs to accelerate development (PA.05); however, it should minimally spend time and effort in defining the best possible software architecture (T-REC.02), adopting an evolutionary approach (PA.06), and prioritizing the attributes of quality (PA.09).

The startup can reduce rework and avoid technical debt. While involving the approach of delegating complexity to third-party applications (CAT5.1), focusing on product efficiency after launch (CAT6.2), and prioritizing user experience (CAT9.1) during architecture design, software should focus on minimally analyzing the best technology to use. The team should streamline the architecture design process, avoiding unnecessary complexity and focusing on the quality attributes essential to the product's success.

By spending minimal time defining the architecture, the team can focus on building the software agilely, reducing architectural design debt and documentation debt, allowing the team to deliver initial versions of the software rapidly, and conducting rapid brainstorming and prototyping sessions to test and validate the solutions. An evolutionary approach allows the team to adapt and evolve the software architecture based on user needs and feedback. The product can be continuously improved, meeting quality requirements and providing a satisfactory user experience.

Observed behavior: Startups have sought to accelerate development through an agile approach and minimization of time spent defining the software architecture, allowing the rapid delivery of initial product versions. This evolutionary approach (PA.06) allows startups to be flexible and adapt to changes and user feedback. In this way, the product can continually evolve, meeting market demands and keeping up with customers' ever-changing needs.

By avoiding unnecessary complexity and prioritizing software quality, teams reduce technical debt and avoid problems stemming from poor design and development decisions, reduce rework, and improve software maintainability, making it more efficient and sustainable.

By focusing on product efficiency after launch (CAT6.2), startups optimize software performance, scalability, and usability, providing a superior user experience. Furthermore, by prioritizing user experience (CAT9.1) during software architecture design, startups create products that meet user expectations and needs, increasing their satisfaction and loyalty.

These approaches adopted by startups aim to boost their growth, improve product quality and strengthen their competitive position in the market. By being agile, adaptable, and focused on the user experience, they are more likely to achieve success and customer preference.

6.2.2.3 T-REC.03

Recommendation: Define software development coding practices and policies with the team.

Recommendations description: One of the measures to deal with the challenges related to technical debt (CAT4.1), documentation debt (CAT4.2), process debt (CAT4.3), requirements debt (CAT4.4), and testing debt (CAT4 .5) is the definition of software development practices and policies together with the team (T-REC.03). This approach seeks to mitigate the problems arising from the slowdown in software development, promote product quality and efficiency, and accumulate these debts.

The team establishes code, architecture, documentation, processes, and requirements guidelines by defining transparent software development practices and policies that facilitate collaboration between team members, ensure development consistency, and improve communication and knowledge sharing. The team can streamline development and mitigate software-related challenges by adopting integrated tools.

The definition of practices and policies also contributes to the quality of the product. The team can automate processes, improve efficiency, and reduce errors using DevOps. Considering software quality attributes during development, such as usability, user experience (UX), and product-market fit, ensures that the product meets customer expectations and stands out in the marketplace.

By involving the customer in the development process, the team can obtain feedback and ensure that the product is aligned with the needs and desires of the users, contributing to achieving a good product-market fit and improving the probability of product success in the market.

The definition of software development practices and policies can also help to deal with the challenges related to the accumulation of technical debt, innovation, and human aspects. The team can mitigate risk and facilitate product evolution by taking an evolutionary approach and utilizing available technologies. Furthermore, implementing a simple and informal workflow and seeking external expertise when needed helps streamline development and address software-related challenges.

The definition of software development practices and policies with the team (T-REC.03) is an important measure to face challenges related to technical debt, guarantee product quality, and promote innovation.

Observed behavior: The main benefit observed in startups that defined development practices and policies was the improvement of development efficiency. The team is productive, reducing errors and accelerating the product development and release cycle. When the team adopts software development practices and policies, it ensures that the product meets desired quality attributes such as usability, user experience (UX), and market suitability by being highly functional, attractive, and meeting customer expectations, highlighting it if on the market.

With the client in the development process, the team obtains valuable feedback to align (pivot) the product to the needs and desires of the users, increasing the probability of achieving an excellent product-market fit since the product will be adapted to the demands and preferences of the users. Customers stimulate innovation and the competitiveness of the startup.

The definition of development practices and policies helps to avoid the accumulation of technical debt, preventing future problems means that the team can avoid rework and focus on continuous improvements, facilitating the evolution of the product in a sustainable way. The definition of clear guidelines for the code, architecture, documentation, processes, and requirements helps to avoid the accumulation of technical debt (CAT4.1), reducing problems related to harmful practices or low quality, resulting in more robust and reliable software.

6.2.2.4 T-REC.04

Recommendation: Include customers as soon and frequently as possible in the software development.

Recommendations description: A key measure to accelerate software development (PA.05) and ensure customer satisfaction is to include customers as early and often as possible in the development process (T-REC.04). This approach strengthens the relationship of trust with the customer, allowing the team to understand better and respond to their needs.

By involving customers early on, the startup can gather valuable feedback by understanding their emotions, expectations, and desires. Understanding the customer's pain allows the startup to focus on developing features and functionality that matter to the customer, avoiding wasting time and resources on less relevant aspects.

Customers should frequently be included in quick discussion sessions to align expectations and clarify requirements informally or verbally, streamlining the requirements elicitation process and avoiding delays and misunderstandings.

Prototyping is an essential practice in this context. Creating rapid prototypes allows the team to visualize and test ideas with the client, verifying their feasibility, and adapting with an iterative and interactive approach allows the team to adapt and pivot quickly, ensuring that the final product a with the customer's needs.

Including customers in the development process also contributes to innovation. By working closely with the customer, the startup can identify opportunities for improvement and new features that drive product differentiation and competitiveness.

Observed behavior: Startups that included customers from the beginning of the development process were able to understand their needs and expectations, allowing the team to develop a product that met their demands, resulting in greater satisfaction and loyalty. This feedback on the product is precious for the startup, as it is based on it that adjustments and improvements are made to ensure that the product meets market demands, increasing its chances of success. The startup quickly identified and responded to changes in market needs and demands, involving customers in the development process, and allowing the team to adapt and make agile adjustments, ensuring that the product evolves.

Collaborating closely with customers provided valuable input for innovation and differentiation. By understanding the needs and pains of customers, the startup can develop creative and innovative solutions to stand out in the market and have a competitive advantage. Thus, they avoided rework and product modifications by including customers early in the software development process. Continuous feedback allowed adjustments and improvements throughout the process, avoiding additional costs and saving time.

6.2.2.5 T-REC.05

Recommendation: Define a minimum software testing process.

Recommendations description: The startup must seek to accelerate development (PA.05), generating the minimum accumulation of technical debt (PA.04), and define a minimum software testing process (T-REC.05) with an evolutionary approach (PA.06) . This minimal software testing process (T-REC.05) should allow a startup to focus on the essentials and avoid wasting resources on complex and bureaucratic testing processes.

The definition of a minimum software testing process aims to guarantee that the main aspects of the software are tested with guarantee and to avoid or at least reduce the test debt (CAT4.5). This process should include an action plan for unit, integration, and acceptance testing. Tests should be carried out by a team member who has not developed, and then a beta version is released for end users. This simple workflow (CAT5.2) is encouraged for testing and should allow minimal manual and automated exploratory test logs and bug reports.

Different tests, such as controlled stress, functional, and manual tests, can be performed. The choice of test types depends on the needs and characteristics of the project. Running tests before meetings with customers or users is common practice, ensuring that the software is reliable and error-free during the stakeholder experience.

By defining a minimum software testing process, a startup seeks to balance product quality and development efficiency. This allows the team to meet critical quality requirements while avoiding excessive delays and costs related to extensive and detailed testing. It is essential to note that the software testing process can be adapted and improved over time, following an evolutionary approach. A startup can continually iterate and improve the testing process based on customer feedback, market needs, and lessons learned from previous projects.

Observed behavior: We observed that startups adopted some minimum software testing practices, with which they believe accelerating the pace of development is only possible because customers are tolerant of beta versions of software.

However, these strategies lead to the accumulation of technical debt related to testing. The software is not always adequately tested. It is just functionally aligned with the customer's expectations. The team focuses more on the development aspects of the software. In this way, the team's time and efforts are directed to the critical areas of development.

Startups only seek to ensure that the main quality aspects of the software are tested for usability and reliability, meeting the expectations of customers and users. Startups adapt software with customer feedback over time, market needs, and lessons learned throughout software development. This dynamic ensures that the product meets market demands.

6.2.2.6 T-REC.06

Recommendation: Make [it] simple.

Recommendations description: Simplicity is how startups accelerate development (PA.05), support innovation (PA.07) and make the team the catalyst for development (PA.01). "Making something simple" (T-REC.06) is an entrepreneurial behavior to respond to the challenges faced by the startup in different aspects such as business, technical and human.

A startup can only face challenges with skill and efficiency with a high-performance team (CAT1.4) by looking for simple solutions to its complex problems. This ability to simplify more complex technical aspects while the team remains focused on its work is one of the hallmarks of startups. Simple and sometimes informal workflow (CAT5.2) promotes agility and efficiency. The startup strives to avoid excessive overhead by focusing on essential work and collaborating more fluidly using integrated tools (CAT5.4), streamlining work and making it easier to perform external tasks.

The search for innovation opportunities (CAT7.1) is directly related to the startup's ability to simplify because it is necessary to simplify software development processes, such as configuration management, software development, and requirements specification so that the team can accelerate the construction of new ideas and ensure the delivery of innovative products. In software development, simplicity is vital at all stages: a simple and informal development process, combined with simple tools, contributes to the efficiency and quality of work. Even the execution of the tests is carried out, covering only the most relevant aspects of the software and ensuring that the requirements are reliably met.

Observed behavior: Simplicity is vital for the startup to accelerate development and quickly deliver new features and updates to customers. The startup performs tasks more quickly and efficiently by eliminating unnecessary complexity and focusing on the core work (AP.05).

The quest for simplicity is directly linked to the quest for innovation opportunities. By simplifying development processes, they have more flexibility to explore new ideas and implement innovative solutions, which makes them stand out in the market (PA.07, CAT7.1).

The teams were skilled and efficient in facing challenges, looking for simple solutions to complex problems. Synergy and collaboration between team members are enhanced, driving our development (PA.01, CAT1.4). Indeed, precisely this integration, simplification capacity, technical knowledge, and integrated tools allied startups to overcome their limitations.

Startups adopted a simple and sometimes informal workflow that promoted agility and efficiency. We avoid unnecessary burdens and bureaucracy, allowing our team to focus on core work and collaborate seamlessly. This results in a productive and efficient workflow (CAT5.2).

They also recognize the importance of choosing easy-to-use, straightforward tools to facilitate our daily work. These tools simplify tasks, increase efficiency and get rid of unnecessary complexities. Thus, we can focus on software development and delivering customer value (CAT5.4).

Simplicity also plays a crucial role in our software testing. By taking a simple and efficient approach to test execution, they could cover the most relevant aspects of our software, ensuring that our requirements were met reliably and contributing to delivering a high-quality product and customer satisfaction (T-REC.06).

6.2.2.7 T-REC.07

Recommendation: Make [it] integrated.

Recommendations description: Tool integration is vital to driving development (PA.05) and making the team the catalyst for our software development (PA.01). By adopting tools with the potential for integration, the startup can achieve high performance (CAT1.4) capable of facing the challenges related to software development, because the use of integrated and simple tools (CAT5.4) allows it to work more efficiently reducing the complexity of activities. The integration of tools is critical to ensure smooth collaboration and effective execution of software-related tasks. By making everything integrated, startups can overcome the challenges related to software development and maintain a cohesive and productive team. The integration allows the startup to work more efficiently and achieve high-quality results.

Observed behavior: Startups that managed to integrate tools strengthened collaboration and teamwork, creating a cohesive and highly efficient team. Using well-integrated and straightforward tools could streamline the development process with more efficient execution of tasks, reducing the time spent on unnecessary and bureaucratic activities. Integration of tools simplified the team's daily work, eliminating unnecessary complexities: daily meetings. Some of them replaced meetings with robots in Slack that questioned the developer about the three Scrum issues. They forwarded the information to the project manager, allowing us to work more efficiently, focusing on software development and delivering value to our customers.

Integration helped them face and overcome the unique challenges of software development. With built-in tools, we can more effectively handle technical issues, improve collaboration between team members, and ensure the quality of our product. The integration even strengthened the relationship between team members, promoting greater synergy and technical collaboration and creating a positive and encouraging work environment where everyone is aligned with the startup's goals.

6.2.2.8 T-REC.08

Recommendation: Make [it] automated.

Recommendations description: Automation is crucial to startup success. The integration of well-designed, easy-to-use tools can streamline processes and increase efficiency. Automation empowers staff to focus on high-value tasks, leading to better performance and productivity. It significantly accelerates the development process by eliminating repetitive and time-consuming activities by performing tasks quickly and accurately.

In addition, automation reduces errors and minimizes the need for rework, optimizing resources and maximizing team efficiency. In particular, automated testing plays a vital role in software assurance. We could assess our software's capabilities by automating these tests and identifying potential bottlenecks or weaknesses. This allows us to make informed decisions and improve our product's quality and reliability.

Observed behavior: Automating routine and low-value tasks frees up the team so that it has more time and energy to dedicate to strategic and innovative activities, increasing productivity and allowing it to launch itself into new challenges.

The elimination of repetitive and time-consuming tasks resulted in a significant increase in operational efficiency, optimizing the use of resources and accelerating development. In some cases, automation has reduced human errors as tasks are completed, minimized the need for rework, and improved work quality by ensuring that products were delivered with fewer issues and failures.

It was also observed that automation accelerated the development process, eliminating bottlenecks and delays caused by time-consuming activities and allowing us to deliver new features and updates faster and more efficiently, keeping them competitive in the market. By automating testing activities, they were able to identify issues and failures more quickly and accurately, allowing them to take immediate corrective action resulting in more responsive products and greater customer satisfaction.

6.2.2.9 T-REC.09

Recommendation: Find the toolset that helps your team to accelerate software development.

Recommendations description: Finding the right toolset, although difficult, is essential to accelerate software development (T-REC09), especially in startups with a small teams. Choosing the right tools can make all the difference in team performance and efficiency. There is no standard set of tools for software startups. There are specific tools that help with some operational tasks. However, the team needs to know how to use them so that the tools are a software development accelerator.

To have a high-performance team (CAT1.4), the startup must offer integrated and straightforward-to-use tools (CAT5.4) to support. By finding the right toolset, the startup might optimize design architecture, business modeling, communications, customer relationship management (CRM), databases, and other critical aspects of software development. Adopting design and project management tools that meet the team's specific needs can facilitate the planning, monitoring, and coordinating activities, ensuring that development progresses efficiently. The choice of configuration and automation tools in software development is also essential because they simplify tasks such as version control, continuous integration, and automated deployment, streamlining the development process and reducing errors. Software testing tools are crucial in product validation and verification because they allow the team to run automated tests, identify issues, and improve software quality before release. It is critical to consider the team's specific needs and look for integrated, simple-to-use, effective tools that work for the startup's success.

Observed behavior: The startups that managed to choose a set of tools that the team knew and knew how to use managed to have a positive response regarding efficiency and productivity, achieving faster and more effective development. Integrating the tools and their ease of use (CAT5.4) allowed a team to work more fluidly and collaboratively, optimizing time and resources. Despite the scarcity that characterizes startups, they overcame these limitations through technology and achieved speed and excellence in the software product they offered.

Adopting integrated and simple tools was also essential to improve the team's performance (CAT1.4). The tools assist in project management, communication, business modeling, database, and other critical areas of software development. With them, the team can efficiently plan, monitor and coordinate activities, ensuring that development occurs organizationally and is aligned with the startup's objectives. Configuration and automation tools were essential to simplify repetitive and time-consuming tasks, such as version control and continuous integration, streamlining the development process, generating error tolerance, and increasing the quality of the delivered software. Software testing tools were also crucial in product validation and verification. The team can identify and fix issues before release with controlled testing, ensuring reliability and high-quality software.

6.2.3 Organizational Recommendations

6.2.3.1 O-REC.01

Recommendation: Make organizational culture change.

Recommendations description: The startup that seeks innovation opportunities (CAT7.1) must promote change in the organizational culture (HA-REC.05) by promoting change in organizational culture (HA-REC.05), the startup creates an environment that encourages creativity and innovation, as it is open and receptive to new ideas, encourages people to think creatively, challenge the status quo and seek innovative solutions to challenges. A culture that values experimentation and continuous learning allows employees to take risks and test new approaches, even if it involves the risk of making mistakes. Mistakes are seen as part of learning and an opportunity for growth and improvement.

The change in organizational culture (HA-REC.05) can promote collaboration and knowledge exchange among team members and increases their skills and experience, resulting in synergies that generate more complete and innovative solutions. This organizational culture generates a capacity for change and adaptation of the team that allows them to respond quickly to market changes and customer demands, bringing opportunities for innovation (CAT7.1).

An organizational culture (HA-REC.05) that promotes innovation (CAT7.1) and the search for opportunities attracts and retains top talent because these professionals seek a stimulating and challenging environment that values innovation and offers space for growth personal and professional growth.

Observed behavior: Startups that provided cultural change or were born encouraging people to think creatively and be willing to challenge established standards created an environment conducive to generating new ideas and innovative solutions to the challenges faced by the startup.

This "new" culture values experimentation and continuous learning, encouraging members to take risks and test new approaches. Mistakes are seen as learning and growth opportunities, driving constant improvement. It promotes collaboration and knowledge exchange among team members, resulting in synergy and generating complete and innovative solutions, taking advantage of each one's experiences and skills.

An organizational culture that values innovation and the pursuit of opportunities allows the team to respond quickly to market changes and customer demands and allows the startup to be always prepared to identify and take advantage of innovation opportunities, remaining competitive in the market.

An organizational culture that fosters innovation and provides space for personal and professional growth attracts top talent. Qualified professionals seek a stimulating and challenging environment where they can contribute with their skills and develop their careers.

6.2.3.2 O-REC.02

Recommendation: Promote diversity.

Recommendations description: Startups looking for innovation opportunities (CAT7.1) must promote diversity (O-REC.02).

Diversity brings a number of benefits to organizations, including startups. Companies recognize the importance of having diverse teams made up of people from different backgrounds, cultures, skills, experiences, and perspectives. The team will have members with diverse backgrounds, skills, and knowledge that favor various perspectives and ideas, which can drive creativity and innovation (CAT7.1). Different points of view can generate unique solutions and solve problems in innovative ways. Diversity (O-REC.02) brings a variety of opinions and approaches to decision-making, which helps avoid group mentality and promotes critical thinking. Different perspectives can lead to a more thorough analysis of challenges and more informed and accurate decision-making. Diversity (O-REC.02) also allows the team to put themselves in the customer's shoes more effectively, better understand an audience's needs and expectations, and get to what is customarily applied to the term "feeling the customer's pain". This is especially relevant in startups looking for innovation (CAT7.1), as they can create products and services that meet the demands of different market segments and are scalable.

The startup creates an inclusive environment where all team members feel valued and respected, bringing a positive work climate where people feel encouraged to share their ideas and contributions. An inclusive environment encourages collaboration and knowledge exchange, strengthening the culture of innovation (CAT7.1). Startups that promote diversity (O-REC.02) gain a positive reputation among their employees and customers, leading them to attract diverse talent and customers who value equality and inclusion. A positive reputation can also open doors to strategic partnerships and business opportunities.

In addition to the benefits of performance and innovation, promoting diversity is a matter of social justice and equality. Companies recognize the importance of providing equal opportunities for all individuals, regardless of origin, gender, race, or other personal characteristics.

Observed behavior: Startups that foster diversity in backgrounds, skills, and knowledge gained from various perspectives and ideas can drive creativity and innovation and lead to unique solutions and innovative approaches to problem-solving. Diversity brought them a variety of opinions and approaches to the decision-making process, helping to avoid groupthink and promote critical thinking, allowing a more comprehensive analysis of challenges and facilitating decision-making.

Teams at these startups better understood customer needs and expectations by considering multiple perspectives. This empathetic understanding, often called "feeling the customer's pain," helps startups create products and services that cater to different market segments and scales. Promoting diversity created an inclusive environment where all team members felt valued and respected, generating a positive work climate where individuals are encouraged to share their ideas and feel their contributions are valued. An inclusive environment also encourages collaboration and knowledge exchange, strengthening the culture of innovation.

Startups that promoted diversity gained positive reputations among employees and customers. This reputation can attract or retain diverse talent, as individuals are drawn to organizations that prioritize diversity and provide an inclusive work environment. Customers who value diversity and inclusion may be more inclined to support and engage with the startup. A positive reputation for promoting diversity has opened the door to strategic partnerships and business opportunities. Other organizations and stakeholders may be more inclined to collaborate with a startup that values diversity and inclusion, leading to mutually beneficial partnerships and growth opportunities.

6.3 FINDINGS' DISCUSSION

This symbiotic relationship between our research questions and the recommendations furnished herein serves to elucidate the practical pathways toward excellence in early-stage software development within startup ecosystems. Through diligent adherence to these recommendations, startups cannot only navigate the turbulent waters of uncertainty but do so confidently and competently, poised for success in an ever-evolving technological landscape. The benefits of these recommendations are not just theoretical, but they are tangible, offering startups a sense of reassurance and confidence in their software development endeavors.

- **Research Question (RQ1): How do startups develop software in the early stages?** In addressing this pivotal question, we delve into the technical realm, where our recommendations serve as guiding lights through the intricacies of software development within startup environments. The Technical Recommendations

(T-REC.01 to T-REC.09) are not just beacons, but they illuminate pathways toward enhanced processes, architectures, policies, and tool selections. By delving into these recommendations, we offer valuable insights into how startups can refine and optimize their software development practices, empowering them with knowledge and understanding.

- **Research Question (RQ1.1): What is the context for startup software development?** Here, our focus shifts to the human aspects integral to startups' software development landscape. Embarking on a journey through the Human Aspects Recommendations (HA-REC.01 to HA-REC.07), we explore the intricacies of mindset, learning culture, resilience in the face of failure, diversity, and fostering trust within teams. These recommendations unveil the human tapestry woven into the fabric of startup environments, shedding light on the nuanced context within which software development unfolds.
- **Research Question (RQ1.2): What software engineering practices are most pertinent for early-stage startups?** In pursuit of this inquiry, we navigate through the labyrinth of software engineering practices, guided by the compass of Technical Recommendations (T-REC.01 to T-REC.09). These directives are not just theoretical, but they are practical and actionable, aimed at streamlining processes, fostering collaboration with customers, ensuring robust testing methodologies, and embracing simplicity, integration, and automation. These recommendations offer a roadmap for startups seeking to navigate the complex terrain of software engineering with finesse and efficacy, equipping them with the tools they need for success.
- **Research Question (RQ1.3): What quality attributes do startups prioritize in software development?** Within this realm of inquiry lies a singular focus on quality, encapsulated in the Technical Recommendation T-REC.05. Here, the emphasis is on defining a minimum software testing process, underscoring the paramount importance of quality assurance in software development endeavors. By adhering to this recommendation, startups can imbue their software products with the resilience and reliability necessary for success in today's competitive landscape.

6.4 CHAPTER CONCLUSION

In this section, we provide a concise summary of our data analysis findings. Figure 6.1 offers an overview of our analysis outcomes. Our investigation identified three core categories: Human Factors, Technical Factors, and Organizational Factors.

Throughout our research, we observed that some startups exhibited a degree of negligence in their software engineering practices, while others only sporadically utilized specific practices. The intersection of these factors highlights the multifaceted challenges startups face in their software development endeavors.

Chapter 7

CONCLUSIONS AND FUTURE WORK

The main question that this research sought to answer was: “How do early-stage startups develop software?” In order to understand this context, we carried out a multiple-case study with the participation of fourteen software startups located in Bahia, Brazil. The data analysis identified that Human, Technical, and Organizational Aspects influence software development.

Several factors and challenges characterize the context of software development for startups. Startups operate in a dynamic, fast-paced environment where speed, innovation, and customer focus are essential. The development process is focused on creating software products to meet market and customer needs. Startups face various software-related challenges: ensuring software quality attributes such as scalability, usability, and User Experience (UX) and dealing with accumulated technical debt, which refers to compromises made during software development that can result in future problems. Also, startups pursue innovation in various aspects, including business, product, service, and technology. They must continually adapt and evolve their software to meet ever-changing market demands.

Customer involvement is crucial for software development startups to receive customer feedback and generate a relationship of trust. During development, customer needs and feedback are considered to ensure the product meets their expectations. In addition to the technical challenges, the startup faces challenges related to the human aspects of software development, including team dynamics, communication, and flexible working hours. Startups often have small, self-managed teams with multiple roles, relying on tacit knowledge and trust. Team members collaborate closely, work under constant pressure, and often work overtime to meet deadlines. To accelerate development and overcome challenges, startups use various accelerators, including well-integrated and straightforward tools, an evolutionary approach to development, architectural design, DevOps practices, and the adoption of third-party frameworks and APIs. Finally, the context of software development for startups is characterized by speed, innovation, customer involvement, and various challenges. Startups rely on a small, well-integrated team, use accelerators, and strive to deliver high-quality software that meets market needs.

7.1 SCIENTIFIC CONTRIBUTIONS

The contributions of the work include the development of specific and innovative technologies related to the context of software development in startups, includes Creating a model that describes this context. Identifying sets of quality attributes relevant to early-stage startups, and Defining a set of tools commonly used by these early-stage startups. A set of recommendations explicitly aimed at early-stage startups.

These contributions aim to provide valuable insights and practical guidance for startups in the software development process, helping them succeed in this crucial phase.

These results have practical applicability to the software industry. Industry professionals can use the results of this study to learn about their initiatives and avoid pitfalls that can lead to early death for software startups. For example, the first activity that the startup must carry out, along with the short-term planning of the business, is a brief economic, technical, and market feasibility study. The team's potential should be analyzed, and mentoring should be sought for points with a deficit. Before starting software development, they should minimally define a simple software development process, with records of critical decisions. The development team must be involved in the organizational culture to avoid noise in communication and rework. Mentorships can help define a valuable process for the startup, but the development team's commitment, communication, and teamwork will lead the startup to success. The effective combination of agile methods and software development techniques known by the team helps overcome the natural challenges faced by this type of company.

From a scientific point of view, the research contributed information about the context of software development in startups, the main pitfalls they fall into when treading this path, and provides recommendations for starting professionals or startups. Although several pieces of research related to software engineering in startups, this research presented an aspect that still needs to be explored in other research. It refers to what happens to startups when they develop software and provides recommendations for improving these identified flaws.

During the construction of this thesis, the following research papers were published, contributing to the body of knowledge in software engineering practices within the context of startups:

- Renata Souza, Orges Cico, Ivan Machado: "A survey on the software engineering practices in Brazilian software startups" at CIbSE 2021.
- Orges Cico, Terese Besker, Antonio Martini, Anh Nguyen-Duc, Renata Souza, Jan Bosch: "Toward a Technical Debt Relationship with the Pivoting of Growth Phase Startups" at PROFES 2021.
- Orges Cico, Renata Souza, Letizia Jaccheri, Anh Nguyen-Duc, Ivan Machado: "Startups Transitioning from Early to Growth Phase - A Pilot Study of Technical Debt Perception" at ICSOB 2020.
- Renata Souza, Larissa Rocha Soares, Franklin Silva, Ivan do Carmo Machado: "Investigating Agile Practices in Software Startups" at SBES 2019.

- Renata Souza, Karla Malta, Roselane Silva, Paulo César Masiero, Eduardo Santana de Almeida, Ivan Machado: "A Case Study about Startups' Software Development Practices: A Preliminary Result" at SBQS 2019.

- Renata Souza, Karla Malta, Eduardo Santana De Almeida: "Software Engineering in Startups: A Single Embedded Case Study" at SoftStart@ICSE 2017.

7.2 RESEARCH LIMITATIONS

While this study presents a diverse database of software startups with various profiles, business models, and segments, it is essential to acknowledge that the challenges identified may be specific to the startup ecosystem in Bahia, Brazil. To validate the generalizability of these findings, further research should explore whether similar challenges exist in startups from different ecosystems. It would be interesting to confirm whether the Human, Technical, and Organizational factors identified in this study hold true in other regions and entrepreneurial contexts.

Another limitation pertains to the choice of interviewees, who were primarily CEOs and CTOs. Startups, particularly in their early stages, often rely heavily on the tacit knowledge and expertise of their founders or senior developers. While these key roles were the primary focus of this study due to their substantial influence on the development process, a broader examination that includes the entire software development team could provide additional insights. Gathering diverse perspectives from the team may contribute to a more comprehensive understanding of software development practices in startups.

Qualitative research is susceptible to interpretation bias by the researcher. To mitigate this limitation, a second researcher was invited to participate in the interviews, data analysis, and the interpretation of results. This collaborative approach aimed to reduce the potential for bias and ensure a more robust analysis.

7.3 FUTURE WORK

This study opens up several avenues for future research and potential applications of the findings. One promising direction is to apply the recommendations developed in this study to startups that are in the early stages of their software development process. This application can help these startups improve their software development practices and enhance their chances of success. Collaborative efforts with startup founders and teams to implement and evaluate these recommendations would be an exciting prospect.

Another area of interest is the investigation of technical debt management in startups. Future research could explore strategies to prevent and effectively manage technical debt in startup software development. Collaborating with experts in this field could yield valuable insights and practical solutions.

Furthermore, there is potential to conduct case studies on software startups that are dedicated to sustainability. Analyzing the software development practices and the impact of their solutions on sustainability could provide a unique opportunity for interdisciplinary collaboration with experts in environmental and sustainable development fields.

These future developments and applications extend the reach of this research, allowing for the continued exploration of new areas, practical implementation of the findings, and collaboration with diverse domains of expertise.

BIBLIOGRAPHY

ABASTARTUP. *Associação Brasileira de Startups*. 2023. April, 2023. Disponível em: <https://abstartups.com.br/>.

ABDULLAH, S. H.; RAMAN, S. Quantitative and qualitative research methods: Some strengths and weaknesses. *Jurnal Pendidik dan Pendidikan, Jilid*, v. 17, p. 1–15, 2001.

ABRAHAMSSON, P.; OZA, N. V.; SIPONEN, M. T. Agile software development method, A comparative review1. *CoRR*, abs/1903.10913, 2019. Disponível em: <http://arxiv.org/abs/1903.10913>.

ALBUQUERQUE, R. F. L. de. *Uma Investigação sobre o Abandono e a Continuidade de Programas de Melhoria de Processos de Software*. Tese (Doutorado) — Pontifícia Universidade Católica do Paraná, 2021.

ANALYTICA, O. Brazil's nubank plots expansion after successful ipo. *Emerald Expert Briefings*, Oxford Analytica, n. oxan-es, 2021.

BAJWA, S. S. Pivoting in software startups. *Fundamentals of software startups: Essential engineering and business aspects*, Springer, p. 27–43, 2020.

BAXTER, P.; JACK, S. et al. Qualitative case study methodology: Study design and implementation for novice researchers. *The qualitative report*, v. 13, n. 4, p. 544–559, 2008.

BERG, V. et al. Software startup engineering: A systematic mapping study. *J. Syst. Softw.*, v. 144, p. 255–274, 2018. Disponível em: <https://doi.org/10.1016/j.jss.2018.06.043>.

BESKER, T. et al. Embracing technical debt, from a startup company perspective. In: IEEE. *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. [S.l.], 2018. p. 415–425.

BJARNASON, E. Prototyping practices in software startups: Initial case study results. In: IEEE. *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. [S.l.], 2021. p. 206–211.

BLANK, S. *What's A Startup? First Principles*. 2010. September,2022. Disponível em: <https://steveblank.com/2010/01/25/whats-a-startup-first-principles/>.

BLANK, S. *What Do I Do Now? The Startup Lifecycle*. 2015. April, 2015. Disponível em: <https://steveblank.com/2015/02/12/what-do-i-do-now/>.

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

BLANK, S.; DORF, B. *The startup owner's manual: The step-by-step guide for building a great company.* [S.l.]: John Wiley & Sons, 2020.

BOSCH, J. et al. The early stage software startup development model: a framework for operationalizing lean principles in software startups. In: SPRINGER. *International Conference on Lean Enterprise Software and Systems.* [S.l.], 2013. p. 1–15.

BOTELHO, T. L.; FEHDER, D.; HOCHBERG, Y. *Innovation-driven entrepreneurship.* [S.l.], 2021.

BOURQUE, P.; FAIRLEY, R. E. Swebok guide v3.0: Guide to the software engineering body of knowledge. IEEE Computer Society, 2014.

BRERETON, P. et al. Using a protocol template for case study planning. In: CITESEER. *EASE.* [S.l.], 2008. v. 8, p. 41–48.

CANTAMESSA, M. et al. Startups' roads to failure. *Sustainability*, Multidisciplinary Digital Publishing Institute, v. 10, n. 7, p. 2346, 2018.

CARMEL, E. Time-to-completion in software package startups. In: *27th Annual Hawaii International Conference on System Sciences (HICSS-27), January 4-7, 1994, Maui, Hawaii, USA.* [S.l.]: IEEE Computer Society, 1994. p. 498–508.

CAVALCANTE, B. H. et al. Technical aspects of software development in startups: A systematic mapping. In: IEEE. *2018 XLIV Latin American Computer Conference (CLEI).* [S.l.], 2018. p. 100–109.

CHARMAZ, K. *Constructing grounded theory: A practical guide through qualitative analysis.* [S.l.]: sage, 2006.

CICO, O. et al. Startups transitioning from early to growth phase-a pilot study of technical debt perception. In: SPRINGER. *Software Business: 11th International Conference, ICSOB 2020, Karlskrona, Sweden, November 16–18, 2020, Proceedings 11.* [S.l.], 2021. p. 102–117.

COLEMAN, G.; O'CONNOR, R. V. An investigation into software development process formation in software start-ups. *Journal of Enterprise Information Management*, Emerald Group Publishing Limited, v. 21, n. 6, p. 633–648, 2008.

COLEMAN, G.; O'CONNOR, R. Using grounded theory to understand software process improvement: A study of irish software product companies. *Information and Software Technology*, Elsevier, v. 49, n. 6, p. 654–667, 2007.

CORBIN, J.; STRAUSS, A. *Basics of qualitative research: Techniques and procedures for developing grounded theory.* [S.l.]: Sage publications, 2014.

CORBIN, J. M.; STRAUSS, A. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative sociology*, Springer, v. 13, n. 1, p. 3–21, 1990.

CRESWELL, J. W. Research designs: Qualitative, quantitative, and mixed methods approaches. *Callifornia: Sage*, 2009.

CROWNE, M. Why software product startups fail and what to do about it. evolution of software product development in startup companies. In: IEEE. *IEEE International Engineering Management Conference*. [S.l.], 2002. v. 1, p. 338–343.

CUKIER, D.; KON, F.; KRUEGER, N. Towards a software startup ecosystems maturity model. *Department of Computer Science-University of São Paulo Technical Report RT-MAC*, 2015.

CUKIERMAN, H. L.; TEIXEIRA, C.; PRIKLADNICKI, R. Um olhar sociotécnico sobre a engenharia de software. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 199–219, 2007.

CUSUMANO, M. A. The changing software business: Moving from products to services. *Computer*, v. 41, n. 1, p. 20–27, 2008. Disponível em: <https://doi.org/10.1109/MC.2008.29>.

DAGENAIS, B. et al. Moving into a new software project landscape. In: *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*. [S.l.: s.n.], 2010. p. 275–284.

DANDE, A. et al. Software startup patterns-an empirical study. Tampere University of Technology. Department of Pervasive Computing, 2014.

DASH, A. Scaling the innovativeness of start-ups in india. *SEDME (Small Enterprises Development, Management & Extension Journal)*, SAGE Publications Sage India: New Delhi, India, v. 46, n. 3, p. 196–204, 2019.

DYBÅ, T. et al. Qualitative research in software engineering. *Empir. Softw. Eng.*, v. 16, n. 4, p. 425–429, 2011. Disponível em: <https://doi.org/10.1007/s10664-011-9163-y>.

EBERT, C. et al. Devops. *IEEE Softw.*, v. 33, n. 3, p. 94–100, 2016. Disponível em: <https://doi.org/10.1109/MS.2016.68>.

EDISON, H. et al. Lean internal startups for software product innovation in large companies: enablers and inhibitors. *Journal of Systems and Software*, Elsevier, v. 135, p. 69–87, 2018.

EDISON, H.; WANG, X.; ABRAHAMSSON, P. Lean startup: why large software companies should care. In: ACM. *Scientific Workshop Proceedings of the XP2015*. [S.l.], 2015. p. 2.

EDISON, H. et al. Innovation initiatives in large software companies: A systematic mapping study. *Information and Software Technology*, Elsevier, v. 95, p. 1–14, 2018.

ENGWARD, H. Understanding grounded theory. *Nursing Standard (through 2013)*, BMJ Publishing Group LTD, v. 28, n. 7, p. 37, 2013.

FARIAS, R. S. et al. Designing smart city mobile applications. *Empirical Software Engineering*, Springer, v. 24, n. 6, p. 3255–3289, 2019.

FOWLER, M.; FOEMMEL, M. *Continuous integration*. 2006. September, 2022. Disponível em: <https://martinfowler.com/articles/continuousIntegration.html>.

FRIESENDORF, C.; UEDELHOVEN, L. Digital transformation of global mobility markets. In: *Mobility in Germany*. [S.l.]: Springer, 2021. p. 1–9.

GAUTHIER, J. F.; MORELIX, A. The impact of covid-19 on global startup ecosystems. *Startup Genome*, 2020.

GHEZZI, A.; CAVALLO, A. Agile business model innovation in digital entrepreneurship: Lean startup approaches. *Journal of Business Research*, Elsevier, v. 110, p. 519–537, 2020.

GIARDINO, C. et al. Software development in startup companies: The greenfield startup model. *IEEE Trans. Software Eng.*, v. 42, n. 6, p. 585–604, 2016. Disponível em: <https://doi.org/10.1109/TSE.2015.2509970>.

GIARDINO, C.; WANG, X.; ABRAHAMSSON, P. Why early-stage software startups fail: A behavioral framework. In: LASSENIUS, C.; SMOLANDER, K. (Ed.). *Software Business. Towards Continuous Value Delivery - 5th International Conference, ICSOB 2014, Paphos, Cyprus, June 16-18, 2014. Proceedings*. Springer, 2014. (Lecture Notes in Business Information Processing, v. 182), p. 27–41. Disponível em: <https://doi.org/10.1007/978-3-319-08738-2_3>.

GLASER, B. G. *Theoretical sensitivity: Advances in the Methodology of Grounded Theory.*. [S.l.]: CA: Sociology Press, 1978.

GLASER, B. G. *Basics of grounded theory analysis: Emergence vs forcing*. [S.l.]: Sociology press, 1992.

GLASER, B. G. *Doing grounded theory: Issues and discussions*. [S.l.]: Sociology Press, 1998.

GLASER, B. G. *The grounded theory perspective III: Theoretical coding*. [S.l.]: Sociology Press, 2005.

GLASER, B. G.; STRAUSS, A. L.; STRUTZEL, E. The discovery of grounded theory; strategies for qualitative research. *Nursing research*, LWW, v. 17, n. 4, p. 364, 1968.

GRALHA, C. et al. The evolution of requirements practices in software startups. In: CHAUDRON, M. et al. (Ed.). *Proceedings of the 40th International Conference on Software Engineering, ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018*. ACM, 2018. p. 823–833. Disponível em: <https://doi.org/10.1145/3180155.3180158>.

HILMOLA, O.-P.; HELO, P.; OJALA, L. The value of product development lead time in software startup. *System Dynamics Review*, Wiley Online Library, v. 19, n. 1, p. 75–82, 2003.

HODA, R. Socio-technical grounded theory for software engineering. *CoRR*, abs/2103.14235, 2021. Disponível em: <https://arxiv.org/abs/2103.14235>.

HODA, R.; NOBLE, J.; MARSHALL, S. Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering*, IEEE, v. 39, n. 3, p. 422–444, 2012.

HOLTON, J. A. The coding process and its challenges. *The Sage handbook of grounded theory*, v. 3, p. 265–289, 2007.

HYYTINEN, A.; PAJARINEN, M.; ROUVINEN, P. Does innovativeness reduce startup survival rates? *Journal of business venturing*, Elsevier, v. 30, n. 4, p. 564–581, 2015.

JACOBSON, I. et al. *The Essentials of modern software engineering: free the practices from the method prisons!* [S.l.]: Morgan & Claypool, 2019.

KEMELL, K.-K. et al. Startup metrics that tech entrepreneurs need to know. In: *Fundamentals of Software Startups*. [S.l.]: Springer, 2020. p. 111–127.

KLOTINS, E. et al. Exploration of technical debt in start-ups. In: IEEE. *40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*. [S.l.], 2018. p. 75–84.

KLOTINS, E. et al. A progression model of software engineering goals, challenges, and practices in start-ups. *IEEE Transactions on Software Engineering*, IEEE, 2019.

KLOTINS, E.; UNTERKALMSTEINER, M.; GORSCHEK, T. Software engineering knowledge areas in startup companies: a mapping study. In: SPRINGER. *International Conference of Software Business*. [S.l.], 2015. p. 245–257.

KO, A. J. A three-year participant observation of software startup software evolution. In: IEEE. *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*. [S.l.], 2017. p. 3–12.

KON, F. et al. A panorama of the israeli software startup ecosystem. *Available at SSRN 2441157*, 2014.

LEEDY, P. D.; ORMROD, J. E. *Practical research*. [S.l.]: Pearson Custom Saddle River, NJ, USA, 2005.

LETHBRIDGE, T. C.; SIM, S. E.; SINGER, J. Studying software engineers: Data collection techniques for software field studies. *Empirical software engineering*, Springer, v. 10, n. 3, p. 311–341, 2005.

MARTIN, A.; BIDDLE, R.; NOBLE, J. The xp customer team: A grounded theory. In: *2009 Agile Conference*. [S.l.: s.n.], 2009. p. 57–64.

MASOOD, Z.; HODA, R.; BLINCOE, K. How agile teams make self-assignment work: a grounded theory study. *Empirical Software Engineering*, Springer, v. 25, n. 6, p. 4962–5005, 2020.

MATT, C.; HESS, T.; BENLIAN, A. Digital transformation strategies. *Business & information systems engineering*, Springer, v. 57, p. 339–343, 2015.

MELEGATI, J.; GOLDMAN, A. Seven patterns for software startups. In: *Proceedings of the 22nd Conference on Pattern Languages of Programs*. [S.l.: s.n.], 2015. v. 30.

MELEGATI, J. et al. A model of requirements engineering in software startups. *Information and software technology*, Elsevier, v. 109, p. 92–107, 2019.

MELEGATI, J. et al. A board game to teach team composition in software startups. In: SPRINGER. *Software Business: 10th International Conference, ICSOB 2019, Jyväskylä, Finland, November 18–20, 2019, Proceedings 10*. [S.l.], 2019. p. 321–335.

MELEGATI, J.; GUERRA, E.; WANG, X. Hymap: Eliciting hypotheses in early-stage software startups using cognitive mapping. *Information Software Technology*, v. 144, p. 106807, 2022. Disponível em: <https://doi.org/10.1016/j.infsof.2021.106807>.

MÜLLER, R. M.; THORING, K. Design thinking vs. lean startup: A comparison of two user-driven innovation strategies. *Leading through design*, v. 151, n. 2, 2012.

NGUYEN-DUC, A.; ABRAHAMSSON, P. Minimum viable product or multiple facet product? the role of MVP in software startups. In: SHARP, H.; HALL, T. (Ed.). *Agile Processes, in Software Engineering, and Extreme Programming - 17th International Conference, XP 2016, Edinburgh, UK, May 24-27, 2016, Proceedings*. Springer, 2016. (Lecture Notes in Business Information Processing, v. 251), p. 118–130. Disponível em: <https://doi.org/10.1007/978-3-319-33515-5_10>.

NIELSEN. *Nielsen Norman Group (NNg)*. 2023. April, 2023. Disponível em: <https://www.nngroup.com/>.

OKANOVIĆ, M. Ž.; JEVTIĆ, M. V.; STEFANOVIĆ, T. D. Organizational changes in development process of technology startups. In: SPRINGER. *International Conference of Experimental and Numerical Investigations and New Technologies*. [S.l.], 2020. p. 203–219.

O'CONNOR, R. Using grounded theory coding mechanisms to analyze case study and focus group data in the context of software process research. In: *Research methodologies, innovations and philosophies in software systems engineering and information systems*. [S.l.]: IGI Global, 2012. p. 256–270.

PATERNOSTER, N. et al. Software development in startup companies: A systematic mapping study. *Information and Software Technology*, Elsevier, v. 56, n. 10, p. 1200–1218, 2014.

PETERSEN, K.; WOHLIN, C. Context in industrial software engineering research. In: IEEE. *3rd International Symposium on Empirical Software Engineering and Measurement*. [S.l.], 2009. p. 401–404.

PLATTNER, H.; MEINEL, C.; LEIFER, L. *Design thinking: understand–improve–apply*. [S.l.]: Springer Science & Business Media, 2010.

POMPERMAIER, L. *Modelo PRESS: Evoluindo a Adoção de Práticas de Engenharia de Software em Startups Digitais*. [S.l.]: Pontifícia Universidade Católica do Rio Grande do Sul, 2021.

POMPERMAIER, L.; PRIKLADNICKI, R. Brazilian startups and the current software engineering challenges: The case of tecnopuc. *Fundamentals of Software Startups: Essential Engineering and Business Aspects*, Springer, p. 331–345, 2020.

RAVASKA, V. The essence of software startup: an empirical study on the application of essence framework. 2020.

REINEHR, S. d. S.; PESSÔA, M. Reuso sistematizado de software e linhas de produto de software no setor financeiro: estudos de caso no brasil. 2008.

REN, M.; DONG, Z. *What do we know about Testing practices in Software Startups?* 2017.

RIES, E. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. [S.l.]: Crown Books, 2011.

RUNESON, P. et al. *Case study research in software engineering: Guidelines and examples*. [S.l.]: John Wiley & Sons, 2012.

SEAMAN, C. B. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering*, IEEE, v. 25, n. 4, p. 557–572, 1999.

SEBASTIAN, K. Distinguishing between the strains grounded theory: Classical, interpretive and constructivist. *Journal for Social Thought*, v. 3, n. 1, 2019.

SILVA, E. L. D.; MENEZES, E. M. Metodologia da pesquisa e elaboração de dissertação. *UFSC, Florianópolis, 4a. edição*, v. 123, 2005.

SJØBERG, D. I. et al. Building theories in software engineering. In: *Guide to Advanced Empirical Software Engineering*. [S.l.]: Springer, 2008. p. 312–336.

SOUSA, L. et al. Identifying design problems in the source code: A grounded theory. In: *Proceedings of the 40th International Conference on Software Engineering*. [S.l.: s.n.], 2018. p. 921–931.

SOUZA, R.; CICO, O.; MACHADO, I. A survey on the software engineering practices in brazilian software startups. In: *CIbSE*. [S.l.: s.n.], 2021. p. 304–317.

SOUZA, R. et al. A case study about startups' software development practices: A preliminary result. In: ALBUQUERQUE, A. B.; BARROS, A. L. B. de P. (Ed.). *Proceedings of the XVIII Brazilian Symposium on Software Quality, SBQS 2019, Fortaleza, Brazil, October 28 - November 1, 2019*. ACM, 2019. p. 198–203. Disponível em: <https://doi.org/10.1145/3364641.3364663>.

SOUZA, R. et al. Investigating agile practices in software startups. *Simpósio Brasileiro de Engenharia de Software*, ACM, 2019.

STOL, K.; RALPH, P.; FITZGERALD, B. Grounded theory in software engineering research: a critical review and guidelines. In: DILLON, L. K.; VISSER, W.; WILLIAMS, L. A. (Ed.). *Proceedings of the 38th International Conference on Software Engineering, ICSE 2016, Austin, TX, USA, May 14-22, 2016*. ACM, 2016. p. 120–131. Disponível em: <https://doi.org/10.1145/2884781.2884833>.

STRAUSS, A.; CORBIN, J. M. *Basics of qualitative research: Grounded theory procedures and techniques*. [S.l.]: Sage Publications, Inc, 1990.

SUTTON, S. M. The role of process in software start-up. *IEEE Software*, IEEE, v. 17, n. 4, p. 33–39, 2000.

TANVEER, B. Sustainable software engineering-have we neglected the software engineer's perspective? In: IEEE. *36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*. [S.l.], 2021. p. 267–270.

TIM, B.; BARRY, K. Change by design: How design thinking transforms organizations and inspires innovation. *MUT Journal of Business Administration*, v. 9, n. 2, p. 190–194, 2009.

TOLVANEN, J.-P. Incremental method engineering with modeling tools: Theoretical principles and empirical evidence (diss., jyväksylä, univ. 1998): Jyväksylä studies in computer science, economics and statistics. University of Jyväksylä (Jyväksylä), 1998.

UNTERKALMSTEINER, M. et al. Software startups–a research agenda. *e-Informatica Software Engineering Journal*, v. 10, n. 1, 2016.

VIAL, G. Understanding digital transformation: A review and a research agenda. *Journal of Strategic Information Systems*, Elsevier, v. 28, n. 2, p. 118–144, 2019.

WANG, X. et al. Key challenges in software startups across life cycle stages. In: SPRINGER. *International Conference on Agile Software Development*. [S.l.], 2016. p. 169–182.

WATERMAN, M.; NOBLE, J.; ALLAN, G. How much up-front? a grounded theory of agile architecture. In: IEEE. *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. [S.l.], 2015. v. 1, p. 347–357.

WHITWORTH, E.; BIDDLE, R. The social nature of agile teams. In: IEEE. *Agile 2007 (AGILE 2007)*. [S.l.], 2007. p. 26–36.

YIN, R. K. *Case study research and applications: Design and methods*. [S.l.]: Sage publications, 2017.

APPENDIX A

INFORMED CONSENT FORM

INFORMED CONSENT FORM
Case Study on Software Engineering in Startups

PURPOSE OF THE STUDY
This study aims to conduct a case study about practices of Software Engineering in Startups.

DECLARATION
I declare to have more than eighteen (18) years old, and I agree to participate in the case study conducted by Renata Souza and Karla Malta, members of the Postgraduate Program in Computer Science (PGCOMP) of the Federal University of Bahia (UFBA) under the advice of Prof. PhD. Ivan do Carmo Machado.

PROCEDURE
In this study, you will participate in an interview that aims to identify the software engineering practices adopted by startups. All documents used in this study will be presented to the participant. The interview will be conducted by two researchers involved in the research project: Renata Souza and Karla Malta. We will ask your special participation to collaborate with: (1) information on your experience; (2) answer interview questions made by researchers based on the startup's current software engineering practices; (3) allow verification of the software artifacts generated by the software development team; and (4) a final questionnaire with your impressions. After data collection, your name will be removed from the forms, and it will not be used at any time during the presentation of results.

CONFIDENTIALITY
I am aware that my name will not be released under any circumstances. I am also aware that the data obtained through this study will be kept confidential. Similarly, I agree not to communicate my findings as the study is completed and to maintain the confidentiality of technical documents submitted and are part of the experiment.

BENEFITS AND WITHDRAWAL OF FREEDOM
I understand that once the case study has been completed, the work I developed will be studied to understand the software engineering practices used by startups. The benefits we receive from this study are limited to learning the distributed and taught material. I also understand that I am free to make inquiries at any time, request that any information related to my person is not included in the study, or communicate my dropping out

without penalty. Finally, I participate in free will with the sole aim of contributing to the advancement and development of Software Engineering.

RESEARCHERS IN CHARGE

Karla Malta
Phone number | +55 71 9 9211-0787
Lattes | http://lattes.cnpq.br/6452560379894828

Renata Souza
Phone number | +55 71 9 9110-6926
E-mail | renata.rmss@gmail.com
Lattes | http://lattes.cnpq.br/8154336739411524

RESPONSIBLE TEACHER

Prof. PhD. Ivan do Carmo Machado
Graduate Program in Computer Science (PGCOMP)/UFBA

____-
—
Name (please print)

____-
—
Signature

____
Date

APPENDIX B

INTERVIEW PROCESS

This section includes the process used to conduct the interviews, the package with the support artifacts, and the questionnaire.

The interviews always took place at the startup's workplace. The first interview point with the participants was the request to read and sign a consent form (SUP04) with the interview agreement regarding participating and recording the interview. We prepare two identical copies of the consent form, one for the participant(s) and another for the interviewers. We asked all participants to sign the document. Once the participant signed the consent form, the interview began following the process illustrated in Figure B.1.

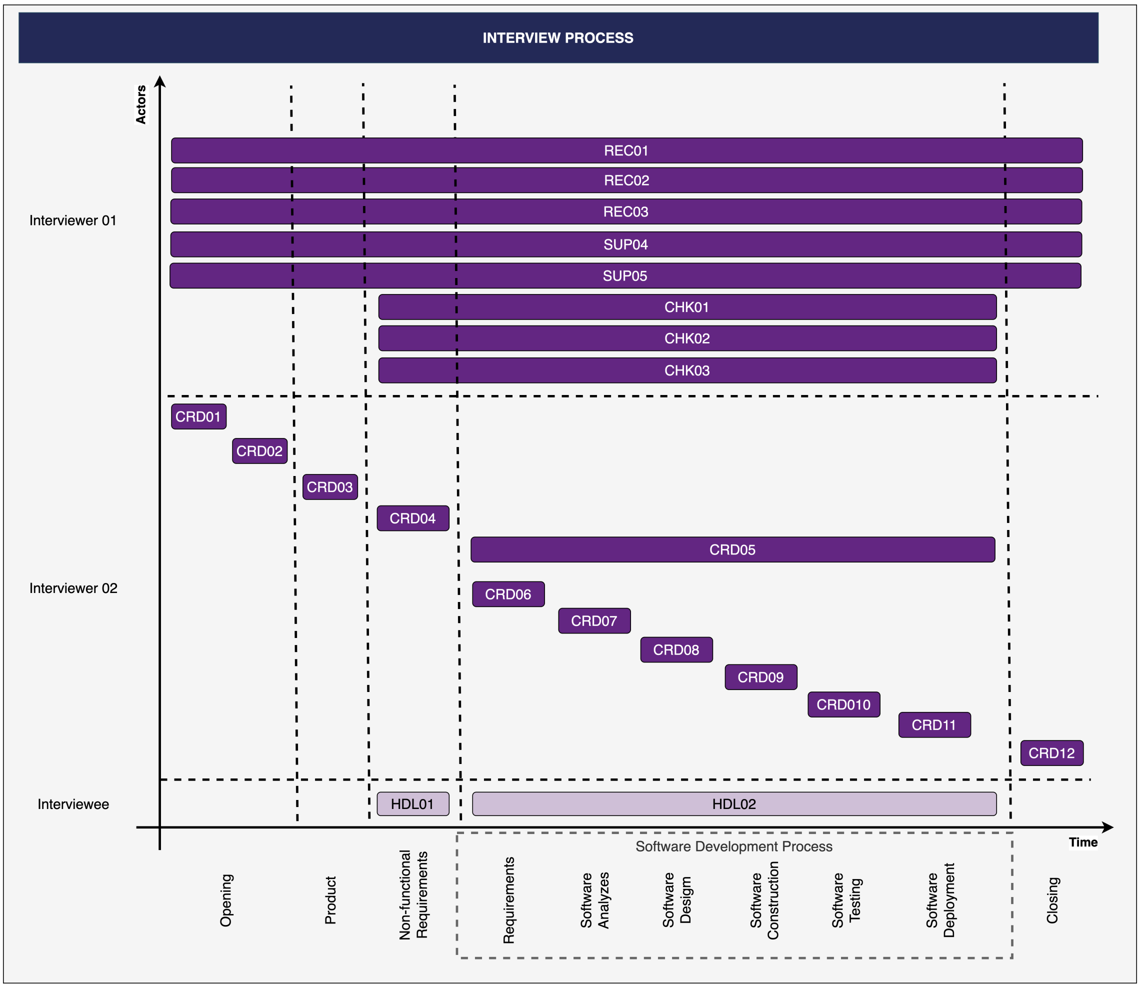


Figure B.1 Interview Process adapted from Giardino et al. (2016)

The interviewers placed two types of interview records: the audio (REC01 and REC02) and the notes (REC03). The first interviewer was responsible for conducting the interview and asking the questions planned in the questionnaire (SUP05, CRD01 to CRD12). The second interviewer made observations and notes and provided checklists (CHK01 to CHK03) when necessary. Some questions needed a support list (HDL01 and HDL02). At the end of the interviews, we finished the recording. At this moment, we asked permission to take pictures of the environment, to check the artifacts and tools described in the interview.

B.1 INTERVIEW PACKAGE

The interview package is structured to contain the artifacts used in all activities involved in the interview process (Table B.1). The content of the interview package is at http://t.ly/Hv3e.

Table B.1: Interview Package

ID	Item	Description
SUP	Support Material	Material that gives support to the researchers
TMP	Templates	Documents templates that might be adapted to be used with each startups.
CRD	Cards	The Interview Script grouped by cards (questions, definition, etc.) to the Interview
CHK	Checklists	Useful lists with Software Engineering concepts that might use to help the researcher during the interview.
HLT	Handlist	Material provided to the startup (documentation, artefacts, non-functional Requirements).
REC	Recordings	Material used to help recordings made during the interview.
TOL	Tools	Tool-set that support the interview process.

Table B.1 presents the following items: general templates to conduct and record information of the sampled companies; support materials aim to help researchers; providing tools to control the process and structure the interview conduction; topic cards used during the interview; well-known checklist practices; tools and methodologies which, the company might have used; hand lists to provide to practitioners possible engineering elements they might have used but not mentioned during the interviews; and, a tool for managing the interview conduction, data collection, and analysis.

Table B.2: Interview Package Items

ID	Item	Description
		Support Material
SUP01.en	Package Content	List of all available documents to conduct the interview.
SUP02.en	Package Usage	Model represents the usage time frame of each artifact in the interview package.
SUP03.en	Definition Vocabulary	List of definitions of engineering terms used during the interview.
SUP04.en	Consent Form	Form that needs to be signed by the interviewee agreeing that he accepts to participate in the interview.
SUP05.en	Startup Info	Information of a single startup related to interviewee and company data; developed product features and qualities.
		Template
TMP01.en	Startup Overview	Startups' information related to name, category, website, and a brief description of their main product or service.
TMP02.en	Invitation for Participation	Invitation for interview by email for informing candidate startups to take part in our research.
TMP03.en	Thanks Mail	Mail to answer to startups that have confirmed their participation in the interviews.
TMP04.en	Cold Call	Script describing how to call startups that have not answered Ack Mail.
		Cards
CRD01.en	Kick-off	Script to start the interview session introducing the research topic and interviewers.
CRD02.en	Opening Question	Script describing the first questions to ask for making the interviewee confident.
CRD03.en	Feature Elicitation	Questions to elicit the main features of the developed product.
CRD04.en	Non-Functional	Questions to elicit the main quality aspects considered during the development process.
CRD05.en	Process	Script to elicit if standard processes and methodologies have been considered during development.
CRD06.en	Requirements	Script to elicit main methods, tools, and measures used during requirements specifications.
CRD07.en	Analysis	Script to elicit main methods, tools, and measures used to analyze critical parts of the project and feasibility assessment.

Continue on the next page

Table B.2 – *Interview Questions (Continuation)*

ID	**Item**	**Description**
CRD08.en	Design	Questions about the high-level architecture decision made during the development process, methods, tools, and measures.
CRD09.en	Implementation	Questions to elicit the implementation methods, tools, and measures.
CRD10.en	Testing	Questions to elicit the methods, tools, and measures for validating and verifying the developed product.
CRD11.en	Deployment	Questions to elicit methods, tools, and measures during product deployment.
CRD12.en	Cool-down	Script for thanking the interviewee for his participation and recalling the follow-up questionnaire.
	Checklists	
CHK01.en	Practices	List of best practices in software engineering.
CHK02.en	Tools	List of the most used tools by practitioners.
CHK03.en	Methodologies	List of the software development methodologies described in software engineering.
	Hands Lists	
HDL01.en	Qualities List	List of qualities mentioned from ISO 9126.
NDL02.en	SE Artifacts List	List of well-known engineering artifacts most software engineers use.
	Follow-up	
FLU01.en	Questionnaire	Follow-up questions for rating adapted methods' accuracy and engineering artifacts' accuracy.
FLU02.en	Result Summary	Analysis of obtained results from the interview.
	Recordings	
REC01.pt	Audio	Audio records of the interview (computer record).
REC02.pt	Audio	Audio records of the interview (smartphone record).
REC03.en	Notes	Written notes obtained during the interview.

End of table

B.2 INTERVIEW QUESTIONS

During the interview process, we developed scripts to conduct the interviews. Table B.3 presents the questions grouped by themes.

Table B.3: Interview Questions

ID	Question
	Kick-off
Q1.0	Do you have any questions before we start?
	Opening Questions
Q2.0	When was *company-name* founded?
Q2.1	How was the initial team composed? And now?
Q2.2	What was your role initially?
Q2.3	How long did it take to release *product-name* to the public the first version of your product?
	Product
Q3.0	Is *product-name* company name's your first product?
Q3.1	Does *product-name* still represent a good part of your current core business?
Q3.2	Can you briefly describe it? What does it do?
Q3.3	Could you help me to write a list of this product's main features on this whiteboard? Let's try to write your system's essential functionalities (around 3-5).
	Non-functional Requirements
Q.4.0	For example, you have a *something* in place... I guess that, in this case, is important to *some-important-non-Functional-stuff*?
Q4.1	Why was this important?
Q4.2	How did you realize it?
	Software Development Process
Q5.0	Did you use any specific development methodology?
Q5.1	Did you use any project management process? How do you schedule the progress of your project?
Q5.2	If any, who was the manager? (critical decision)
Q5.3	Working hours?
Q5.4	Speed and time pressure?
	Software Requirements
Q6.0	Where does the idea behind *product-name* come from?
Q6.1	Did you discuss the idea with other founders/team members? Are any other stakeholders involved in the discussion? Did you document it?
Q6.2	Have you formalized the behavior/requirements you wanted to implement in the first release? How?
Q6.3	Did you structure/organize the requirements?
Q6.4	What happened when requirements were changed, added, or deleted? How did you manage them? Any tools?
Q6.5	Did you trace functionalities/requirements during subsequent activities?
	Software Analyzes

Continue on the next page

Table B.3 – *Interview Questions (Continuation)*

ID	Question
Q7.0	Did you analyze the project's main challenges from the development perspective and how to mitigate them? Did you document it?
Q7.1	Did you consider the skills and time needed for realizing the project? Did you document it?
Q7.2	Have you applied any measure to assess feasibility? And potential risks?
Q7.3	Any particular considerations for critical requirements?
	Software Design
Q8.0	Have you considered any design architecture before implementing the actual code? Have you structured your system in different parts? How do they interact?
Q8.1	Have you created models or diagrams for documenting those decisions?
Q8.2	Have you considered measuring your architectural decisions? Such as the maintainability effort required to change a component of your system.
Q8.3	Have you considered well-known standards to adopt (such as design or architectural patterns)? Did you document them?
Q8.4	Have you utilized any particular tools for designing your system?
	Software Construction
Q9.0	How did you approach coding the first days? You had the idea, and you had the requirements and the design... and then?
Q9.1	Have you considered any workflow guidelines before or during the coding phase?
Q9.2	How did you divide the work between team members?
Q9.3	How did you manage the code base?
Q9.4	What documentation have you produced during coding?
Q9.5	Have you considered any configuration process for the development environment? Did you document it?
Q9.6	How did you manage issues and bugs?
Q9.7	Did you monitor aspects of your development such as team productivity, code size/complexity, etc?
Q9.8	Which are the tools that have helped you produce code?
Q9.9	What programming language did you use?
	Software Testing
Q10.0	Did you perform any kind of tests for the implemented code? Such as acceptance, unit, integration, and system tests.
Q10.1	When did you write the tests? Are they documented?
Q10.2	Quality assurance was an important concern? (to deduct also from the discussed qualities)
Q10.3	Have you conducted any verification and validation process? Did anyone try your product before the first release? Any reports and analysis of results?
Q10.4	Have you conducted any measurements for assessing the validation and verification results?

Continue on the next page

Table B.3 – *Interview Questions (Continuation)*

ID	**Question**
Q10.5	Have you utilized any specific tools for performing testing?
	Software Deployment
Q11.0	How did you deploy your project?
Q11.1	Have you utilized any specific tools?
	Closing Questions
Q12.0	Have you considered improving the development process (in terms of efficiency and effectiveness)?
Q12.1	Have you experienced a drop-down performance during the development, if any? What could be the reason, and when did it happen?
Q12.2	What are the most valuable improvements you would apply with perfect hindsight?

End of table

APPENDIX C

CASE STUDY ANALYSIS NETWORK

This appendix presents the synthesis of the analysis networks of the fourteen organizations investigated to know the context of software development in startups by the investigated research propositions, which were the factors: Human, Organizational, and Technical. Notably, new factors were identified in these categories, identified by the NPA code, plus an identifying number followed by the name. For example, NPA.XX - <name>. Furthermore, a new proposition emerged from the data, called external factors (challenges and trade-offs).

C.1 HUMAN-ASPECTS RECOMMENDATIONS

There are human factors that influence software development in startups.

C.2 TECHNICAL RECOMMENDATIONS

There are technical factors that influence software development in startups.

C.3 ORGANIZATIONAL RECOMMENDATIONS

There are organizational factors that influence software development in startups.

Here we present the results of the analysis points and several points of view that we find relevant to provide valuable insights for researchers and professionals in the field. Table C.10 presents the results of the codification of Points of Analysis through Categories (axial codes) and the Individual Cases of Software Startups. Table C.11 presents the results of the codification of Points of Analysis through Categories (axial codes) and the Individual Cases of Software Startups. Table C.12 presents the results of the codification of Points of Analysis through Categories (axial codes) and Software Startups Actual Stages. Table C.13 presents the results of the codification of Points of Analysis through Categories (axial codes) and the Human Aspects.

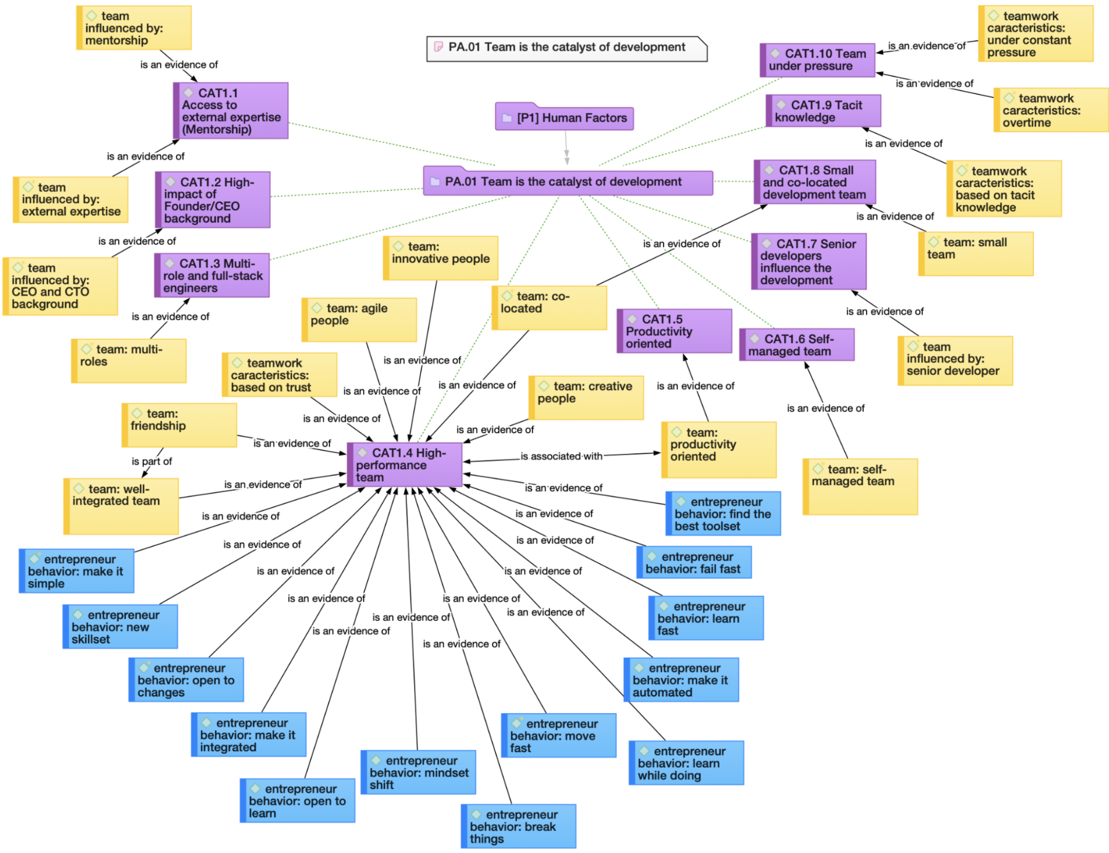


Figure C.1 PA.01 Team is the catalyst of development.

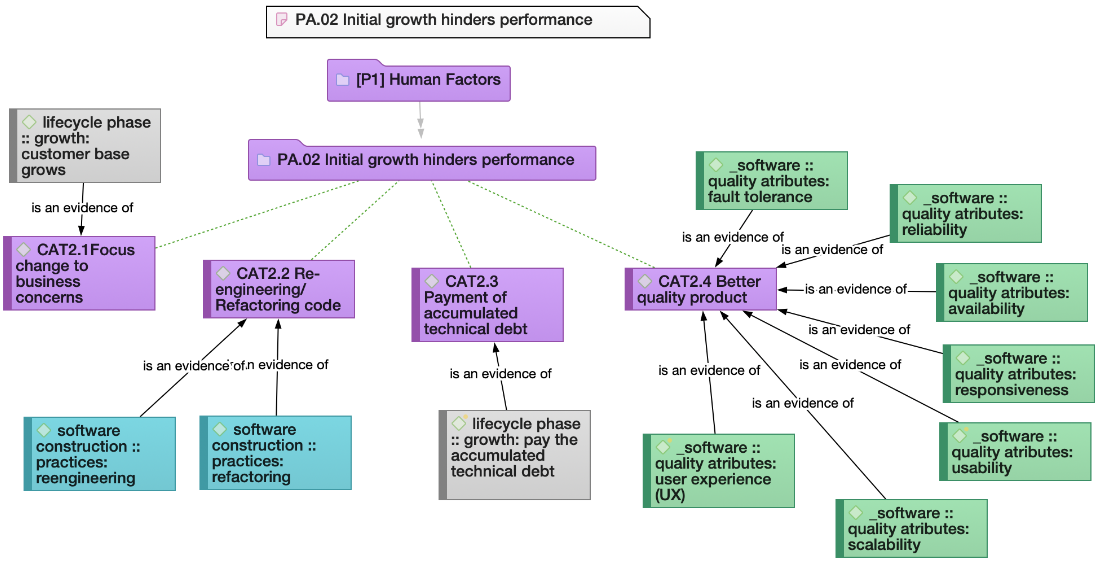


Figure C.2 PA.02 Initial growth hinders performance.

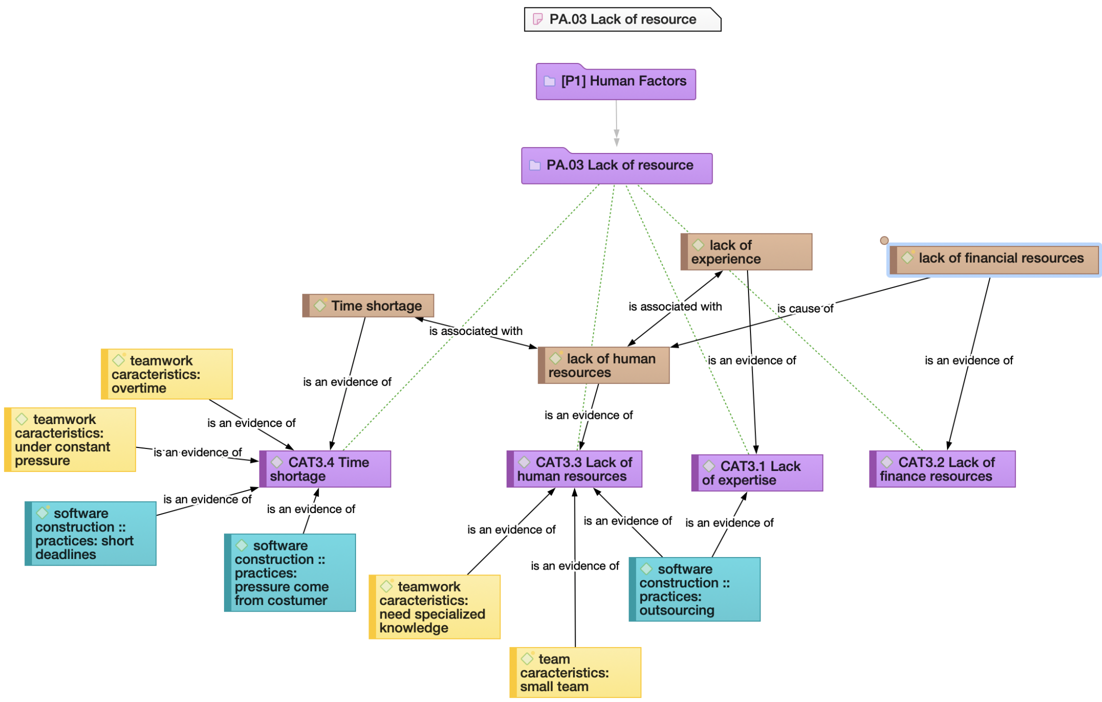


Figure C.3 PA.03 Lack of resources.

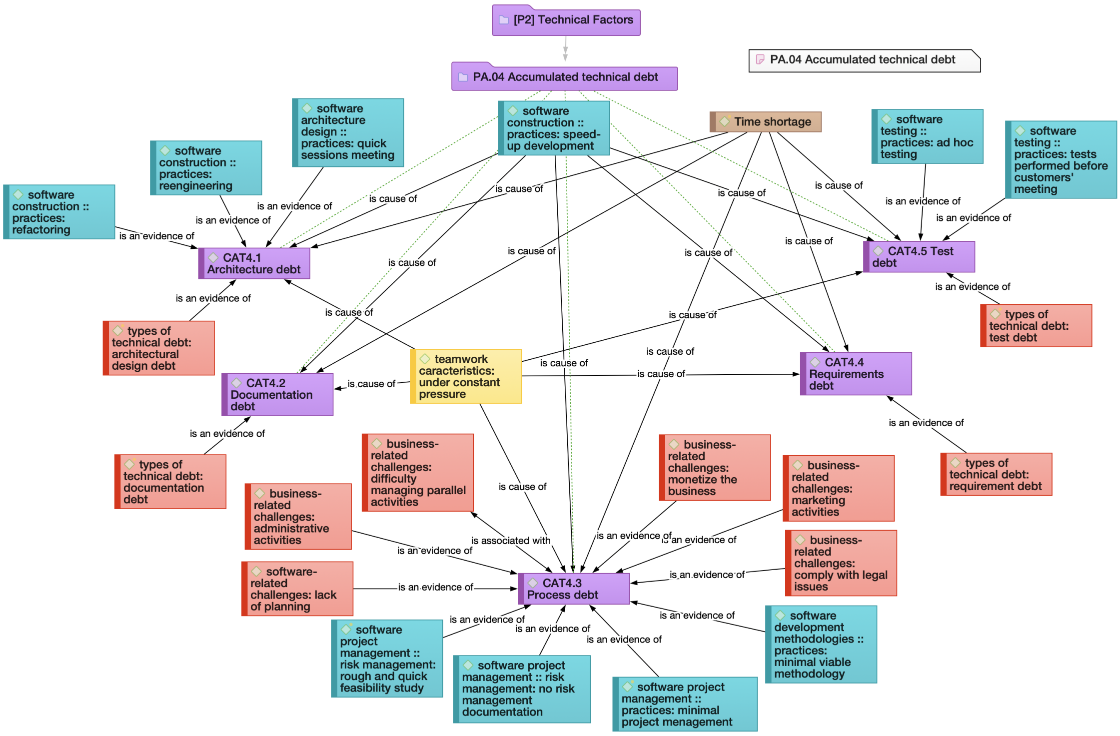


Figure C.4 PA.04 Accumulated Technical Debt.

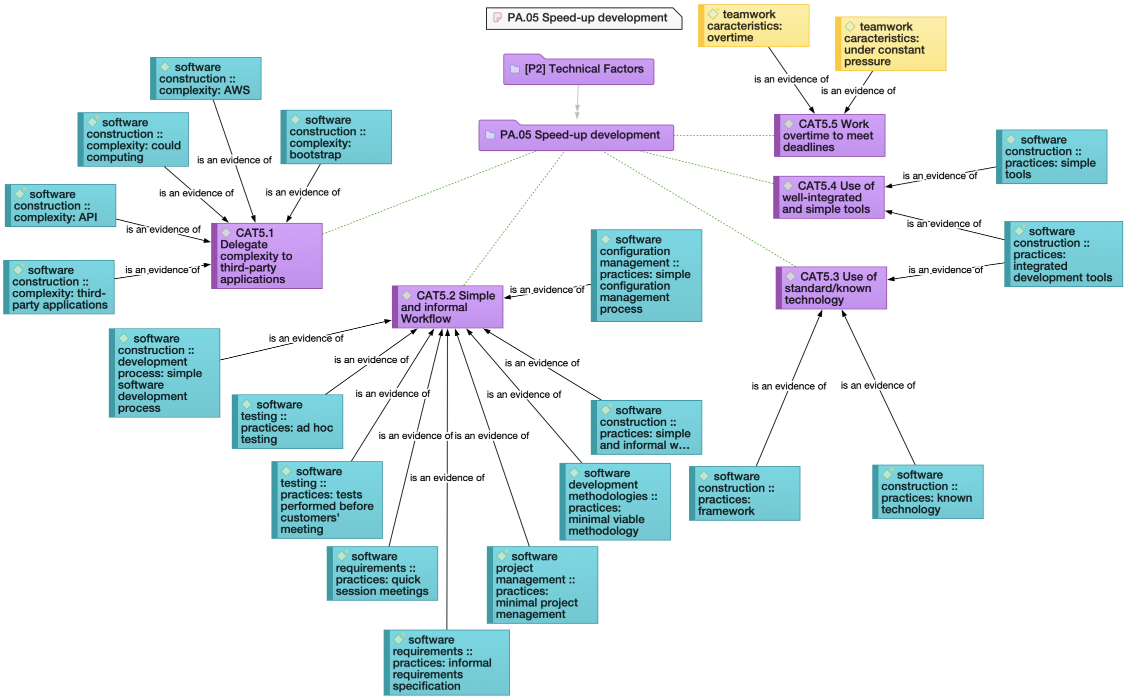


Figure C.5 PA.05 Speed-up Development

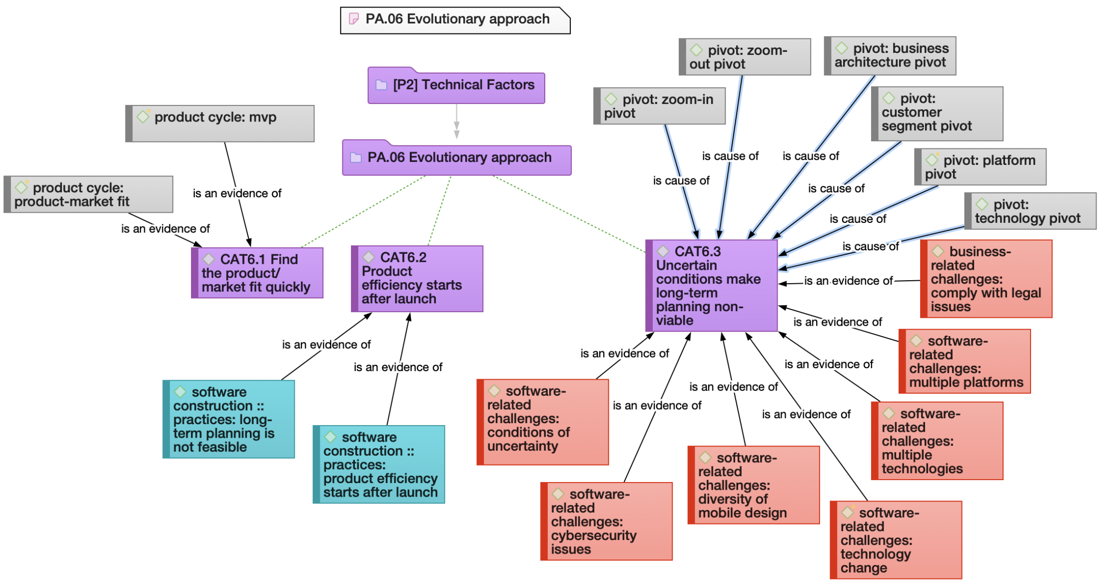


Figure C.6 PA.06 Evolutionary Approach.

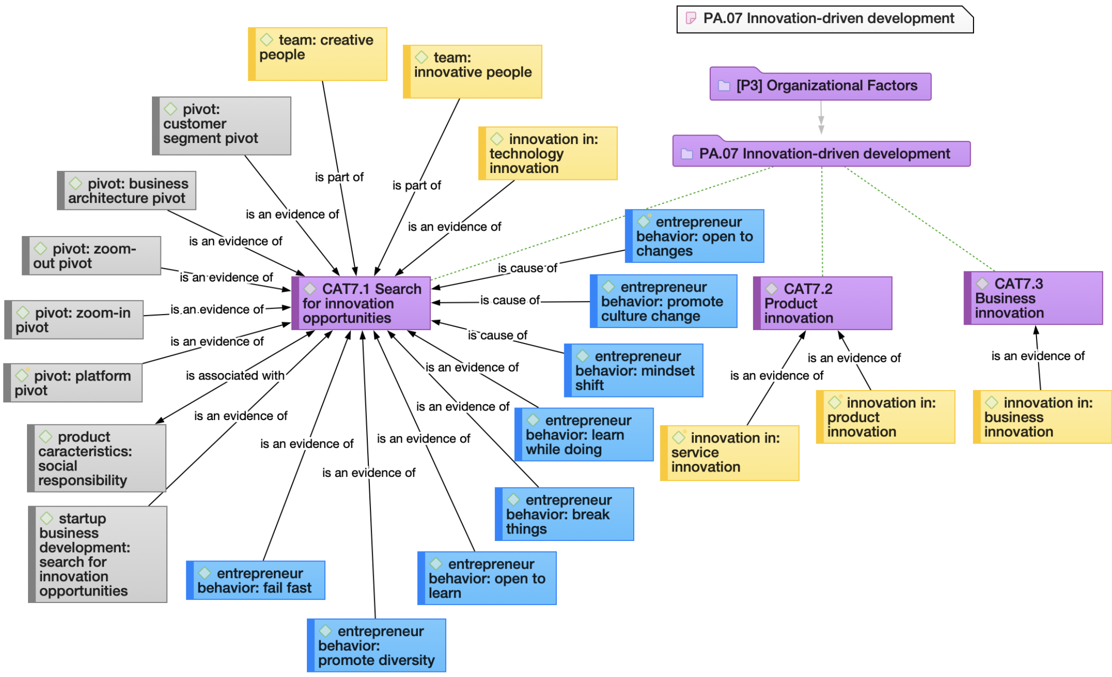


Figure C.7 PA.07 Innovation-driven Development.

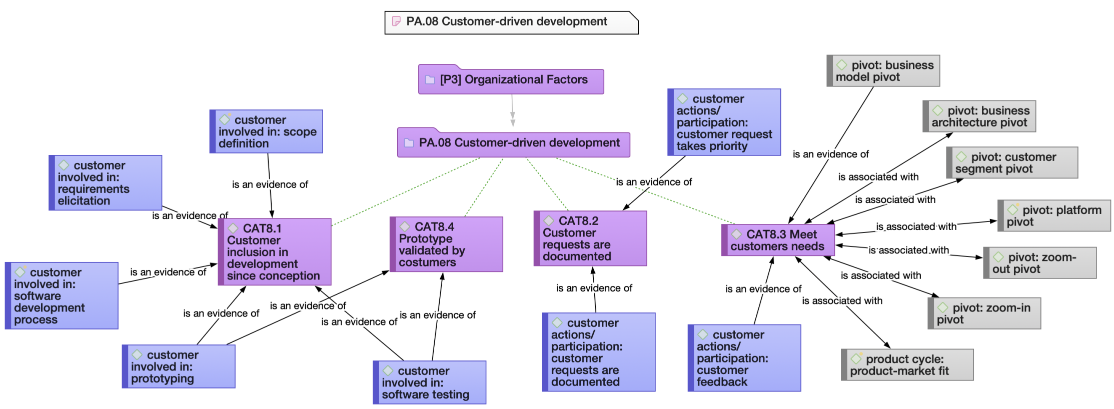


Figure C.8 PA.08 Customer-driven Development.

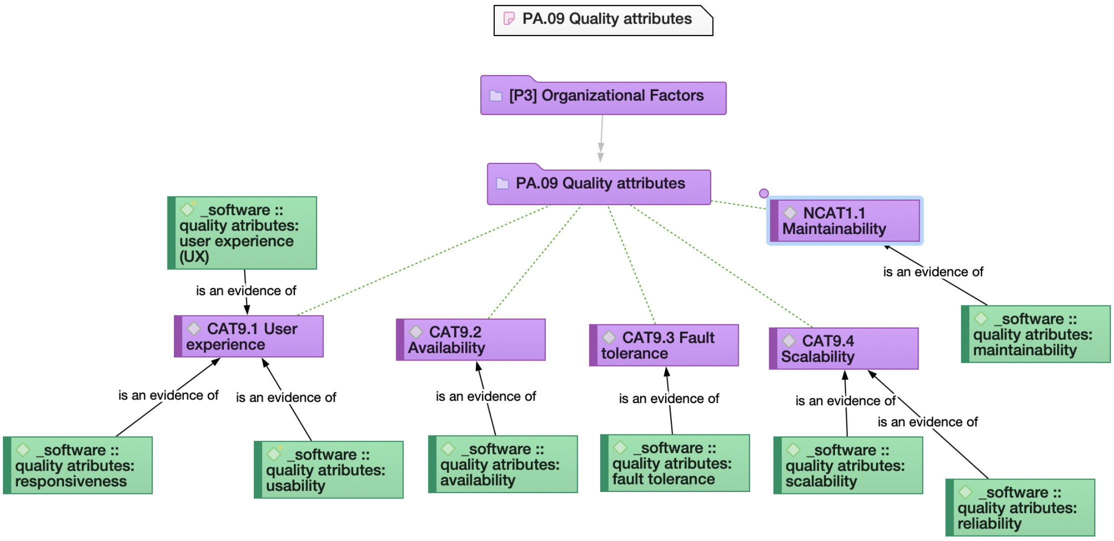
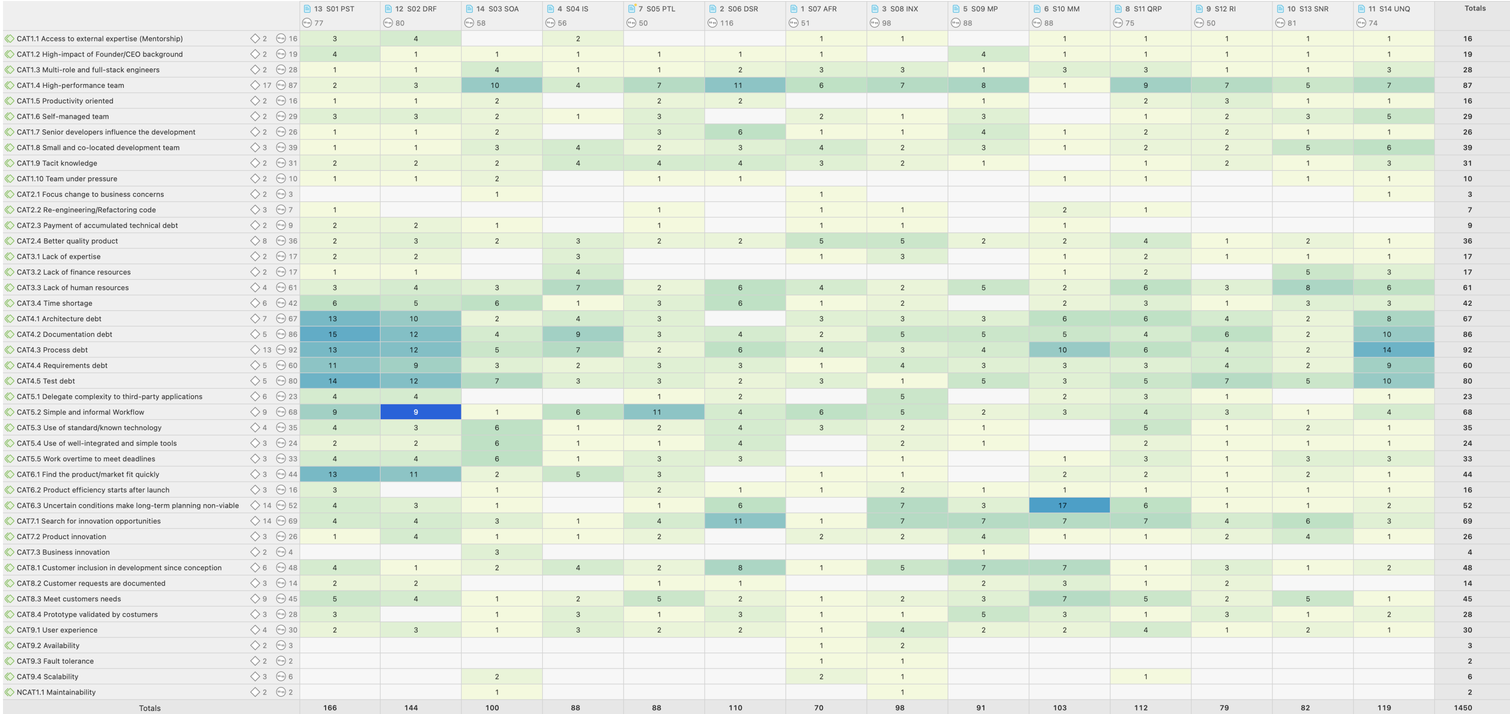


Figure C.9 PA.09 Quality Attributes.



			13 S01 PST	12 S02 DRF	14 S03 SOA	4 S04 IS	7 S05 PTL	2 S06 DSR	1 S07 AFR	3 S08 INX	5 S09 MP	6 S10 MM	8 S11 QRP	9 S12 RI	10 S13 SNR	11 S14 UNQ	Totals
			77	80	58	56	50	116	51	98	88	88	75	50	81	74	
CAT1.1 Access to external expertise (Mentorship)	2	16	3	4		2			1	1		1	1	1	1	1	16
CAT1.2 High-impact of Founder/CEO background	2	19	4	1	1	1	1	1	1		4	1	1	1	1	1	19
CAT1.3 Multi-role and full-stack engineers	2	28	1	1	4	1	1	2	3	3	1	3	3	1	1	3	28
CAT1.4 High-performance team	17	87	2	3	10	4	7	11	6	7	8	1	9	7	5	7	87
CAT1.5 Productivity oriented	2	16	1	1	2		2	2			1		2	3	1	1	16
CAT1.6 Self-managed team	2	29	3	3	2	1	3		2	1	3		1	2	3	5	29
CAT1.7 Senior developers influence the development	2	26	1	1	2		3	6	1	1	4	1	2	2	1	1	26
CAT1.8 Small and co-located development team	3	39	1	1	3	4	2	3	4	2	3	1	2	2	5	6	39
CAT1.9 Tacit knowledge	2	31	2	2	2	4	4	4	3	2	1		1	2	1	3	31
CAT1.10 Team under pressure	2	10	1	1	2		1	1				1	1		1	1	10
CAT2.1 Focus change to business concerns	2	3			1				1							1	3
CAT2.2 Re-engineering/Refactoring code	3	7	1				1		1	1		2	1				7
CAT2.3 Payment of accumulated technical debt	2	9	2	2	1		1		1	1		1					9
CAT2.4 Better quality product	8	36	2	3	2	3	2	2	5	5	2	2	4	1	2	1	36
CAT3.1 Lack of expertise	2	17	2	2		3			1	3		1	2	1	1	1	17
CAT3.2 Lack of finance resources	2	17	1	1		4						1	2		5	3	17
CAT3.3 Lack of human resources	4	61	3	4	3	7	2	6	4	2	5	2	6	3	8	6	61
CAT3.4 Time shortage	6	42	6	5	6	1	3	6	1	2		2	3	1	3	3	42
CAT4.1 Architecture debt	7	67	13	10	2	4	3		3	3	3	6	6	4	2	8	67
CAT4.2 Documentation debt	5	86	15	12	4	9	3	4	2	5	5	5	4	6	2	10	86
CAT4.3 Process debt	13	92	13	12	5	7	2	6	4	3	4	10	6	4	2	14	92
CAT4.4 Requirements debt	5	60	11	9	3	2	3	3	1	4	3	3	3	4	2	9	60
CAT4.5 Test debt	5	80	14	12	7	3	3	2	3	1	5	3	5	7	5	10	80
CAT5.1 Delegate complexity to third-party applications	6	23	4	4			1	2		5		2	3	1		1	23
CAT5.2 Simple and informal Workflow	9	68	9	9	1	6	11	4	6	5	2	3	4	3	1	4	68
CAT5.3 Use of standard/known technology	4	35	4	3	6	1	2	4	3	2	1		5	1	2	1	35
CAT5.4 Use of well-integrated and simple tools	3	24	2	2	6	1	1	4		2	1		2	1	1	1	24
CAT5.5 Work overtime to meet deadlines	3	33	4	4	6	1	3	3		1		1	3	1	3	3	33
CAT6.1 Find the product/market fit quickly	3	44	13	11	2	5	3		1	1		2	2	1	2	1	44
CAT6.2 Product efficiency starts after launch	3	16	3		1		2	1	1	2	1	1	1	1	1	1	16
CAT6.3 Uncertain conditions make long-term planning non-viable	14	52	4	3	1		1	6		7	3	17	6	1	1	2	52
CAT7.1 Search for innovation opportunities	14	69	4	4	3	1	4	11	1	7	7	7	7	4	6	3	69
CAT7.2 Product innovation	3	26	1	4	1	1	2		2	2	4	1	1	2	4	1	26
CAT7.3 Business innovation	2	4			3						1						4
CAT8.1 Customer inclusion in development since conception	6	48	4	1	2	4	2	8	1	5	7	7	1	3	1	2	48
CAT8.2 Customer requests are documented	3	14	2	2			1	1			2	3	1	2			14
CAT8.3 Meet customers needs	9	45	5	4	1	2	5	2	1	2	3	7	5	2	5	1	45
CAT8.4 Prototype validated by costumers	3	28	3		1	3	1	3	1	1	5	3	1	3	1	2	28
CAT9.1 User experience	4	30	2	3	1	3	2	2	1	4	2	2	4	1	2	1	30
CAT9.2 Availability	2	3							1	2							3
CAT9.3 Fault tolerance	2	2							1	1							2
CAT9.4 Scalability	3	6			2				2	1			1				6
NCAT1.1 Maintainability	2	2			1					1							2
Totals			166	144	100	88	88	110	70	98	91	103	112	79	82	119	1450

Figure C.10 Points of Analysis X Startup Cases.

	Codes	Quotations	Startup :: Game (2, 155)	Startup :: IoT Solution (2, 155)	Startup :: Mobile Application (1, 88)	Startup :: Open Data Platform (1, 50)	Startup :: Software as a Service (3, 262)	Startup :: Web Application (5, 332)	Totals
CAT1.1 Access to external exp...	2	16	2	5	1			8	16
CAT1.2 High-impact of Found...	2	19	2	2	1	1	6	7	19
CAT1.3 Multi-role and full-sta...	2	28	4	4	3	1	7	9	28
CAT1.4 High-performance te...	17	87	12	12	1	7	29	26	87
CAT1.5 Productivity oriented	2	16	2	3		2	5	4	16
CAT1.6 Self-managed team	2	29	8	4		3	5	9	29
CAT1.7 Senior developers infl...	2	26	2	3	1	3	12	5	26
CAT1.8 Small and co-located...	3	39	11	3	1	2	9	13	39
CAT1.9 Tacit knowledge	2	31	4	3		4	7	13	31
CAT1.10 Team under pressure	2	10	2	2	1	1	3	1	10
CAT2.1 Focus change to busi...	2	3	1				1	1	3
CAT2.2 Re-engineering/Refac...	3	7		1	2	1		3	7
CAT2.3 Payment of accumula...	2	9		2	1	1	1	4	9
CAT2.4 Better quality product	8	36	3	7	2	2	6	16	36
CAT3.1 Lack of expertise	2	17	2	4	1			10	17
CAT3.2 Lack of finance resou...	2	17	8	3	1			5	17
CAT3.3 Lack of human resour...	4	61	14	10	2	2	14	19	61
CAT3.4 Time shortage	6	42	6	8	2	3	12	11	42
CAT4.1 Architecture debt	7	67	10	16	6	3	5	27	67
CAT4.2 Documentation debt	5	86	12	16	5	3	13	37	86
CAT4.3 Process debt	13	92	16	18	10	2	15	31	92
CAT4.4 Requirements debt	5	60	11	12	3	3	9	22	60
CAT4.5 Test debt	5	80	15	17	3	3	14	28	80
CAT5.1 Delegate complexity t...	6	23	1	7	2	1	2	10	23
CAT5.2 Simple and informal...	9	68	5	13	3	11	7	29	68
CAT5.3 Use of standard/know...	4	35	3	8		2	11	11	35
CAT5.4 Use of well-integrate...	3	24	2	4		1	11	6	24
CAT5.5 Work overtime to mee...	3	33	6	7	1	3	9	7	33
CAT6.1 Find the product/mark...	3	44	3	13	2	3	2	21	44
CAT6.2 Product efficiency sta...	3	16	2	1	1	2	3	7	16
CAT6.3 Uncertain conditions...	14	52	3	9	17	1	10	12	52
CAT7.1 Search for innovation...	14	69	9	11	7	4	21	17	69
CAT7.2 Product innovation	3	26	5	5	1	2	5	8	26
CAT7.3 Business innovation	2	4					4		4
CAT8.1 Customer inclusion in...	6	48	3	2	7	2	17	17	48
CAT8.2 Customer requests ar...	3	14		3	3	1	3	4	14
CAT8.3 Meet customers needs	9	45	6	9	7	5	6	12	45
CAT8.4 Prototype validated b...	3	28	3	1	3	1	9	11	28
CAT9.1 User experience	4	30	3	7	2	2	5	11	30
CAT9.2 Availability	2	3						3	3
CAT9.3 Fault tolerance	2	2						2	2
CAT9.4 Scalability	3	6		1			2	3	6
NCAT1.1 Maintainability	2	2					1	1	2
Totals			201	256	103	88	301	501	1450

Figure C.11 Points of Analysis X Business Model.

			Actual Stage :: Dead	Actual Stage :: Early-stage	Actual Stage :: Growth-Satge	Actual Stage :: Mature	Actual Stage :: Scaling	Totals
			2 136	9 704	3 202	0 0	0 0	
CAT1.1 Access to external expertise (Mentorship)	2	16	6	6	4			16
CAT1.2 High-impact of Founder/CEO background	2	19	2	11	6			19
CAT1.3 Multi-role and full-stack engineers	2	28	2	21	5			28
CAT1.4 High-performance team	17	87	7	62	18			87
CAT1.5 Productivity oriented	2	16	1	10	5			16
CAT1.6 Self-managed team	2	29	4	18	7			29
CAT1.7 Senior developers influence the development	2	26	1	19	6			26
CAT1.8 Small and co-located development team	3	39	5	29	5			39
CAT1.9 Tacit knowledge	2	31	6	18	7			31
CAT1.10 Team under pressure	2	10	1	6	3			10
CAT2.1 Focus change to business concerns	2	3		3				3
CAT2.2 Re-engineering/Refactoring code	3	7		4	3			7
CAT2.3 Payment of accumulated technical debt	2	9	2	4	3			9
CAT2.4 Better quality product	8	36	6	22	8			36
CAT3.1 Lack of expertise	2	17	5	8	4			17
CAT3.2 Lack of finance resources	2	17	5	9	3			17
CAT3.3 Lack of human resources	4	61	11	39	11			61
CAT3.4 Time shortage	6	42	6	24	12			42
CAT4.1 Architecture debt	7	67	14	31	22			67
CAT4.2 Documentation debt	5	86	21	43	22			86
CAT4.3 Process debt	13	92	19	52	21			92
CAT4.4 Requirements debt	5	60	11	32	17			60
CAT4.5 Test debt	5	80	15	43	22			80
CAT5.1 Delegate complexity to third-party applications	6	23	4	11	8			23
CAT5.2 Simple and informal Workflow	9	68	15	29	24			68
CAT5.3 Use of standard/known technology	4	35	4	20	11			35
CAT5.4 Use of well-integrated and simple tools	3	24	3	16	5			24
CAT5.5 Work overtime to meet deadlines	3	33	5	18	10			33
CAT6.1 Find the product/market fit quickly	3	44	16	10	18			44
CAT6.2 Product efficiency starts after launch	3	16		10	6			16
CAT6.3 Uncertain conditions make long-term planning non-via...	14	52	3	38	11			52
CAT7.1 Search for innovation opportunities	14	69	5	49	15			69
CAT7.2 Product innovation	3	26	5	17	4			26
CAT7.3 Business innovation	2	4		4				4
CAT8.1 Customer inclusion in development since conception	6	48	5	36	7			48
CAT8.2 Customer requests are documented	3	14	2	8	4			14
CAT8.3 Meet customers needs	9	45	6	24	15			45
CAT8.4 Prototype validated by costumers	3	28	3	20	5			28
CAT9.1 User experience	4	30	6	16	8			30
CAT9.2 Availability	2	3		3				3
CAT9.3 Fault tolerance	2	2		2				2
CAT9.4 Scalability	3	6		5	1			6
NCAT1.1 Maintainability	2	2		2				2
Totals			232	852	366	0	0	1450

Figure C.12 Points of Analysis X Startups Actual Stage.

		Actual Stage :: Dead (2, 136)	Actual Stage :: Early-stage (9, 704)	Actual Stage :: Growth-Satge (3, 202)	Actual Stage :: Mature (0, 0)	Actual Stage :: Scaling (0, 0)	Totals
entrepreneur behavior: break things	1		1				1
entrepreneur behavior: fail fast	12		8	4			12
entrepreneur behavior: feel like you belong/contribute	1			1			1
entrepreneur behavior: find the best toolset	9		9				9
entrepreneur behavior: learn fast	18		15	3			18
entrepreneur behavior: learn while doing	19	1	15	3			19
entrepreneur behavior: make it automated	15		9	6			15
entrepreneur behavior: make it integrated	15		10	5			15
entrepreneur behavior: make it simple	31	6	16	9			31
entrepreneur behavior: mindset shift	14		11	3			14
entrepreneur behavior: move fast	14		11	3			14
entrepreneur behavior: new skillset	8		8				8
entrepreneur behavior: open to changes	15		10	5			15
entrepreneur behavior: open to learn	16	1	12	3			16
entrepreneur behavior: promote culture change	10		9	1			10
entrepreneur behavior: promote diversity	5		5				5
Totals		8	149	46	0	0	203

Figure C.13 Points of Analysis X Human Aspects.

APPENDIX D

ATLAS.TI REPORTS

D.1 ATLAS.TI DOCUMENTS AND GROUPS REPORT

ATLAS.ti Report
Startups & Software Engineering Practices
Documents grouped by Document Groups
Report created by Renata Maria de Souza Santos on 9 Jul 2023

Table D.1: ATLAS.ti Documents Grouping Structure

Documents Folder Structure
1st Round Interviews (4 Documents)
∟ 4 S04
∟ 12 S02
∟ 13 S01
∟ 14 S03
2nd Round Interviews (4 Documents)
∟ 1 S07
∟ 2 S06
∟ 3 S08
∟ 7 S05
3rd Round Interviews (2 Documents)
∟ 5 S09
∟ 6 S10
4th Round Interviews (2 Documents)
∟ 8 S11
∟ 9 S12
5th Round Interviews (2 Documents)
∟ 10 S13
∟ 11 S14
Actual Stage :: Dead (2 Documents)
∟ 4 S04
∟ 12 S02
Actual Stage :: Early-stage (9 Documents)
∟ 1 S07
∟ 2 S06

Continue on the next page

Table D.1 – *Documents Folder Structure (Continuation)*

Documents Folder Structure
∟ 3 S08
∟ 5 S09
∟ 6 S10
∟ 9 S12
∟ 10 S13
∟ 11 S14
∟ 14 S03
Actual Stage :: Growth-Satge (3 Documents)
∟ 7 S05
∟ 8 S11
∟ 13 S01
Startup :: Game (2 Documents)
∟ 10 S13
∟ 11 S14
Startup :: IoT Solution (2 Documents)
∟ 8 S11
∟ 12 S02
Startup :: Mobile Application (1 Documents)
∟ 6 S10
Startup :: Open Data Platform (1 Documents)
∟ 7 S05
Startup :: Software as a Service (3 Documents)
∟ 2 S06
∟ 5 S09
∟ 14 S03
Startup :: Web Application (5 Documents)
∟ 1 S07
∟ 3 S08
∟ 4 S04
∟ 9 S12
∟ 13 S01

End of table

D.2 ATLAS.TI MEMOS AND GROUPS REPORT

ATLAS.ti Report
Startups & Software Engineering Practices
Memos grouped by Memo groups
Report created by Renata Maria de Souza Santos on 8 Jul 2023

Table D.2: ATLAS.ti Memos Grouping Structure

Memos Folder Structure
Cases (14 Memos)
∟ S1
∟ S2
∟ S3
∟ S4
∟ S5
∟ S6
∟ S7
∟ S8
∟ S9
∟ S10
∟ S11
∟ S12
∟ S13
∟ S14
P1 Human factors (3 Memos)
∟ PA.01 Team is the catalyst of development
∟ PA.02 Initial growth hinders performance
∟ PA.03 Lack of resource
P2 Technical factors (3 Memos)
∟ PA.04 Accumulated technical debt
∟ PA.05 Speed-up development
∟ PA.06 Evolutionary approach
P3 Organizational factors (3 Memos)
∟ PA.07 Innovation-driven development
∟ PA.08 Customer-driven development
∟ PA.09 Quality attributes
Recommendations (16 Memos)
∟ HA-REC.01 Mindseft shift
∟ HA-REC.02 Learn while doing
∟ HA-REC.03 Accept failure, Fail fast
∟ HA-REC.04 New skill sets
∟ HA-REC.05 Trust each other
∟ O-REC.01 Cultural change
∟ O-REC.02 Promote diversity
∟ T-REC.01 Minimum Viable Process
∟ T-REC.02 Architecture Design
∟ T-REC.03 Practices and Policies
∟ T-REC.04 Requirements with Customers

Continue on the next page

Table D.2 – *Memos Folder Structure (Continuation)*

Memos Folder Structure
∟ T-REC.05 Test Soon, and with Customers
∟ T-REC.06 Make it Simples
∟ T-REC.07 Make it integrated
∟ T-REC.08 Make it automated
∟ T-REC.09 Right Toolset for Your Team
Research Question (5 Memos)
∟ RQ 1: How do startups develop software?
∟ RQ1.1: What is the context of startup software development?
∟ RQ1.2: What software engineering practices are relevant for early-stage startups?
∟ RQ1.3: What quality attributes do startups consider in software development?
∟ RQ1.4: What software development supporting tools do startups use?
Supporting Memos (5 Memos)
∟ Case - Model
∟ Codes
∟ Ideas
∟ Research Diary
∟ To Do

End of table

D.3 ATLAS.TI NETWORK AND GROUPS REPORT

ATLAS.ti Report
Startups & Software Engineering Practices
Network grouped by Network groups
Report created by Renata Maria de Souza Santos on 9 Jul 2023

Table D.3: ATLAS.ti Network Grouping Structure

Network Folder Structure
Proposition 1 (P1) - Human Factores (3 Networks)
∟ PA.01 Team is the catalyst of development
∟ PA.02 Initial growth hinders performance
∟ PA.03 Lack of resource
Proposition 2 (P2) - Technical Factors (3 Networks)
∟ PA.04 Accumulated technical debt
∟ PA.05 Speed-up development
∟ PA.06 Evolutionary approach
Proposition 3 (P3) - Organizational Factors (3 Networks)

Continue on the next page

Table D.3 – *Network Folder Structure (Continuation)*

Network Folder Structure
∟ PA.07 Innovation-driven development
∟ PA.08 Customer-driven development
∟ PA.09 Quality attributes
Recommendations (16 Networks)
∟ HA-REC.01
∟ HA-REC.02
∟ HA-REC.03
∟ HA-REC.04
∟ HA-REC.05
∟ O-REC.01
∟ O-REC.02
∟ T-REC.01
∟ T-REC.02
∟ T-REC.03
∟ T-REC.04
∟ T-REC.05
∟ T-REC.06
∟ T-REC.07
∟ T-REC.08
∟ T-REC.09
Research Questions (4 Networks)
∟ RQ1.1: What is the context of startup software development?
∟ RQ1.2: What software engineering practices are relevant for early-stage startups?
∟ RQ1.3: What quality attributes do startups consider in software development?
∟ RQ1.4: What software development supporting tools do startups use?
Supporting networks (1 Networks)
∟ Propositions [P1, P2, P3]

End of table

APPENDIX E

RECOMMENDATIONS NETWORK

This appendix presents a set of recommendations to support software startups in software engineering practices and the justifications that support adopting such recommendations. These recommendations emerged from the analysis memos made in the interviews.

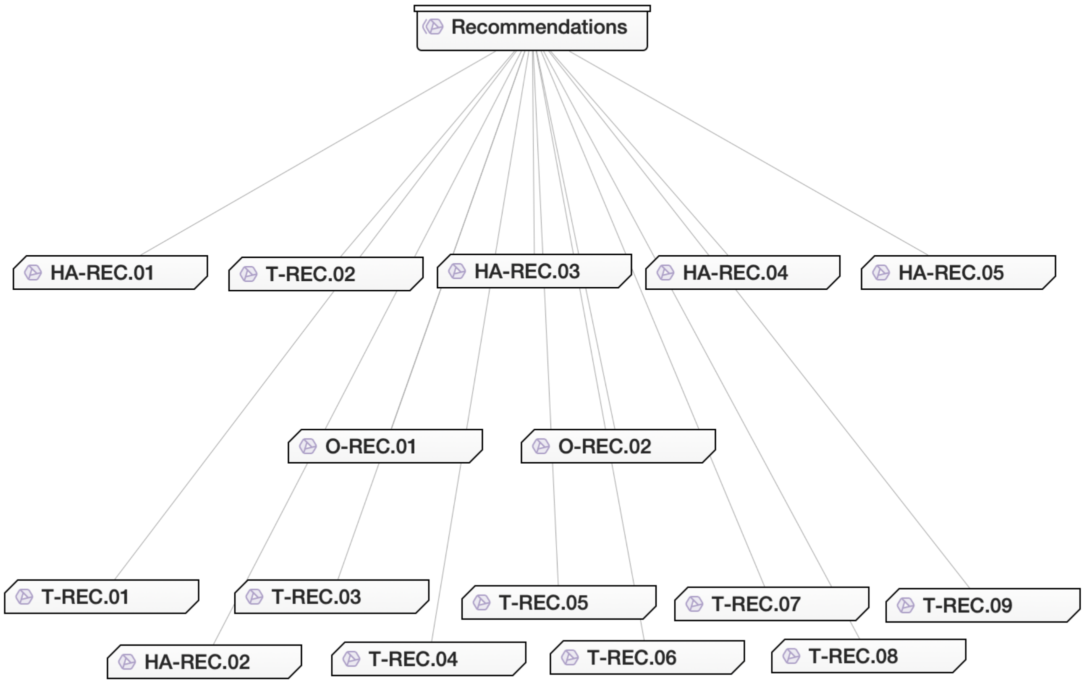


Figure E.1 A Set of Recommendations for Software Startups.

E.1 HUMAN-ASPECTS RECOMMENDATIONS

`HA-REC.01` **Make a mindset shift.** Figure E.2 shows the analysis networks HA-REC.01 recommendations.

`HA-REC.02` **Learn while doing.** Figure E.3 shows the analysis networks HA-REC.02 recommendations.

`HA-REC.03` **Accept failures, fail fast, and learn from them.** Figure E.4 shows the analysis networks HA-REC.01 recommendations.

`HA-REC.04` **Have a team with new skill sets.** Figure E.5 shows the analysis networks HA-REC.05 recommendations.

`HA-REC.05` **Trust each other.** Figure E.6 shows the analysis networks HA-REC.07 recommendations.

E.2 TECHNICAL RECOMMENDATIONS

`T-REC.01` **Define a minimum viable process.** Figure E.7 shows the analysis networks T-REC.01 recommendations.

`T-REC.02` **Spend minimal time and effort on defining the best possible software architecture.** Figure E.8 shows the analysis networks T-REC.02 recommendations.

`T-REC.03` **Define software development coding practices and policies with the team.** Figure E.9 shows the analysis networks T-REC.03 recommendations.

`T-REC.04` **Requirement** Figure E.10 shows the analysis networks T-REC.04 recommendations.

`T-REC.05` **Test** Figure E.11 shows the analysis networks T-REC.05 recommendations.

`T-REC.06` **Make [it] simple.** Figure E.12 shows the analysis networks T-REC.06 recommendations.

`T-REC.07` **Make [it] integrated.** Figure E.13 shows the analysis networks T-REC.07 recommendations.

`T-REC.08` **Make [it] automated.** Figure E.14 shows the analysis networks T-REC.08 recommendations.

`T-REC.09` **Find the toolset that helps your team to accelerate software development.** Figure E.15 shows the analysis networks T-REC.09 recommendations.

E.3 ORGANIZATIONAL RECOMMENDATIONS

`O-REC.01` **Make organizational culture change.** Figure E.16 shows the analysis networks O-REC.01 recommendations.

`O-REC.02` **Promote diversity.** Figure E.17 shows the analysis networks HA-REC.04 recommendations.

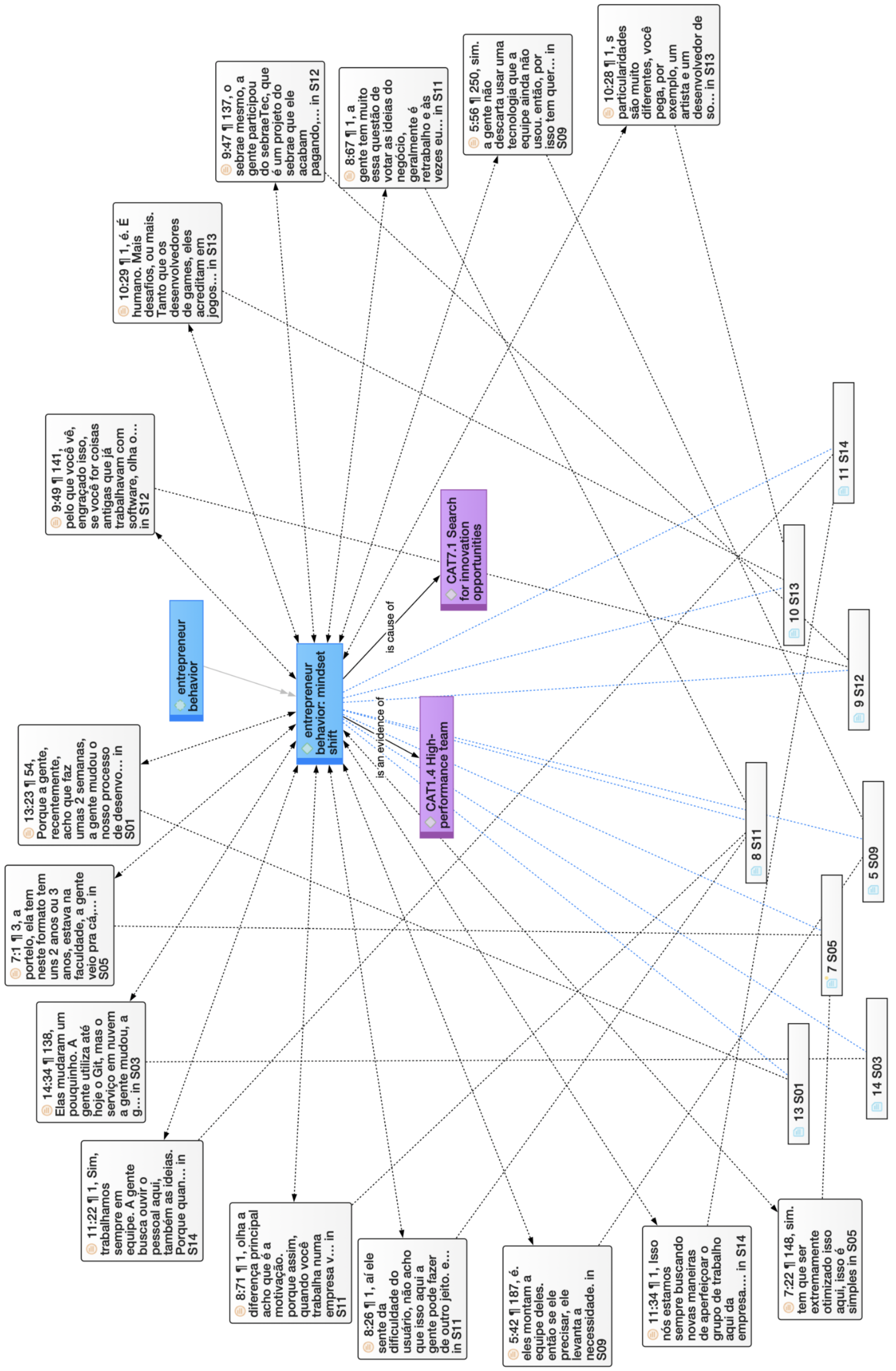


Figure E.2 HA-REC.01 Make a mindset shift.

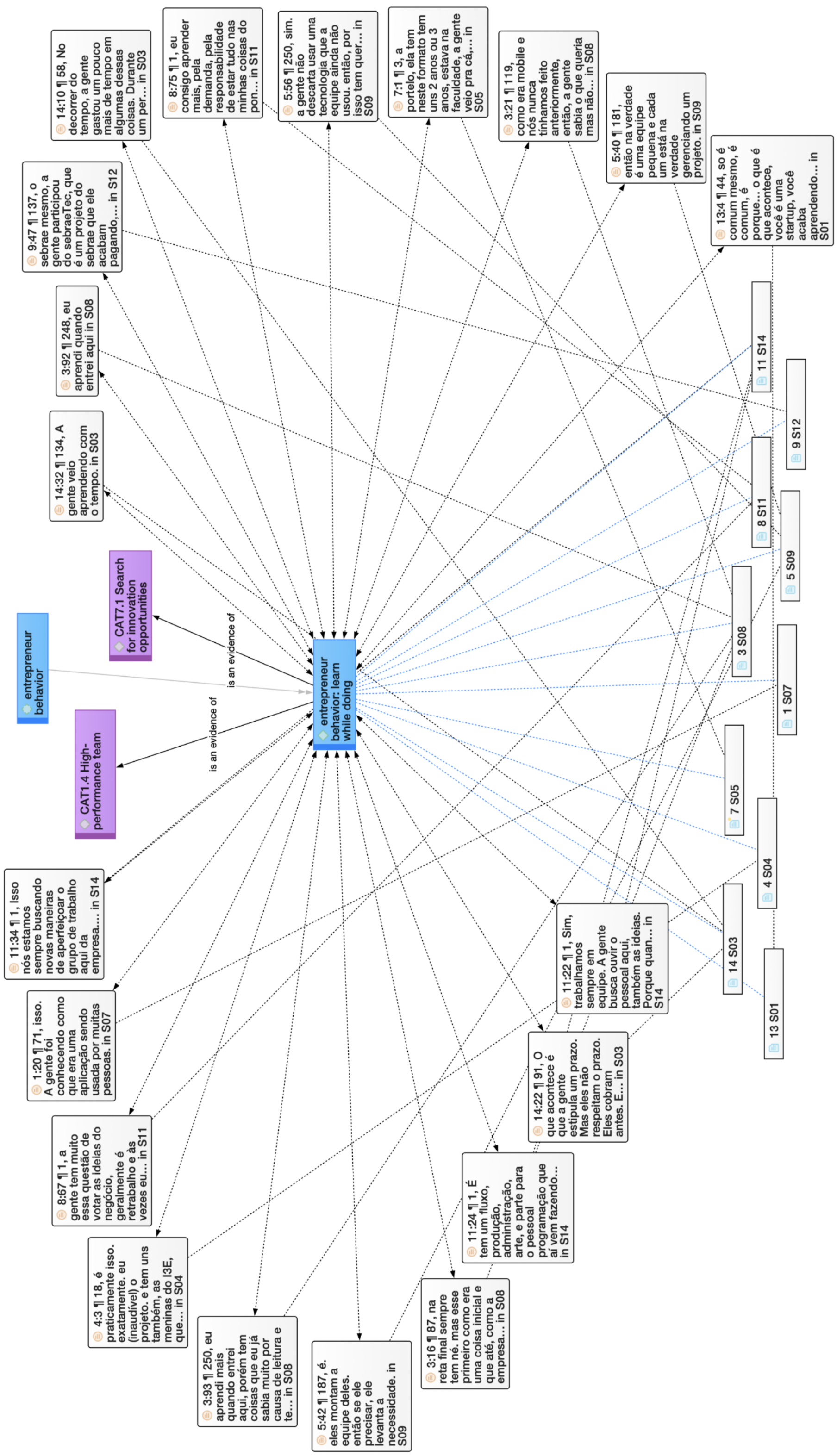


Figure E.3 HA-REC.02 Learn while doing.

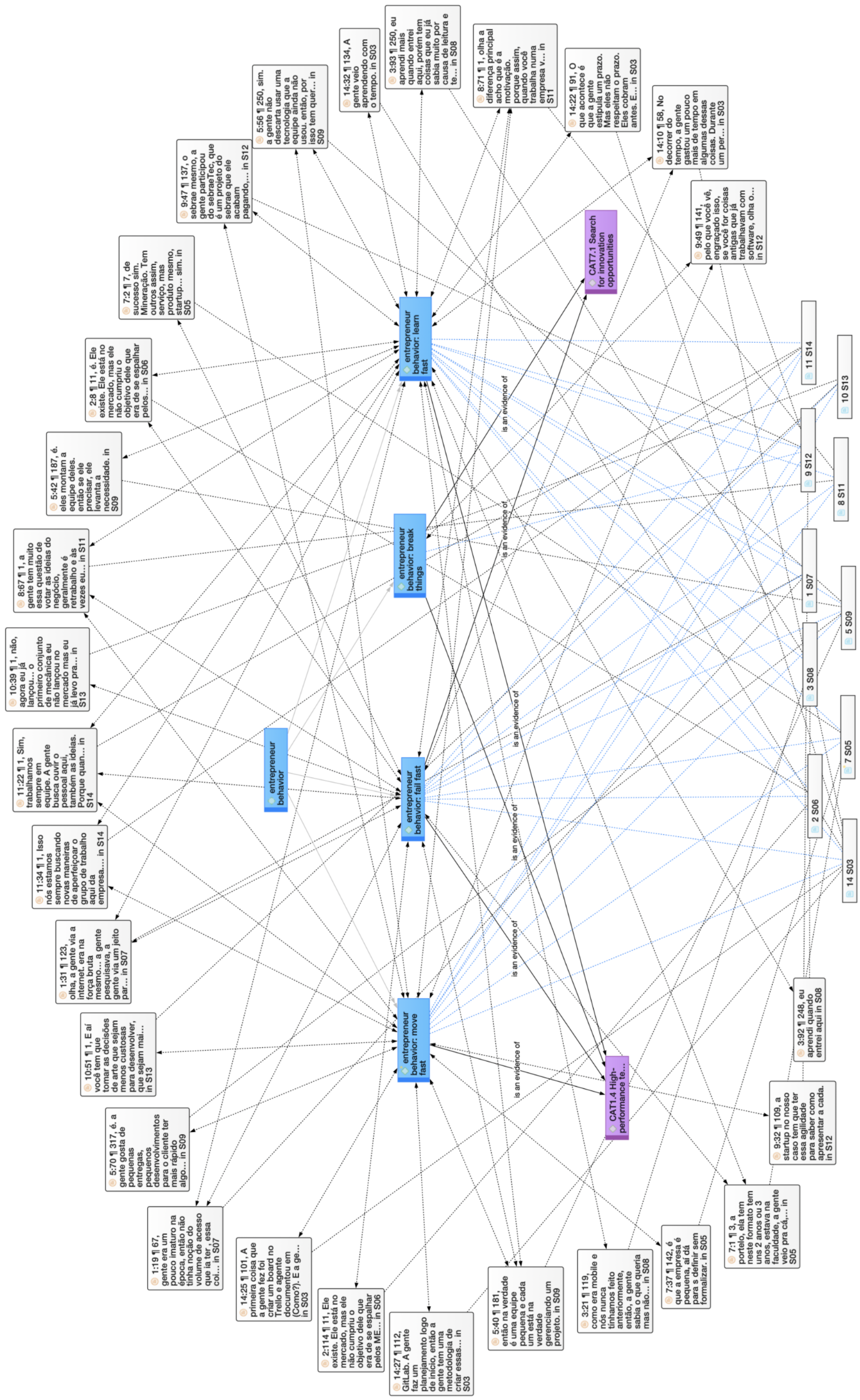


Figure E.4 HA-REC.03 Accept failures, fail fast, and learn from them.

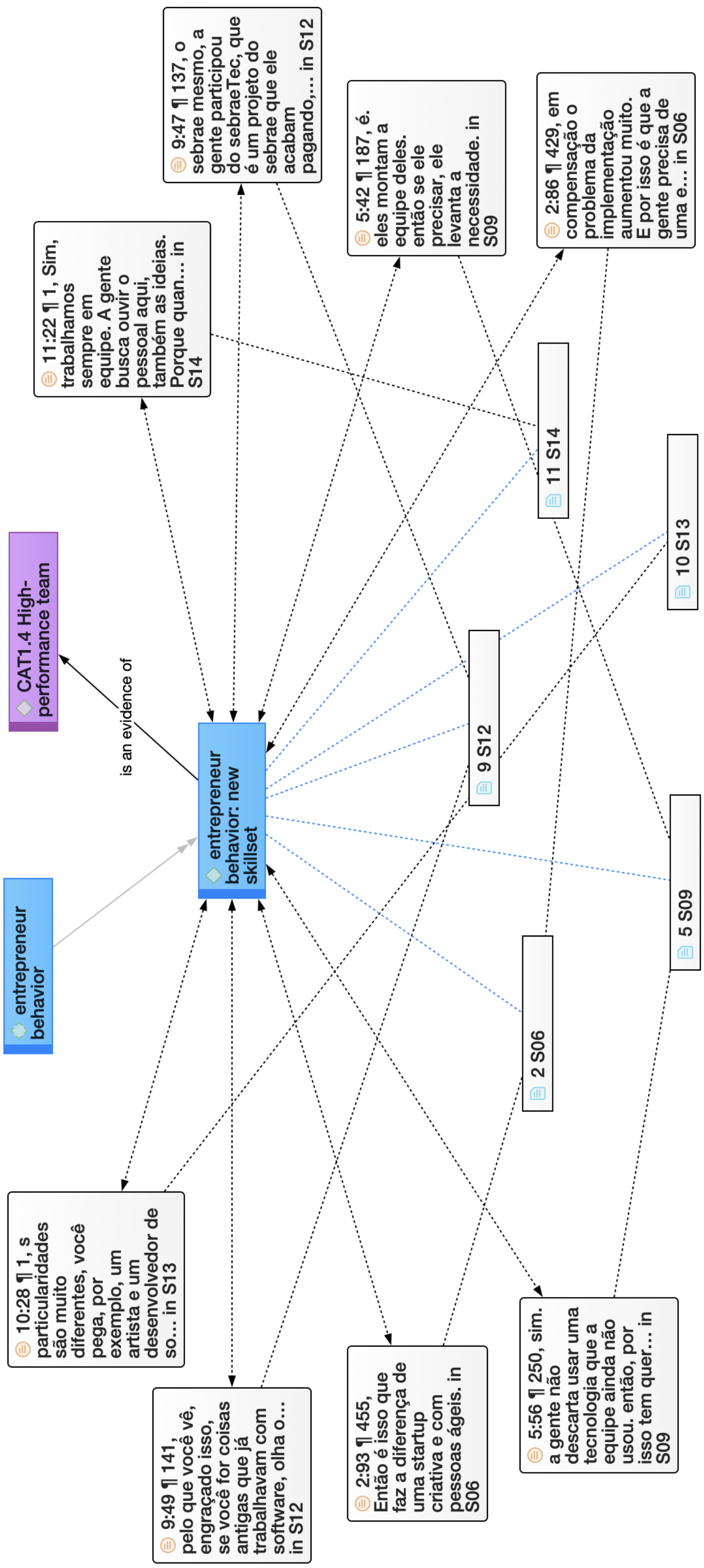


Figure E.5 HA-REC.06 Have a team with new skill sets.

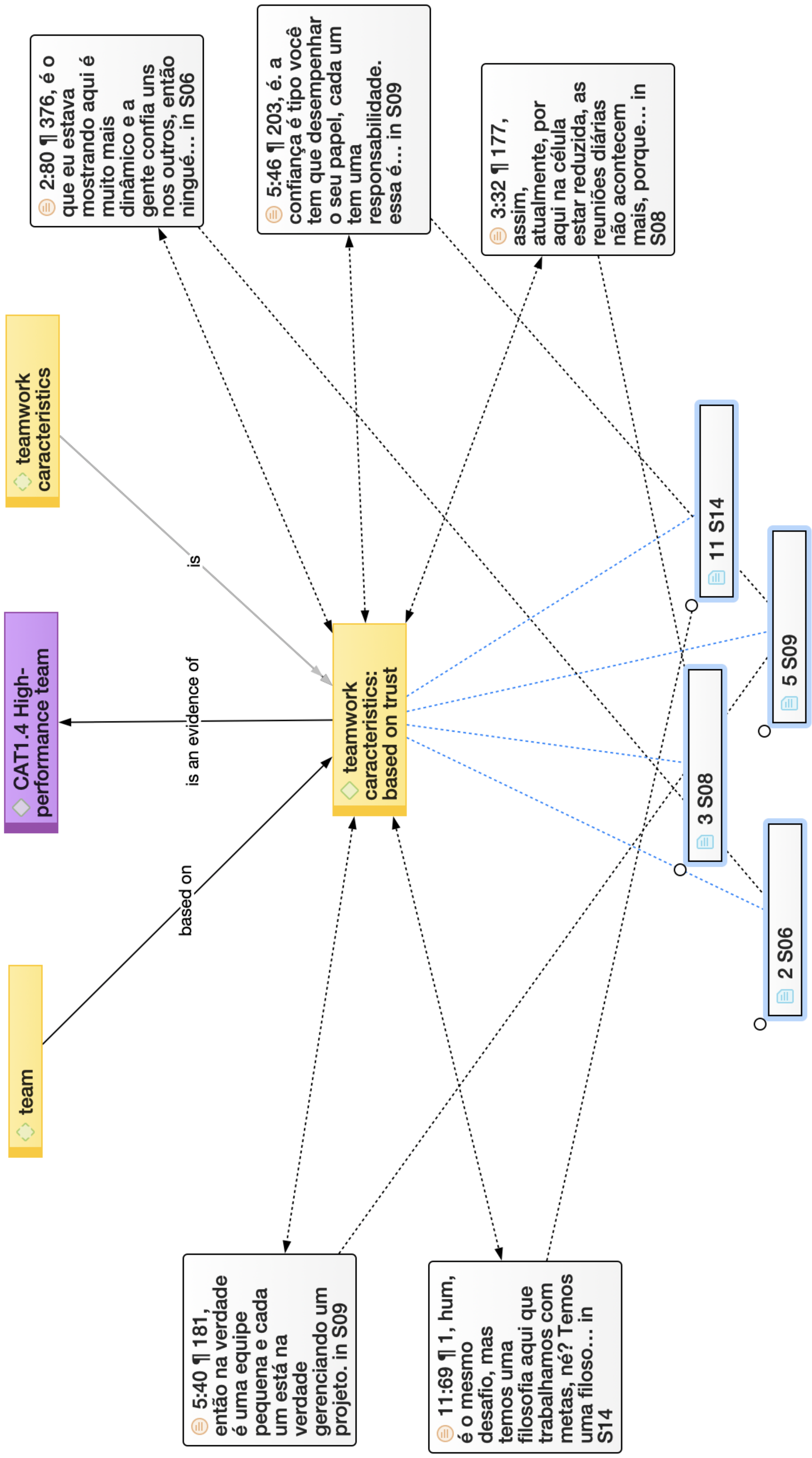


Figure E.6 HA-REC.06 Trust each other.

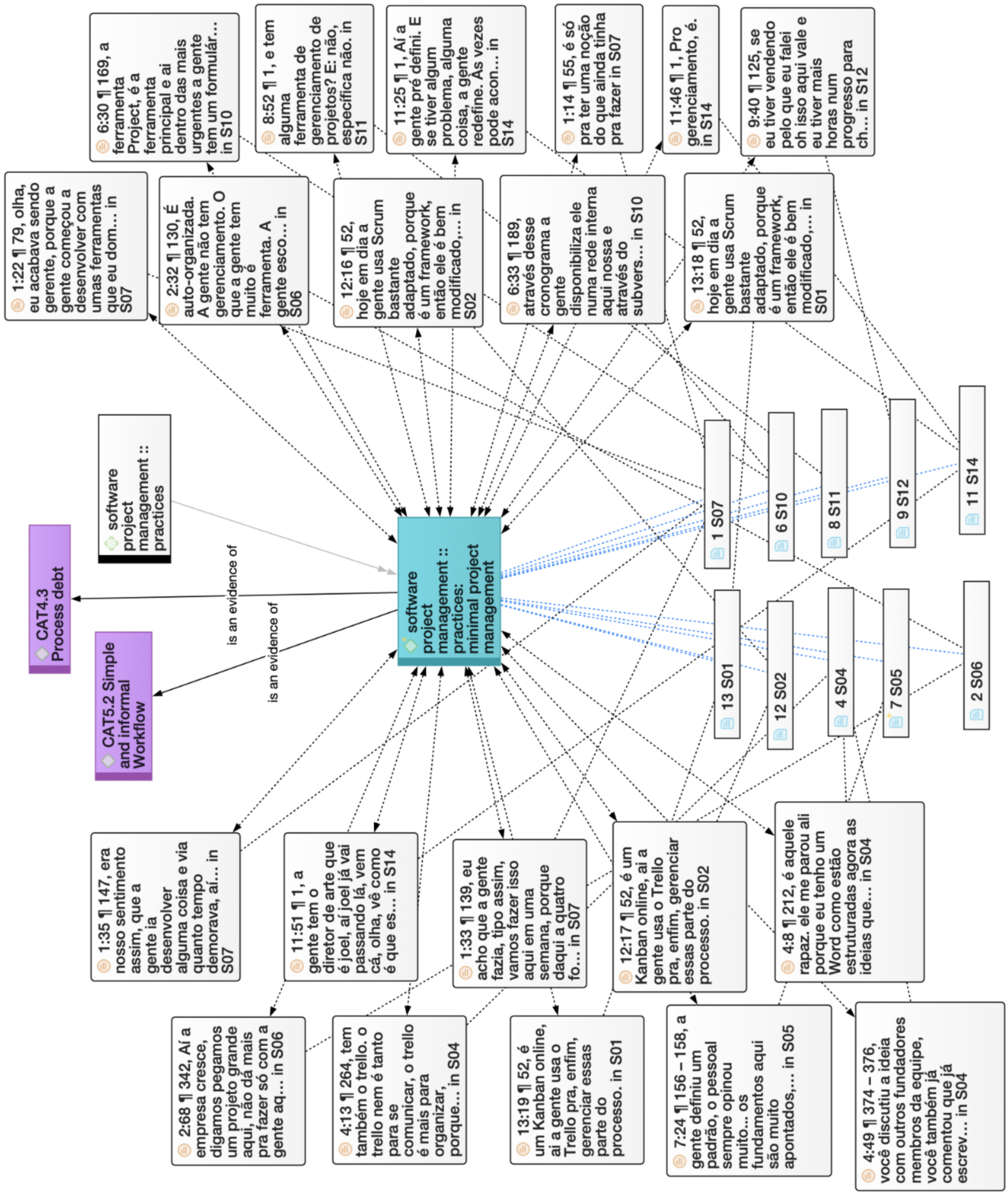


Figure E.7 T-REC.01 Define a minimum viable process.

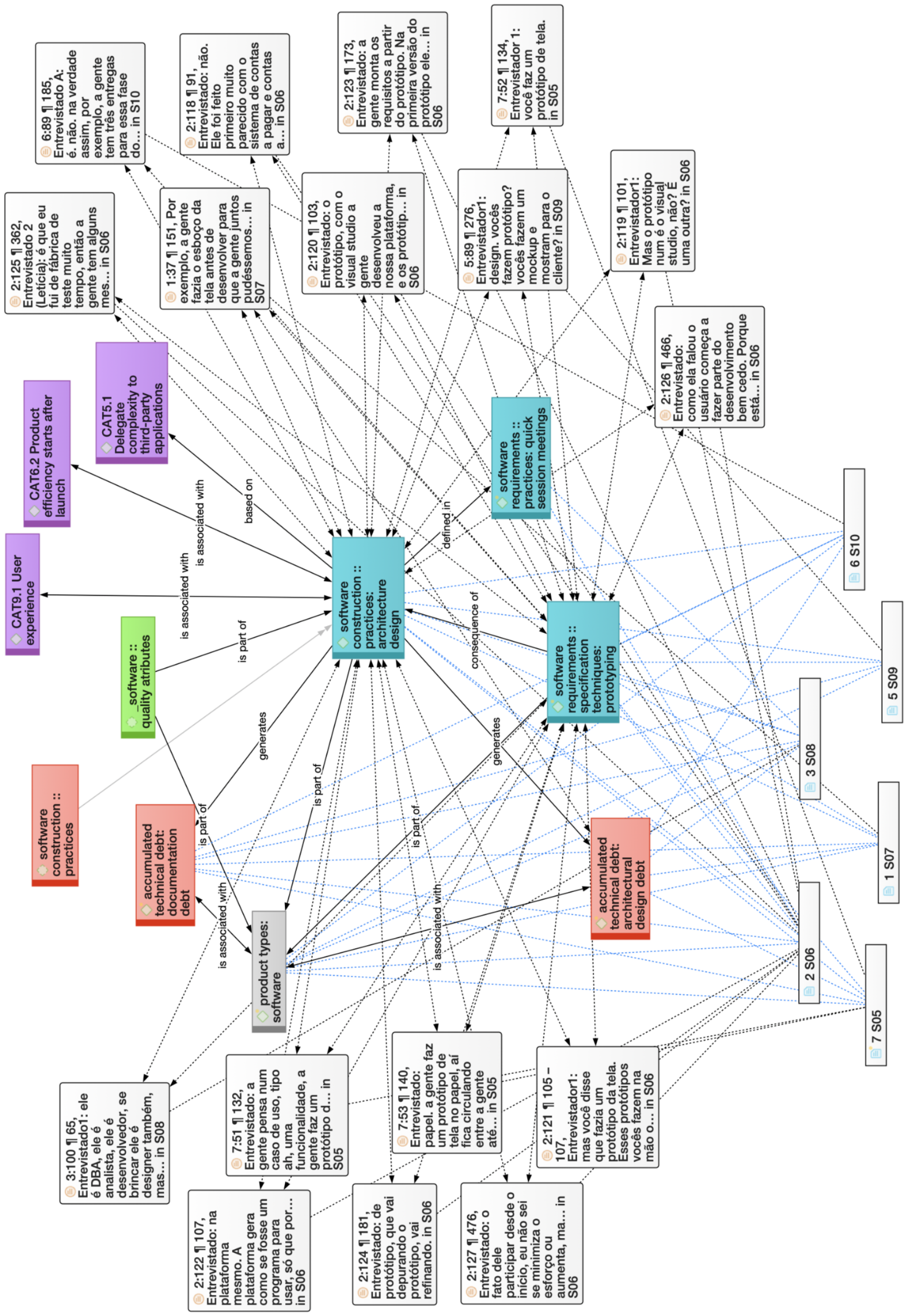


Figure E.8 T-REC.02 Spend minimal time and effort on defining the best possible software architecture.

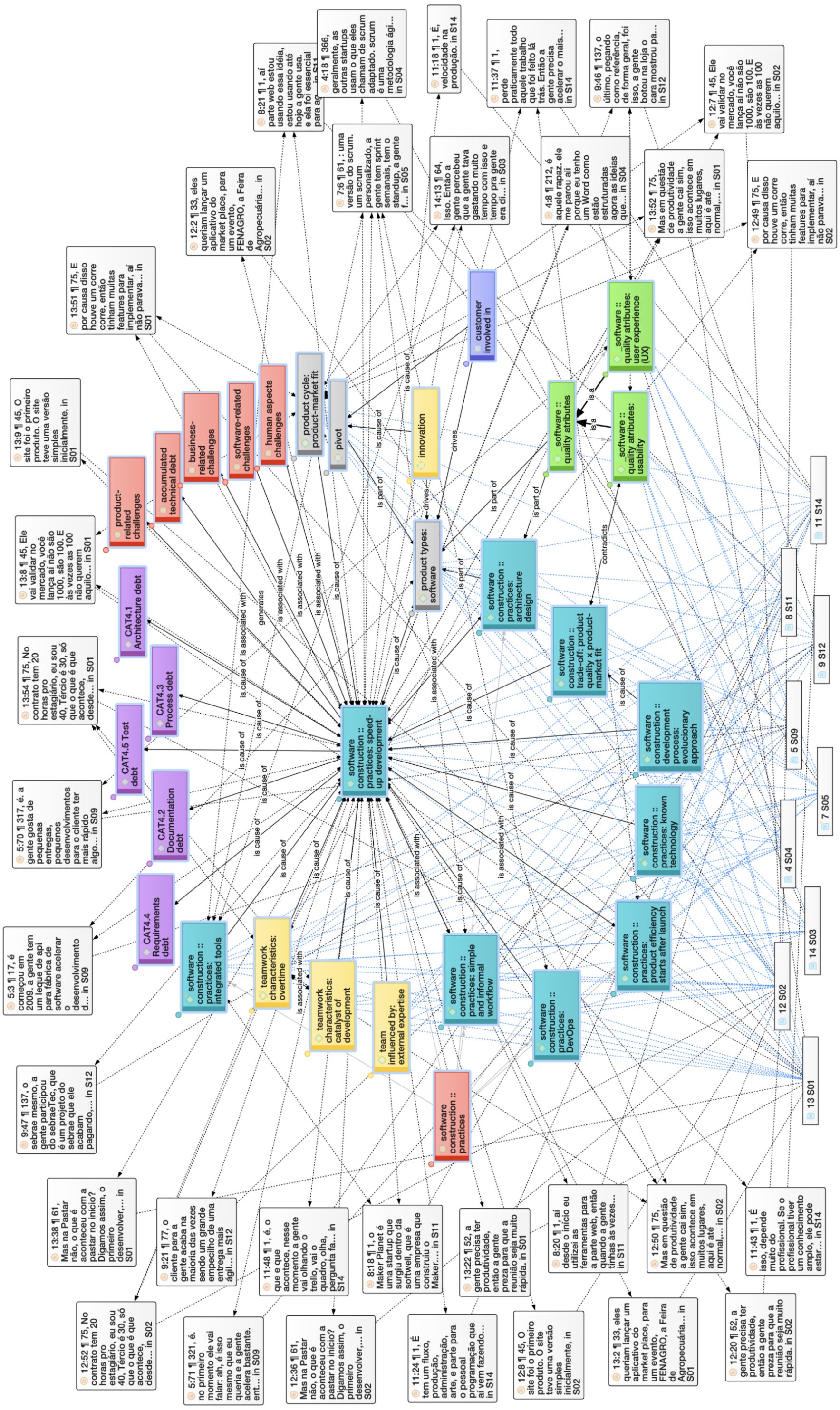


Figure E.9 T-REC.03 Define software development, coding policies, and pattern with the team.

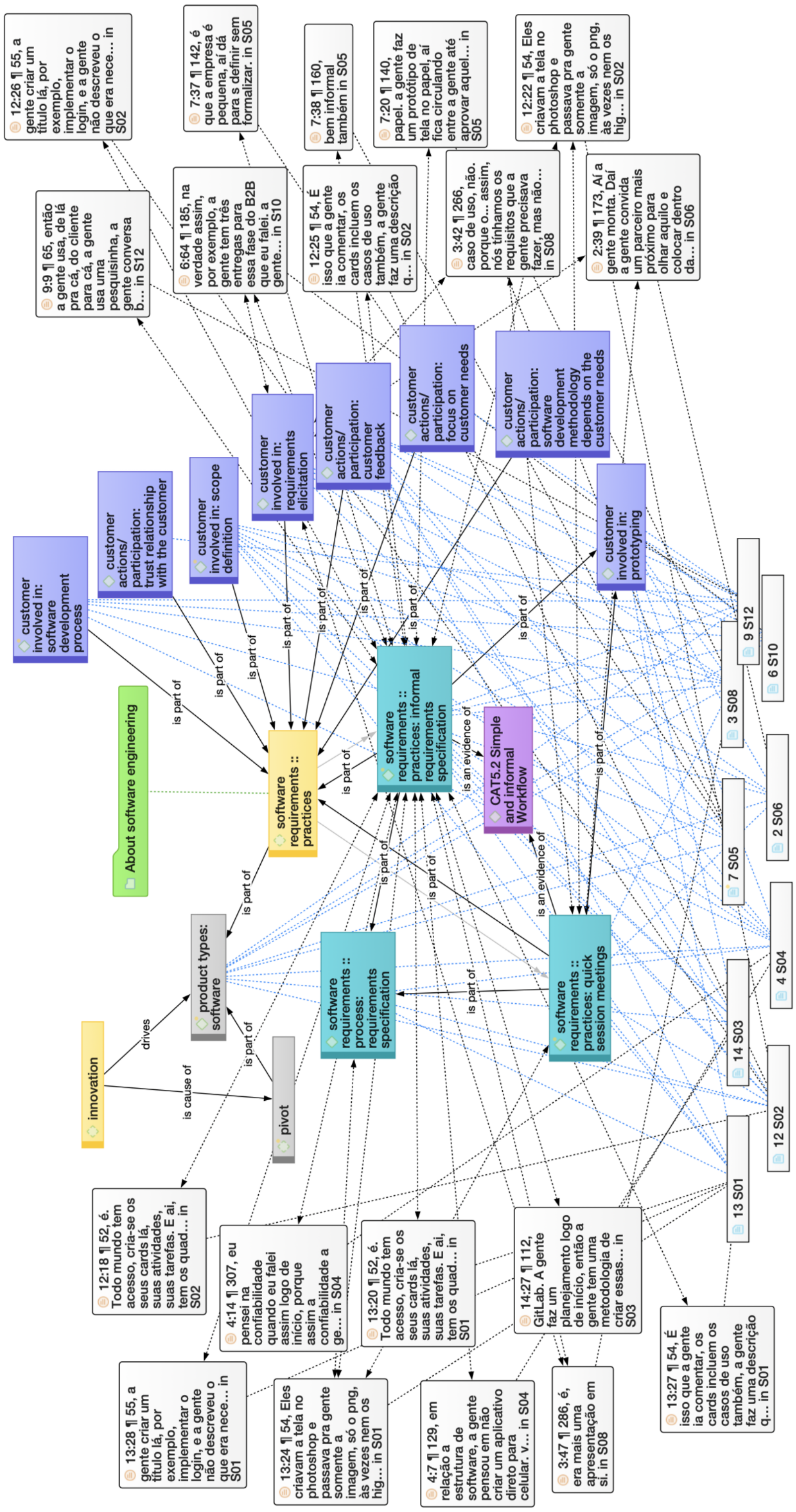


Figure E.10 T-REC.04 Requirement.

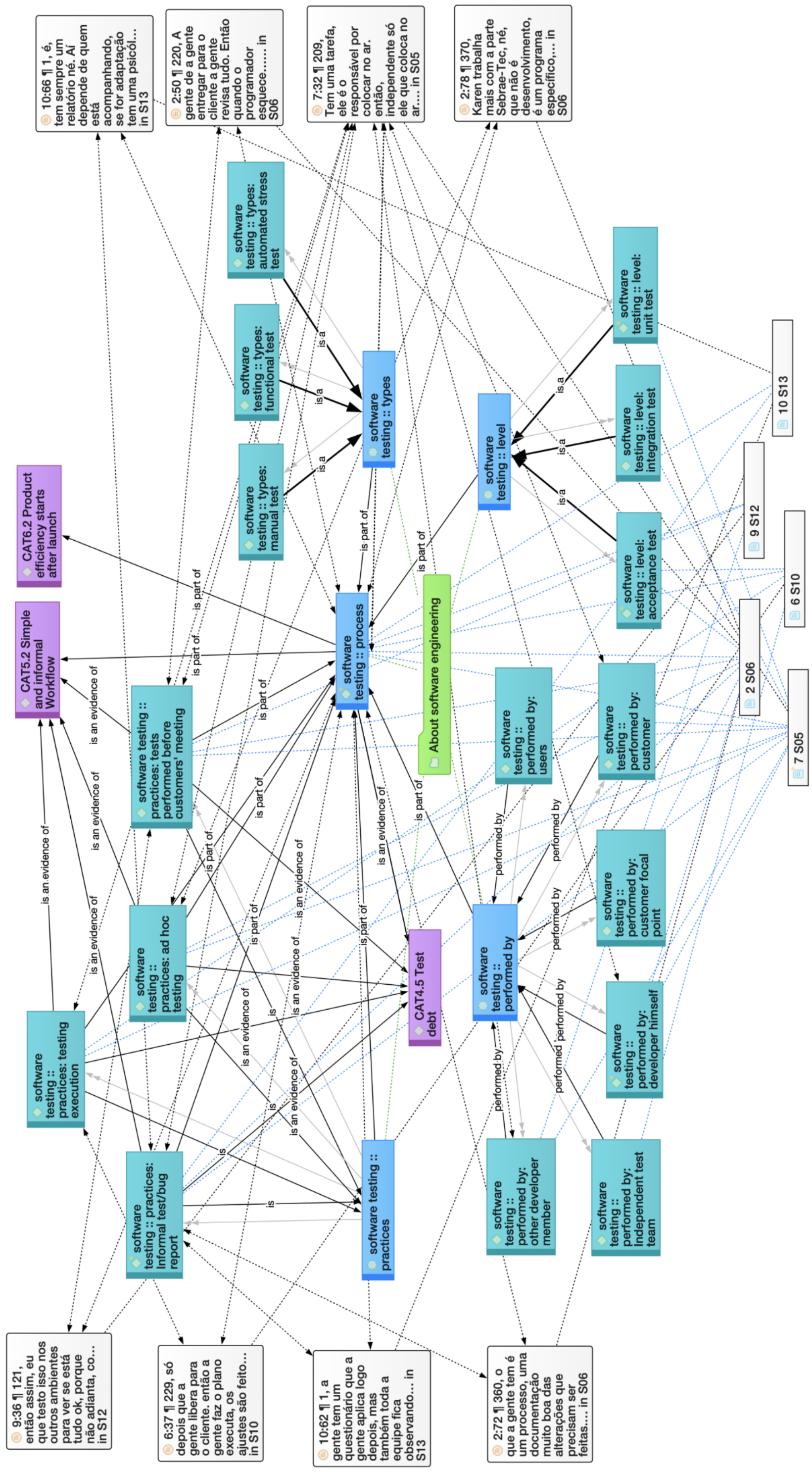


Figure E.11 T-REC.05 Test.

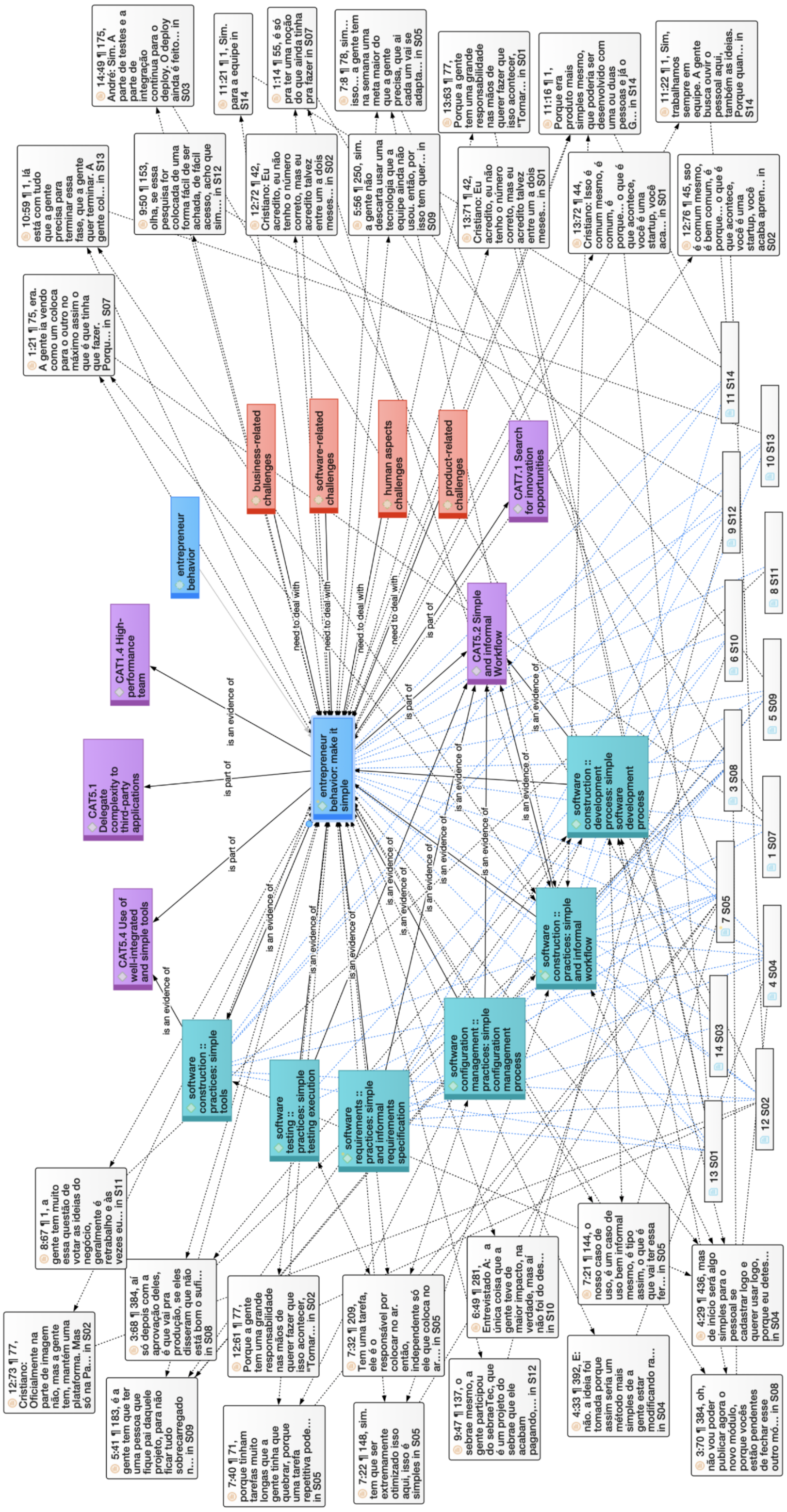


Figure E.12 T-REC.06 Make [it] simple.

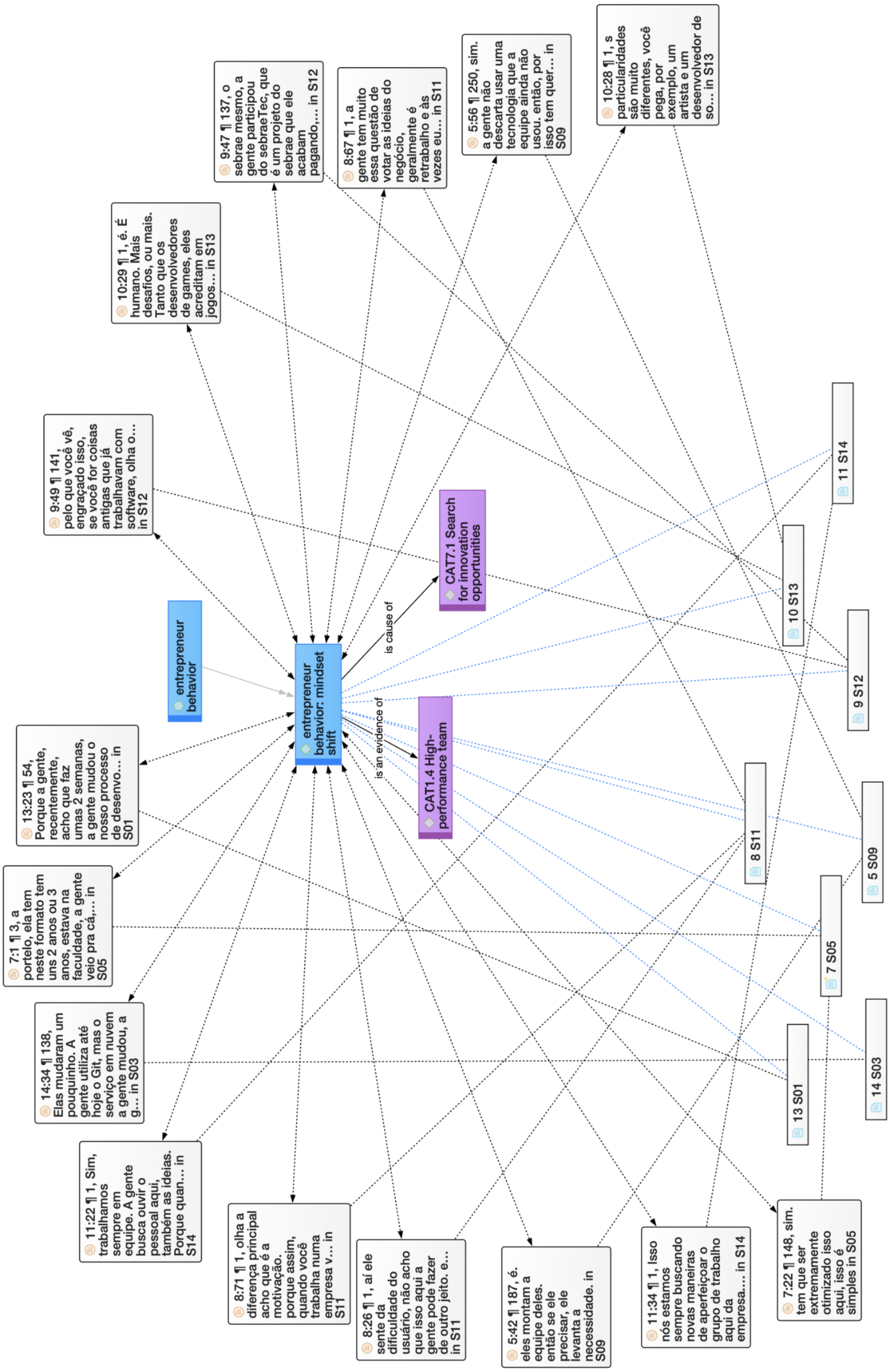


Figure E.13 T-REC.07 Make [it] integrated.

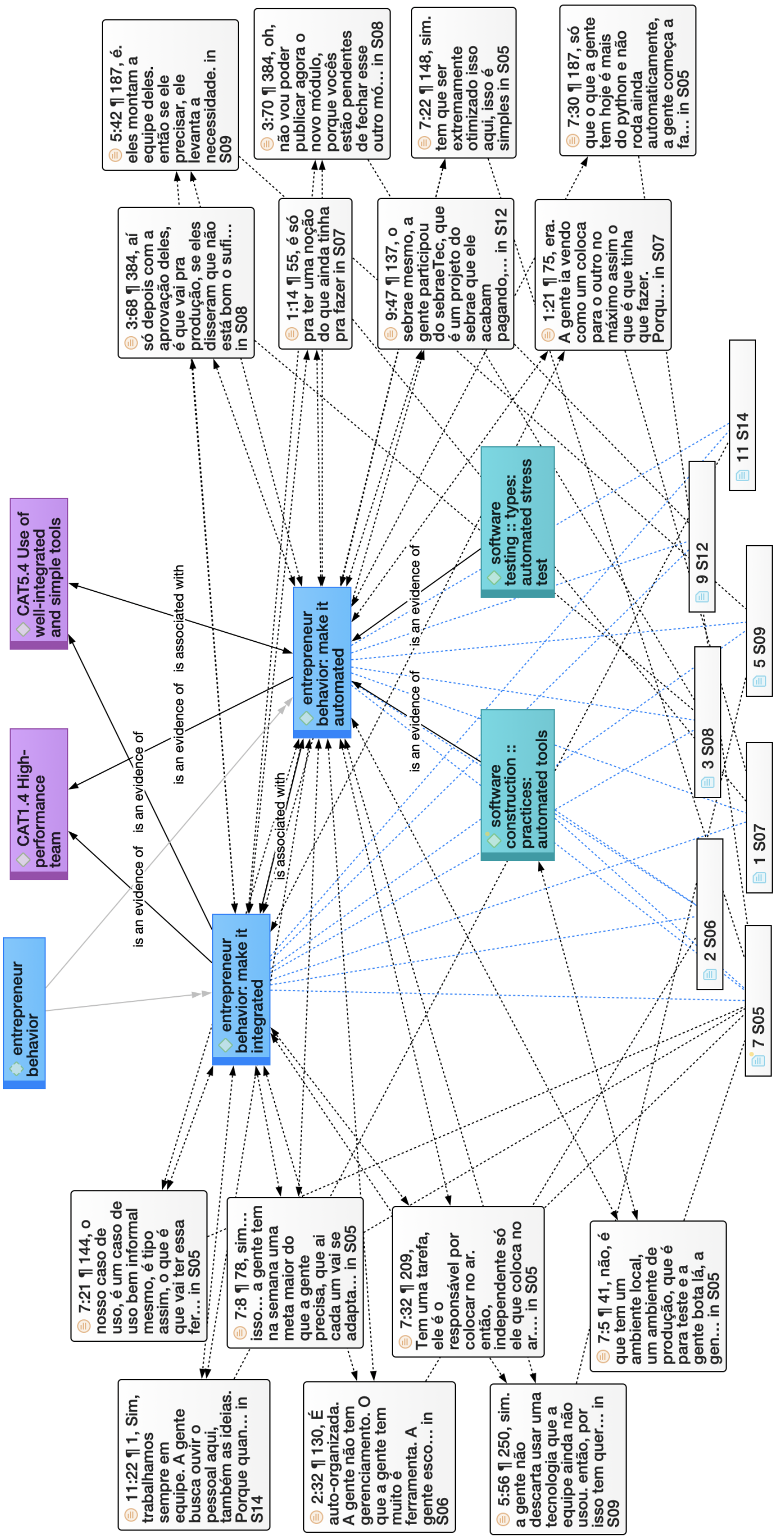


Figure E.14 T-REC.08 Make [it] automated.

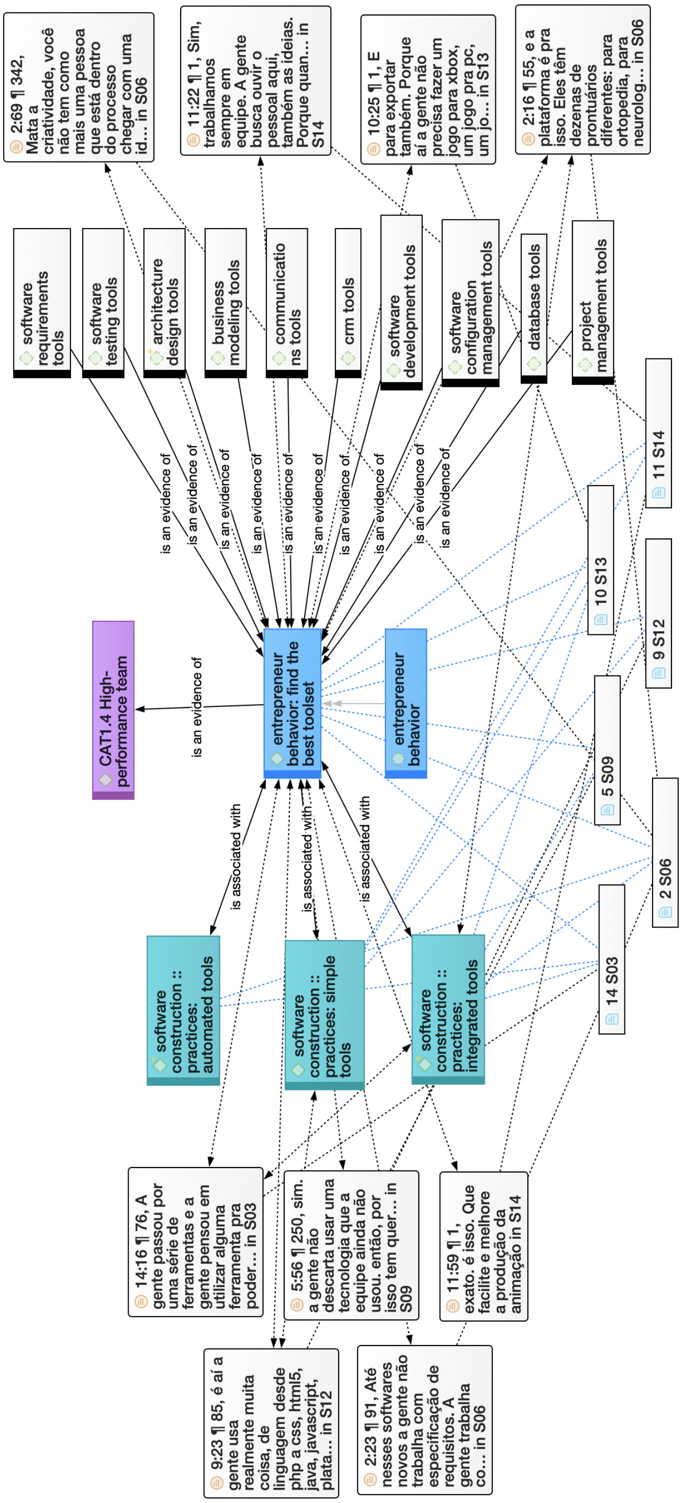


Figure E.15 T-REC.09 Find the toolset that helps your team to accelerate software development.

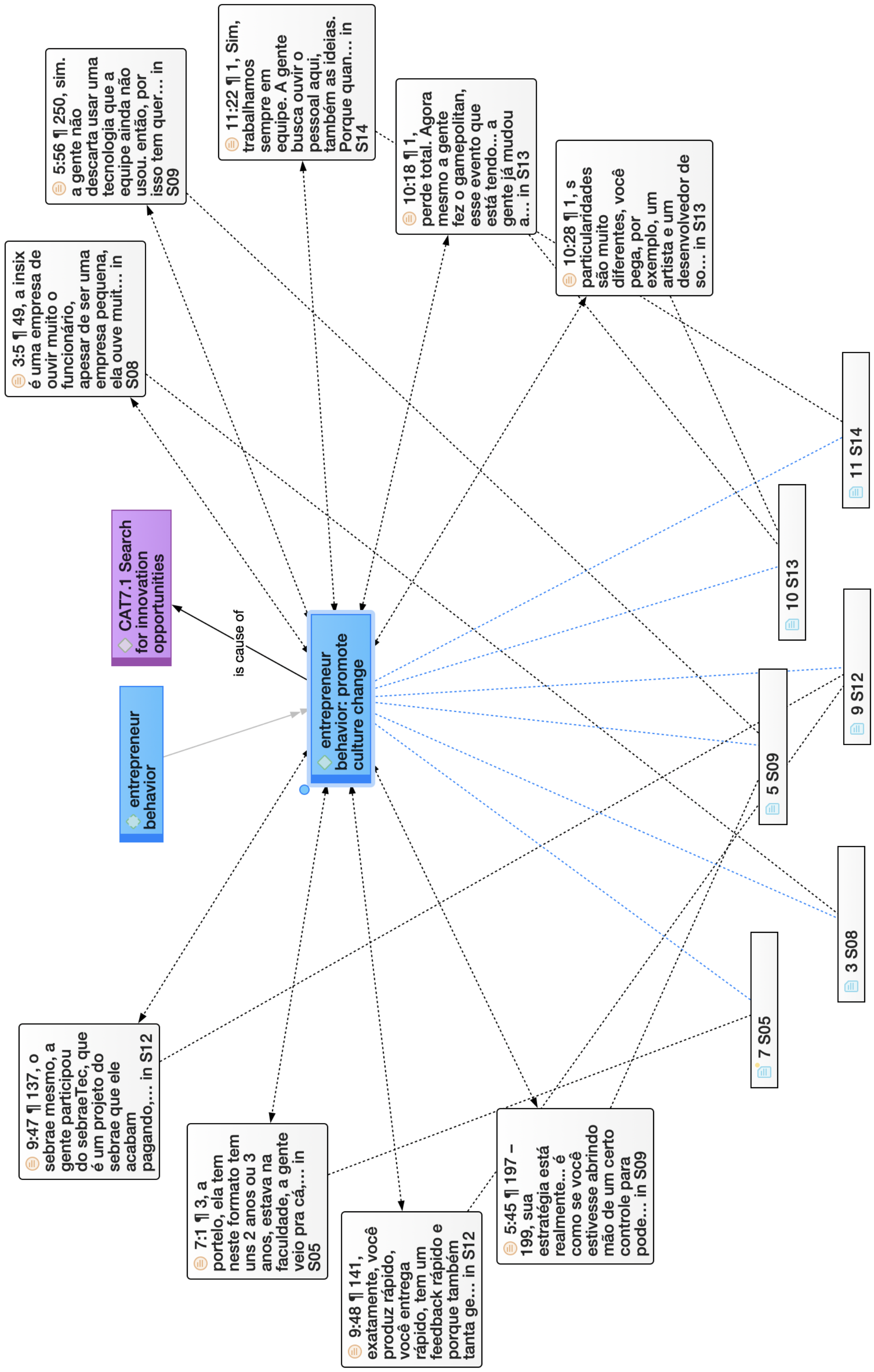


Figure E.16 O-REC.01 Make organizational culture change.

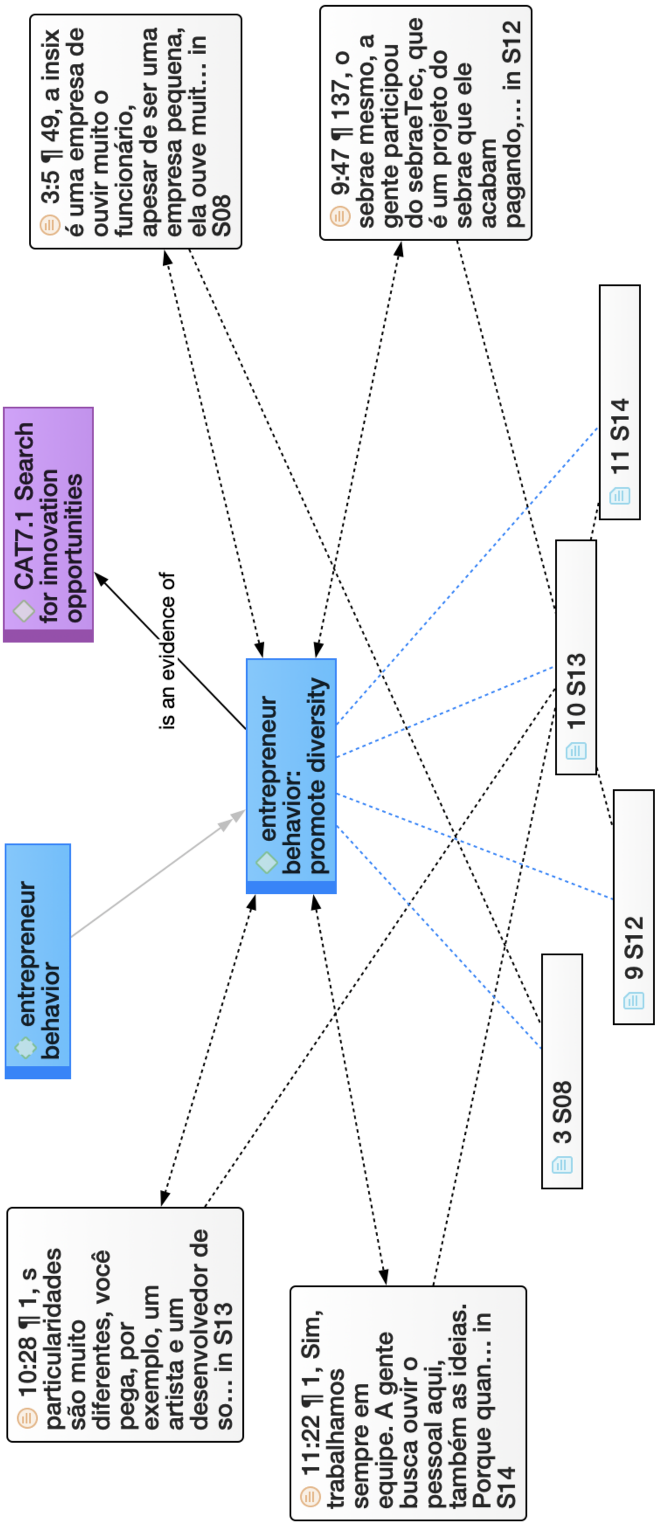


Figure E.17 O-REC.02 Promote diversity.