PUCRS

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

INGRID WALDY MANFRIM NETA SIGNORETTI

Characterizing the Combined Use of Agile, User-Centered Design and Lean Startup

A Case Study of Two Software Teams

Porto Alegre
2020

PÓS-GRADUAÇÃO - *STRICTO SENSU*

Pontifícia Universidade Católica
do Rio Grande do Sul

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

CHARACTERIZING THE COMBINED USE OF AGILE, USER-CENTERED DESIGN AND LEAN STARTUP: A CASE STUDY OF TWO SOFTWARE TEAMS

INGRID WALDY MANFRIM NETA SIGNORETTI

Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Prof. Sabrina Marczak, Ph.D.

Porto Alegre
2020

Ficha Catalográfica

S578c Signoretti, Ingrid Waldy Manfrim Neta

Characterizing the Combined Use of Agile, User-Centered Design and Lean Startup : A Case Study of Two Software Teams / Ingrid Waldy Manfrim Neta Signoretti. – 2020.
124.
Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientadora: Profa. Dra. Sabrina dos Santos Marczak.

1. Agile. 2. User-Centered Design. 3. Lean Startup. 4. Agile Transformation. 5. Case Study. I. Marczak, Sabrina dos Santos. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

Ingrid Waldy Manfrim Neta Signoretti

Characterizing the Combined Use of Agile, User-Centered Design and Lean Startup: A Case Study of Two Software Teams

This Master Thesis/Doctoral Thesis has been submitted in partial fulfillment of the requirements for the degree of Master of Computer Science, of the Graduate Program in Computer Science, School of Technology of the Pontifícia Universidade Católica do Rio Grande do Sul.

Porto Alegre, 03rd April 2020.

COMMITTEE MEMBERS:

Prof. Dr. Tiago Silva da Silva (UNIFESP)

Prof. Dr. Afonso Henrique Corrêa Sales (PUCRS)

Prof. Dr. Sabrina Marczak (PPGCC/PUCRS - Advisor)

This study is dedicated to my beloved husband and parents.

“And whatever you do, whether in word or deed, do it all in the name of the Lord Jesus, giving thanks to God the Father through him.” (Colossians 3:17)

ACKNOWLEDGMENTS

First of all, I thank God that was my strength for the last two years. I am also grateful for my husband's support and encouragement. I want to thank my parents for being unconditionally supporting and for always encouraging me to chase my dreams. I want to thank all my friends for the outstanding support and for their patience in my absence. I am also thankful for my brothers and sisters in Christ who have sustained me with prayers and love. My advisor who was caring and inspired me to discover the research world. She also encouraged me each step of the way when at times I believed that doing research was harder than I could manage. You are a wonderful person, thank you! To Professor Ricardo Bastos, who acted as my co-advisor and unconditionally shared his wisdom and devoted his precious time to my work and to my development. For my research group, you guys are amazing! You helped me to get here. Special thanks to Larissa Salerno, who I have the chance to work closely together during my Master's. I want to also thank all the PPGCC faculty for their share on this journey. And finally, I thank the study participants and acknowledge that this research is sponsored by Dell Brazil using incentives of the Brazilian Informatics Law (Law no 8.2.48, year 1991).

CARACTERIZANDO O USO COMBINADO DE MÉTODOS ÁGEIS, USER-CENTERED DESIGN E LEAN STARTUP: UM ESTUDO DE CASO SOB A PERSPECTIVA DE DUAS EQUIPE DE DESENVOLVIMENTO DE SOFTWARE

RESUMO

Apesar dos muitos benefícios de uma transformação ágil, o envolvimento do usuário e o valor agregado ainda são relatados como desafios no desenvolvimento de software. O uso combinado de *User-Centered Design* e *Lean Startup* juntamente ao Desenvolvimento Ágil é argumentado como uma alternativa para auxiliar na minimização desses desafios. Diante desse cenário, esta dissertação de Mestrado buscou caracterizar o uso combinado dessas metodologias e entender como ocorre a transformação para uma abordagem combinada em uma empresa multinacional. Também, identificar benefícios e desafios encontrados em uma transformação deste porte. Esta pesquisa qualitativa foi composta por dois estudos: um Mapeamento Sistemático da Literatura e um Estudo de Caso com duas equipes da área financeira pertecente a uma multinacional de tecnologia da informação. O primeiro teve como objetivo identificar na literatura o uso da abordagem combinada e o segundo, caracterizar o uso dessa abordagem em um caso real na indústria. Os resultados do mapeamento da literatura revelaram que poucos estudos estão relatando o uso da abordagem combinada. Além disso, nenhum deles caracteriza e relata a transformação para o uso da abordagem em empresas multinacionais. O estudo de caso permitiu explorar o entendimento de como o processo de transformação ocorre em uma empresa multinacional, bem como a caracterização de elementos que compõe o dia-a-dia de trabalho (por exemplo, papéis, atividades, técnicas, cerimônias) de equipes que utilizam a abordagem combinada. Foi possível identificar estratégias como a criação de uma equipe dedicada para o processo e o surgimento de um conjunto de iniciativas proposta por esta equipe, como maneira de auxiliar as equipes que estão adotando o uso da abordagem. Além disso, foram mapeados benefícios e desafios desta adoção. A contribuição mais significativa desta pesquisa é a caracterização da abordagem combinada, no contexto de um processo de transformação. Ainda o estudo provê novas ideias, enriquecendo a literatura da área que perecia nesses aspectos, conforme mapeado na revisão da literatura.

Palavras-Chave: Desenvolvimento Ágil, *User-Centered Design*, *Lean Startup*, Transformação Ágil, Mapeamento Sistemático da Literatura, Estudo de Caso.

CHARACTERIZING THE COMBINED USE OF AGILE, USER-CENTERED DESIGN AND LEAN STARTUP: A CASE STUDY OF TWO SOFTWARE TEAMS

ABSTRACT

Despite the claimed benefits of an agile transformation, user involvement and added value are still reported as challenges in software development. The combined use of User-Centered Design and Lean Startup with Agile is argued as an alternative to minimize these challenges. Given this scenario, this Master Thesis aims to characterize the combined use of these methodologies and also to understand how the transformation of a combined approach into a multinational company takes place. We also point out the benefits and challenges to undergo through such transformation. This qualitative research was composed of two studies: a Systematic Mapping Review and a Case Study with two teams from the financial area of the multinational company. The first one has the purpose of identifying studies that report the use of the combined approach. The second study aims to characterize the use of the combined approach in a multinational company. The results of the systematic mapping revealed that only a few studies report on the use of the combined approach. Moreover, none of these studies report on the transformation strategy in a multinational context. Aware of this gap in literature, the conducted case study details how the combined approach transformation process took place in a multinational company. We identified the company's strategy to conduct the transformation, for instance, to establish a dedicated team to lead the process; and also a set of initiatives defined to help the teams in transformation to go through the adoption of new practices and of a new mindset. Additionally, the study provides the combined approach characterization including the elements (e.g., roles, activities, techniques, ceremonies) which comprises the teams' daily work in the use of the approach. The study also provides a list of benefits and challenges of such adoption. The more significant contribution of this research is the combined approach characterization through the transformation process. This research provides new insights for the area, filling in the lack of knowledge on these specific topics, as mapped in the literature review.

Keywords: Agile, User-Centered Design, Lean Startup, Agile Transformation, Literature Systematic Mapping, Case Study.

LIST OF FIGURES

1.1 Methodological Approach Summarized 17
1.2 Research Design 18
2.1 HCD/UCD Activities 24
2.2 Build-Measure-Learn Loop 26
3.1 Systematic Mapping Flow 30
3.2 Snowballing Procedure 33
3.3 Systematic Mapping Procedure 34
3.4 P1 - Software Development Process 40
3.5 Nordstrom Model 42
3.6 Converge Model 43
3.7 InnoDev Model 45
4.1 Case Study Research Design 53
4.2 Case Study Stages 56
4.3 Data Collection Timeline - First Stage 57
4.4 Data Analysis Example 58
4.5 Transformation Package and Outcomes 60
4.6 Team A and B perceptions 65
4.7 Benefits Overview 75
4.8 Challenges Overview 77
5.1 Combined Approach Life Cycle 82
5.2 Combined Approach Detailed 83
5.3 Systematic Mapping x Case Study 87

LIST OF TABLES

3.1 Search Results 32
3.2 Inclusion and Exclusion criteria 32
3.3 Classification Scheme 35
3.4 Systematic Mapping Overview 36
3.5 Data Extraction Form 37
3.6 Literature Findings 48
4.1 Participants' Profile 55
5.1 Benefits: Literature Findings x Case Study Findings 84
5.2 Challenges: Literature Findings x Case Study Findings 86

LIST OF ACRONYMS

BML – Build-Measure-Learn

C.RQ – Case Study Research Question

DSDM – Dynamic Systems Development Method

DT – Design Thinking

HCD – Human-Centered Design

L.RQ – Literature Research Question

MVP – Minimum Viable Product

PICO – Population, Intervention, Comparison and Outcomes

RQ – Research Question

SE – Software Engineering

SLR – Systematic Literature Review

SMR – Systematic Mapping Review

UCD – User-Centered Design

UCSD – User-Centered System Design

XP – eXtreme Programming

CONTENTS

1 INTRODUCTION 15
1.1 Research Problem 16
1.2 Research Design 16
1.3 Contribution 18
1.4 Publications 19
1.5 Thesis Outline 20

2 THEORETICAL FOUNDATION 22
2.1 Agile 22
2.2 User-Centered Design 23
2.3 Lean Startup 25
2.4 Transformation Process 27
2.5 Agile, User-Centered Design, and Lean Startup 29

3 SYSTEMATIC MAPPING 30
3.1 Research Questions Definition 30
3.2 Database Search 31
3.3 Paper Screening 32
3.4 Snowballing Sampling 33
3.5 Classification Scheme 34
3.6 Data Extraction 37
3.7 Systematic Mapping Results 37
3.7.1 L.RQ1: Published Studies 37
3.7.2 L.RQ2: Context of Use 38
3.7.3 L.RQ3: Benefits 45
3.7.4 L.RQ4: Challenges 46
3.8 Discussion 49
3.8.1 Published Studies (L.RQ1) 49
3.8.2 Context of Use (L.RQ2) 49
3.8.3 Benefits (L.RQ3) 50
3.8.4 Challenges (L.RQ4) 50
3.9 Threats to Validity 51

4 CASE STUDY **52**
4.1 Case Study Research Design 52
4.2 Case Study Protocol 52
4.3 Case Setting 53
4.4 Data Collection Methods 55
4.5 Data Analysis Procedure 57
4.6 Results 58
4.6.1 Combined Approach Adoption and Characterization (C.RQ1) 59
4.6.2 Combined Approach Benefits (C.RQ2) 71
4.6.3 Combined Approach Challenges (C.RQ3) 75
4.7 Threat to Validity 78

5 DISCUSSION **79**
5.1 Combined Approach Adoption and Characterization (RQ1) 79
5.2 Combined Approach Benefits (RQ2) 84
5.3 Combined Approach Challenges (RQ3) 85
5.4 Empirical Approach x Literature Models 86
5.5 Recommendations for Practice 88

6 CONCLUSION **89**
6.1 Limitations 90
6.2 Future Work 91

REFERENCES **92**

APPENDIX A – Search String **99**

APPENDIX B – Case Study Protocol **100**

APPENDIX C – Questionnaire **102**

APPENDIX D – Interview Script - I **103**

APPENDIX E – Interview Script - II **104**

APPENDIX F – Consent Form **105**

APPENDIX G – Code Analysis **106**

APPENDIX H – Conceptual Model Instance **119**

APPENDIX I – Workflow Elements **120**

1. INTRODUCTION

Motivations for an agile transformation are distinct. Dikert et al. [17] highlight the need to align software development with corporate strategies, the need to respond to market changes rapidly, and the teams' dissatisfaction with the current way of working. Such transformation might be challenging since it brings organizational changes, including structural (e.g., roles and responsibilities), technical (e.g., techniques), and cultural (e.g., trust) aspects [39]. Difficulties are even more challenging when considering a large-scale agile development setting (e.g., inter-team coordination [8], knowledge sharing [42]).

Despite the adopted strategy to guide the transformation (e.g., big bang or gradual introduction [38]) or the effort put in, customer involvement [4] and added value [28] are still among the main challenges. Vilkki [71] argues that agile needs to be combined with other methodologies. He claimed that this combination could provide better guidance for agile teams to improve the problem understanding and to provide more aligned solutions as well as to keep the customer engaged.

The combined use of Agile Development, User-Centered Design (UCD), and Lean Startup has been argued as a manner to tackle the before mentioned agile limitations [75]. For instance, Lean UX [27] argues for the need of designers, engineers, and product managers 'act as one' to build a shared understanding around customers and what they need most. Jeff Gothelf, Lean UX co-author, says that a team needs to concern about whom they are building a solution for and what success looks like, counter-arguing that often teams *'ship the feature off and do not look back'* [27].

Regarding the methodological aspects, while UCD [50] puts the user at the center of the discussion, aiming for creativity and empathy for designing user-centric solutions and helping developers to change their mindset on how to approach a problem and envision its solution [29], Lean Startup focuses on adding value to business stakeholders by looking for the best solution through experimentation. A hypothesis about a satisfactory solution is defined and validated with users, and solutions pivoted until an available fit resolution is achieved. These systematic manners of experiments reduce the waste of resources, time, and financial investments.

Despite the industry interest in using the three methodologies altogether in order to boost agile development, the knowledge about the subject is limited. The existent literature studies did not approach how the methodologies are combined. Also, no empirical studies are exploring how the transformation takes place in a multinational company or advise how to do so. The studies did not examine the benefits of undergoing such transformation or the challenges faced while doing it. Moreover, none of the studies discuss whether large organizations have special needs or should act differently than others.

1.1 Research Problem

Inspired by the previously scenario, we understand that there is a lack of studies describing how the combined approach of Agile, UCD, and Lean Startup is adopted by companies, as well as how the transformation takes place.

Driven by the problem statement, the goal of this research is *to characterize the combined use of the Agile, UCD, and Lean Startup approach*. We posed three research questions to guide our study.

Research Question

- RQ1. How the combined approach of Agile, UCD, and Lean Startup is adopted in the software development context?
- RQ2. What are the benefits of the combined adoption of Agile, UCD, and Lean Startup?
- RQ3. What are the challenges on the combined adoption of Agile, UCD, and Lean Startup?

RQ1 is our main question. We seek to understand how the combined approach is applied in a software development context. Also, we aim to characterize the three methodologies that are combined - and map what the activities, techniques, and artifacts produced from the combined approach usage are. The answer to this question will provide new insights for the academy, given a more detailed overview of the combined use. Also, the RQ1 aim to map how the transformation process can affect and determine how the combined approach is used in a company. RQ2 and RQ3 will point out the benefits and challenges of the combined approach usage. The answer to these three questions (RQ1, RQ2, and RQ3) could provide significant knowledge about how the transformation occurs, given recommendations of adoption for academy and industry.

1.2 Research Design

This study has a qualitative research nature. Strauss and Corbin [69] explain that this kind of research seeks to understand people's life experiences, behaviors, and feelings, and that can be in an organizational context, social movements, or culture phenomenons. Guided by the decision to conduct a qualitative study, we define the research characteristics in Figure 1.1. Led by the research problem and the research question defined in Section 1.1, we also illustrate our research design in Figure 1.2.

Methodological Approach	
Research Nature	Qualitative
Secondary Study Method	Systematic Mapping Study
Research Method	Case Study
- Purpose	Exploratory
- Data Collection Techniques	Questionnaire, Interview, Observation, and Focus group

Figure 1.1: Methodological Approach Summarized
[Source: Adapted from [60, 78]]

Literature Systematic Mapping: We conducted a literature systematic mapping review aiming to understand the state-of-art about what is addressed in the literature related to the combined use of agile, ucd, and lean startup, as recommended by Creswell and Creswell [14]. Also, we aim to know the benefits and challenges pointed out in the literature. We detail the study protocol, search strings, and data extraction in Section 3. The study results show that in literature, the combined approach is named as software process or design process models. Few studies are reporting the combined approach usage in a detailed manner in a multinational context. We also found out the benefits and challenges of using the models.

Case Study Protocol: We performed an empirical study seeking to understand in practice how these three methodologies combined are used. We adopted as a research method case study. The study's purpose is exploratory since we need to understand what is happening about the phenomena, to look for new possibilities, and to generate ideas and hypotheses for the area to be researched [78].

We developed a case study protocol, which contains the research goal, the data collection methods, and analysis procedure. We adopt a set of different methods to achieve the goal. We use a questionnaire at the beginning of the study, seeking to map the profile of each participant. We also conduct follow-up interviews, observation, and focus group sessions. Section 4 presents a detailed overview of the case study.

Case Study Execution: We conducted the study with two teams from the financial area which have an internal product in their scope. The teams were the firsts company teams to start the adoption of the combined approach. The case study was executed in two stages. First, we followed the two teams through the training process in the approach adoption, collecting data through questionnaires, interviews, observation, and focus group sessions. In the second stage, we use interviews and focus group sessions on confirming the data gathered in the first stage.

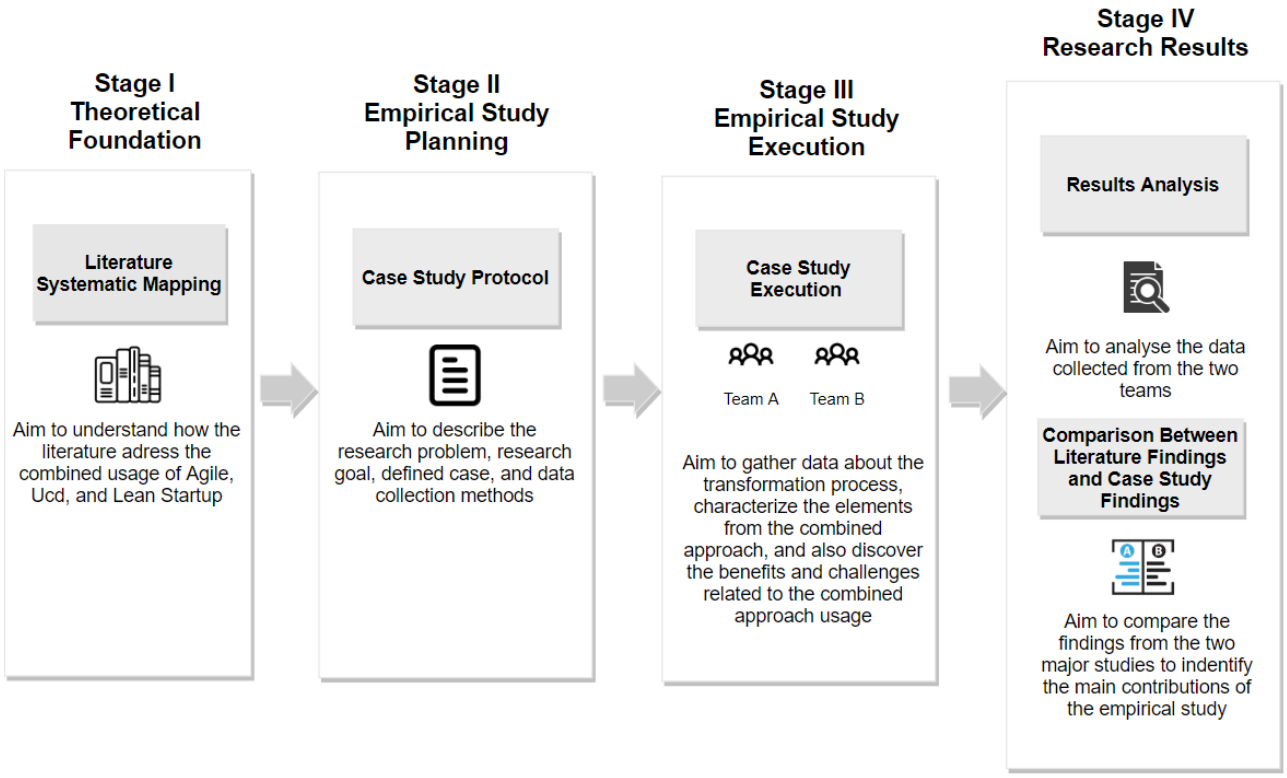


Figure 1.2: Research Design
[Source: Author (2019)]

Results Analysis: The case study results revealed the strategies defined to conduct such a transformation for the combined approach. Also, it was possible to identify how the three methodologies are combined, and we mapped the activities, techniques, and artifacts used by teams that adopt the approach. Also, it was pointed out the benefits and challenges of such adoption. The results from the case study allow us to characterize the combined use of the three methodologies.

Comparison Between Literature and Case Study Findings: Finally, with the systematic mapping and the case study data, we were able to compare the existing reports in the literature about the combined approach with the case study findings, pointing out the similarities and differences, which lead us to contribute with the research area.

1.3 Contribution

This Master Thesis main contribution is on the characterization of the combined use of Agile, UCD, and Lean Startup in software development. This includes the roles, activities, techniques, and the outcomes that compose such combination.

Additionally, this thesis also provides:

- The understanding about how such approach was adopted in a multinational company, including a set of recommendations such as:
  - Create a new team dedicated to lead the transformation process
  - Define a set of initiatives as workshops and cookbooks to teams that are still working with older methodologies
  - Promote hands-on training to enable teams to work using the combined approach
- Qualitative evidence on the benefits of such transformation to the combined approach, including:
  - Introduction of practices derived from eXtreme Programming (XP), as continuous delivery, and pair programming
  - Change for a problem-solving mindset, getting closer for the users and through techniques as user interviews, and personas
  - Use of techniques and practices derived from Lean Startup methodology, as experimentation, and build-measure-learn loop
- A list of challenges faced by the teams that underwent the transformation, including those related to organizational matters.

1.4 Publications

We have published four papers in Software Engineering conferences from the studies reported in this Master thesis. These papers are listed below.

- *“Boosting Agile by Using User-Centered Design and Lean Startup: A Case Study of the Adoption of the Combined Approach in Software Development" [65]*

  Authors: **Ingrid Signoretti**; Sabrina Marczak; Larissa Salerno; Augusto de Lara; Ricardo Bastos. International Symposium on Empirical Software Engineering and Measurement (ESEM), 2019. Qualis: A2.

  This paper reports on the first stage of the conducted case study. We present how the transformation takes place for the combined approach, and also point out the benefits and challenges of such adoption.

- *“Repensando Papéis em Equipes Ágeis: Um Estudo de Caso no Uso de uma Abordagem Combinada de Desenvolvimento Ágil, User-Centered Design e Lean Startup" [61]*

Authors: Larissa Salerno; **Ingrid Signoretti**; Sabrina Marczak; Ricardo Bastos. Brazilian Symposium on Software Engineering (SBES), 2019. Qualis: B2.

This paper also reports on results of the case study. Here we explored how the inclusion of UCD and Lean Startup impacts the team structure. We characterized the introduction of the product designer role in an agile team and the change from the product owner role to the product manager new role.

- *“On the Understanding of Experimentation in Lean Startup in a Large-Scale Software Development Context" [70]*

  Authors: Bruna Vargas; **Ingrid Signoretti**; Maximilian Zorzetti; Sabrina Marczak; Ricardo Bastos. International Conference on Evaluation and Assessment in Software Engineering (EASE), 2020. Qualis: B1.

  This paper reports on the relevance of experimentation in the software development process. This study is also part of the case study. However, it contrasts results from this study with those from literature. We additionally conducted a snowballing literature review and compare perspectives here.

- *“Combining User-Centered Design and Lean Startup with Agile Software Development: A Case Study of Two Agile Teams" (camera-ready)*

  Authors: **Ingrid Signoretti**; Larissa Salerno; Sabrina Marczak; Ricardo Bastos. International Conference on Agile Software Development (XP), 2020. Qualis: B1.

  This paper reports on the main findings of the case study. We present the teams’ perspective about the combined approach adoption and its elements such as activities, techniques, ceremonies, and outcomes.

1.5 Thesis Outline

The remainder of this Master Thesis is structured as follows.

Chapter 2: Theoretical Foundation: This chapter describes the main concepts that are the foundation of this work. The text presents the main concepts underlying the topics of Agile methods, User-Centered Design, and Lean Startup, including values, principles and activities of each one of the approached. It also includes studies that address how an agile transformation is often conducted, and how the three before mentioned approaches work when put together.

Chapter 3: Systematic Mapping: This chapter presents a systematic mapping study that aimed to map the state-of-art on the models that describe a combined use of Agile, UCD and Lean Startup. The study was guided by Petersen guidelines [57] for mapping studies. To supplement the mapping and make sure we had covered all current literature,

we conducted a supplementary snowballing review. In total, we identified 7 papers, and their studies revealed that the combined approach is defined as a model in literature. Also, the papers present a set of benefits and challenges under the combined approach usage.

Chapter 4: Case Study: This chapter describes the case study, including the case setting, participants profile, and the data collection techniques applied. The case study revealed the strategies adopted by the company to conducting the transformation from agile to the combined approach such as creating a team dedicated to the transformation, suggest a set of initiatives to the rest of the company that is not adopting yet the combined approach. It also presents the elements that comprise the approach, such as roles, activities, techniques, ceremonies, and outcomes. Last but not least, results also include the benefits and challenges of moving to the combined approach.

Chapter 5: Discussion: This chapter discusses the case study results, including the transformation process, the combined approach characterization, the success factors, and the challenges faced to go through the transformation. It also introduces a comparison between the empirical study results and the literature-based models. Finally, the chapter introduces a set of recommendations for practice to those companies that aim to conduct this kind of transformation.

Chapter 6: Conclusion: The conclusion chapter presents the final conclusions of this Master Thesis. The chapter also includes the limitations and future work.

2. THEORETICAL FOUNDATION

This chapter presents the main concepts that comprise the background content for this thesis work. We explore the Agile methods definition with its values and principles. We also present the User-Centered Design and Lean Startup definition. As this research context is observed in a transformation process, we present the literature studies about agile transformation. Finally, we briefly introduce the combined use of Agile, User-Centered Design, and Lean Startup here. Section 3 details the topic as a result of a literature review on the subject.

2.1 Agile

The Agile Manifesto was defined in early 2001 for a group that later was named as Agile Alliance. The Agile Manifesto emerges as an alternative to the heavyweight software development processes at the time [5].

These selected group of seventeen participants and representatives from eXtreme Programming (XP), SCRUM, Dynamic Systems Development Method (DSDM) and other Agile Methods, defined four values which drive the agile methods [5]:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

To better understand the four values in practice was created the 11 principles. Agile methods prioritize the high interaction between individuals. This interaction is the reason for the relationship between team members, and customers are one of the most relevant principles of agile. Also, the teams must work looking to achieve customer satisfaction through the continuous delivery of the products. Additionally, the teams must be flexible to changes, more than following a closed plan. Another critical element is to promote better communication between team and customers [5].

To achieve these objectives is crucial that the teams be self-organized in which the members are not only collocated but also work at a pace that promoting their creativity and productivity [19]. They must have the freedom to evaluate themselves and make the necessary adjustments.

Besides the social aspects, agile teams must deliver working software as a metric to indicate their progress. Deliver a working product can be achieved through continuous attention to the technical aspects, prioritize a good design and code quality. Moreover, having simplicity to work only on the requirements that are prioritized is a manner of adding value to the deliveries [5].

Dingsøyr et al. [19] argue that these principles are not a definition for agility. They are guidelines for delivering high-quality software in an agile manner. For this purpose, many companies started the process of agile methods instead of traditional ones, and this is related to flexibility and the benefits such as handling requirements changes, productivity gains, and business alignment [9].

Although agile methods can achieve better performance in small and co-located teams [45], many traditional and large-scale organizations started the adoption of these methodologies [7]. Large-scale teams, as defined by Dingsøyr et al. [18] taxonomy, claim that agile teams must have between 7 and 9 members. When the project needs to have more members involved, the teams are divided and arises a new category of large-scale teams.

Despite the many benefits that agile methods bring for small or large-scale companies, many studies are relating to the lack of user involvement [4], and difficulties to address business value into the products [28]. Vilkki [71] argues that agile needs to be combined with other approaches to provide better guidance for agile teams, to improve the understanding of the problem at hand and to provide more aligned solutions as well as to keep the customer engaged.

2.2 User-Centered Design

User-Centered Design or User-Centered System Design (UCSD) coined by Norman and Draper [50] is an approach oriented by business objectives and user needs, limitations, and preferences. UCD seeks to achieve the correctness of the problem. The approach can be applied for design, testing, and implementation of products and services [26].

Norman and Draper [50] affirm in their books the proposal of UCD:

> *“[...] user-centered design emphasizes that the purpose of the system is to serve the user, not to use a specific technology, not to be an elegant piece of programming. The needs of the users should dominate the design of the interface, and the needs of the interface should dominate the design of the rest of the system."*
> [50]

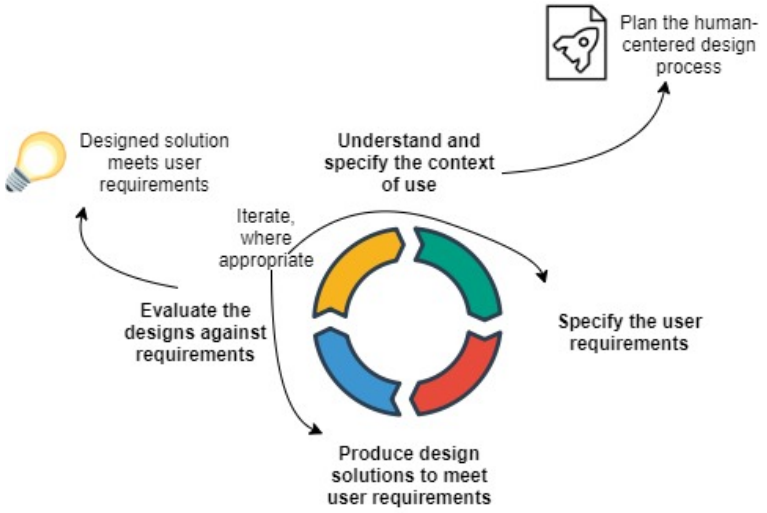


Figure 2.1: HCD/UCD Activities
[Source: Adapted from ISO 9241-210 [35]]

Norman [49] mentioned that engineers and business people are trained to solve problems, and designers are trained to discover the real problems. However, the solution must attend the right problem. Problem discovering and solution definition must be aligned.

The ISO 9241-210 [35] defined the human-centered activities for a development life-cycle, emphasizes that the term Human-Centered Design (HCD) is a synonymy of UCD. Due to this fact, the set of activities defined in the ISO is considered UCD activities as well.

The ISO 9241 defines four UCD main activities in its process (see Figure 2.1):

- Understand and specify the context of use: Identify the users and task characteristics that are the context of product development. It is relevant that in the final of this activity, the following aspects are understood: characteristics of the end-user, the tasks the users will perform, and the environment in which the users will use.

- Specifying the user requirements: Identify the requirements and conduct the analysis is requested in this activity. The ISO 9241 defined set of elements that should be cover in the specification: (1) Identify relevant users, (2) Define the design goals, (3) Define appropriate priorities for the requirements, (4) Provide measurable benchmarks against the design, (5) Provide evidence of acceptance of requirements by the representatives, and finally, (6) Provide acknowledgment of any statutory or legislative requirements.

- Produce design solutions to meet user requirements: Exploration of the design of the solution by creating simple mock-ups of the proposed system and the later presenting them to a representative sample of users.

- Evaluate the designs against requirements: The evaluation through usability testing with actual users. The users' evaluation is an essential development activity that allows it to confirm the extent of the user and organizational objectives that have been met.

UCD also has four main principles defined by ISO 9241. These principles are: *1) an appropriate allocation of function between user and system*: which aspects of a job or a task should be handled by people and can be handled by software and hardware. *2) the active involvement of users* - as one of the critical strengths of UCD: the involvement depends on the nature of the design activities. Involving end-users can also enhance the acceptance and commitment to the new software. *3) iteration of design solutions* - entails the feedback of end-users following their use of new design solutions. These can be range by simple paper mock-up of screen layouts to prototypes. *4) multi-disciplinary design teams* - user-centered software is a collaborative process that benefits from the active users' involvement. It is relevant to each perspective on the development team. The team can include managers, usability specialists, software engineers, quality assurance representatives, and the end-user, the people who will use the final product.

2.3 Lean Startup

Lean Startup is a methodology idealized by Eric Ries [58] in 2011. The methodology is based on Lean Manufacturing and Lean Thinking. The Lean Startup's primary goal is to eliminate the waste of the products building process for remains innovative. Ries [58] defines the methodology as a set of practices that helps entrepreneurs to increase their chances to develop successful products. To guide the methodology, Ries [58] defined five principles:

1. Entrepreneurs are everywhere: The methodology considers as an entrepreneur any person who aims to develop new products under an extreme uncertainty scenario. In lean Startup the entrepreneurs are more than just a role, it is a mindset

2. Entrepreneurship is management: Ries [58] argues that the old way of managing business is not so successful in the context of startups. It is necessary to look for precise metrics, consider the customer feedback, and be aware whether the product is still feasible and addressing the value

3. Validated learning: The methodology defines validated learning as a manner to companies learn through applying experiments, what the customers want, and not only developing what the customers say that they want. As a consequence of the results of the experiment, the learning is validated by collecting data empirically using real customers named as early adopters.

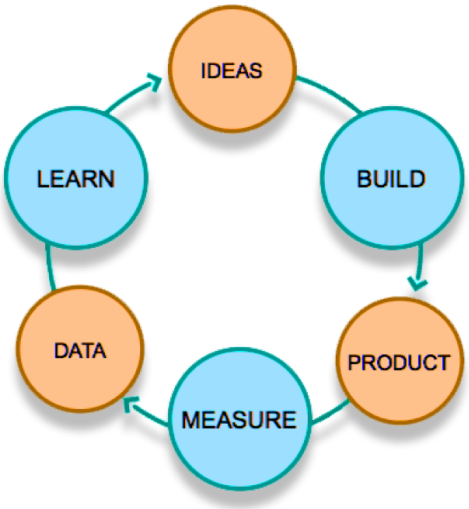


Figure 2.2: Build-Measure-Learn Loop
[Source: Adapted from Ries [58]]

4. Build-Measure-Learn (BML): The 4th principle is the core of the methodology (Figure 2.2).

   > *"The fundamental activity of a startup is to turn ideas into products, measure how customers respond, and then learn whether to pivot or persevere. All successful startup processes should be geared to accelerate that feedback loop."* Ries [58]

   It is relevant to mention that the build-measure-learn has an experiment at the core of the loop.

5. Innovation accounting: The 5th principle is associated with the fact of measuring the progress of the product and prioritize work. The innovation data comes from a quantitative approach, contrasting the presence of actionable metrics over vanity metrics. Ries [58] affirm that for companies remain innovative is needed to define actionable metrics that usually had the proposal of observing the customer behavior, in contrast to the vanity metrics that only give a superficial overview about the product.

As mentioned before, the core of the methodology is the build-measure-learn loop (Figure 2.2). However, it is possible to affirm that the use of experiments drives the entire bml cycle. Systematically exploring the loop is observed that the **ideas** about the product becomes an assumption, which will drive the whole experiment. First, the assumption is defined to guide the experiments. From the assumption definition, it is possible to define the hypothesis in order to validate the assumption. The results can refute or accepted the assumption and end the experimentation loop.

The methodology also defines a concept called leap-of-faith, which is the riskiest element of the experiment plan. The leap-of-faith is translated into the two main assump-

tions: value assumptions and growth assumptions. One has the concern of the delivered value into the product and whether the customer perceives that value. With the first one defined and confirmed by customers, it needs to perceive how to grow the solution.

So, the loop starts with those ideas translated into a hypothesis for **product** that is defined as a minimum viable product (MVP), in the build phase. The MVP now acts as a tool to collect the customers' feedback, which allows the measurement of the product assertive. The **data** collected from the customers' feedback will serve as metrics to test the hypothesis. Finally, with the first version of the MVP and with the collected data, the loop gets into the learning phase. This phase allows the learning about the product, will get information about whether the assumption was accepted or refuted, and gives input to improve the product based on the metrics collected and validated.

The loop also give inputs for companies about the direction toked. At this moment, it is possible to decide to pursue a new direction of the business, which means change the strategy (pivot) or continue and scale it the product(persevered) [58].

The focus on the customer value addressed into the products, and the systematic methods as experiments, the build-measure-learn loop that leads the methodology have gained the attention of the software development industry. There are some studies [22], [48] reporting the use of Lean Startup in the software development process and the benefits of use.

Edison, Wang, and Abrahamsson [22] conduct a case study and points out the benefits and challenges of the use of the methodology by companies. Among the benefits pointed out in the study, it was highlighted the build-measure-learn loop application, which is indicated in the whole software development process, which allows the continuous process of learning what gives value to customers. Also, the use of lean Startup helps the company to address value into the product and to find the right market segment faster. The study also points out the difficulties of implement lean startup in companies that have bureaucratic characteristics. They explain that there is a need to promote autonomy and freedom to teams that are applying the methodology.

2.4 Transformation Process

An agile transformation is introduced in companies for several reasons: align the product with corporate strategies, respond to market changes, teams' not engaged to the development method, and changes, teams' highlight the need to align software development with corporate strategies, the need to rapidly respond to market changes, and the teams' dissatisfaction with the current work process and culture.The transformation process comes with a set of changes (e.g., structural, technical, and cultural changes [53]) that the organizations need to employ. The difficulty which many companies have to software project

management issues, as people management and managing schedules [53] is also a motivator. Also, other motivators are the extra bureaucracy [52], process gates [12], change management overhead [72] and excess documentation [32].

The use of a strategy to guide the transformation process (generally either a “big bang" approach, adopting all practices by-the-book; or a “gradual" approach, gradually integrating agile practices into the organization), is essential to conduct a transformation process [38]. The “big bang" strategy, start learning the agile practices by-the-book, and then begin to modify those practices. Meanwhile, the gradual approach remain using the non-agile practices at first, and gradually integrate the agile practices in their process. The second strategy could be a little bit longer than the first one.

Rohunen et al. [59] pointed out in their study the use of bottom-up and top-down strategy to conduct an agile transformation. The bottom-up adoption usually could be by the team level that must be self-organized and empowered to adopt agile practices. The top-down strategy on the other hand cover the operating lean development, drive the business objectives and the transformation process itself. Despite the strategies for adoption and transformation, the study highlight that observing the response to changes and adjust the adoption process are prerequisite for a successful transformation process.

Aside from the need of a strategy to guide a transformation, it is crucial that teams have a mechanism, typically an agile transformation model, that sets up and supports an improvement process during the adoption phase, while also enabling the full benefits of agile practices and techniques to take place. These models are often organized in a 2 by 2 matrix that defines the specific aspects that the model will map (e.g., attributes and dimensions of a formal software development process [55]) and are oftentimes divided in levels or stages [64] while emphasizing key characteristics that result in a successful agile adoption (e.g., people, process, project, and product characteristics [68]).

The success factors of an agile transformation were mapped in the study of Dikert et al. [17]. Among the many initiatives that were considered successful are the commitment to change, which means showing strong commitment during the transformation process, create a pilot team to lead the process, and use the benefits gathered by these first team to replicate in others teams, engaging people defining “agile supporters", and change the team mindset concentrating on agile values and promoting social events.

As expected there is an extensive number of challenges during this kind of transformation. The change of resistance, the needed of coordination and communication through several teams and among different organization units, hierarchical management and organizational boundaries, and requirements engineering challenges are some of the challenges mapped on [17].

2.5 Agile, User-Centered Design, and Lean Startup

In the past two decades, agile methods have taken over in popularity over more traditional methods for software development [33]. A crucial agile advantage is the ease of interaction and collaboration among team members and customers, which is supported by a range of ceremonies that promote such interactions, as daily meetings, stakeholders meetings, and so on [38]. Still, despite the incentives to team-user collaboration, there is still a lack of user involvement [62] and difficulty in addressing business value into the product. Vilkki [71] argues that agile needs to be combined with other approaches.

There are a set of benefits reported on literature about the use of such approach. The advantages are related to have the user closer to the development process [75], [21], [29], which are more encourage by the use of UCD or DT techniques, that some studies reveal that the use of these techniques increase the assertiveness and the development of a user-centered product. Also, the change of mindset for a problem-oriented development was related as a gain, as well as the use of experimentation, due the fact that the hypothesis, and assumptions definition and conduct the experiments contributes with the change of mindset focusing on the problem understanding. As a consequence, the study conducted by reveals that teams felt a significant advance in the relationship between them and the stakeholders.

The challenges associated with the use of the combined approach were identified as difficulties to build a cross-functional teams [44], which is a consequence of the organizational culture do not understand the transformation that occurs in teams that adopt the combined approach. Although, one of the most challenge identified in the studies which reports the combined approach is the difficulty to illustrate the combination and elements that composed of. The studies often reports that UCD, and Lean Startup brings benefits complementing agile methods, however is difficult to affirm where UCD started, or where Lean Startup takes place, what activities belong to each methodology (e.g., where the build-measure-learn loop begins or how the UCD activities cover the agile gaps).

3. SYSTEMATIC MAPPING

This systematic mapping aims to investigate the state-of-art under the use of Agile, User-Centered Design, and Lean Startup altogether in the context of software development.

We decide to conduct a systematic mapping review (SMR) using Petersen et al. [57] guidelines to accomplish the goal. Petersen et al. [57] define a systematic mapping study as a particular type of Systematic Literature Review (SLR). The SMR concentrates on locating and classifying the literature. Systematic mappings also can be used to structure a research area [57]. Aiming to cover all studies of the area, since we have few studies accepted from the SMR, we also conducted a Snowballing sampling using Wholin [73] recommendations.

Figure 3.1 illustrates the study flow. Next, we explore in detail each step of the systematic mapping.

3.1 Research Questions Definition

We define as L.RQ the research questions of this literature study to facilitate the identification and separation of the different type of studies conducted in the research. Following the SMR characteristics, we define the first question aiming to map the area and know the conferences or journals that are publishing the subject and timeline of publication. The L.RQ2, L.RQ3, and L.RQ4 on the other hand, aim to understand in depth how the combined approach is adopted and used, and also the benefits and challenges of such usage.

- L.RQ1. Where and when were combined approach of Agile, UCD, and Lean Startup studies published?

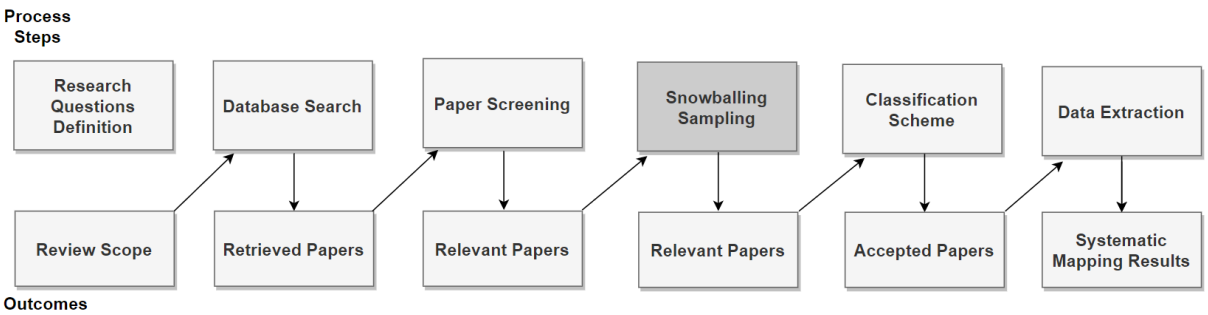


Figure 3.1: Systematic Mapping Flow
[Adapted by Petersen et al. [56]]

- L.RQ2. How the combined approach of Agile, UCD, and Lean Startup is used in the software development context?

- L.RQ3. What are the benefits of the combined adoption of Agile, UCD, and Lean Startup?

- L.RQ4. What are the challenges on the combined adoption of Agile, UCD, and Lean Startup?

Following, we describe the database search and the string definition.

3.2 Database Search

We define the string using the Population, Intervention, Comparison, and Outcomes (PICO) criteria, as suggested by Kitchenham and Charters [40].

- Population: In the context of software engineering, the population refers to a specific software engineering role, category, an application area, or an industry group [40]. In this study, the population is studies that report empirical studies of the combined approach application on the context of software development

- Intervention: Intervention means in the context of software engineering to software methodology, tool, technology, or procedure. In this study, the intervention is associated to a software methodology, seeking for studies that use the combined approach for software development

- Comparison: This study did not have the sense to make any comparison with other methods, because the focus is to understand how these concepts works and not compare to any other;

- Outcomes: No measurable outcome is considered in our study, as we do not focus on studies evaluating the combined approach usage.

- Context: For Software Engineering, this is the context in which the comparison takes place [40], and this study will not make any comparison.

The main keywords identified for this study are **Centered Design**, **Lean Startup** and **Agile**. Their synonyms were considered and used on the search string.

Database	Papers	Duplicated
IEEE	36	2
ACM	10	0
Springer	41	12
Science Direct	140	13
Scopus	110	14
Total:	337	41

Table 3.1: Search Results

Inclusion Criteria	Exclusion Criteria
Papers written in English	Papers not fully available online
Papers that present the integration of the three concepts together	Papers not related to software development context
	Books, Extended Abstracts and Editorial Papers
	Duplicated papers

Table 3.2: Inclusion and Exclusion criteria

We formulate the search string with the keywords identified, and performed on the IEEExplore[1], ACM Digital Library[2], Springer Database[3] and Science Direct[4], Scopus[5] databases. The search string of each database can be found in Appendix A. Following, we show an example of the search string used on the Scopus database.

> *ALL (“Design Thinking" OR “*Centered Design" OR “*Centred Design") AND ALL (“Lean Startup" OR “Lean Start-Up" OR “Lean UX") AND ALL (“Agile" OR “Agile Practice" OR “Agile Method*" OR “Agile Development" OR “Agile Software*" OR “Extreme Programming" OR “Scrum" OR “Kanban") AND (“Software Development")*

We utilized StArt[6] as a tool to support the whole process of systematic mapping.

The search retrieved 337 papers on the total, 41 of the 337 were duplicated. Table 3.1 brings the results individually for each database, including the duplicated papers.

3.3 Paper Screening

First, we conducted a database search that resulted in 337 papers retrieved. We apply the inclusion and exclusion criteria based on the titles, abstracts, and keywords read-

[1] http://ieeexplore.ieee.org/Xplore/home.jsp
[2] https://dl.acm.org/
[3] https://link.springer.com/
[4] https://www.sciencedirect.com/
[5] https://www.scopus.com/home.uri
[6] http://lapes.dc.ufscar.br/tools/start_tool

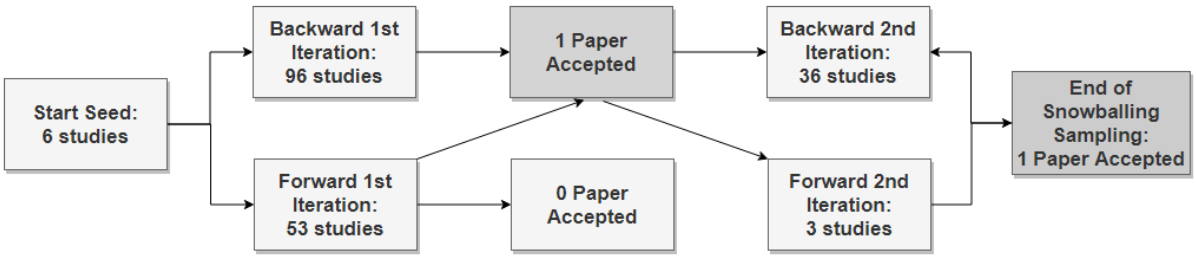


Figure 3.2: Snowballing Procedure

ing (Table 3.2). In the first phase, we accepted seven studies. From that, we performed a full-text reading as a second filter, and we rejected one more paper, concluding in six accepted papers. Seeking to increase the number of accepted papers and concerned to cover the existing studies, we conducted a snowballing sampling, using the six papers accepted on the database search, as start seed [73]. We detail the process performed next.

3.4 Snowballing Sampling

We decide to conduct a snowballing sampling since we accepted only six studies from the systematic mapping search. Wholin [73] suggests the use of snowballing to increase the studies coverage in the study area, and as a tool to supplement the results from another literature review approaches. Wohlin [73] brings attention to the challenges of applying a snowballing search. The need to have an initial set is defined as one of these challenges. In our case, the definition of start seed it is the six studies accepted on the systematic mapping. We conducted two iterations of backward and forward snowballing based on the start seed defined. Figure 3.2 illustrates the whole process.

We start to conduct the iterations from the start seed defined. The backward iteration which consists of investigating the references from the starter seed. The forward iteration consists of the search for studies that have cited the start seed articles [73]. We used Google Scholar [7] to search for all referenced and cited studies, as Wholin [73] recommends.

We identified from the first backward iteration 96 studies, and the forward 53, totalizing 149 articles. We filtered the papers using the same criteria defined to the mapping study. We read title-abstract-keywords and applied the inclusion and exclusion criteria (Table 3.2). We selected one paper from the first backward iteration (see paper id S1 in Table 3.4), and for the forward, we did not add any paper.

Based on the one paper included in the first iteration, we conducted one more sampling of backward and forward. From the backward second iteration were retrieved 36

[7] https://scholar.google.com.br/

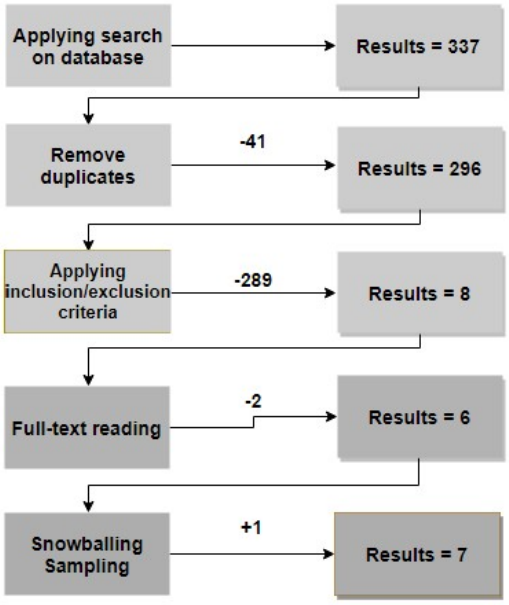


Figure 3.3: Systematic Mapping Procedure
[Adapted from Petersen et al. [57]]

studies. Meanwhile, the forward iteration returned only three studies, totalizing 39 studies. In this iteration, none paper was accepted, and the process ended.

With the snowballing sampling finalized, we were able to start the categorization through keywording using abstract, and also the data extraction. Figure 3.3 presents the whole process of systematic mapping + snowballing conduction and the results retrieved.

From the snowballing sampling, one more paper was added to the accepted papers list. Afterward, we finish the searches with a total of seven papers accepted (Table 3.4).

3.5 Classification Scheme

We discuss in this section how we create categories to translate them into a classification scheme for this study. Petersen et al. [57] mentioned that the process of keywording helps us to reduce the time on the development of the classification scheme. As mentioned before, we decided to read title-abstract-keywords and not only the abstract, as recommended by Petersen et al. [57]. Over this process, we could already identify the main contribution of the paper and categorize it. The categories allow us to have a more straightforward interpretation and use for classification without evaluating each paper in detail [57]. We adapted the categories from Petersen et al. [56] classification of *Research Type Facet*. From that, we defined our own categories based on the papers first reading. The classification scheme is illustrated in Table 3.3.

Category	Description
Research Method	
Case Study	A report a specific phenomena studied
Evaluation	Evaluation of a product or a model
Development	Development of a product or a model
Report	Report about the application of the approach
Contribution	
Advice	Recommendations based on personal opinions
Framework/Model	Model used to develop a new software or creation of a new model
Lessons learned	Actionable advice derived from real experiences.
Focus	
Development of a new process	Development of a new model or process using the three approaches
Application in real projects	Using the three approaches for a real project in a real company
Educational	Applied the model for development of software for students

Table 3.3: Classification Scheme
[Adapted by Petersen et al. [56]]

We used this classification in Table 3.4, which contains the columns *research method*, *contribution*, and *focus*.

Paper ID	Title	Author	Venue	Publication Year	Research Method	Contribution	Focus
P1	Lean UX: The Next Generation of User-Centered Agile Development?	Lassi A. Liikkanen; Harri Kilpiö; Lauri Svan; Miko Hiltunen.	Nordic Conference	2014	Evaluation	Lesson Learned	Application in Real Projects
P2	Applying Lean Startup: An Experience Report – Lean & Lean UX by a UX Veteran: Lessons Learned in Creating & Launching a Complex Consumer App	Beverly May	Agile Conference	2012	Case Study	Lesson Learned	Application in Real Projects
P3	Pet empires:Combining Design Thinking, Lean Startup and Agile to Learn from Failure and Develop a Successful Game in an Undergraduate Environment	Danielly F. O. de Paula; and Cristiano C. Araújo	International Conference on Human-Computer Interaction	2015	Case Study	Advice	Application in Real Projects
P4	Software Project Management Combining Agile, Lean startup and Design Thinking	Bianca H. Ximenes; Isadora N. Alves; Cristiano C. Araújo	International Conference of Design, User Experience, and Usability	2016	Development and Evaluation	Framework /Model	Educational/Application in Real Projects
P5	The Best of Three Worlds - The Creation of InnoDev a Software Development Approach that Integrates Design Thinking, Scrum and Lean Startup	Franziska Dobrigkeit and Danielly de Paula	International Conference on Engineering Design	2017	Development	Framework /Model	Development of a New Process
P6	InnoDev: A Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup	Franziska Dobrigkeit; Danielly de Paula; Mathias Uflacker	Design Thinking Research	2019	Development	Framework /Model	Development of a New Process
S1	Skip the Silver Bullet: Driving Innovation Through Small Bets and Diverse Practices	Ben Grossman-Kahn and Ryan Rosensweig	Design Management Institute: International Research Conference	2012	Development and Evaluation	Framework /Model	Development of a New Process

Table 3.4: Systematic Mapping Overview

Data item	Value	RQ
Paper Id	Integer	
Title	Name of the study	
Author Name	Names of the authors	RQ2
Year of publication	Year	RQ1
Journals/Conferences	Name of the journals/ conferences	RQ1
Integration of the methods	How the integration works, models	RQ3
Benefits and challenges of the integration	Benefits and challenges in the application of the integrated approach	RQ4 and RQ5

Table 3.5: Data Extraction Form

3.6 Data Extraction

We developed a template for the extraction of data that can be observed in Table 3.5 that have the information that we desire from the study and the value.

We decided to compute the coefficient of the agreement under the selected papers as well. The coefficient of the agreement was performed by two researchers and was evaluated by the Kappa statistical test [13]. Another research analyzed 25 papers in the first iteration (from the papers retrieved before the selection phase), and the percent was classified as almost perfect agreement (100%), as Landis and Koch [43] defined and suggested.

3.7 Systematic Mapping Results

We present now the extracted findings from the seven papers selected in this study. We organize the answers according to the RQ's defined.

3.7.1 L.RQ1: Published Studies

Publication Venue

Table 3.4, column *Venue* provides an overview of the articles per publication venues. We observed that the studies often were published on conferences associated with the design, user experience, and human-computer interaction area. The studies were mostly published in conferences, and only one of them was published as a book chapter by Springer.

Frequency of Publication

Table 3.4, column *Publication Year* presents the number of studies reporting the combined approach identified within the years 2012–2019. The first study was published by May [46], and Dobrigkeit et al. [20] published the last one, which also have a study published in 2017 [21].

3.7.2 L.RQ2: Context of Use

Table 3.4 presents the authors' articles, also presenting the paper id, title, the research method used in the article, the contribution, and focus (See Table 3.3 to see the categories description).

We identified on literature seven studies that explore the combination of Agile, UCD or Design Thinking (DT), and Lean Startup. We were able to identify that the combined approach could be described as a *design model*, since Gothelf defined Lean UX as a design process, and a mindset [27], and *software process model* [75], [29], and [21], since those models are a representation of a software process [67].

We also found that teams using this combination should have a member configuration that meets the three primary needs in building a software product. Next, we explore in-depth models.

Lean UX

Lean UX [27] is a design process model that is grounded on the methodologies of agile software development, design thinking, and lean startup. Lean UX applies the four principles of the Agile Manifesto to product design. Besides that, Lean UX has its manifesto with 15 principles:

- Cross-functional teams: The teams must be made up of various disciplines involving, for example, software engineers, product managers, interaction designers, visual designers, and other roles.
- Small, dedicated, co-located: The teams must be small- no more than ten members. Moreover, they must be dedicated to one project in the same location.
- Progress = Outcomes, not output: The business goals must be achieving generating outcomes. the outputs are associated with features and services, which just do not deliver value to the business.
- Problem-focused teams: The teams must have the mindset of work on the business problems, and not on a set of features to implement

- Removing waste: Anything that does not contribute to the business goals, which means the outcomes are considered waste and should be removed from the team's process.
- Small batch size: Create only the design that is necessary to move forward
- Continuous discovery: This principle aims to make the teams understand that it is necessary to continue discovering what the users are doing with their products and why they are doing it.
- Goob: The new user-centricity: Goob means an expression "Get out of the building". In the context of Lean UX, it is to encourage the teams to do customer research, observe the problem in the user context, give potential customers the chance to provide feedback to the ideas as soon as possible.
- Shared understanding: The knowledge about the space, product, and customers is not individual. It is necessary to build up collective knowledge as the team works together
- Anti-Pattern: rockstars, gurus, and ninjas: The team must not have an individual that solves all the problems. Lean UX emphasizes the team-based mentality.
- Externalizing your work: The teams must to externalizing their work using collaborative tools like whiteboards, artifact walls, and sticky notes, exposing the work in progress
- Making over-analysis: In Lean UX, there is more value on making a first version of the product over discussing and debating for hours the possibilities
- Learning overgrowth: Lean UX prioritize the learning about an idea over a scale that idea rampantly
- Permission to fail: The teams must to experiment with the ideas and have permission to fail in a safe environment
- Getting out of the deliverable business: Lean UX does not focus on documenting everything from the design process. The teams must focus on delivery right products over-focus on documentation

The whole model that has focused on the design process incorporate into the development of a product had defined the principles based on the concepts of the three methodologies (e.g., cross-functional teams from DT, permission to fail from Lean Startup, and getting out of the deliverable business from agile).

We found in our systematic mapping, two studies that have applied Lean UX to develop their products. The papers are P1 by Liikkanen et al. [44] and P2 by May [46].

P1 [44] introduces Lean UX in their software development process. The article reports a case study observed in a software agency. The authors reported that the company

Figure 3.4: P1 - Software Development Process
[Source: Liikkanen et al. [44]]

started integrating Lean UX to the current agile method - Scrum. Using the sprint concept, the company divided the sprint planning in design and feedback, and two weeks of development with two feedback cycles between the two weeks (Figure 3.4).

The sprints begin with sketching and ideation exercises - this produces a first draft for the design outputs that will be developed during the sprints and the next ones. Furthermore, to reinforce the user involvement, they apply user testing in each sprint, instead of a single test at the end of the feature development. The test frequency change leads the teams to use the MVP concept developing the most significant and valuable pieces of the product and receiving quick feedback from the users.

The authors mentioned that the teams use the personas definition as a manner to have a user representative. The personas are used in user stories as a manner to identify the desired of the user. Moreover, other factors are the use of experiments, defining a hypothesis, and testing it to understand whether the teams are on the right path or not.

Unlike P1, **P2** [46] is an experience report of the application of Lean UX under the development of an application named as MiniDates.com, which is a consumer dating application. The lessons learned reported in the paper was from the experience of a senior UX product and a software development consultant.

P2 author reported that the agile method chosen was a scrum, implementing the concepts of sprints, user stories, short release cycles, and frequents releases, and also the extreme programming practices as test-driven development, and pair programming. To cover the user side, the author said that the experts had used the UCD concepts, designing the product architecture based on the user experience and needs. Also, the experts have concerned about develop product-oriented by the 14 principles of Lean UX.

The author does not mention specific steps where every methodology is applied. However, the article reported the use of techniques from UCD, like personas, card sorts, user research; wireframes; user flow; prototypes. Furthermore, from Lean Startup, the use of experiments; define growth hypothesis; creating MVP, and emphasizing the validate learning in each delivery.

NordStrom

The Nordstrom innovation lab developed the Nordstrom model, and the paper **S1** [29] reports the development of the model. The Nordstrom is an approach that combines Agile, DT, and Lean Startup. The company that owns the lab has a focus on innovation and was facing problems associated with the relationship with the customers.

The model uses the activities, techniques, and concepts from DT at the beginning, seeking to understand the users' needs. With a problem vision defined, the model proposes the assumptions and hypothesis definition to start the experiments, which enable the start for the build-measure-learn loop. Inside of the build-measure-learn loop, the iteration started, using an agile method, which leads the teams to execute activities, apply techniques, conduct ceremonies, and use practices from agile.

The study S1 [29] describes in detail the process of creation of the model that starts integrating DT and Lean Startup to the already stated Agile method. The model started with only one software developer who proposes to include a set of agile practices (e.g., pair programming), incorporate agile ceremonies (e.g., daily stand-up, retrospective), and begin a customer-centered prioritization to features.

After the changes associated with agile, the lab realizes the need to attend the business needs in order to gain support from the stakeholders, and this leads it to include the BML cycle to the process. The authors mentioned that the BML cycle is at the core of the process, providing a learning cycle through ideas and prototypes, testing, and iterating trough experiments that allow the teams to measure the users' behaviors.

The authors' feelings about how the practices of Lean Startup were applied during the process of the model are described in the following quotation:

> *"We pair on work, test-drive our ideas and develop iteratively. Lean reminds us to visualize our work and reduce cycle time. At the heart of all three approaches are an iterative mindset, a relentless focus on the needs of the customer, and a bias towards rapid experimentation, prototyping, and testing."*

With the Agile and Lean Startup integration, the lab experimented with external customers seeking to validate their learnings through the first version of a product; this first version was the MVP. They reported as a lesson learned the need to put the user at the center of the product development.

The authors use the IDEO Human-Centered Design Toolkit[34] diagram, to start implementing Design Thinking as a way to develop more profound customer empathy and frame problems in a way that reflected the needs of the customer rather than the business[29]. As they already have used Agile and Lean Startup, they integrate the methods with Design Thinking, including a bunch of techniques as ethnographic research, customer interviews, brainstorming sessions, and low fidelity prototypes.

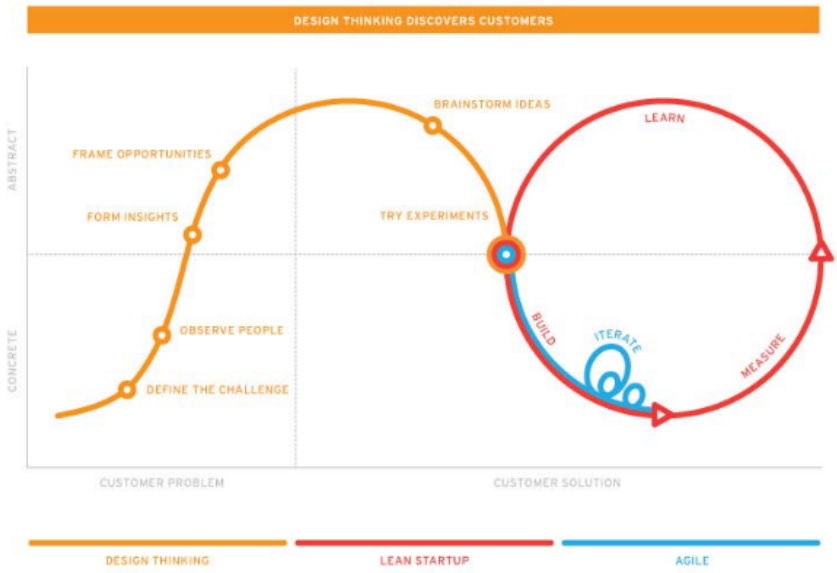


Figure 3.5: Nordstrom Model
[Source: Grossman-Kahn and Rosensweig [29]]

Figure 3.5 presents the integration among the areas proposed by Grossman-Kahn and Rosensweig [29].

Another study reports the Nordstrom model usage in the software development process. De Paula and Araújo [15] reported on **P3**, the application of the Nordstrom model by a team of three computer science undergraduate students to develop a game. The team was composed of developers and one designer. The team used the model sequentially (as proposed in Figure 3.5, starting with DT, after Lean Startup and Agile.

The authors aimed to offer insights about how startups can benefit from the combined approach usage. The study reports two cycles of development. In the first cycle, the team applied the Nordstrom model applying the methodologies sequentially. After a focus group session with a local group of game developers, the game received a bad evaluation, which made the team consider to modify the manner of how the model was applied.

In the second cycle, the team decides to implement the use of DT practices during the whole process, different from the Nordstrom model was proposed. The team made the first evaluation through a questionnaire in an event to collect quick feedback about the game. The team reported on the study that DT practices usage in the entire process offered rapid feedback to solve micro problems. They also identified that the team stops guessing what are the user's needs, through the use of DT techniques, they can be sure about the real user's necessity.

Converge

P4[75] proposes the converge model aiming to help teams on the teams project management. The model was inspired and based on the Nordstrom model [29], and the

Figure 3.6: Converge Model
[Source: Ximenes and Araújo [75]]

study reports the evaluation of the model on the application development by an undergraduate team. Figure 3.6 illustrates the model structure.

The model puts agile at the center of the software development process. The agile method chosen was Scrum, using the ceremonies proposed by the method, and also the use of practices as pair programming and collective code ownership from XP. With the agile method, ceremonies and practices defined, the model start to define and prioritize the hypothesis and assumptions, concepts from the Lean Startup methodology, as well as the definition of critical metrics, and MVP developed. Also relevant is the product vision definition and the constant feedback cycle.

The authors propose the model as a manner to put the user at the center of the development. They argued that exists a gap in the literature regarding methodologies that do not prioritize the user's need. The model implemented DT techniques and activities. The authors mentioned that the activities structure used is from Stanford's d.School model for DT, which has five types of activities: empathizing, problem definition, ideation, prototyping, and testing. DT is present during the project beginning, helping the teams to identify the problem and validate the ideas.

The authors explicitly report the need for a mindset change, associated with how the requirements arise for the development teams, they argue that the requirements must be a mix about the teams' belief and the business desire. The authors conclude that the key to this model is that the team is responsible for the idea and design. Also, the team is fully responsible for the development of the final product.

An undergraduate team empirically validated the model, and the results from that evaluation reveal that the teams started defining potential users for a design challenge proposed by the researchers. These results enable the team to initiate the DT activities, the team conducted brainstorming sessions, user interviews, and gathered inputs to identify the problem and leading it to define the lean canvas and define the product vision. The whole

ceremonies occur during the whole software development, and the hypothesis creation and validation were constants.

The authors reported that the model was sufficient for app development, mainly because of the team inexperience. However, they point out the relevance of improving the model before trying to scale it. Besides, all team members take part in the creative process from the very beginning until the end. Developers do not receive ready-made requirements from others, taking part in the making of the product concept, being able to innovate[75].

InnoDev

The Innodev Model was reported in two studies **P5** [21] and **P6** [20]. This model was created based on a comparison of two other models, MoIT [16], which is a model that combined DT and Lean Startup and DT@Scrum [31], which is a combination of DT and Scrum. The authors analyzed the similarities and differences among the models, and based on this comparison was developed the InnoDev model.

The InnoDev model has three phases: Design Thinking, Initial Development, and Development phase. Figure 3.7 illustrates the model. The starting point for the process often is a challenge or a general area of interest. The goal of the Design Thinking phase is to understand user needs and related products. The authors applied the DT model proposed by Wölbling et al. [74], which have defined the understand, observe, synthesis, ideate, prototype, and test activities. Besides that, it is expected that this phase produces the following outcomes: problem and solution exploration, define a product vision addressing at least one of the identified problems.

Initial Development aims to refine and test the product vision from the previous phase and start the Minimum Viable Product (MVP). At this phase, some DT activities are conducted, especially the prototype and test activities, since the user interface (UI) concepts are tested and implemented in this phase. This phase also includes the identification of metrics.

Furthermore, the Development Phase, the MVP, is tested and gradually incorporated into the product feature through the build-measure-learn loop. In parallel, the team must conduct agile sprints. The results from the learning phase give inputs for the team to decide to persevere with the strategy or pivot the idea.

The authors considered that the InnoDev model could be implemented in different company settings, like startups or large organizations. However, it is relevant to say that real teams did not evaluate the model either in P5 [21] nor in P6 [20]. The differences between the two studies were the granularity of information from one article to another. P5 reports an overview of the model, and P6 explores in detail the development of the model.

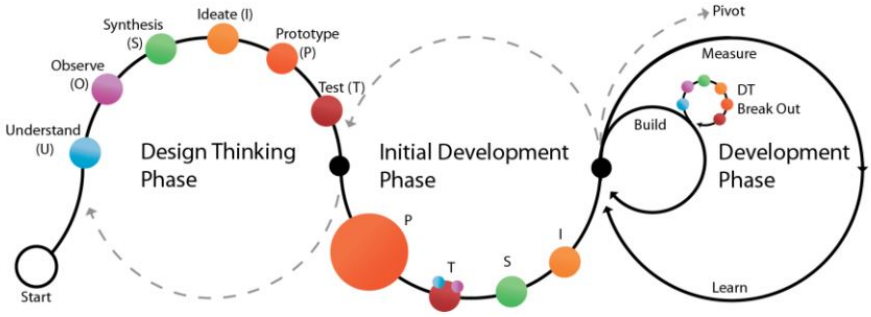


Figure 3.7: InnoDev Model
[Source: Dobrigkeit et. al [21]]

3.7.3 L.RQ3: Benefits

P2 author [46] cited the relevance of gathered feedback with potential customers during the product development. The experts went to some events and ask for potential users to test the product. This practice is one of the principles of Lean UX, the GOOB: get out of the building. The author also mentioned the continuous test as a benefit. The experts have recommended during the entire process, and test from their point of view means *"trying, evaluating and researching".*

For **P3** authors [15], the use of design thinking during the whole process was considered as greater benefit.

> *"Design thinking should be used in the entire process of the Nordstrom Model. Understanding the challenges faced by IT teams to combine Design Thinking, Agile, and Lean Startup is critical to help the industry and literature on how to improve software development. For IT teams, knowing how to overcome the challenges when adopting design thinking will help them to improve their software development processes and launch more innovative products [15]."*

Also, the authors considered a benefit gathered feedback with potential customers during the product development. This feedback enables the teams to identify problems in the product's first version and fix that aiming to align the product to the users' expectations.

P4 [75] also identify the use of design thinking during the whole process as a significant benefit. The teams argued that have the users closer to the development process since the product conception, guarantees that the user's needs and expectations are met in the product delivered. The use of techniques as user research, empathy maps, and others, were highly recommended.

The results of the case study reported in P4[75] bring the possibility to combine the methodologies Agile, Lean Startup, and Design Thinking in rounds. The authors also mentioned that the combination of the three methodologies help teams focus on the user

problem beyond only requirements and tasks. One of the earnings of this model is that having inexperienced developers and designers did not influence the team performance on the use of the combined approach [75].

The **P5** [21] and P6 [20] authors identify that the exploration of the concepts of Lean Startup, as BML cycle and experiments, for example, can enhance the software development process. They also identify that the use of Design Thinking is essential for software development, which provides the approximation with the users. The authors of P5 [21] and P6[20] explain that the InnoDev model is considered a generic model and could be applied to different companies settings, be they startups or large companies.

The authors recommended the InnoDev model fits for large organizations even though they do not have the opportunity to evaluate in this company setting. However, they claim that the model could benefit from the use of combining Lean Startup with the other methodologies as the main point. Based on the BML cycle, a company could enhance its software development process by making use of lean concepts such as MVP, actionable metrics, and pivot [21].

S1 [29] reported some interesting benefits on the use the concepts of Lean Startup altogether with Design Thinking and Agile. The authors explain that bringing the Lean mindset brings relevant benefits to improve the relationship with the clients, which started to getting closer with the IT area and began to believe in the release, as the product is aligned with the business needs.

3.7.4 L.RQ4: Challenges

P1 [44] reported as a challenge is the difficulties to define a cross-functional team. The teams observed in the study, cannot allocate people for all the roles, as indicated by Lean UX. The teams do not have a UX Designer position, for instance. They faced problems in the outsourcing culture because the outsourcing team wanted to keep the old team structure of design, development, and user testing separated by acquiring each of these services from a different provider. The authors argued that this kind of structure is directly against Lean UX advocates. The authors also mentioned barriers at the organizational level, as, convert clients to waterfall for Lean UX, facing difficulties to help them to understand the new way of work, and changing the clients' mindset. Organizational culture and the decision making power about the product be concentrated in the business area were also pointed out as challenges faced associated with the team autonomy.

The author of **P2** [46] has reported as challenges that the team work in multiple projects, which not use the same approach that they were trying to apply to develop the app. The experts of the study mentioned that this impact on the software architecture decisions. Also, they have reported that they overestimate the UX knowledge that the UX

experts already have, and this generates a lack of use UX techniques that could be helpful to the product (e.g., wireframes, prototypes).

> *“Given our specialty in this area, we should have gotten this right, but as noted previously, we tried to cut corners here to save some time and money. There are many things we did “right": early wires of some views, using a UCD design process, personas and wires, and plenty of market and user research. Things we skipped: rapid prototyping, maintaining the wireframes and completely diagramming the app and all flows based on HTML5 standards, and early and frequent user testing of all parts of the app and its demand (see teal dots on the diagram for the UX and design process)." [46]*

P3 [15] brings of the team misunderstood the quickly launches concepts. The team just followed the concept of rapid deliveries. However, the product was not aligned with business needs. The teams have the anxiety to make the quick releases and get the product ready, but when the real users receive the product was different from what the users expected.

S1 [29] have challenges in the business audience. The authors explain that there is a resistance of business side related to the changes proposed by the IT teams. The ideas are generated, but the business area does not approve of the beginning of the process. **P4** [75] **P5** [21] and P6 [20] do not report challenges related on the use of the models, due the fact that the studies does not report a model evaluation.

Table 3.6 summarizes the main findings of the systematic mapping, including benefits and challenges on the application of the combined approach models.

Combined Approach Definition	Combined Approach Nomenclature	Combined Approach Description	Phase	Team Structure	Mapped Benefits	Mapped Challenges
Software Design Process	Lean UX [46], [44]	The model are grounded on the combined approach, having as priority the junction of lean startup an UX. It have defined 14 principles, and is focus on the design process.	There is no phases defined	Small and cross-functional teams.	Gathered feedback with potential customers [46] Continuous test [46]	Define a cross-functional teams [44], [46] Organizational Culture [44] Team working in multiple projects [46] Overestimate the UX knowledge [46]
Software Process Model	Nordstrom [29], [15]	The model was developed by the nordstrom innovation lab and focus on the software development process aiming to create innovative process. Focus on the problem oriented and seek to understand the user needs, identify the problems and propose solutions.	**Design Thinking:** Start the development process using DT activities and techniques. However, DT is present during the whole development process. **Lean Startup:** With the problem defined, the use of Lean Startup activities, as experiments and the build-measure-learn loop start to appear in the model. **Agile:** The whole iteration using agile are inside on the build-measure-learn loop and by executing experiments. * The whole process is oriented by the agile meetings/ceremonies.	Small and cross-functional teams.	Gathered feedback with potential customers [15] Use of DT during the whole process [15] Use of Lean Startup concepts [29]	Organizational Culture [29] Decision making power [29] Misunderstood the quickly launches concept [15] Resistance of business side [29]
Software Process Model	Converge [75]	The model was developed inspired on the nordstrom model. Converge model is applicable to development teams, in order to provide creative solutions for the products.	**Agile:** It is presented during the whole process and is the foundation of it. The definition of a feature-by-feature implementation is strictly followed, as well as the extreme programming practices. **Lean Startup:** With the XP practices defined and applied, the team also concentrate of define a lean canvas aiming to define the product vision, and later work on the experiments by defining hypothesis and assumptions, all of this inside of build-measure-learn loop. **Design Thinking:** At the end, the Design Thinking activities and techniques are incorporated in the model.The authors use the Stanford's d.school DT model to define the activities. * The whole process is oriented by the agile meetings/ ceremonies.	There is no definition or recommendation for team structure.	Focus on the user problem [75] Use of DT during the whole process [75]	-
Software Process Model	InnoDev [21], [20]	The InnoDev model was developed based on the need of a general model that could be applicable for different company settings. Arose from a comparison of two models that combine Agile with DT.	**Design Thinking Phase:** The model starts seeking to explore the problem and solution sides, and to define a product vision. Uses the DT model to define its activities. **Initial Development Phase:** From the product vision and the problem defined, this phase aim to refine the product vision, and define the MVP. **Development Phase:** This phase is oriented by the build-measure- learn loop, and aims to test the MVP, collecting metrics and by the feedback. The agile sprints takes place on this phase, combined with lean practices.	There is no definition or recommendation for team structure.	Use of Lean Startup concepts [21],[20] Model fits for large organizations [21], [20]	-

Table 3.6: Literature Findings

3.8 Discussion

3.8.1 Published Studies (L.RQ1)

The findings provide a brief understanding of how the area is structured. Indeed this is a growing area; the publications started in 2012 and have some papers published until the year 2019. The most of the papers were published in conference of design area, only 1 of the seven papers is from a Software Engineering (SE) conference, and thinking about it, arise a question: The area starts to generate interest on SE now or is because the model has more gains used for the design of the products/software?

3.8.2 Context of Use (L.RQ2)

The studies show that any agile methods (Scrum, XP, Kanban) can be combined with Design Thinking (inspired by UCD activities) and Lean Startup. However, all papers use only SCRUM as a methodology.

Another important finding is that exist several models that use the three methodologies together for different contexts, as for a UX design team or a whole software development process. The most exciting thing is the fact that this new approach can be used in a different context and bring good results for all of them. The fact of none study was conducted in large scale companies was a finding of this mapping. The models were applied in startups or academic labs. This lack of not explore such company configuration can be identified as a gap and something to look with more attention. This kind of curiosity leaves us to think if the combined approaches are suitable for large-scale companies and whether it will be faced problems when applied in some complex environment.

One of the studies related that for the model application developed, the teams are composed of inexperienced developers and designs, still, this was not a problem for the process management. This leads us to question: This model is feasible for teams that have most experienced engineers and designs and few inexperienced? or vice and versa.

The results are not precise, regarding the fact of when each approach fits in the process of software development. What is the techniques and practices used, where the business people get involved, and what is the roles needed in a team that wants to use this kind of approach, there is not a definition for it?

3.8.3 Benefits (L.RQ3)

The results reveal that the transformation for teams that already have implemented in the agile method as a software development process is more natural and brings benefits under the adoption of the combined approach [44].

Associated with the use of UCD/DT techniques, the studies revealed that using concepts as GOOB and went to gathered feedback with potential customers seems to be very useful for product development. The use of the UCD/DT techniques and UCD activities encourage teams to change the mindset focused on the use problem, and not only on the requirement. Nevertheless, two of the seven studies recommended the use of Design Thinking techniques and activities during the entire development process. De Paula et al. [15] reported that the team observed in the case study, achieve better results from the users when applied DT in the whole product development process. The authors reported that using DT on the entire process increases the assertiveness of the product, due to the fact the team works closely and continuously with the user, always gathering feedback through the techniques from DT.

Another essential benefit reported in the studies was implementing and continuous test during the whole process. May [46] reported in the study, that the team of experts highly recommended the continuous test as a mindset. Furthermore, by test, the experts mean all kinds of tests. They argue that using this kind of mindset helps teams to develop a more certain product.

The use of the BML loop, techniques, and concepts from Lean Startup also was cited as benefits in the papers. The studies reveal that the use of a Lean startup promotes a better relationship with the stakeholders. This improvement occurs because the teams are seeking to focus on the stakeholders' problems.

3.8.4 Challenges (L.RQ4)

The challenges associated with the use of the combined approach were identified as difficulties to build a cross-functional team, and this is related to the fact of the organizational culture does not want to change and fit on the new way of work [44], [46]. The issues are also related to the decision making power, which is strictly taken by business people, that violates the teams' autonomy.

The business resistance was also a challenge faced and put in risk this autonomy. Grossman-Kahn and Rosensweig [29] reported that they put a significant effort into showing the benefits of the combined approach used for the stakeholders, and the results just arose after the first MVP version. This lack can be associated to the misunderstanding concept

of quickly launches. P3 describes that the team faced such a problem, and this impact the whole team's confidence.

May [46] was the only study that reported difficulties in applying Lean UX by the book, even that the team was composed of senior development and a UX expert. The author reported that they do not expect to face this kind of challenge. However, they struggled to make launches quickly or conduct user research.

It is relevant to mention that all of the studies retrieved in the systematic mapping were applicable for small teams, and most of the time was startups—one of the studies reported as a benefit, the fact that the model fits for large organizations. However, the study was not evaluated, and there is no empirical evidence of such affirmative. Also, the studies reported the use of techniques and a sequential timeline of where every methodology begins and ends, although there is no clear evidence related to how the methodologies complement each other.

3.9 Threats to Validity

We understand that exist several threats in the execution of these systematic mapping. The whole process can be invalidated; for example, if the search strategy is not correct or for the researcher bias. We try to mitigate that getting a second opinion from other researchers, first reviewing the protocol, reviewing the studies retrieved by conducting the kappa coefficient and also reviewing the selected studies. We must pay attention to the fact that only because the area is brand new, we cannot accept any paper that brings the concepts of Agile, UCD, and Lean Startup, due to our interest is on the methodologies combination.

4. CASE STUDY

We conducted an empirical study aiming to characterize the combined approach adoption and the transformation process by teams from a multinational company. As we observed, there is none study describing the combination of Agile, UCD, and Lean Startup usage in such a context. We present in this chapter, the case study design, case study protocol, case setting, data collection methods, and the analysis process. After that, we present the results derived from the case study research questions (C.RQ). At the end of the section, we presented the study limitations.

4.1 Case Study Research Design

Yin [78] characterizes the case study method by providing a twofold definition. First, *Scope* where is defined as the phenomenon to be investigated in-depth and in a real-world context, and *feature*, which arise from the phenomenon and context. We were motivated by these definitions to conduct a single case study aiming to understand our phenomenon of interest, which takes place in a real-world scenario.

Our scope is the *phenomena* or *case bounding* to be studied, which are two teams from a multinational company that is adopting a combined approach of Agile, UCD, and Lean Startup. The case study features, once again reaffirm our decision to research in light of this empirical method. We are interested in many variables and not only data points, and we rely on using multiple sources of evidence to triangulate the collected data.

Aiming to structure the research design, Figure 4.1 illustrates the process under the case study conduction. Guided by design, we present in section 4.3 the case setting, contextualizing the company, and also providing a team description. Section 4.4 presents the data collection methods as well. The conduction itself is described in the study. The two teams have the data gathered in parallel, and the data analyzed is in Section 4.6, which presents the case report.

4.2 Case Study Protocol

We defined a case study protocol following Runenson's guideline [60] to guide the research. The protocol was presented to ORG that approved the research conduction using the company as a case. The protocol presented can be found in Appendix B. Next, we explore the protocol in detail with the case settings - including the teams' description, data collection methods, and data analysis.

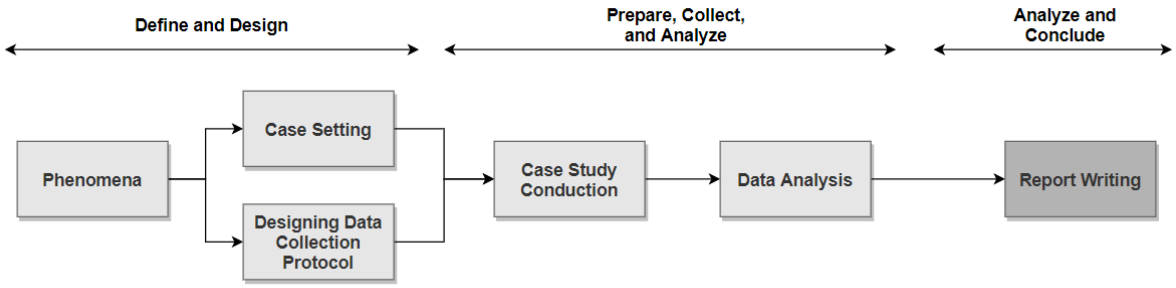


Figure 4.1: Case Study Research Design
[Adapted by Yin [78]]

4.3 Case Setting

ORG has development sites in the USA (headquarters), India, and Brazil. With over 7,000 employees and responsible for about 1,200 internal software products, the IT department started its agile transformation in 2015 and moved to the combined use of Agile, UCD, and Lean Startup principles in late 2017. Before adopting the combined approach, ORG had a well defined roadmap for software product improvements based on an annual budget negotiated among business department and organized into software projects. High-level business features were prioritized and decided upon business personnel to later be transformed into software requirements by IT software project teams. The project deadlines were strict and defined by quarter, i.e., every four months the project teams delivered a set of software features to existing or new software products to the company internal customers.

Associated with the business features definition negotiation, the company had Business Representatives responsible for defining the business needs. Once approved those needs were translated into business features, elected as the starting point for the IT project teams. Mostly, IT Business Analysts transformed these features into software requirements with the help of the Business Representatives and used these to drive software development.

We observed *in-loco* two teams from the financial area located in Brazil. These teams were co-located in a dedicated lab at the PUCRS University prepared for ORG teams following the Pivotal Labs' collaborative work environment recommendations (e.g., single large Table for pair-wise work, large screen TV for reports and news, large whiteboards for ideas' development and information sharing, and a meeting room that turns into an entertainment space for leisure time).

With the introduction of the agile transformation in 2015, project teams used Scrum as the guiding development framework. From this time and on, it become common but not company-wide spread to get more team members (e.g., developers, software architects, testers) engaged into the business feature-to-software requirement translation. Some teams

move then to a more product-oriented view while others are still guided by project time slots. The company starts then to discuss how to move from a world-wide roadmap to a product development organization when they realize help was need. This is when they decide to board the agile, UCD and Lean Startup combined journey and hire Pivotal consulting to support such transformation.

Overall, Pivotal brings the Pivotal Labs[1] methodology at core of the transformation. This methodology proposes a 'team rhythm' (or work flow) composed of principles and ceremonies based on the three before-mentioned approaches. It also suggests the adoption of a cross-functional team composed of three leading roles: Product Designer, Product Manager, and Software Engineer. The Pivotal Labs' main goal is to help teams to build software products that deliver meaningful value for users and their businesses. Thus, it offers a framework and initial starting point for any team to discuss the client/user specific needs and define its way towards software development.

Next, we describe the teams' product characteristics, as well as the teams' composition.

- *Team A* is responsible for a internal software product that manages, calculates, and generates data about company projects related to equipment (e.g., peripherals and computers for personal or server use) and service delivery (e.g., machine installation, support, and replacement). The product manages general project information, such as personnel assignments and time spent on tasks. It also calculates the associated costs of services offered by the products sold by ORG and displays this information to internal ORG consumers. The application generates profit data for each project which is consumed (along with the rest of the data) by the accounting department.

  Team A is composed of one Product Designer (enabler), two Product Managers (enabler and learner), and four Software Engineers (two enablers and two learners).

- *Team B* is also responsible for a internal software product that consumes data from multiple ORG applications (including Team A's) to calculate the average cost of equipment developed in Brazil. The application generates reports for internal accounting, such as inventory reports for tax purposes. The team is also working on automating the validation process for the data coming from each source.

  Team B is composed of one Product Designer (enabler), two Product Managers (enabler and learner), and four Software Engineers (two enablers and two learners).

In addition to the 14 team members, we also counted with the Brazilian Transformation Leader perspective in our study (see profile in Table 4.1).

[1] https://pivotal.io/Labs

Table 4.1: Participants' Profile

ID	Role	Training	IT Work Exp	Company Exp
P1	Software Engineer	Enabler	10	4
P2	Product Manager	Enabler	19	0,5
P3	Software Engineer	Learner	6	1
P4	Software Engineer	Learner	15	11
P5	Product Designer	Enabler	27	10
P6	Software Engineer	Enabler	21	8
P7	Software Engineer	Learner	7	7
P8	Product Manager	Enabler	21	6
P9	Product Designer	Enabler	5	4
P10	Product Manager	Learner	16	7,5
P11	Product Manager	Learner	23	10,5
P12	Software Engineer	Learner	5,5	4
P13	Software Engineer	Enabler	20	11
P14	Software Engineer	Enabler	5	5
P15	BR Transformation Lead	-	12	7

4.4 Data Collection Methods

The study was conducted in two stages in which Team A and B participated and lasted six months. Figure 4.3 illustrates which data was collected in the two stages, which in turn help us explain how data collection methods were applied in this research.

We used multiple data sources. We present them organized by each of the case study stages.

Month 1 to Month 3

- The *Questionnaire* was used to collect the participants' profile (name, role, main responsibilities, time in years working in IT and at ORG, and whether the person participated of the immersion training in the US—labeled Enabler in Table 4.1 or is being trained by the enablers in Brazil—labeled Learner);

- *Semi-Structured Interviews* to gather information on their perceptions about the combined transformation, the training experience, and benefits and challenges. The interviews were also used when we needed to confirm the collected evidence during the observations session. Beyond the teams participants, we also interviewed the transformation leader in Brazil, aiming to gather more information about the transformation strategy. Interviews were voice recorded and transcribed for analysis. They lasted in average 30 min and were conducted in weeks 3 and 4 of the 12 weeks of the teams in the lab;

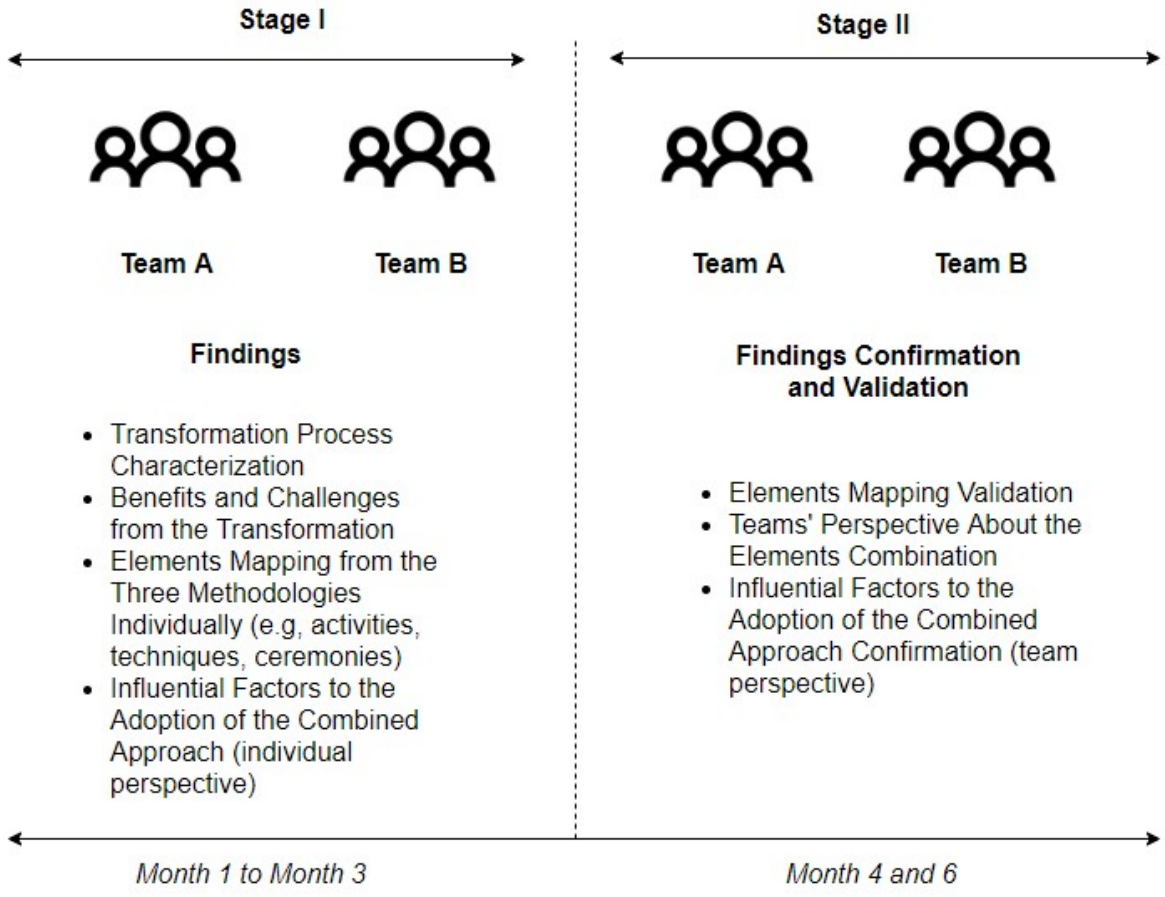


Figure 4.2: Case Study Stages
[Source: Author (2019)]

- *Daily Observations* of team ceremonies (e.g., daily standup, retrospective, iteration planning), meetings with stakeholders (user interviews, demos), and work routine. We also conducted shadowing of roles (e.g., product manager, product designer, and software engineer) seeking in-depth knowledge about the responsibilities of each role;

- *Focus Group* sessions were conducted to collect data from the combined approach elements. We performed four sessions at this stage. Those sessions promote discussion among the elements from each methodology (e.g., activities, techniques, and outcomes from UCD methodology), and also about specific topics that emerged from the previous data collection sources (e.g., to discuss the Product Designer new role). The focus group session participants were the product designer, product manager, and software engineers enablers since they had the training and have more experience with the combined approach usage. Focus group sessions lasted on average 1.5h and were also voice recorded and transcribed.

Figure 4.3 presents a data collection timeline for the study first stage.

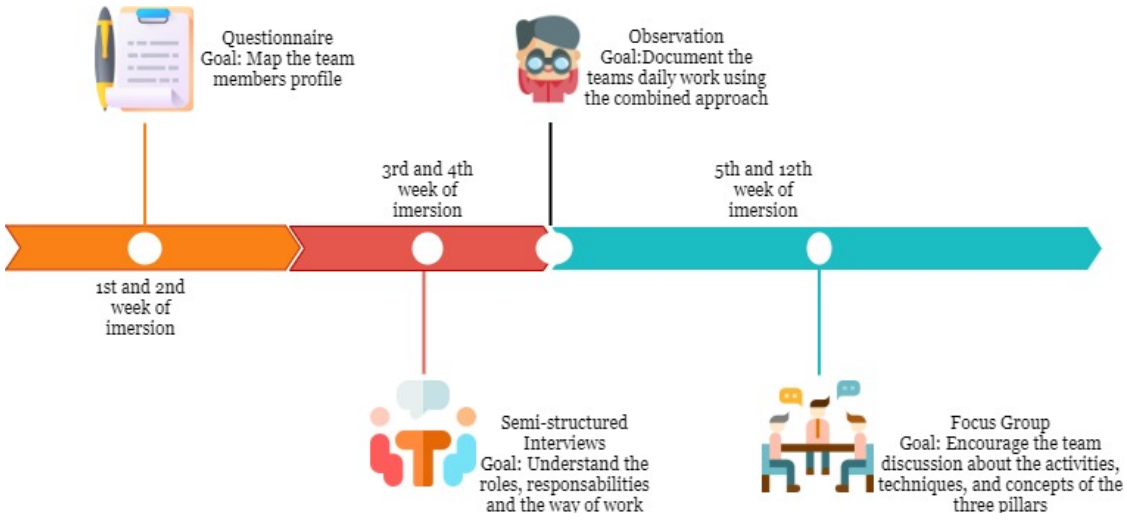


Figure 4.3: Data Collection Timeline - First Stage
[Source: Author (2019)]

Month 4 and 6

In the study second stage, we aimed to confirm and validate the pieces of evidence collected in the first period. It is relevant to emphasize that from month 1 to month 4 occurred the hands-on immersion training, and the data collected correspond to the time that teams are co-located in the university lab. In the current stage, the teams returned to their companies, experienced a different environment; at this moment, we started stage 2.

- *Semi-structured Interviews* was used two confirm data collected from month 1 and 3 about factors that could influence the combined approach adoption. These interviews emerged question-related to team engagement, stakeholder relationships. Once again, the interviews were voice recorded and transcribed for analysis. They lasted on average 30 min.

- We conducted two *Focus Group* sessions at this stage. The first session was conducted aiming to consolidate a group vision about the benefits and challenges of the combined approach adoption. In the last one, we aimed to confirm the elements mapped from each methodology combined into one approach. We ask them to confirm and to illustrate how they visualize agile, UCD, and Lean Startup combined in their daily work. Focus group sessions lasted on average 1.5h and were also voice recorded and transcribed.

4.5 Data Analysis Procedure

Regarding data analysis, we conducted the content analysis procedure by Krippendorff [41], using a qualitative approach to the ethnographic content analysis, where we are

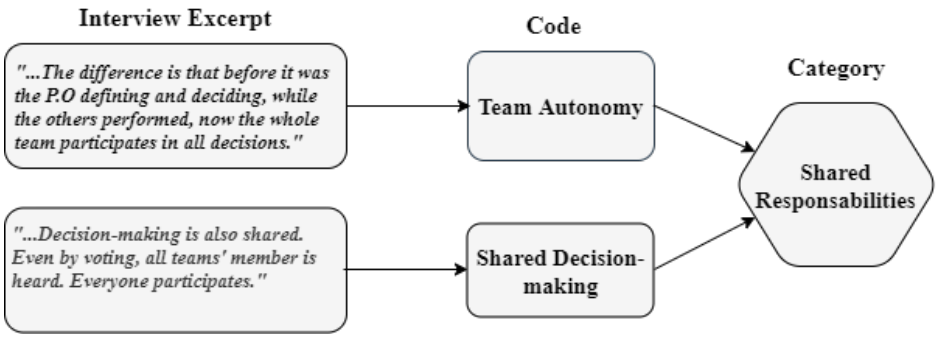


Figure 4.4: Data Analysis Example
[Source: Author (2019)]

focused on the narrative description of the situations, settings, as well as the perspective by the actors involved in the phenomena [41]. Also, as we use recording/coding units, we organized the analysis into the following steps: organization and pre-analysis, reading and categorization, and recording the results. We first read the dataset, extracted text excerpts, and marked them as codes. Figure 4.4 illustrates the analysis process. We use Atlas.TI[2] tool to conduct the analysis. Appendix F contains the network for the interviews analyzed.

These codes were revisited and grouped into more extensive codes, for at the end form categories. To accomplish the empirical validity procedure, we conducted a series of reviews with seniors researchers, aim to mitigate the study limitations. The two senior researchers reviewed the questionnaire and interview scripts - assisting the content of the question, as well as the order and terminology. The iterative analysis process was conducted by two juniors researchers and continuously revised by two senior ones.

The data triangulation was performed by two juniors researchers who analyzed the results of each other to find missing points in the findings. The discrepancies of interpretation or follow-ups identified were discussed with the two senior researchers for deliberation and planning at what time was most appropriate for clarification with the teams.

Following, we present the results from the case study conducted.

4.6 Results

In this section, we present the findings of the case study. The data exposed is a result of the analysis of teams A and B. We organized the findings in order to respond to the C.RQs defined.

[2]atlasti.com

4.6.1 Combined Approach Adoption and Characterization (C.RQ1)

To answer the C.RQ1, we present the transformation process in the first part, which explains how the process occurred, the strategies selected, and so on. In the second part, we present the approach characterization, which corresponds to the adoption in their daily work.

Transformation Process

ORG made two significant decisions on how to drive the combined transformation in its IT department. First, a Digital Transformation Team was created to lead the transformation. Second, this team defined a Transformation Package Toolkit as a means to help the teams to kick-off and work towards the transformation. As a consequence, major changes outcomes involve roles and responsibilities reshaped, and working processes, practices and activities, tools, and artifacts modified (either new or adjusted). These are explained in detail next.

Digital Transformation Team

The CIO office created a dedicated transformation team to lead the IT transformation process. Composed of a Transformation Team Leader Head, a Lead Representative from the USA, Brazil, and India sites, and a senior consulting member from Pivotal, the main responsibility of this team is to define strategies and roll out several actions towards the transformation worldwide. They also liaison with business senior managers and representatives who champion the participation and engagement of business (in-house customers and end-users) personnel. Moreover, the team also leads the discussion of which product teams should first get engaged in the transformation and be prioritized to participate in the 'learn-on-the-job' hands-on immersion training in the headquarters. This transformation team defined a Transformation Package Toolkit, which has been slowly being adopted by the prioritized teams worldwide, as discussed next.

Transformation Package

As a means to help the teams to understand the fundamentals of UCD and Lean Startup as well as to brush up Agile Development knowledge, and provide them with a skill set to work towards the combined transformation, a set of resources is made available, namely: workshops, cookbooks, learn-on-the-job hands-on immersion training, and health-check assessment tools. Figure 4.5 illustrates these resources and promoted changes.

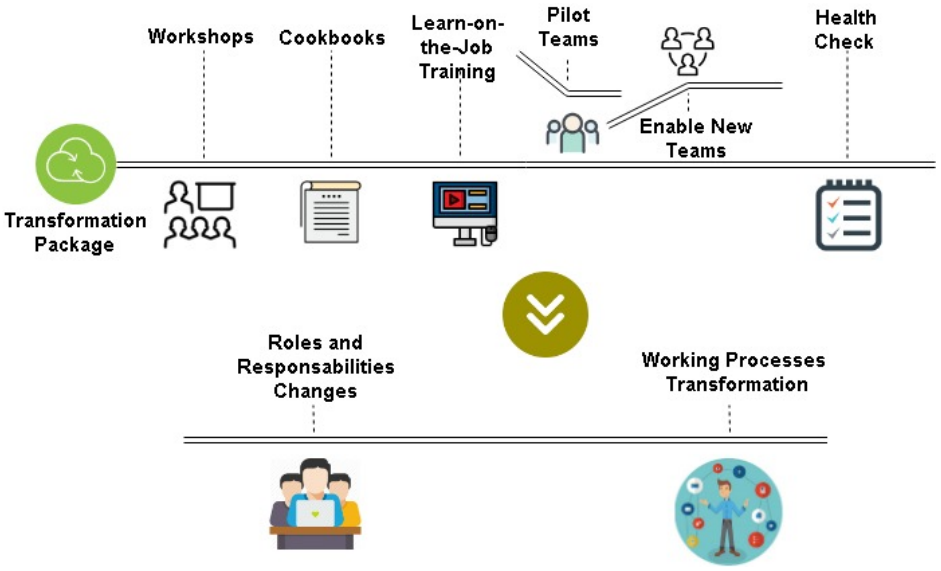


Figure 4.5: Transformation Package and Outcomes
[Source: Author (2019)]

Workshops: Targeting middle-management but also welcoming development team members, the workshops aim to provide basics on the individual approaches, their combined use, and offer a forum for discussion of expected changes. As a consequence of informed knowledge and debate, the Digital Transformation Team seeks for the engagement of middle-management with this activity, which also tackles diverse additional topics (e.g., how-to engage users, assess progress, prioritize user needs; roles and responsibilities changes, etc.).

Cookbooks: Produced by ORG, the *cookbooks* are quick guides to techniques (e.g., persona) and practices (e.g., behavior-driven development) targeting mainly development members who are yet not engaged in the transformation. This strategy is an attempt to keep people up-to-date with new terms and activities. The Digital Transformation Teams expects that, by making these cookbooks available, development members will reduce resistance to changes and keep others motivated and eager to learn more when the time comes.

Learn-on-the-Job Hands-on Immersion Training: Led by Pivotal Labs consulting personnel who are software developers, architects, team leads, etc. themselves, this activity is at the core of the Transformation Package Toolkit and is meant to promote mindset and cultural change, and shape new skills. For 12 weeks, selected team members travel to the USA headquarter office to work side-by-side with Pivotal Labs representatives in a certain ORG software product backlog of their own. By acting as team members and indeed contributing to the product development, Pivotal Labs representatives coach and mentor ORG team members, forming “Pilot Teams”. A Software Engineer highlighted:

> *“It was not like a traditional training. The Pivotal guys executed a certain activity, and we just followed them, learning by observing and doing ourselves. when we*

were done, they would give us links and videos for us to study the topic further." (P14)

Another member concluded:

“These guys were teaching us to learn how to learn and change our mindsets. They were always asking us to say why we thought they had done a certain thing and discussing our responses with us. This helped shape our new mindset. This was indeed a driven culture change for us." (P9)

Daily work involves activities from understanding the user needs to decide on the best solution, as well as hands-on software development and deployment. A software engineer mentioned the trainers’ lightness to introduce new concepts:

“... the trainers’ easiness to introduce new techniques aiming to find a better solution, got us to start thinking about this process in a different way. They brought every day a new technique, and we evaluated whether the technique was good to the given context or not." (P14)

During the training period, role and responsibilities are also revisited, including the definition of new roles—such as the Product Designer (described next), as part of teams’ evolving maturing process.

Pilot Teams Acting as Enablers: To scale up the learn-on-the-job immersion strategy, the Digital Transformation Team defined a strategy of snowballing training: those directly trained by Pivotal Labs personnel—the Enablers—are now acting as coaches and mentors of new learning teams–the Learners. Upon their return, the Enablers were straight-up allocated to work with new Learners from their product group, forming new working teams. The Brazilian Transformation Lead mentioned:

“We selected the best of the best to participate in the hands-on immersion. These highly skilled guys will be the seeds of the transformation [in the financial area] in Brazil." (P15)

A Product Manager recalled:

“We were quite concerned at first. We realize our responsibility and how risky the snowballing effect might be, but we are seeing it working. Users are happy with results, and senior managers are positive; this will work in the long-run." (P2)

A software engineer explains the mindset required from an enabler in this type of immersion:

> *“As enablers are our responsibility to make the learners change their mindset and behaviors, more than just teach how to use techniques or practices. For example, encourage the feedback between the team members is a relevant achievement for us, as enablers."* (P13)

Health-Check Assessment: Kicked-off using an artifact provided by Pivotal Labs, the teams self-assessed their progress and evolution with the help of Pivotal Labs personnel, including practices, product quality, team ownership, and user involvement. The results of this self-assessment are consolidated amongst all worldwide teams from a certain business area (e.g., all product teams from the financial area) and used by the Digital Transformation Team as input to reconsider the transformation goals and toolkit strategies and resources.

Roles and Responsibilities Changes

Looking to address all the three pillars through the roles, which now will be important for working in this perspective, is necessary as a Product Designer (User-Centered Design) role, which the teams define as a team facilitator.

> *“The product designer helps the team conducting user interviews to identify the problems, suggest the use of techniques to map the user flow, to create team empathy with the user, and helps the team to address the pain points of the user in the product. One work closely to the product manager"* (P5)

The Product Manager (Lean Startup) role is the business vision inside the team, as one of the interviewees defined.

> *“This role helps the team to address the business need in the product through value map, help to create and to validate the assumptions, manage with the team the product backlog, and also to help during the experiments. One helps the team to focus on the user/business problem and not working with done requirements anymore"* (P2)

Moreover, a Software Engineer (Agile) role now participates in all decisions which involve the product.

> *“The software engineer has the responsibility of guarantee the environment to the solution developing, make the pipeline implementation using continuous delivery and integration. Also, the software engineer participates in each decision in the team since the conception of the product, joining the users’ interviews, stakeholders meeting, and the other ceremonies."* (P1)

The teams also reported that in the hands-on immersion training, the consultants also presented a role named as an anchor. They defined as a specialization of the software engineer role focused on technical, but highly dependent on soft skills, as communication. Team A and B decided not to adopt this role:

> *"During the training, there is one of the consultants that performed the role of anchor. This role is focused on software engineer. However, it focused on having good communication. We decided not to adopt the role yet because we think that is not so required to this moment. We decided to focus on exploring the software engineers in-depth at this moment."* (P14)

Moreover, new techniques were also introduced or revisited, such as working in pairs:

> *"This new setup, where we have a single large table and work in pairs for most of the activities, has increased our productivity despite all the odds."* (P1)

New tools are also in place and are considered keen for supporting the transformation:

> *"Pivotal Tracker*[3] *is indeed central to our work. We do not work based on sprints any longer, so we just open Pivotal Tracker any time we meet with the customer to show them metrics, user stories, code deliver packages, whatever we have to report, or want to discuss. We have a single tool that centralizes all our artifacts."* (P2)

The teams members started to promote meetings named the community of practices (CoP), motivated by the actions proposed by the digital transformation team. These meetings have as subjects the transformation itself and the introduction of the product designer role, as a manner to engage and emphasize the role importance. A product designer explains the initiative:

> *"The CoPs are actions to highlight the role value and show the relevance of the product designer and software development teams. The meetings are for any company member who has the curiosity to understand what we do, and demystify the product designer in the organization."* (P9)

Working Processes Transformation - Combined Approach Characterization

We observed the learn-on-the-job hands-on immersion training in Brazil from the perspective of teams A and B, during the first stage of our research. We gathered aspects

[3]https://www.pivotaltracker.com

related to the combined approach adoption. These aspects were elements as roles, activities, techniques, outcomes produced, and mindset changes. Aiming to confirm and validate the data collected in the first stage, we developed a conceptual model to better illustrate for the teams the elements and its interaction. The model is an instance from a conceptual model that was developed by the research group that the author is part of. We use the model to conduct a focus group session with the teams. The model is available in Appendix H.

The second stage results revealed *methodological aspects*, which explored, the addition of UCD activities, to promote user involvement and participation. Lean Startup concepts as the use of the Build-Measure-Learn (BML) loop as an approach, to create a more certain product, introduce the concepts of experimentation that allows teams to fail and quickly correct, and pivot/persevere strategy—still related to methodological aspects, the change for XP methodology, which aims to provide consistency and quality of the code, reflecting on the product. The findings also revealed mindset changes as the *product is developed under a new perspective*, using a problem-oriented mindset, the teams' adaptation to this new mindset, and roles and responsibilities modifications. Next, we report the changes related to the methodological aspects.

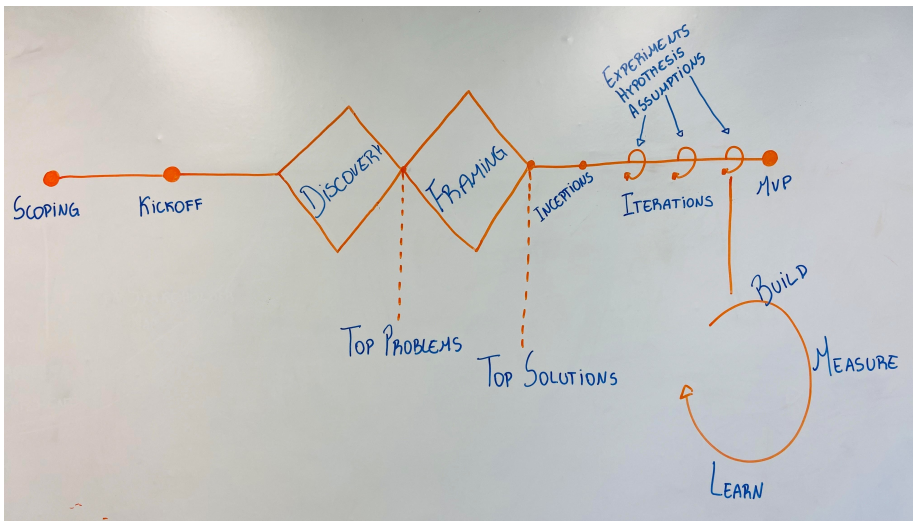
Methodological Aspects

In one of the focus group sessions, we ask the teams to illustrate how they visualized the changes related to methodological aspects in their daily work with the addition and change of these methodologies. Also, they were inspired by the conceptual model instance presented, which contained the activities, techniques, outcomes produced mapped on the first stage. Based on that, the teams illustrate their perspective in Figure 4.6.

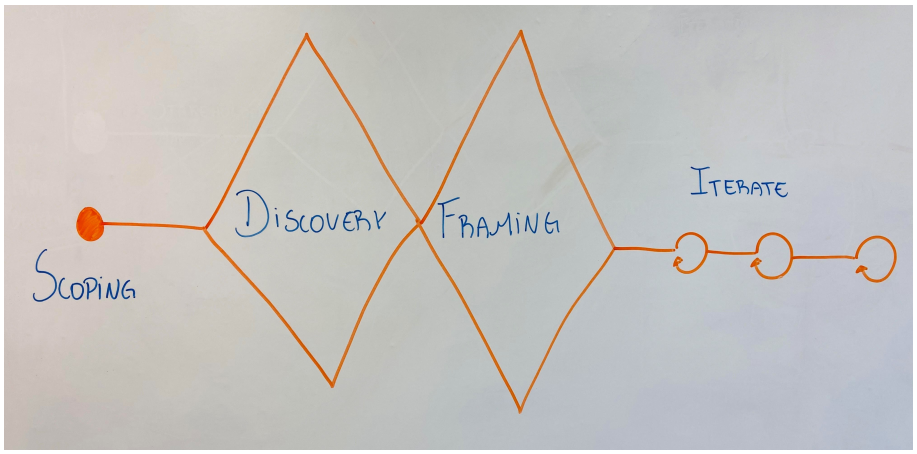
Following the teams' representation, next, we start exploring the aspects related to UCD as a manner to promote user involvement in the product development in the teams' perspective.

UCD to Promote User Involvement and Participation

Although agile methods encourage the relationship between team and stakeholders, the participants felt that in their context, it seems to be not enough because the product still not address the stakeholders' needs. With the change for the combine approach usage, business people and users reported a set of benefits in terms of user participation in the teams' activities and their daily work, and also the user involvement, since with the introduction of UCD activities, techniques and concepts they seem to be more engaged in the product development.



(a) Team A



(b) Team B

Figure 4.6: Team A and B perceptions

The participants considered as a significant modification in the UCD context the adoption of the Discovery and Framing framework designed by Design Council[4]. On the two teams' representation (Figure 4.6), the framework is in the early stages, aiming to explore the problem and possible solutions to it.

Team members emphasize that the use of discovery and framing framework, it is a consequence of work in a problem-oriented, reaffirming once again the mindset change impacts. Another relevant factor for participants is the need to have the whole team working in the framework stages, as problem exploration, user interviews, user research, and other activities. Team A members consider that team engagement to participate in these activities promotes a better relationship with the stakeholders, making them believe in teams' effectiveness:

[4]https://designcouncil.org.uk/news-opinion/what-framework-innovation-design-councils-evolved-double-diamond

> *“We gain their (stakeholders) trust when we talk with them and show interest in providing a product that attends and solves their problems."* (P3)

Team B participants declare:

> *“Using UCD techniques and also a mindset of being more empathetic with our stakeholders, make them feel indispensable in the development process, and consequently, encouraged to talk and to contribute with us. Our stakeholders see us as problem solvers. We gain their trust when we show interest in providing a product that attends the problems."* (P9)

The problem understanding is an outcome of the problem exploration. This outcome allows possible framing solutions to the given problem. The teams reported that the stakeholders' presence is even more necessary at this stage. Team A members affirm that to promotes the stakeholders' engagement is vital to collect stakeholders' feedback all the time and consider it:

> *“We use stakeholder feedback as a tool to refine and redefine problem definition and priority. Being aligned with the stakeholders' needs makes them more confident about our work. We work together with stakeholders, ensure that the developed product is being following the right path."* (P5)

In team B experience, another concept that helps to focus on the problem understanding and provide a more accurate solution to the product is experimentation concept inside the problem discovering:

> *“We aggregate value to our products by using experimentation. We explore the problem that business brings to us, and by the end of it, we address their needs in the product."* (P12)

Also, addressing the users and business needs in the product, demand a change of mindset to guide the teams. They mentioned that a critical value taken for all three methodologies is the BML loop and experiments, which lead us to explore the teams' perspective on the Lean Startup concepts addition.

Lean Startup concepts as a tool to be more assertive

One of the most powerful concepts derived from Lean Startup in the teams' point of view is the BML loop inclusion. The participants have defined BML usage as an approach, and the reason for that is that the loop is applied all the time:

> *“We use BML all the time in any part of our process. For example, a user interview. If we are defining the interview script, we are building the script. We measure the script value by observing after the interview, if we collected the right data or not (e.g., the stakeholders answer the question, but we do not formulate the question for the answer that we aimed.) - and this process allows us to learn from our fails to create a more assertive script to be more accurate in the next one. BML applies to any product development activity."* (P13)

However, BML usage is not so relevant if used alone. The richness of the loop is combined with experimentation, as teams' members reported:

> *“All foundation of the BML brings the experimentation concept in the core of it. We work with a problem-oriented mindset because the experimentation allows it. In the beginning, we have a simple problem view, and this leads us to start making assumptions from that, execute the experiments using prototypes or any technique. The results give us a condition to measure it and to refute or accept our assumptions. At the end of it, we learn from the results and restart the loop, refining our vision."* (P2 and P6)

Team A also experienced an unusual usage of experimentation in a non-software solution. They mentioned that this shows the relevance of the concept used for the teams, stakeholders, and to the company itself:

> *“Our users were claiming a solution to the performance issue in the system. Before we run directly to the code, imagining that the problem in the software solution, we decide to analyze the problem. The stakeholders reported that the use of some spreadsheets contained a significant amount of data, and it was getting a poor performance taking about three days to calculate and return the results. So, we assume that maybe the problem was not in the application, but in the host machines. We decided to run the same application in a more powerful machine, and we have found out that our assumption was right - the problem was in the machine's performance. This experience shows us the relevance of experimentation - and more than that, it shows that sometimes the problem solution could not be a software solution, which for us is a huge breakthrough."* (P2)

Besides, the participants perspective, experimentation gives them room to fail up; however, fail and fix quickly:

> *“Product development is uncertain and very susceptible to failure. Nevertheless, what matters is the speed at which we will react to those. The experimentation as a core of the BML gives us room to fail but also allows us to fail and fix quickly. We do not need to wait until the end of the iteration to discover that we do not understand the stakeholder needs."* (P13 and P11)

The pivot/persevere usage follows the same idea of one of the agile principles, in terms of adaptability for team B members. They affirmed that pivot/persevere reinforce the relevance of refining the product and problem strategy, being adaptable to change or persevere:

> *“Experiments give us conditions to understand if we are in a smart strategy for our product or not. Also, the stakeholders’ relationship with us is an essential factor to persevere in the strategy or start to look another direction, pivoting. Sometimes, the strategy defined in the long-term can not be valid anymore. That is the reason why BML, experimentation, and pivot/persevere perform better together; one depends on the other."* (P11)

Notwithstanding, the addition of UCD and Lean Startup has been the main change. In terms of code development, the teams reported a need to align the changes in a possible technological manner. To attend this modification, participants reported the use of XP instead of Scrum as an agile method. Now, we explore how the insertion of XP affected the process, from the teams’ perspective.

XP to Boost Code Quality

The XP methodology choice as an agile method cames with the Pivotal Labs approach proposal. However, team A members recognize that even that the change was top-down from the Scrum framework to XP was a great fit. They cited that the use of XP practices (e.g., pair programming, TDD, and unit test) boost the development and increase the code quality:

> *“The use of pair programming increases our product development process. We can benefit from using it in many ways: from accelerating the learning process of a new engineer, to promote improvements in the code quality."* (P7)

Continuous Integration (CI)/Continuous Delivery (CD) pipeline was considered as a practice that promotes a problem-oriented mindset in the context of software development, as team B participants mentioned:

> *“CI/CD pipeline was crucial to address the changes. It promotes faster feedback and helps us to validate stories on the production environment. CI/CD inclusion encourages software engineers to feel more proficient."* (P4)

The participants also reported significant modifications in terms of team rhythm. They have changed a set of ceremonies during the daily and the iteration work and also its nomenclature aiming to attend to XP methodology rhythm:

> *“We tried to be more aligned, and the ceremonies are useful for that. We continued doing the standup meeting, retrospective, and planning. However, we now have an office standup to be more connected to other teams — also, the ceremony nomenclature change from sprint to iteration. In the planning sessions, we choose if we must have more than one session, for example, a pre-iteration meeting. Finally, we have weekly sessions with all stakeholders to strengthen our relationship with them further."* (P13 and P4)

Once again, BML shows its relevance, as well as experiment concepts in teams' perspectives. The teams reported that the use of these concepts impact the manner they deal with the iteration directly. It is a common-sense between them, the relevance of developing the product thinking more systematically and investigating the real problem, defining assumptions, executing the experiments, collecting data, and verifying whether the assumptions were accepted or refuted.

Concluding the teams report, the participants attribute the adoption success in terms of mentality, engagement, and modifications related to methodological aspects with UCD and Lean Startup, to an organically approach application:

> *“Even though our drawing represents a sequential or continued vision of the methodologies combination, our daily use is adapted. If we are during the iterations and perceived that the problem is not well defined, we are ok to come back to the discovery and framing framework and start again. Alternatively, if we defined some assumptions and discovered that the product/problem vision is not aligned, we can redefine these assumptions. This is the adoption secret, apply the approach organically."* (P2)

Now, we explore the changes related to mindset and the impacts of the roles and responsibilities.

Product Developing Under a New Perspective

The approach adoption has a crucial motivator for change: a problem-oriented mindset. Teams' members mentioned that one of the most relevant modifications occurred with the adoption was a change for work in a problem-oriented perspective than refining software requirements only:

> *“Before, we usually received a set of predefined requirements. We implemented these requirements and considered done. We did not know whether the problem was solved or not. Now, we do participate and have the opportunity to investigate and understand the problem."* (P1)

The participants also considered that the change in mentality was a challenge at first, as this modification directly affects the team's attitude. The mindset change required that teams start acting as main actors in the development of the product and not just as those who operationalize it. However, it is crucial for a new attitude from the teams to fit into this new mentality.

Team Engagement

The teams' commitment to the entire software development process has increased considerably since the adoption of the problem-oriented mindset. In fact, during the approach training, when the teams start to recognize the need to move to an improved way to provide more business-aligned products, changing at the core the manner of understanding the product was considered as one of the most relevant achievements. This forces them realized that many aspects are new and are directly dependent on the involvement of teams. One example is the relevance of having a shared product vision:

> *"Everyone needs to understand the product, not just the product designer or the product manager - the software engineer is no longer isolated. The entire team needs to know why the products are working and having an understanding of the product vision. Everyone is always up-to-date."* (P7)

Shared Responsibilities

With a shared product vision is essential that teams have shared responsibilities. The whole team participates from activities as the problem understanding - where is discussed the product's needs. By establishing a relationship between them and the stakeholders, the team can define a stakeholder map - which allows the teams to be more effective in the next phases of the product development, as well. This change requires a different position from the software engineers since the product designer and product manager already have this participative role with the stakeholders due to the nature of the roles. Now, the software engineers affirm that they need to adapt to a more collaborative attitude in all decisions that involve the product:

> *"We have the responsibility to guarantee the environment to the solution developing, make the pipeline implementation using continuous delivery and integration. However, we are now responsible for participating in each decision in the team since the conception of the product, joining the users' interviews, stakeholders meeting, and the other ceremonies."* (P1 and P13)

In the teams' perspective, in terms of methodological aspects, the combined approach adoption depends strongly on the first two elements discussed above. Establish

that; we can describe the aspects related to the teams' way of working on the adoption of UCD and Lean Startup concepts, also the change from the Scrum framework to the Extreme Programming (XP) methodology.

Next, we answered the C.RQ2 presenting the combined approach benefits.

4.6.2 Combined Approach Benefits (C.RQ2)

Team A and B reported a set of benefits related to the *cross-functional team* structure, improvements on the *stakeholder relationship*, how ucd and Lean Startup addition are *boosting agile*, the change to scrum for XP impacts in *technical aspects*, and how the combined approach adoption benefits *team autonomy*.

Cross-functional Teams

With the redesigned responsibilities of the Product Manager and the introduction of the Product Designer role, the newly defined cross-functional team co-shares responsibilities towards the product under development:

> *"Having both roles working closely together is key. While Product Managers focus on the business, Product Designers focus on the user; we bring the engineering perspective."* (P4)

The co-shares responsibilities lead to shared knowledge and vision about the problem and the product that the team is working on:

> *"Before we were only present at the [product] scope definition meetings. In the end, the requirements came 'chewed' our way. It was like 'do it.' Now, we do participate in the understanding of the user needs, and, as a consequence, we all are well aware of what has to be done. Everyone is always up-to-date."* (P3)

A Product Manager adds that they are the co-owners of the solution. This ownership also fosters trust among the team members, since they felt very proud because the customer does want to hear the teams' opinion. All these changes can contribute to stakeholder vision about the team:

> *"We are very excited with the velocity and progress that the team achieves. You guys can be a success case under the combined adoption to other teams and business people."* (Stakeholder - team A)

Boosting Agile

Focusing on the problem understanding is of greater value than refining software requirements only, as mentioned before. The participants mentioned that by discussing the problem, they have a range of different solutions to be considered. As a consequence, the teams can now conduct experimentation:

> *“Whatever we see fit, the Product Designer and I [Product Manager] hypothesize about a possible solution and have the freedom to test it. We also use experimentation to validate the answers to our questions and make sure we got the right problem. In the end, we need to be sure that we understand the problem correctly and not only do what we are told." (P8)*

Nevertheless, another benefit of using experimentation is having room to fail upfront from development:

> *“We used to work based on sprints and release plans; there was no room whatsoever to experiment and fail. With our new continuous development and release approach, we can explore, test, and pivot candidate solutions. Time slot gives room for value-driven development."* (P1)

Including ucd and Lean Startup also impact the relationship with the stakeholders. The participant affirms that having a frequent contact between the team and stakeholders, collaborates with the relationship of trust:

> *"We gain their trust when we talk with the user and understand their needs."* (P5)

As a consequence, the participants identify the importance of the have team empathy with the users

> *"We must be empathetic with the users, make with they feel important in our processes of developing a product. They see that we are working to solve their problems."* (P2)

For the participants, frequent communication is a factor that contributes to the stakeholders and team communication.

> *“Meetings are an important tool, promoting the stakeholder and team communication, but must be used only when necessary. Decision-making can not be only in meetings; we communicate with the stakeholders as soon as a decision must be made."* (P2)

Having face-to-face meetings with the stakeholders was mentioned as a factor as well:

> *“ The team and stakeholders are benefited with the use of face-to-face meetings, creating intimacy and generate a better communication."* (P13)

The participants mentioned that have the stakeholders involved since the product conception due to the team empathy with them, promotes communication among the team and stakeholders.

Deliveries with added value in production is of greater value to the stakeholders understand the effort and the concern of the team with their needs:

> *“They (stakeholders) observe our effort to do deliveries with added value. They are informed about everything. This collaborates with the relationship of trust."* (P11)

This confirms that the participants felt about the need for mutual transparency between stakeholders and team:

> *“We just need to build this relationship, showing to the stakeholders what we are doing and the results.* (P2)
>
> *“The team must look for the users’ feedback constantly. The teams and the stakeholders are more confident with each other since both sides are aligned with what they need and what the team is producing."* (P9)

Team pro-activity was also mentioned as a factor that contributes to the stakeholders’ communication:

> *“We must understand the problems that the user has on the application and not only wait for them to see what we have to do. "* (P12)

Technical Aspects

The participants also considered that the combined transformation helped them to improve technical-related aspects. For instance, they believe that by having a continuous delivery, as opposed to work based on sprints, was an interesting change:

> *“We finished, reviewed the code with the customer and users, and deployed it. We did not have to wait for a release date or milestone. This continuous approach also promotes transparency to stakeholders, who are constantly seeing progress, understood as the return of investment."* (P13)

Improved code quality was also mentioned as a side effect of adopting practices from Extreme Programming such as Test-Driven Development, Pair Programming, Continuous Integration, and Code Refactoring:

> *“We always watch out for good code quality, but this is different from Scrum. XP offers a new mindset on software development."* (P4)

The teams identified as a benefit, the fact of the software engineers have a vision about the application architecture and can contribute to the architectural decisions:

> *“ I have the feeling that the whole team ends up knowing the architectural design that sometimes this information stays in silos, usually in my old model it was in silos, today if you reach for anyone there on the team, they will know."* (P13)

The team’s members explain how the concepts of continuous integration and continuous delivery, reflect on quickly deliveries:

> *“ In 3 days, we went there and already did deploy in production. So, we do not discuss a release; we do not have an expense in discussing release[...] This is decided at the time that needs to be decided." (P13)*

Team autonomy

For the participants having the middle managers trust is essential to the team autonomy:

> *“We must build the relationship of trust with middle managers because we need to have they in our side"* (P9)

As a consequence, the trust helps the team gain the middle management support since they understand and recognize the teams’ effort.

Made small deliveries in production was considered a factor since this adding value to the product:

> *“There is a considerable effort on the process of delivery code in production. However, this is important to the team autonomy since the deliveries are only in production, and this is the way that we show the added value to the product. Delivery small allows us not to break the deploy; for this reason, we need this freedom."* (P5)

This implies to the factor of having the autonomy to made teams decision about their scope:

Cross-Functional Teams	
Promotes	Co-shares Responsabilities Shared Knowledge and Vision Trust Among the Team Members
Boosting Agile	
Promotes	Focusing on the Problem Understanding Experimentation Having Room to Fail Frequent Contact Team Empathy with the Users Frequent Communication Face-to-Face Meetings with the Stakeholders Stakeholders Involved Since the Product Conception Deliveries with Added Value Mutual Transparency Team Pro-Activity
Technical Aspects	
Promotes	Continuous Delivery Improved Code Quality Vision about Application Architecture Quickly Deliveries
Team Autonomy	
Promotes	Middle Managers Trust Middle Management Support Small Deliveries in Production Teams Decision about Their Scope

Figure 4.7: Benefits Overview
[Source: Author (2019)]

> *"We must have this free pass to make our own decisions. Decide the solution, what is the best for the user - of course, with the agreement. The point is that the team is the owner of the product, and this decision is ours."* (P5)

Figure 4.7 presents an overview of the benefits of adopting the combined use of agile, ucd, and Lean Startup.

Next, we report on the challenges of adopting the combined approach.

4.6.3 Combined Approach Challenges (C.RQ3)

The challenges faced by ORG teams A and B are majority related to *mindset and cultural changes* and *organizational issues*, which are not directly related to the combined approach usage. However, the combined approach *higher dependence of soft skills* and the fact that the teams must *work co-located* were pointed out as challenges of the adoption itself.

Mindset and Cultural Changes

Middle-management is used to be the focal point for negotiations with in-house customers. ORG is working towards a more flat organization. Participants believe that taking away middle-management power might be, at least in the long-run, of a great need for discussion:

> *"We are now going straight to the customers and end-users. The middle-management, at some point, will realize that their main job is relocated."* (P1, P12)

Some are concerned with it will likely be to changing working habits [to the new combined approach]:

> *"We have colleagues that are here for over 20 years. It was challenging enough to introduce agile to these guys. Things are way more dynamic and less structured (in the good sense, I mean) than before. We have to go slow."* (P14)

Introducing the role of Product Designer at large is one of the main challenges ahead, considers a Product Designer:

> *"ORG does not have a company-wide job position for Product Designers. Now that we are working with this role, I cannot grasp how we survived without it for this long."* (P5)

Organizational Issues

The challenge of institutionalizing the Product Designer role was also seen as an organizational issue:

> *"It will be a long run to show the value of having this role and defining how it will be part of the ORG structure. For starters, we need to convince the CIO office."* (P5)

A Software Engineer brings to attention the coordination need with infrastructure personnel when going company-wide with the transformation:

> *"We got the 'free pass' from our middle-management regarding infrastructure for continuous deployment and delivery. The higher managers gave us the resources we needed. However, usually, we have to go through the release management people to have access granted to put new code in production."* (P9)

Funding for portfolio management is yet another major issue company-wide:

Mindset and Cultural Changes	
Promotes	Taking Away Middle-Management Power Changing Working Habits Introducing the Role of Product Designer at Large
Organizational Issues	
Promotes	Coordination Need with Infrastructure Personnel Funding for Portifolio Management Lack of Deliveries Dates
Higher Dependence of Soft Skills	
Work Co-Located	
Promotes	Home Office Freedom

Figure 4.8: Challenges Overview
[Source: Author (2019)]

> *"Nowadays, we receive a certain amount of money to fund a business area. People are allocated to projects to fit this annual budget. We need to think in terms of the capacity of delivery from now on."* (P2, P15)

In contrast to what is perceived by others, a Software Engineer considers that the lack of delivery dates might be an issue:

> *"Our stakeholders keep asking about the delivery dates. We are trying to explain that now we work based on solving issues, there is no target timeline. We are problem-solution and value-driven now."* (P6)

Higher dependence of soft skills

The participants mentioned that using the combined approach requires soft skills more than technical skills sometimes. This seems to concern the teams:

> *"I think the big challenge is related to behavioral issues. For example, there is always that person who thinks he knows more than everyone else, and that can create conflict within the team. "* (P13)

Work co-located

Another concern is related to the change on the home office freedom:

> *"There are many people who understand that they have to go to work every day at the same place, that having to go to a place to work is a setback because many people nowadays work remotely."*(P4)

Figure 4.8 presents an overview of the challenges faced with the adoption of the combined approach.

4.7 Threat to Validity

Yin [78] affirms the relevance of defined four tests (Construct, Internal and External Validity, and Reliability) to judge the quality of the case study. We observed two teams in a real setting, which offers them a new setup that aims to promote collaboration. Also, the teams are composed of members playing distinct roles and with different backgrounds and experiences. Moreover, we used interchangeably and overtime multiple data sources aiming to triangulate our findings, which were reviewed continuously by senior researchers. Therefore, although we cannot claim that our results apply to distinct scenarios, these strategies helped reduce limitations. Next, we present the four tests in our research context.

- Construct Validity: As a premise of construct validity, we defined our subject of change by conducting a first interview with each study participant. One of the questions was regarding how they were used to develop software. This was essential to mitigate construct validity, since we have a parameter to understand what really changes with the combined approach adoption. Ans as expected, we use as case study tactics multiple sources of evidence as described on data collection section. Also, we had multiple researchers conducting observation and focus group session, constructing a chain of evidence. Finally, we have senior researchers reviewing the draft case study reports and validating with two keen study participants.
- Internal Validity: We conducted an exploratory study, and for this reason, this test is inapplicable.
- External Validity: The study was generalized in the same context since we presented a case study. The same RQs were used to observe the two teams and provided similar findings. However, it is essential to highlight that generality was verified only in this context. In regards to generalization, we can not claim that our results apply to distinct scenarios, since the teams' maturity, organizational vision, and their instance of the combined approach are factors that need to be well-considered during a large-scale adoption.
- Reliability: The study steps were properly documented, from the research protocol until the data analysis procedure. All documents related to this research can be observed in the Appendix.

5. DISCUSSION

In this section, we present a discussion of our findings in light of studies reported in the literature. We organized the section presenting the transformation process and how the strategies were made to adopt the combined approach - the strategy, the initiatives, and how the roles, responsibilities, and the teams' daily work were modified to support the change. We also discuss the benefits and challenges. The systematic mapping findings and the case study are compared to understand the research contribution clearly. Finally, in the light of our findings, we present a list of recommendation of practices for organizations and practitioners that aims to conduct a transformation in the same context.

5.1 Combined Approach Adoption and Characterization (RQ1)

Transformation Process

Regarding the transformation process, we found that two significant decisions were made: to put together a dedicated team to lead the transformation and to define a toolkit as a means to guide teams piloting and kicking-off the process. The dedicated team is composed of members from each development site to ensure representation and consideration of local sites' needs. The team decided for a gradual transformation approach, introducing the customized Pivotal Labs approach to a few business areas at a time. Julian et al. [38] mentioned that this type of transformation approach often started introducing practices as iterations and ceremonies. In the team's training, the first concepts introduced were ceremonies as well. These small changes in their daily routine contribute to the mindset change since other skills are explored, as communication improvement (e.g., standup meeting) [38, 3].

Aiming to keep people engaged despite their role in the transformation, the transformation team had the concern of defining strategies as *workshops* to bring middle-management up to speed and gain their support, *cookbooks* to qualify development members still not working with the new approach, and *learn-on-the-job hands-on immersion* to promote culture change and new skills development to those directly appointed to kick off the transformation. Gandomani and Nafchi [23] reported the relevance of defining initial training and workshops - the initial training helps the teams in the transition, and the workshops are relevant to improve communication with people that are not engaged directly in the transformation (e.g., stakeholders). On the other hand, Gandomanin et al. [24] emphasize the relevance of complete package training, including customers, managers, and so on - to avoid a dysfunctional transition.

Regarding the hands-on immersion strategy, the findings show that the company considered it valuable as opposed to traditional training, allowing for a more natural mindset

and culture change. Dikert et al. [17] reported that training and coaching as a success factor during an agile transformation since training improves the chances of conduct a succeeding transformation. The case study highlights a different manner of conduct training since the teams were learned by doing - studies as reported a similar strategy, defining coaches to guide the teams during the adoption. In our case, the coaches were the enablers. This is relevant since agile avoid prescribing and instead emphasize a mindset change [17]. They also defined a strategy to foster the transformation scalability given a large number of employees and the high cost of moving them to the USA site for a 12 weeks long immersion—pilot teams as enablers of newcomers. Dikert et al. [17] points out the pilot team definition as good practice during a transformation process. The study mentioned two main reasons for use pilot teams: to gain acceptance and gather insights from the pilot. This immersion training, combined with the use of health-check assessments, support continuous improvement, considered vital for sustaining the introduction of new working processes and practices [36].

The initiatives proposed by the dedicated team, encourage team members to start a community of practices aiming to emphasize the relevance of the roles, the mindset, and so on. Smite et al. [66] report that many communities of practices are organic and emerge from a group of practitioners concerned with improving or introduce a practice, in our case, a mindset or culture. These communities are also relevant for facilitating inter-team coordination [54].

Related to the roles and responsibilities changes, the teams emphasized the relevance of having a product designer in the team structure. Although that focus on the users' needs is an obligation of the entire team. The product designer is specialized in that specific topic and, with the inclusion of techniques and practices, turns in a team facilitator. The nomenclature changing and responsibilities of the product manager also bring significant impacts. The role is considered as the business view inside the team - and consequently, the improvements in the relationship between business and IT is already perceived. Related to the software engineer role also have suffered modifications about involvement in the early phases of the software development cycle (e.g., user interviews and problem identification). This change emphasizes that the role is not limited to technical skills when the focus is on developing products problem-oriented. Concerning the anchor position defined by pivotal labs methodology, the teams decided not to adopt, since the role does not make sense in their context. Jovanovic et al. [37] mentioned that each agile method defines its roles, ceremonies, and artifacts. However, how will be the roles nomenclature is often defined and customized by the organization. Yilmaz et al. [77] affirm that the organizational roles should be tailored to an individual software project, depending on the development activities.

Working Processes Transformation - Combined Approach Characterization

Reinforcing the development-oriented by user/business problem perspective, there is an extensive effort on the discovery of the right problem and framing the possible solutions to the right solution. The double diamond structure that the teams applied follows the UCD activities defined at ISO 9241-210 [35]. Schön et al. [62] also defines that this is one of the critical aspects under the integration of UCD and Agile, separate product discovery and product solution. Define the discovery and framing usage brings benefits associated with the added value of the product. Alahyari et al. [1] mentioned that one of the factors that can impact the perceived value on the products is the customer relationship, which is highly explored during the discovery and framing since the UCD activities and techniques usage promotes an approximation between team and stakeholders.

Incorporated to the discovery and framing and also in the iteration, the teams make use of the build-measure-learn loop, aiming to produce a better product. The perceived benefits and the reason for the teams choose to use build-measure-learn derived by experimentation was very similar to those reported by Yaman et al. [76], which reduce the development effort, deeper customer insights, and use experimentation as a guide on development decisions. The teams also reported that the use of a build-measure-learn application was a considerable modification since they work only with agile methodologies before, and they feel that agile does not help them to know what product should be developed. Edison, Wang, and Abrahamsson [22] affirm the same, agile prescribes how to develop, but it is not so accurate to answer and to investigate the products' needs.

Another finding on the combined approach adoption is the use of the pivot and persevere concept original from lean startup [58]. Pivot decision could occur at any moment (e.g., problem/solution definition, scope definition), as well as remain in the same strategy, persevering. This is relevant because inputs to the teams and does not allow the teams to work on products that will not add value to the customers and business people, reducing the waste of the process [58].

The change impacts, related to the insertion of XP practices, were lower since the teams were already familiar with agile methods. However, the change for an XP over scrum framework affects their way of work. The inclusion of the build-measure-learn loop and also the XP practices as pair programming, TDD, and continuous delivery bring perceived benefits to the teams and stakeholders [11].

Schön et al. [62] mentioned the barriers of access the stakeholders as a challenge in their study. In the reported study, mitigate this barrier was considered as one of the crucial changes that derive the way that the company works now; it is working as a problem-oriented perspective. The teams changed their mindset to map the user and business problems over only refine pre-defined requirements - solving the difficulty of decrease the creativity to the process of solution-finding.

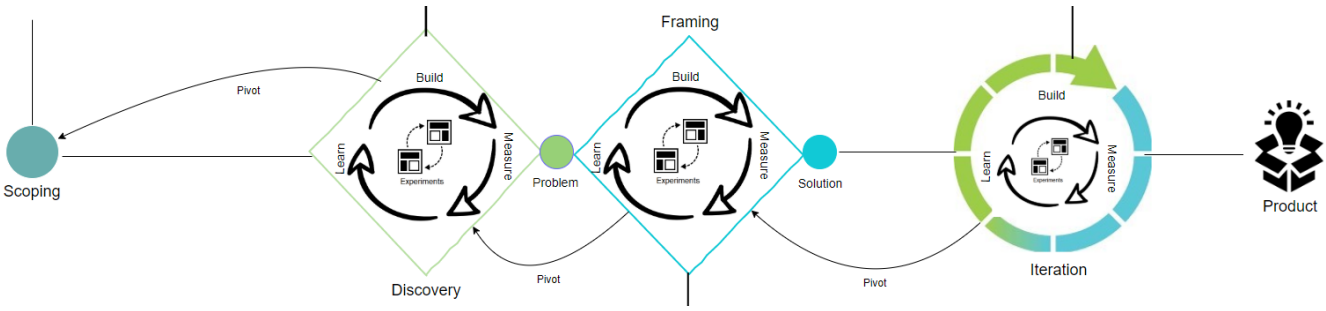


Figure 5.1: Combined Approach Life Cycle
[Source: Author (2019]

Teams' attitude required an adaption to attend the problem-oriented mindset change. All roles became more engaged in activities as product/problem scoping, user interviews, or stakeholder meetings. Nyfjord and Kajko-Mattsson [51] mentioned in their study that the entire team engagement in these activities often was executed by business people and the teams (especially software engineers) only receive the artifact produced from these activities. Once again, these problems are decreased by changing for the problem-oriented mindset.

As reported, the manner of how the combined approach is adopted is essential. It is possible to notice that even that concepts from UCD and Lean Startup are essential in their new way of work, the core of the approach remains in agile value, which is a response to change over following a plan [5], which means use the approach adaptively. Pivot/persevere concepts explore in the core of it, the change of the team rhythm adopting XP ceremonies, which was claimed to promote the engagement and involvement among the team members and stakeholders. From a team's perspective, these modifications ensure adoption success. We decide to translate the development life cycle described by the teams into Figure 5.1, which represents a consolidated view from the the combined approach life cycle by team A and B perspective.

In the early phases (scoping) of the process of problem identification, the starting point, some concepts from UCD, and Lean Startup are already explored. In the discovery and framing, there is more UCD effort, since they explore the UCD activities through UCD techniques, and the whole cycle is derived by the build-measure-learn loop as an approach as the team defined. The iteration covers the implementation of the product that could be software, or a process, or just a recommendation to the users. Finally, all of this work produce the product expected.

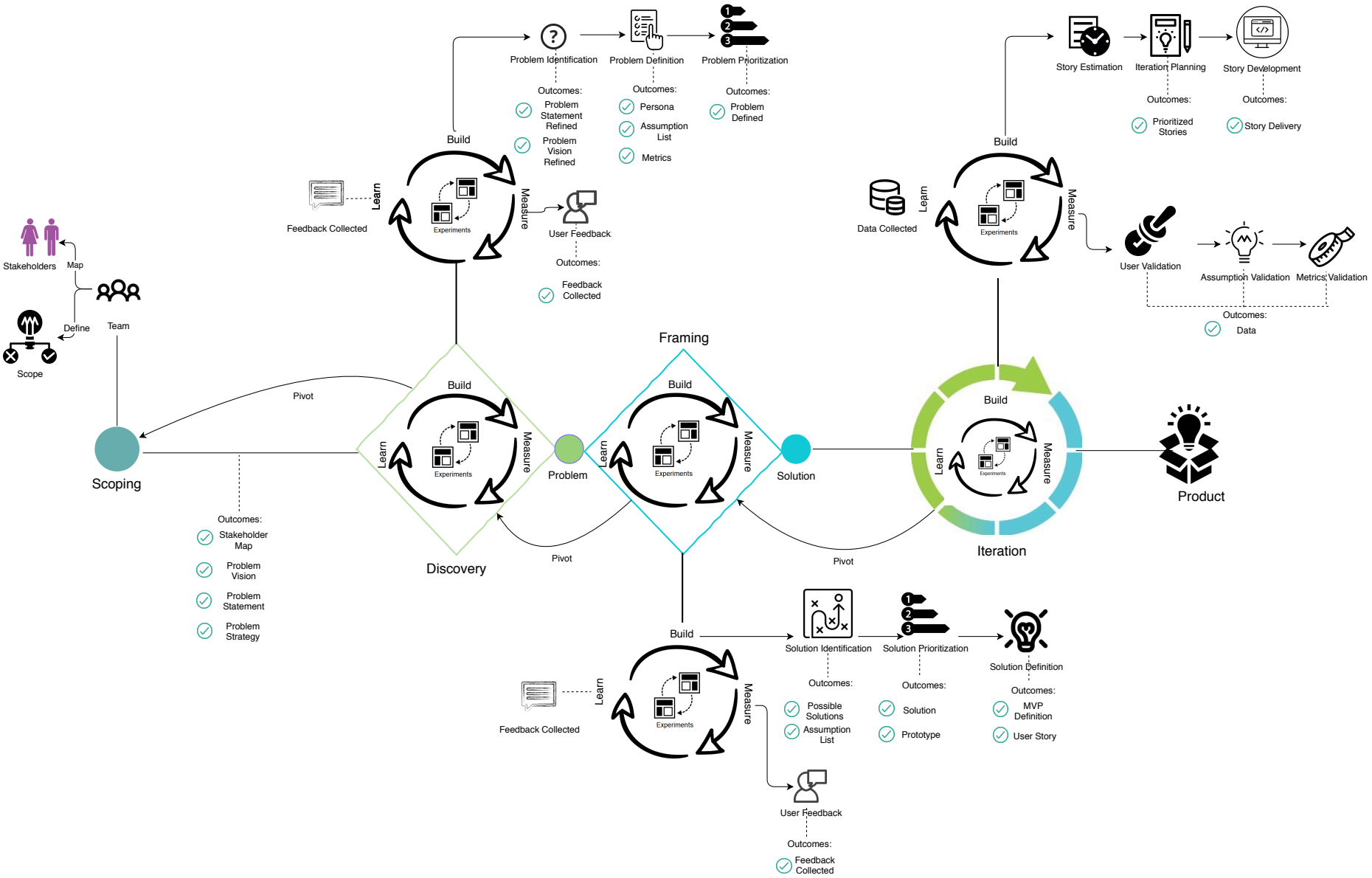


Figure 5.2: Combined Approach Detailed
[Source: Author (2019)]

Benefits		
Case Study Findings	**Introduced by**	**Literature findings**
Cross-functional team	Agile, UCD and Lean Startup	[25]
Shared Vision and Knowledge	Agile, Lean Startup	[42]
Problem solving-driven	UCD, Lean Startup	[37]
Experimentation in software development	Lean Startup	[22], [58]
Stakeholder relationship	UCD, Lean Startup	[4]
XP practices	Agile	[33], [10], [63]
Team autonomy	Agile	[5], [17]

Table 5.1: Benefits: Literature Findings x Case Study Findings

We represent these phases, activities and outcomes through Figure . The sketch illustrates a detailed overview about the combined approach, including the major activities of each phase as explained before and the main outcomes.

5.2 Combined Approach Benefits (RQ2)

RQ2 asked about the perceived benefits of the combined transformation. Participants found relevant to introduce the Product Designer role to compose the cross-functional team, and that the reconsideration of responsibilities between this new role and the Product Manager the team now perceives shared responsibilities with the customer and users and has a shared vision and knowledge [42] about the problem and product. These are considered essential for developing a problem solving-driven mindset [37]. Trust among team members is one benefit that is generated by the aspects before mentioned. The teams felt trust in their colleagues, given the fact that every team member is aligned about vision and responsibilities. Mchugh, Conboy, and Lang [47] reported that trust among team colleagues is an essential factor in agile teams, and highlights that communication is one of the factors that contribute to this trust.

Also key was the introduction of experimentation [58], allowing the teams to explore the problem and to pivot candidate solutions while engaging the customers and end-users, resulting in added value. Gutbrod, Münch, and Tichy [30] report that experimentation enables a better understanding of customers' needs, priorities, behaviors, and continuously better prioritization of development activities. The teams corroborate on this by reporting that using experimentation in the software development process results in products that are more aligned with stakeholders' needs, increasing their perceived added value—this due to experimentation allowing for more room for failure, enabling the search for a better solution.

The improvements in the relationship between the teams and the stakeholders were mentioned as benefits by the participants. The fact that the stakeholders collaborate with the team regularly and participate during the meetings is considered essential during

an adoption process [53], especially in the combined approach, since this iteration between team and stakeholders is crucial to the adoption success.

Associated to the technical aspects, the addition of XP practices on the team brings benefits related to the communication between the team members [33], an improvement on technical issues including practices as test-driven development (TDD) [10], and they also pointed out the gains with the introduction of continuous delivery to project success [29, 63]. All benefits cited are possible by the fact that the teams are autonomous. They decide the way of work and have a free pass to do that by the management level [17]. Tessem [6] discusses in his study that agile teams often felt engaged and empowered when the work practices go beyond programming. Participation in daily, weekly, and monthly meetings with stakeholders and activities of estimation, architecture, and so on, produced this feeling. In our case, the team members are engaged in the entire process cycle, and the autonomy is related to this inclusion in all moments of the software development. This is aligned to the agile principle [5], and this freedom allows them to be committed to the change and remains motivated during the transformation [17].

Table 5.1 illustrates the case study findings compared to the literature findings.

5.3 Combined Approach Challenges (RQ3)

RQ3 aimed to reveal perceived challenges. Interestingly, the cited challenges are mostly related to company-wide related issues, such as defining a fit strategy for IT funding, or organizational-related such as reshaping the middle-management role, or stakeholders misunderstanding the new way of work. Dikert et al. [17] reported that when the transformation is bottom-up, the middle management could be resistant and making the significant organizational change above the team level impossible.

Finding room for the new role of Product Designer as an ORG job position [28] was also pointed out as a challenge. The company used to have only technical and business roles in the team structure, which compromises the position of product designer in future teams that adopt the approach - impacting directly on the transformation process.

Cultural changes such as working habits (e.g., coordination need with infrastructure, lack of delivery dates, and work co-located) [39, 17] are part of the concerns, but they are perceived as handleable at the development teams level. Also, the combined approach adoption has a higher dependency of soft skills since there are some principles as create empathy with the stakeholders, promotes a feedback cycle, and these factors require skilled, self-directed and motivated teams [28].

Table 5.2 illustrates the case study findings compared to the literature findings.

Challenges		
Case Study Findings	**Introduced by**	**Literature findings**
Middle-management resistance to change	Organization	[53]
Change of work habits	Agile, UCD, and Lean Startup	[39], [17]
Introduction of Product Designer role	UCD	[28]
Organizational Issues	Organization	[17]
Dependency of soft skills	Agile, UCD, and Lean Startup	[28]

Table 5.2: Challenges: Literature Findings x Case Study Findings

5.4 Empirical Approach x Literature Models

The outcomes of this study show the relevance of a detailed workflow for the integration of the Agile, UCD, and Lean Startup. The multiple case study findings differ in some aspects from the reported studies on literature.

It is relevant to mention that are similarities comparing the systematic mapping to the multiple case study. The Drobrigkeit et al. [20] map in their model a scoping phase that aims to understand and map the problem that the team will work on, the same that multiple case study found. The use of a double diamond is presented in the two cases, even that the multiple case study workflow use as pillar UCD. Moreover, all of them have a problem-oriented mindset.

The most significant differences between the systematic mapping 3 and the multiple case study 4 (Figure 5.3) are the use of UCD instead design thinking, the build-measure-cycle loop during the whole process instead in just a part of it, and the use of extreme programming as an agile methodology scrum.

Associated to the methodologies choose by the company UCD and extreme programming that were inherited from the Pivotal Labs approach, and this was not a decision made by the teams or company. However, observing the results, the use of this two methodologies to cover the user and technological aspects seems to make sense due the detailed activities, techniques, and practices of each one of them, especially from extreme programming, since scrum framework has its strengths on process management while XP explore in-depth the software engineering practices and techniques [2].

In contrast to the systematic mapping findings, the multiple case study shows that the usage of the build-measure-learn loop during the whole process is the key to work drives by a problem-oriented. The loop allows teams to validate their learnings based on the previous iteration, build their products grounded on the passed learnings, and measure the success of the product built. The usage of the loop during the whole process is essential

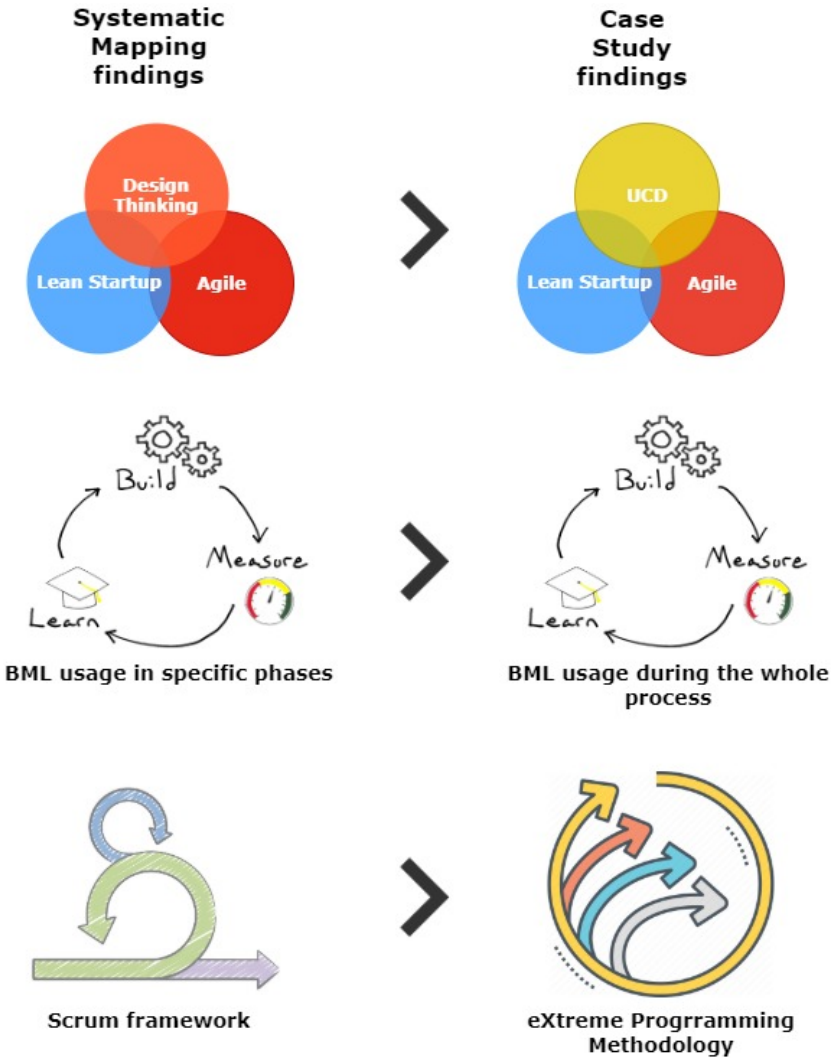


Figure 5.3: Systematic Mapping x Case Study
[Source: Author (2019)]

to avoid the waste on the process and have more certainty that the product built is actually what the user and business need.

The comparison between the two studies show the need for a detailed characterization of the combined approach. Also, reinforce the relevance to observe how this kind of transformation takes place on a large-scale setup, and what are the benefits and challenges faced by these companies. These results allow us to contribute with a set of recommendations for companies who want to adopt and face this kind of transformation. These recommendations are detailed next.

5.5 Recommendations for Practice

Although we know that there is no single solution that fits it all, from the results of our study, we derive functional and practical recommendations for companies that are in a similar transformation process.

- Assign people to lead the transformation
- Provide material to educate those that are directly involved but do not forget those that will be brought together later on in a gradual transformation approach
- Define strategies to gain management support aiming to reduce resistance and facilitate role description changes
- Bring support from experts (e.g., consulting)
- Adopt a hands-on training approach to facilitate culture change and the shaping on new working skills
- Use recently trained people to train others and speed up the cultural and organizational changes
- Consider a cross-functional team composed of a Product Manager, Product Designer, and Software Engineer roles
- Seek for customer and end user engagement
- Use experimentation as a support strategy to problem-solving and value-driven mindsets transformation
- Pivot solutions and do not be afraid to fail avoiding waste
- Introduce continuous delivery as opposite to release-plan driven as yet another strategy to support the migration to value-driven development
- Empower cross-functional teams and give them autonomy to own all working processes, from problem understanding to delivering to production, reducing the need for coordination with outside teams
- Introduce agile software engineering practices (e.g., XP) to improve product quality

6. CONCLUSION

The need for a detailed characterization describing a combined approach usage comprised of Agile, UCD, Lean Startup motivates us to investigate the area. We have noticed that have a lack of studies describing the combined approach and also, the transformation process to use it. Driven by this research problem, our main goal was to characterize the combined use of agile, ucd, and lean startup as an approach. To accomplish that, we conducted two studies: a systematic mapping and a multiple case study.

The systematic mapping was conducted to give us an understanding of the state-of-art in the research area. The results revealed the existence of 4 models - 3 as a software process model, and 1 as a design process model. Those models usually were comprised of agile (using Scrum framework), design thinking (DT), and Lean Startup. The models claimed for a cross-functional team to cover the three methodologies combined, a mindset changing, and feedback culture with customers. Also, the challenges were mostly related to organizational issues. Although, the existence of literature papers illustrating the combined approach, none of the studies were conducted in a multinational company setting - only in startup context which gives us conditions to contribute with the literature, as our second study was in this type of company complexity.

The multiple case study aimed to characterize through an agile transformation on the use of a combined approach of Agile, User-Centered Design, and Lean Startup. Through the perspective of two teams from the financial area, we were able to identify strategies to conduct a transformation as the creation of a dedicated Transformation Team to lead the transformation and also a Toolkit elaboration available to support team members throughout this process. Besides, we were able to characterize the combined approach used during the time that teams pass through one of the initiatives of the Transformation Team - learn-on-the-job hands-on immersion training which lasts 12 weeks and during more 12 weeks, we have the opportunity to validate our findings with those teams. Our study also reveals a set of elements that composes the combined approach workflow. These elements are activities, meetings, techniques/practices of each methodology, as well as the roles and the mindset change. We also mapped the benefits of adopting the combined approach, as a cross-functional team structure, especially with the introduction of the product designer role, and the application of build-measure-learn loop as an approach, guiding the development process through experimentation. The most challenging factors during the transformation were mapped as well, and these factors are associated with the managerial mindset change and team autonomy at risk.

Considering the limitations of our study, we characterize how the three pillars of Agile, User-Centered Design, and Lean Startup are combined, based on an empirical study.

Our study can be useful in different scenarios:

- **Combined Approach Characterization:** To the academy, we provide a detailed description of the combined approach usage, presenting roles, activities, techniques, outcomes, ceremonies that composed the approach
- **Combined Approach Transformation**: We present strategies that are taken to conduct such transformation in the context of a software multinational company. This finding can contribute to the academic area and industry practitioners that aim to start this type of adoption
- **Combined Approach Benefits and Challenges:** We present a set of advantages and disadvantages of adopting such an approach - that can benefit academics and industry
- **Recommendations for Practices:** Especially for industry practitioners, we recommend a set of good practices for those that aim to adopt the combined approach.

6.1 Limitations

We discuss the threats to validity for the systematic mapping and the multiple case study in its sections. However, we observed some limitations in a general view of this thesis:

- We conducted a case study observing only two teams. We cannot guarantee the participants' reports were faithful to the real facts. As we were capturing the participants' perspective which is a piece of tacit knowledge, they could not report any event or information. Aiming to mitigate this limitation, we use a range of data collection methods as interview, observation, and focus group sessions to gather as much true information as we could
- The case study teams' participants work in internal software products (e.g., for ORG use only). We considered this as a restriction to understand the combined approach operation, since we did not know whether the approach fits for external software products (e.g., for customers in general)
- The company setting was considered a limitation, since we cannot affirm that the combined approach adoption works for startups, small or medium size companies. Our study was conducted in a large-scale company and this could influence the adoption itself
- It has a limitation of the interpretation of the result by the researcher. Aiming to mitigate that, we conducted revision sessions with two senior researchers, especially in the data extraction of the systematic mapping and the code analysis on the case study results.

6.2 Future Work

As future work, we suggest some questions that could be considered to improve the studies on this specific subject:

- Conduct a case study where the teams product scope being external. It will be essential to analyze the combined approach operation with a this different input
- Confirm the combined approach elements mapped by applying in other teams to make improvements: It will be important to conduct more case studies to understand whether there are more elements, and how other teams apply the concepts from the three methodologies. These findings could be useful to provide more accurate mapping and also can be a start point in a path to elaborate a combined approach conceptual model
- Conduct a longitudinal case study to investigate the adoption in a long-term: Could be interesting to conduct a longitudinal case study aiming to understand whether the adoption in a long-term is sustained in the companies, and also verify points of improvements
- Conduct another study aiming to understand the transition for teams that work in a project-like structure adopting a product-like mindset. Understand the benefits and challenges of this transformation
- Define a scalable model to accelerate the transformation in large-scale companies who desires to face a combined approach adoption: From a conceptual model defined, it could be relevant to propose a model that provides tools to accelerate the combined approach transformation, to escalate to the rest of the company.

REFERENCES

[1] Alahyari, H.; Svensson, R. B.; Gorschek, T. “A Study of Value in Agile Software Development Organizations”, *Journal of Systems and Software*, vol. 125, Mar. 2017, pp. 271–288.

[2] Anwer, F.; Aftab, S.; Shah, S. M.; Waheed, U. “Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum”, *International Journal of Computer Science and Telecommunications*, vol. 8, Mar. 2017, pp. 1–7.

[3] Barroca, L.; Sharp, H.; Dingsøyr, T.; Gregory, P.; Taylor, K.; AlQaisi, R. “Enterprise Agility: A Balancing Act - A Local Government Case Study”. In: International Conference on Agile Software Development, 2019, pp. 207–223.

[4] Bastarrica, M.; Espinoza, G.; Sánchez, J. “Implementing Agile Practices: The Experience of TSol”. In: International Symposium on Empirical Software Engineering and Measurement, 2018, pp. 1–10.

[5] Beck, K.; Beedle, M.; Van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; et al.. “Manifesto for Agile Software Development”. Source: http://agilemanifesto.org/, 2019 september.

[6] Bjørnar; T. “Individual Empowerment of Agile and Non-Agile Software Developers in Small Teams”, *Information and Software Technology*, vol. 56, Aug. 2014, pp. 873 – 889.

[7] Boehm, B.; Turner, R. “Management Challenges to Implementing Agile Processes in Traditional Development Organizations”, *IEEE Software*, vol. 22, Sep. 2005, pp. 30–39.

[8] Bordin, S. “Towards a More User-Centred Agile Development”. In: International Conference on Agile Software Development, 2016, pp. 307–311.

[9] Campanelli, A. S.; Parreiras, F. S. “Agile Methods Tailoring -– A Systematic Literature Review”, *Journal of Systems and Software*, vol. 110, Dec. 2015, pp. 85 – 100.

[10] Causevic, A.; Sundmark, D.; Punnekkat, S. “Factors Limiting Industrial Adoption of Test Driven Development: A Systematic Review”. In: International Conference on Software Testing, Verification and Validation, 2011, pp. 337–346.

[11] Chen, L. “Continuous Delivery: Huge Benefits, but Challenges too”, *IEEE Software*, vol. 32, Jan. 2015, pp. 50–54.

[12] Chung, M.; Drummond, B. “Agile at Yahoo! From the Trenches”. In: Agile Conference, 2009, pp. 113–118.

[13] Cohen, J. “A Coefficient of Agreement for Nominal Scales”, *Educational and Psychological Measurement*, vol. 20, Apr. 1960, pp. 37–46.

[14] Creswell, J. W.; Creswell, D. J. “Research Design Qualitative, Quantitative, and Mixed Methods Approaches”. SAGE, 2018, 304p.

[15] de Paula, D. F.; Araújo, C. C. “Pet Empires: Combining Design Thinking, Lean Startup and Agile to Learn from Failure and Develop a Successful Game in an Undergraduate Environment”. In: International Conference on Human-Computer Interaction, 2016, pp. 30–34.

[16] de Paula, D. F. O. “Model for the Innovation Teaching (MoIT): Um Modelo Baseado em Design Thinking, Lean Startup e Ágil para Estudantes de Graduação em Computação”, Dissertação de mestrado, Universidade Federal de Pernambuco, 2015, 85p.

[17] Dikert, K.; Paasivaara, M.; Lassenius, C. “Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review”, *Journal of Systems and Software*, vol. 119, Sep. 2016, pp. 87–108.

[18] Dingsøyr, T.; Fægri, T. E.; Itkonen, J. “What is Large in Large-Scale? A Taxonomy of Scale for Agile Software Development”. In: International Conference on Product-Focused Software Process Improvement, 2014, pp. 273–276.

[19] Dingsøyr, T.; Nerur, S.; Balijepally, V.; Moe, N. B. “A Decade of Agile Methodologies: Towards Explaining Agile Software Development”, *Journal of Systems and Software*, vol. 85, Jun. 2012, pp. 1213 – 1221.

[20] Dobrigkeit, F.; de Paula, D.; Uflacker, M. “InnoDev: A Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup”. Springer International Publishing, 2019, chap. 9, pp. 199–227.

[21] Dobrigkeit, F.; de Paula, D.; et al.. “The Best of Three Worlds: The Creation of InnoDev, a Software Development Approach that Integrates Design Thinking, Scrum and Lean Startup”. In: International Conference on Engineering Design, 2017, pp. 319–328.

[22] Edison, H.; Wang, X.; Abrahamsson, P. “Lean Startup: Why Large Software Companies Should Care”. In: International Conference on Agile Software Development, 2015, pp. 1–7.

[23] Gandomani, T. J.; Nafchi, M. Z. “An Empirically-Developed Framework for Agile Transition and Adoption: A Grounded Theory Approach”, *Journal of Systems and Software*, vol. 107, Sep. 2015, pp. 204 – 219.

[24] Gandomani, T. J.; Zulzalil, H.; Ghani, A. A. A.; Sultan, A. B. M.; Parizi, R. M. “The Impact of Inadequate and Dysfunctional Training on Agile Transformation Process: A Grounded Theory Study”, *Information and Software Technology*, vol. 57, Jan. 2015, pp. 295 – 309.

[25] Garcia, A.; da Silva, T. S.; Silveira, M. S. “Artifact-Facilitated Communication in Agile User-Centered Design”. In: International Conference on Agile Software Development, 2019, pp. 102–118.

[26] González, C. S. G.; González, E. G.; Cruz, V. M.; Saavedra, J. S. “Integrating the Design Thinking into the UCD’s methodology”. In: IEEE Education Engineering Conference, 2010, pp. 1477–1480.

[27] Gothelf, J. “Lean UX: Applying Lean Principles to Improve User Experience”. O’Reilly, 2013, 152p.

[28] Gregory, P.; Barroca, L.; Sharp, H.; Deshpande, A.; Taylor, K. “The Challenges that Challenge: Engaging with Agile Practitioners’Concerns”, *Information and Software Technology*, vol. 77, Sep. 2016, pp. 92–104.

[29] Grossman-Kahn, B.; Rosensweig, R. “Skip the Silver Bullet: Driving Innovation through Small Bets and Diverse Practices”, *Leading Through Design*, vol. 14, Aug. 2012, pp. 815–830.

[30] Gutbrod, M.; Münch, J.; Tichy, M. “How Do Software Startups Approach Experimentation? Empirical Results from a Qualitative Interview Study”. In: Product-Focused Software Process Improvement, 2017, pp. 297–304.

[31] Häger, F.; Kowark, T.; Krüger, J.; Vetterli, C.; Übernickel, F.; Uflacker, M. “DT@Scrum: Integrating Design Thinking with Software Development Processes”. Springer International Publishing, 2015, chap. 13, pp. 263–289.

[32] Hansen, M. T.; Baggesen, H. “From CMMI and Isolation to Scrum, Agile, Lean and Collaboration”. In: Agile Conference, 2009, pp. 283–288.

[33] Hoda, R.; Salleh, N.; Grundy, J.; Tee, H. M. “Systematic Literature Reviews in Agile Software Development: A Tertiary Study”, *Information and Software Technology*, vol. 85, May. 2017, pp. 60–70.

[34] IDEO. “Human Centered Design toolkit”. Source: https://www.ideo.com/post/design-kit/, 2018 september.

[35] ISO. “9241-210: Ergonomics of Human System Interaction-Part 210: Human-Centred Design for Interactive Systems”, Ergonomics of human-system interaction 13.180, International Organization for Standardization (ISO), 2010, 108p.

[36] Jalali, S.; Wohlin, C.; Angelis, L. “Investigating the Applicability of Agility Assessment Surveys: A Case Study”, *Journal of Systems and Software*, vol. 98, Dec. 2014, pp. 172 – 190.

[37] Jovanovic, M.; Mas, A.; Mesquida, A.; Lalic, B. “Transition of Organizational Roles in Agile Transformation Process: A Grounded Theory Approach”, *Journal of Systems and Software*, vol. 133, Nov. 2017, pp. 174–194.

[38] Julian, B.; Noble, J.; Anslow, C. “Agile Practices in Practice: Towards a Theory of Agile Adoption and Process Evolution”. In: International Conference on Agile Software Development, 2019, pp. 3–18.

[39] Karvonen, T.; Sharp, H.; Barroca, L. “Enterprise Agility: Why is Transformation So Hard?” In: International Conference on Agile Software Development, May. 2018, pp. 131–145.

[40] Kitchenham, B.; Charters, S.; et al.. “Guidelines for Performing Systematic Literature Reviews in Software Engineering”, Technical report, School of Computer Science and Mathematics, Keele University, Jul. 2007, 65p.

[41] Krippendorff, K. “Content Analysis: An Introduction to Its Methodology”. SAGE, 2018, 456p.

[42] Kuusinen, K.; Gregory, P.; Sharp, H.; Barroca, L.; Taylor, K.; Wood, L. “Knowledge Sharing in a Large Agile Organisation: A Survey Study”. In: International Conference on Agile Software Development, 2017, pp. 135–150.

[43] Landis, J. R.; Koch, G. G. “The Measurement of Observer Agreement for Categorical Data”, *International Biometric Society*, vol. 33, Mar. 1977, pp. 159–174.

[44] Liikkanen, L. A.; Kilpiö, H.; Svan, L.; Hiltunen, M. “Lean UX: The Next Generation of User-Centered Agile Development?” In: Nordic Conference on Human-Computer Interaction, 2014, pp. 1095–1100.

[45] Marinho, M.; Noll, J.; Richardson, I.; Beecham, S. “Plan-Driven Approaches Are Alive and Kicking in Agile Global Software Development”. In: International Symposium on Empirical Software Engineering and Measurement, 2019, pp. 1–11.

[46] May, B. “Applying Lean Startup: An Experience Report–Lean & Lean UX by a UX Veteran: Lessons Learned in Creating & Launching a Complex Consumer App”. In: Agile Conference, 2012, pp. 141–147.

[47] McHugh, O.; Conboy, K.; Lang, M. “Using Agile Practices to Build Trust in an Agile Team: A Case Study”. In: International Conference on Information Systems Development, Sep. 2011, pp. 503–516.

[48] Müller, R. M.; Thoring, K. “Design Thinking vs. Lean Startup: A Comparison of Two User-Driven Innovation Strategies”, *Leading through design*, vol. 151, Aug. 2012, pp. 91–106.

[49] Norman, D. “The Design of Everyday Things”. Verlag Franz Vahlen GmbH, 2016, 347p.

[50] Norman, D.; Draper, S. “User Centered System Design: New Perspectives on Human-Computer Interaction”. CRC Press, 1986, 544p.

[51] Nyfjord, J.; Kajko-Mattsson, M. “Degree of Agility in Pre-Implementation Process Phases”. In: International Conference on Software Process, 2008, pp. 234–245.

[52] O’Connor, C. P. “Anatomy and Physiology of an Agile Transition”. In: Agile Conference, 2011, pp. 302–306.

[53] Paasivaara, M.; Behm, B.; Lassenius, C.; Hallikainen, M. “Large-Scale Agile Transformation at Ericsson: A Case Study”, *Empirical Software Engineering*, vol. 23, Jan. 2018, pp. 2550–2596.

[54] Paasivaara, M.; Lassenius, C. “Communities of Practice in a Large Distributed Agile Software Development Organization–Case Ericsson”, *Information and Software Technology*, vol. 56, Dec. 2014, pp. 1556–1577.

[55] Packlick, J. “The Agile Maturity Map A Goal Oriented Approach to Agile Improvement”. In: Agile Development Conference, 2007, pp. 266–271.

[56] Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. “Systematic Mapping Studies in Software Engineering”. In: International Conference on Evaluation and Assessment in Software Engineering, 2008, pp. 68–77.

[57] Petersen, K.; Vakkalanka, S.; Kuzniarz, L. “Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update”, *Information and Software Technology*, vol. 64, Aug. 2015, pp. 1–18.

[58] Ries, E. “The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses”. Crown Business, 2011, 320p.

[59] Rohunen, A.; Rodriguez, P.; Kuvaja, P.; Krzanik, L.; Markkula, J. “Approaches to Agile Adoption in Large Settings: A Comparison of the Results from a Literature Analysis and an Industrial Inventory”. In: International Conference on Product Focused Software Process Improvement, 2010, pp. 77–91.

[60] Runeson, P.; Höst, M. “Guidelines for Conducting and Reporting Case Study Research in Software Engineering”, *Empirical Software Engineering*, vol. 14, Dec. 2008, pp. 131.

[61] Salerno, L.; Signoretti, I.; Marczak, S.; Bastos, R. “Repensando Papeis em Equipes Ágeis: Um Estudo de Caso no Uso de uma Abordagem Combinada de Desenvolvimento Ágil, User-Centered Design e Lean Startup”. In: Conferência Brasileira de Software: Teoria e Prática, Sep. 2019, pp. 72–77.

[62] Schön, E.-M.; Winter, D.; Escalona, M. J.; Thomaschewski, J. “Key Challenges in Agile Requirements Engineering”. In: International Conference on Agile Software Development, 2017, pp. 37–51.

[63] Sedano, T.; Ralph, P.; Péraire, C. “Sustainable Software Development Through Overlapping Pair Rotation”. In: International Symposium on Empirical Software Engineering and Measurement, 2016, pp. 1–10.

[64] Sidky, A.; Arthur, J.; Bohner, S. “A Disciplined Approach to Adopting Agile Practices: The Agile Adoption Framework”, *Innovations in Systems and Software Engineering*, vol. 3, Jul. 2007, pp. 203–216.

[65] Signoretti, I.; Marczak, S.; Salerno, L.; Lara, A. d.; Bastos, R. “Boosting Agile by Using User-Centered Design and Lean Startup: A Case Study of the Adoption of the Combined Approach in Software Development”. In: International Symposium on Empirical Software Engineering and Measurement, Sep. 2019, pp. 1–6.

[66] Smite, D.; Moe, N. B.; Levinta, G.; Floryan, M. “Spotify Guilds: How to Succeed With Knowledge Sharing in Large-Scale Agile Organizations”, *IEEE Software*, vol. 36, Mar. 2019, pp. 51–57.

[67] Sommerville, I. “Software Engineering”. Pearson, 2016, 796p.

[68] Soundararajan, S.; Balci, O.; Arthur, J. “Assessing an Organization’s Capability to Effectively Implement Its Selected Agile Method(s): An Objectives, Principles, Strategies Approach”. In: Agile Development Conference, 2013, pp. 22–31.

[69] Strauss, A.; Corbin, J. “Basics of Qualitative Research Techniques”. SAGE, 1998, 454p.

[70] Vargas, B. P.; Signoretti, I.; Zorzetti, M.; Marczak, S.; Bastos, R. “On the Understanding of Experimentation Usage in Light of Lean Startup in Software Development Context”. In: Evaluation and Assessment in Software Engineering, 2020, pp. 330–335.

[71] Vilkki, K. “When Agile is Not Enough”. In: International Conference on Lean Enterprise Software and Systems, 2010, pp. 44–47.

[72] Vlaanderen, K.; van Stijn, P.; Brinkkemper, S.; van de Weerd, I. “Growing into Agility: Process Implementation Paths for Scrum”. In: International Conference on Product Focused Software Process Improvement, 2012, pp. 116–130.

[73] Wohlin, C. “Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering”. In: International Conference on Evaluation and Assessment in Software Engineering, 2014, pp. 1–10.

[74] Wölbling, A.; Krämer, K.; Buss, C. N.; Dribbisch, K.; LoBue, P.; Taherivand, A. “Design Thinking: An Innovative Concept for Developing User-Centered Software”. Springer Berlin Heidelberg, 2012, chap. 7, pp. 121–136.

[75] Ximenes, B. H.; Alves, I. N.; Araújo, C. C. “Software Project Management Combining Agile, Lean startup and Design Thinking”. In: International Conference of Design, User Experience, and Usability, 2015, pp. 356–367.

[76] Yaman, S.; Munezero, M.; Münch, J.; Fagerholm, F.; Syd, O.; Aaltola, M.; Palmu, C.; Männistö, T. “Introducing Continuous Experimentation in Large Software-Intensive Product and Service Organisations”, *Journal of Systems and Software*, vol. 133, Nov. 2017, pp. 195 – 211.

[77] Yilmaz, M.; O’Connor, R. V.; Clarke, P. “Software Development Roles: A Multi-Project Empirical Investigation”, *Special Interest Group on Software Engineering*, vol. 40, Feb. 2015, pp. 1–5.

[78] Yin, R. K. “Case Study Research and Applications: Design and Methods”. SAGE, 2017, 350p.

APPENDIX A – SEARCH STRING

In this appendix, we present how the strings were structured for each specific base.

SCOPUS:

110 results (07/08/2019)

ALL (“Design Thinking" OR “*Centered Design" OR “*Centred Design") AND ALL (“Lean Startup" OR “Lean Start-Up" OR “Lean UX") AND ALL (“Agile" OR “Agile Practice" OR “Agile Method*" OR “Agile Development" OR “Agile Software*" OR “Extreme Programming" OR “Scrum" OR “Kanban") AND (“Software Development")

IEEE:

36 results (07/08/2019)

((“Lean Startup" OR “Lean Start-Up" OR “Lean User Experience" OR Lean UX) AND (“Agile" OR Agile Method* OR Agile Development OR Agile Software Development OR “SCRUM" OR “Extreme Programming" OR “Kanban") AND (“Design Thinking" OR “*Centered Design" OR “*Centred Design") AND (“Software Development"))

ACM:

10 results (07/08/2019)

((“Lean Startup") OR (“Lean UX") OR (“Lean Start-Up") OR (“Lean User Experience")) AND ((“Agile") OR (“Agile Method") OR (“Agile Development") OR (“Agile Software Development") OR “Agile Methodology") OR (“SCRUM")OR (“Extreme Programming") OR (“Kanban")) AND ((“Design Thinking") OR(“User-Centered Design") OR (“User-Centred Design") OR (Human-Centered Design) OR (Human-Centred Design)) AND (“Software Development"))

Science Direct:

140 Results (07/08/2019)

(“Design Thinking" OR Centered Design OR Centred Design)AND (“Lean Startup" OR Lean UX OR “Lean Start-Up") AND (“Agile" OR Agile Method OR Agile Development OR “SCRUM" OR “Extreme Programming" OR “Kanban") AND (“Software Development")

Springer:

64 Results (07/08/2019)

“Design Thinking" OR “*Centered Design" OR “*Centred Design" “Lean Startup" OR “Lean UX" OR “Lean Star-up" “Agile" OR “Agile Method*" OR “Agile Development" OR “SCRUM" OR “Extreme Programming" OR “Kanban" “Software Development"

APPENDIX B – CASE STUDY PROTOCOL

Case Study Protocol

1. Document Overview

This protocol aims to clarify how this case study was planned aiming to identify how the combined use of User-Centered Design, Lean Startup and Agile Development takes place in a large-scale multinational company. The following sections describe the purpose of the case study, the case under investigation, the methods to be used to perform data collection, and analysis of such data.

2. Case Study Goal

This case study seeks to characterize how a certain large-scale multinational company, named ORG for confidentiality reasons, is adopting and using in a combined fashion three methodologies, namely User-Centered Design, Lean Startup and Agile to support its software teams' work.

The secondary objectives of the study are:

- To characterize the combined approach
- Map the activities, ceremonies, techniques, and outcomes produced
- To know the roles and responsibilities
- To describe the transformation process to the combined approach
- To identify benefits and challenges of the transformation

The study follows a constructivist research paradigm, which considers that the phenomema under investigation seeking to understand the team perspective about the combined approach adoption Our goal by applying a constructivist paradigm is explore in depth the phenomena by the participants view. It is also exploratory given that the researchers who will conduct the study intend to understand without preconceived thoughts and go open willing to capture how the participants perceive the phenomenona to be studied. Unlike the positivist view that starts from a pre-established theory, researchers will inductively generate or develop a theory or pattern of meaning [2].

3. Data Collection Methods

Four different methods will be used for data collection, characterizing the data triangulation collected from multiple sources.

First, an initial questionnaire will be applied to understand the profile of each team member. Then, observation sessions will be held, and interviews will be conducted, both semi-structured and unstructured. At the end of the 12 weeks of training, semi-structured interviews will be conducted to complete the study. Throughout the case study, focus group sessions will be held, aiming to understand the concepts derived from each of the methodologies separately.

The following sections explore in-depth how each method will be used and applied.

Inspirado no "Guidelines for conducting and reporting case study research in software engineering" - Per Runeson and Martin Höst.

a. Questionnaire

The process begins with the questionnaire application, which aims to map the team members' profiles who will participate in the study. The questionnaire will be applied right at the beginning of the study in order to understand: training, length of experience in IT, how long they work at the company, how long use the approach, whether it was trained by consultants or by the company's members and what team member's role.

b. Observation

The researchers will act as direct observer, that is, they will observe without interfering in the environment or the activities of people. Observations will take place throughout the entire case study period. Observations will be used to collect evidence on activities, artifacts, and techniques adopted by the in-transformation teams. If there is no impediment, the secondary objective is to observe and participate in the teams' ceremonies/rhythm (e.g., daily, iteration planning meeting.).

We aim to conduct another type of observation that is shadowing an specific role from the cross-functional team. We will also be held in order to map the activities, artifacts produced, and specific paper tools. For example, use an 1-hour session with the team's Product Designer and observe his activities.

c. Semi and Unstructured Interview

Semi-structured and unstructured interviews will be conducted with team members at different times, with the following aims:

i. To understand how each team member plays her role, to describe performed activities by her role, and to understand and to describe the software product under developed.
ii. To confirm evidences collected through the observation's sessions.
iii. To understand the team's vision/opinion regarding the application of the combined approach.

d. Focus Group

The focus group sessions will be held to collect a group vision about about the combined approach adoption. We aim to confirm and collect new data about the roles, activities, techniques, ceremonies, and outcomes from a team perspective. The focus group could be essential to validate our previously findings and guarantee the collection consistence and validity.

Reference

[1] Runeson, Per, and Martin Höst. Guidelines for conducting and reporting case study research in software engineering.

[2] *Creswell, John W., and J. David Creswell. Research design: Qualitative, quantitative, and mixed methods approaches. Sage publications, 2017.*

Inspirado no "Guidelines for conducting and reporting case study research in software engineering" - Per Runeson and Martin Höst.

APPENDIX C – QUESTIONNAIRE

Background data

Name:

What is your educational degree?

How long have you worked in IT?

How long have you worked at ORG?

When have you started using Pivotal Labs?

Where were you trained: in the USA or in Brazil?

What is your role on the team and what are your main responsibilities?

APPENDIX D – INTERVIEW SCRIPT - I

Script Interview I

1. What is your role in the team?
2. Could you tell me a little more about what are the role's responsibilities?
3. Were you trained by Pivotal Labs or by people who had been trained by Pivotal Labs?
4. Could you describe a typical day during the training?
5. Did you work with what kind of methodology / approach, before working on the Pivotal labs approach?
6. What was your role in the previous working model?
7. Did the activities you performed on the previous model change? What were these changes (New responsibilities assigned ...)?
8. What difference could you observe in retrospect to the previous model to the current one in transformation so far?
9. What benefits could you observe in transformation so far?
10. What challenges could you observe in transformation so far?

APPENDIX E – INTERVIEW SCRIPT - II

Script Interview

Cross-functional teams

1. In your opinion, what are the factors that influence the team's engagement?
    a. Having a shared product vision
    b. Co-shared responsibilities among team members
    c. Having shared knowledge among all team members

* More factors may be added to the list.

Boosting Agile

2. In your perception, what factors contribute to the relationship of trust established between stakeholders and the team?
    a. The team is working from the problem perspective rather than the predefined requirements
    b. Use of experiments as a way of “ensuring” understanding of what is working

3. In your perception, what factors have contributed to closer contact with stakeholders and users?
    a. Stakeholders are closer, through weekly meetings
    b. Delivery be validated from user feedback

* More factors may be added to the list.

Technical Aspects

4. In your perception, what are the technical factors that should be adopted and practiced to support a good delivery process? (The delivery could be software or not software)
    a. Continuous Delivery (Deploy and Delivery)
    b. Delivery Frequency
    c. Code Quality

* More factors may be added to the list.

Mindset and cultural changes

5. What are the factors that contribute to the team’s autonomy?
    a. Have a “free hand” for decision making
6. What has put team autonomy at risk? What are the challenges in this area?
    a. Middle management power.
    b. Impairments related to the teams that control deploy production

* More factors may be added to the list.

Organizational issues

7. What are the indicators that can influence the decision-making on investments based on the team's delivery?
    a. ROI
    b. Projects’ costs

APPENDIX F – CONSENT FORM

Pontifical Catholic University of Rio Grande do Sul (PUCRS)
School of Technology
Porto Alegre – RS

Consent Form

PUCRS, through the Pivotal Labs Methodology and Practices Scalability Project with the SAFe Framework of the DELL/PUCRS agreement in partnership with the School of Technology, thanks all participants of this study conducted under our responsibility for the invaluable contribution they make to the advancement of research in Software Engineering area.

The project aims to develop a software process capability model using Pivotal Labs as a reference, which encompasses concepts of User-Centered Design, Lean Startup and Agile. In this context, it is part of the project to carry out observations, interviews and group workshops to discuss with the participants their perceptions on the subject. This activity will be recorded on paper and also through audio and video recordings of all or part of it.

We remind you that the objective of the study is not to **evaluate** participants 'knowledge within the proposed activity, but to have the opportunity to understand the participants' point of view of the activity to be applied. The use made of the records made during the interview is **strictly** limited to research and development activities, ensuring that:

1. The participants anonymity will be preserved in any and all documents published in scientific forums (such as conferences, journals, books and the like) or pedagogical (such as course handouts, presentation slides, and the like).
2. Recorded audios and videos will be used for analysis only and will not be released externally.
3. Any participant who feels embarrassed or uncomfortable during a activity situation the can interrupt it and will be doing the team a favor by writing down the reasons or feelings that led to it. The team is obliged to discard the activity for the purpose of the intended evaluation.
4. Participants who are minors must present the consent of their guardian, to participate in the study, who will be declared aware of the study to be conducted by signing this consent form.
5. Each participant has the right to express in writing, on the date of the interview, any additional restrictions or conditions that may appear to apply to the items listed above (1, 2, 3 and 4). The team undertakes to observe them rigorously and understands that, in the absence of such a manifestation, the participant agrees that they govern the ethical behavior of the team only the conditions printed in this document.
6. The team has the right to use the activity data, under the conditions mentioned above, for any academic, pedagogical and / or developmental purposes contemplated by its members.

[to be completed by researchers]
Form:_____ Date: __ / __ / ____

Special conditions (if there are no special conditions, write "none"):

_____
_____
_____
_____
_____
_____

☐ continua no verso

Please indicate your position regarding the above terms:

☐ I fully agree with the above terms.

☐ Attached registration additional conditions for this test.

_____
Participant's Signature

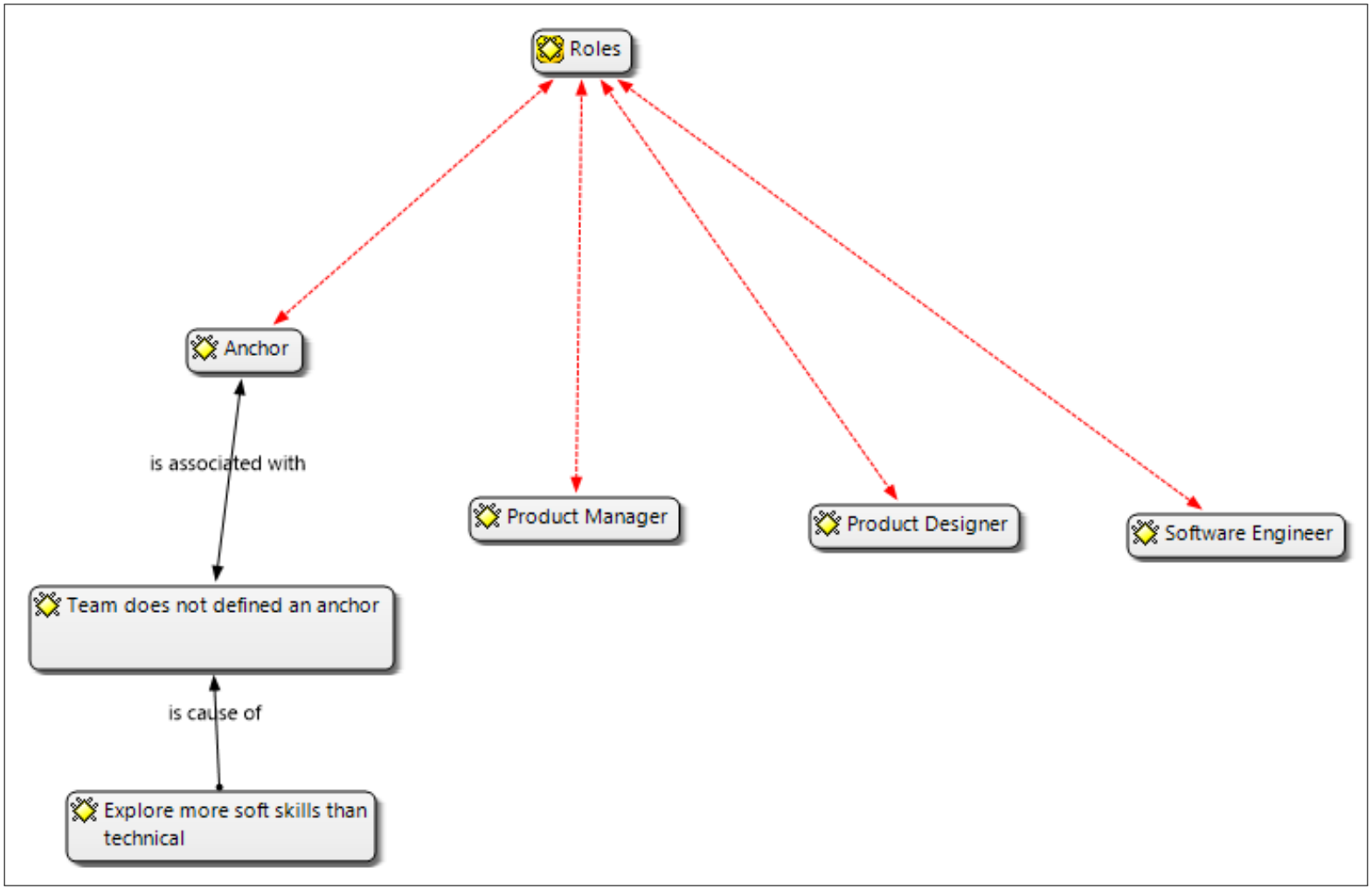
_____
Signature of the responsible
(if the participant is a minor)

_____
Researcher's signature

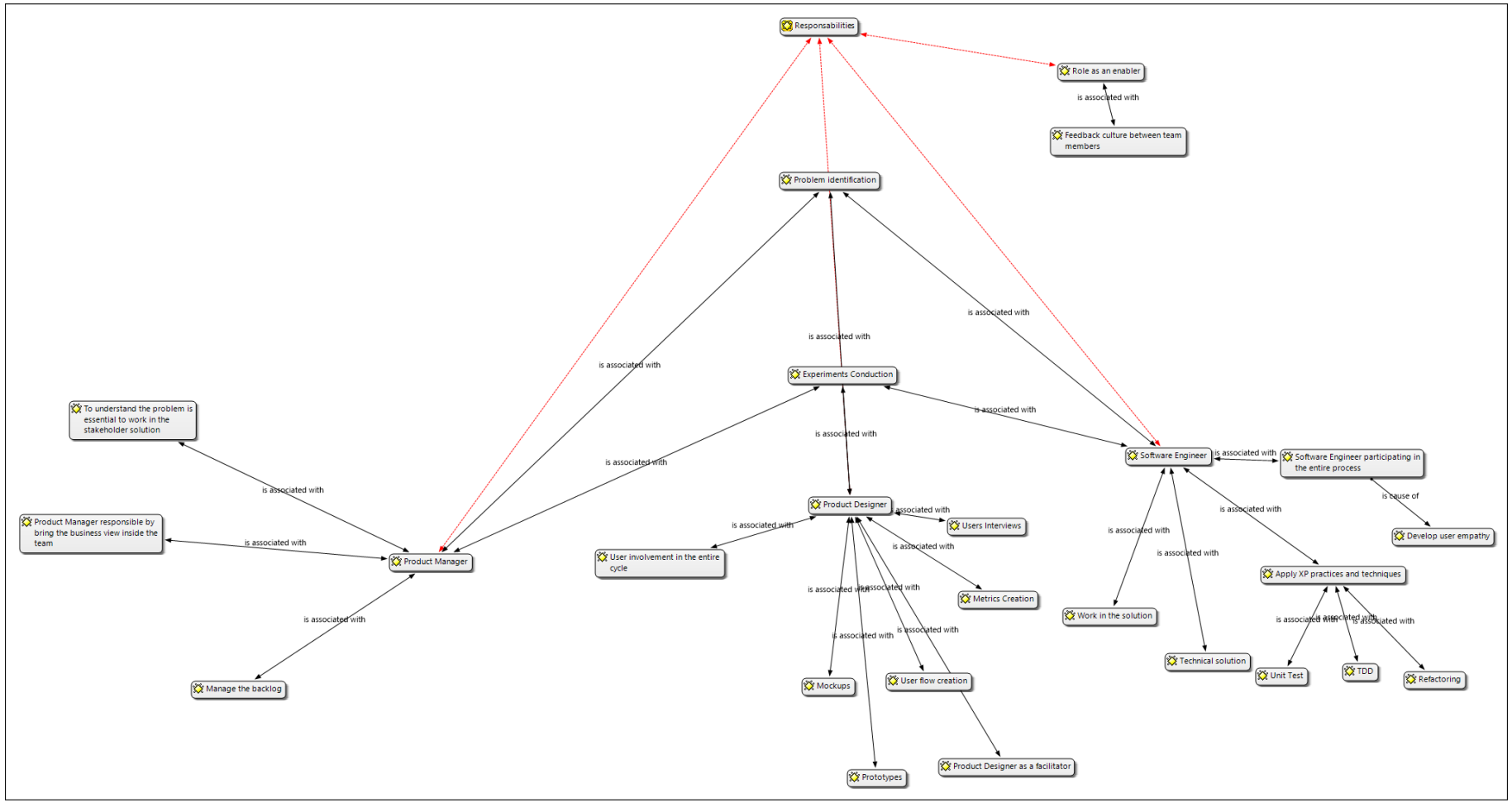
Participant Name:

Researchers: Ricardo Bastos and Sabrina Marczak (Lead Researchers), Cassiano. Moralles (Ph.D Candidate), Ingrid Signoretti, Maximilian Zorzetti, Matheus Vaccaro and Cássio Trindade (Master Students), Bruna Prauchner and Larissa Salerno (Bachelors Students)

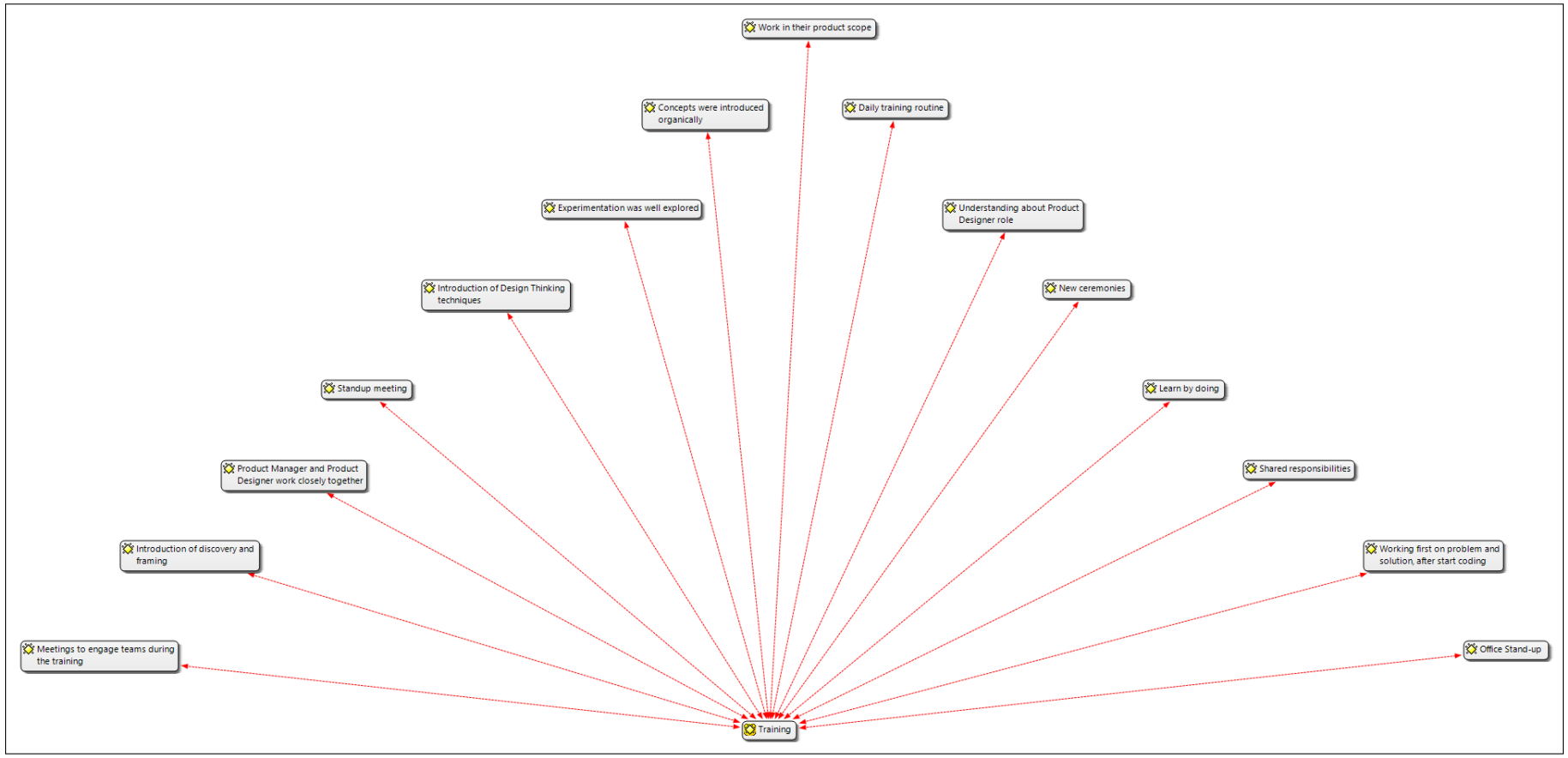
APPENDIX G – CODE ANALYSIS



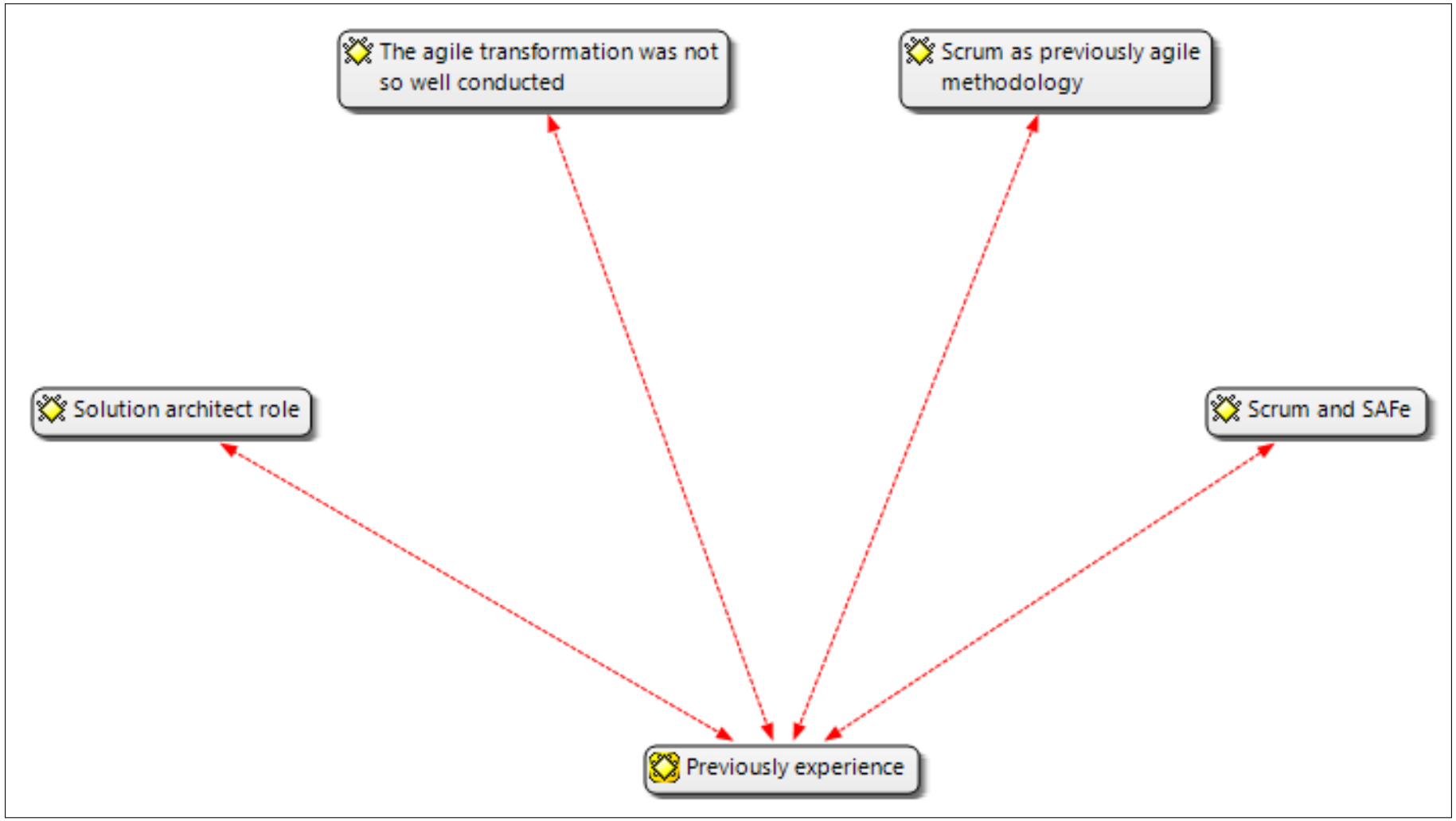
This code analysis corresponds to the first question of the script interview in Appendix D



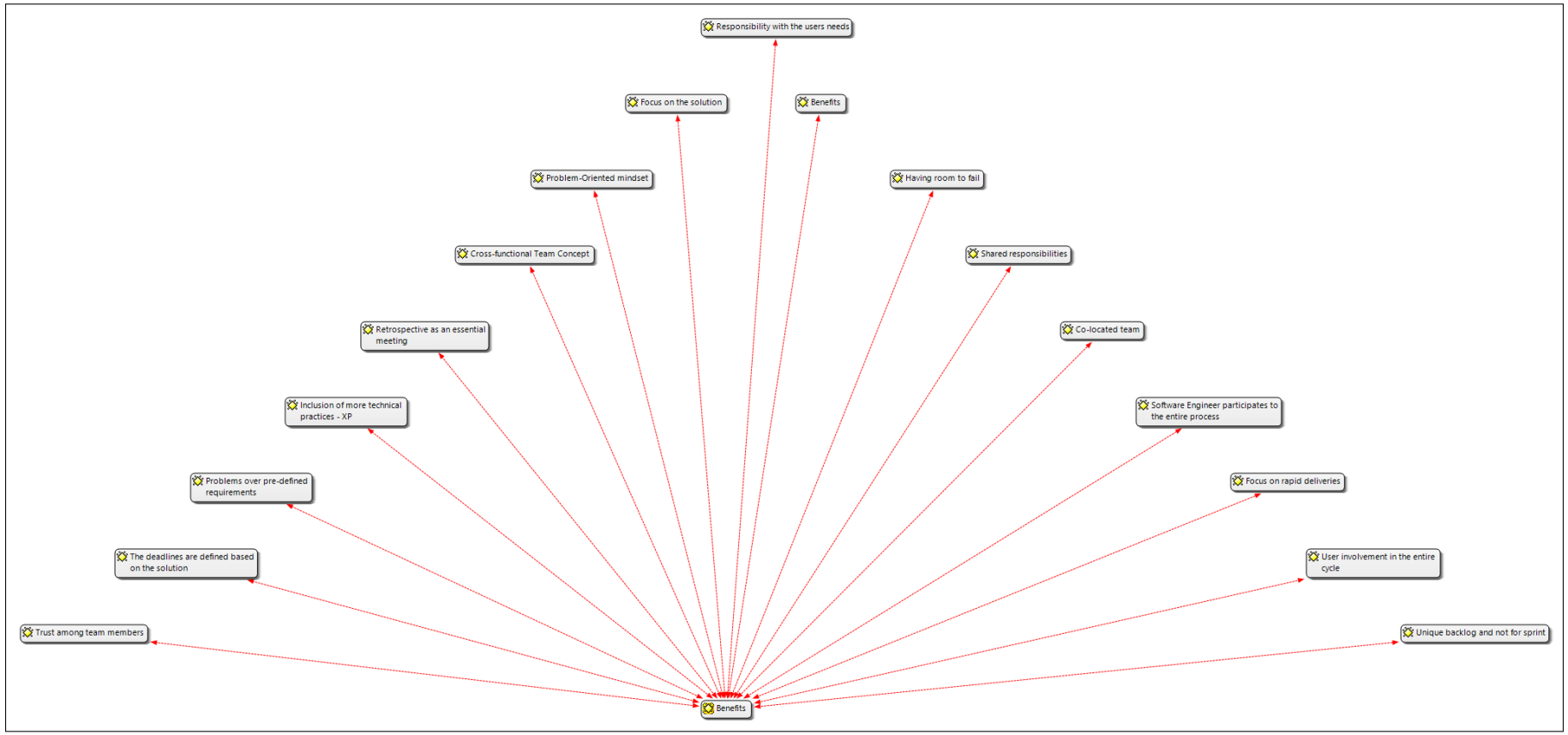
This code analysis corresponds to the second question of the interview script in Appendix D



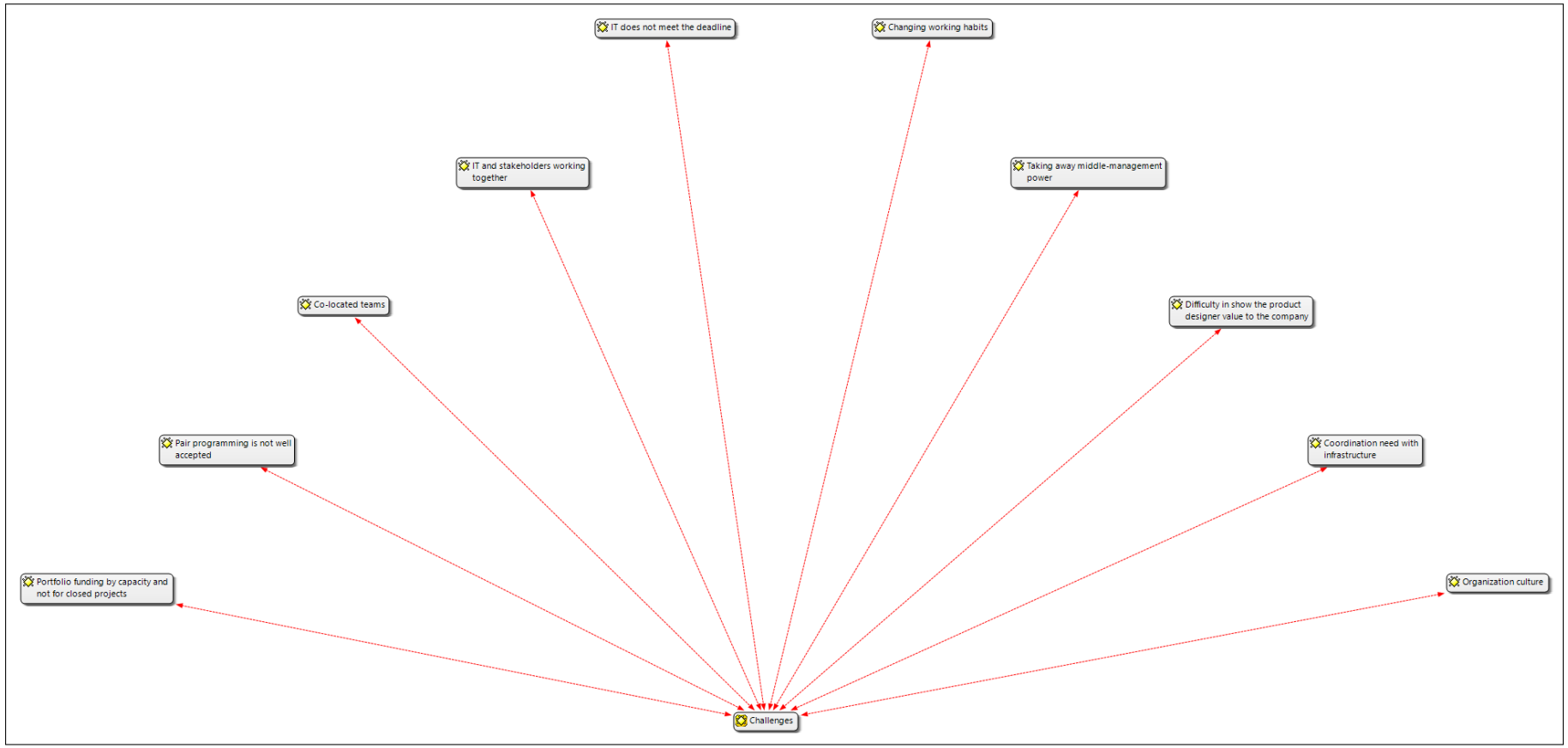
This code analysis corresponds to the third question of the interview script in Appendix D



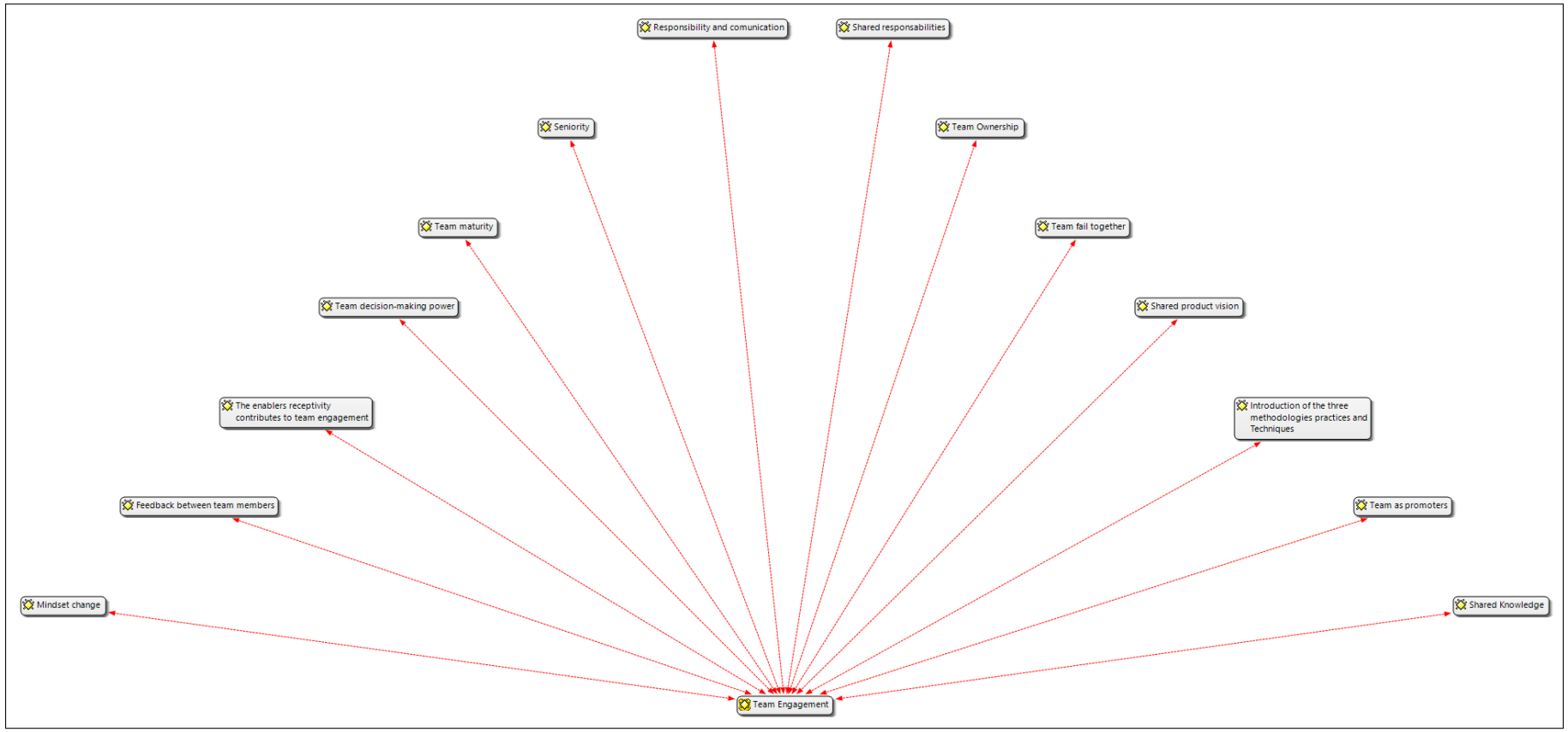
This code analysis corresponds to the fourth, fifth, sixth and seventh question of the interview script in Appendix D



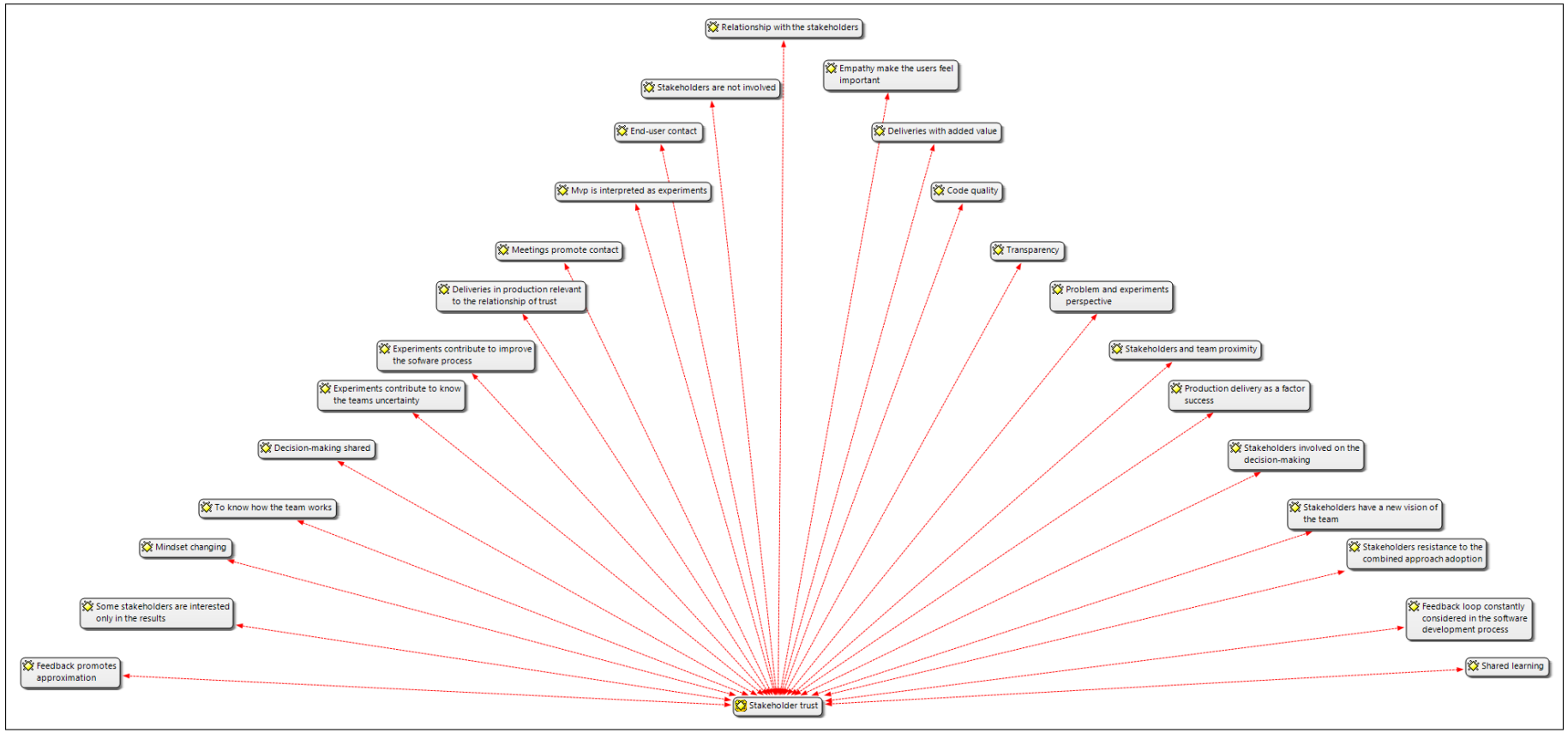
This code analysis corresponds to the eighth and ninth question of the interview script in Appendix D



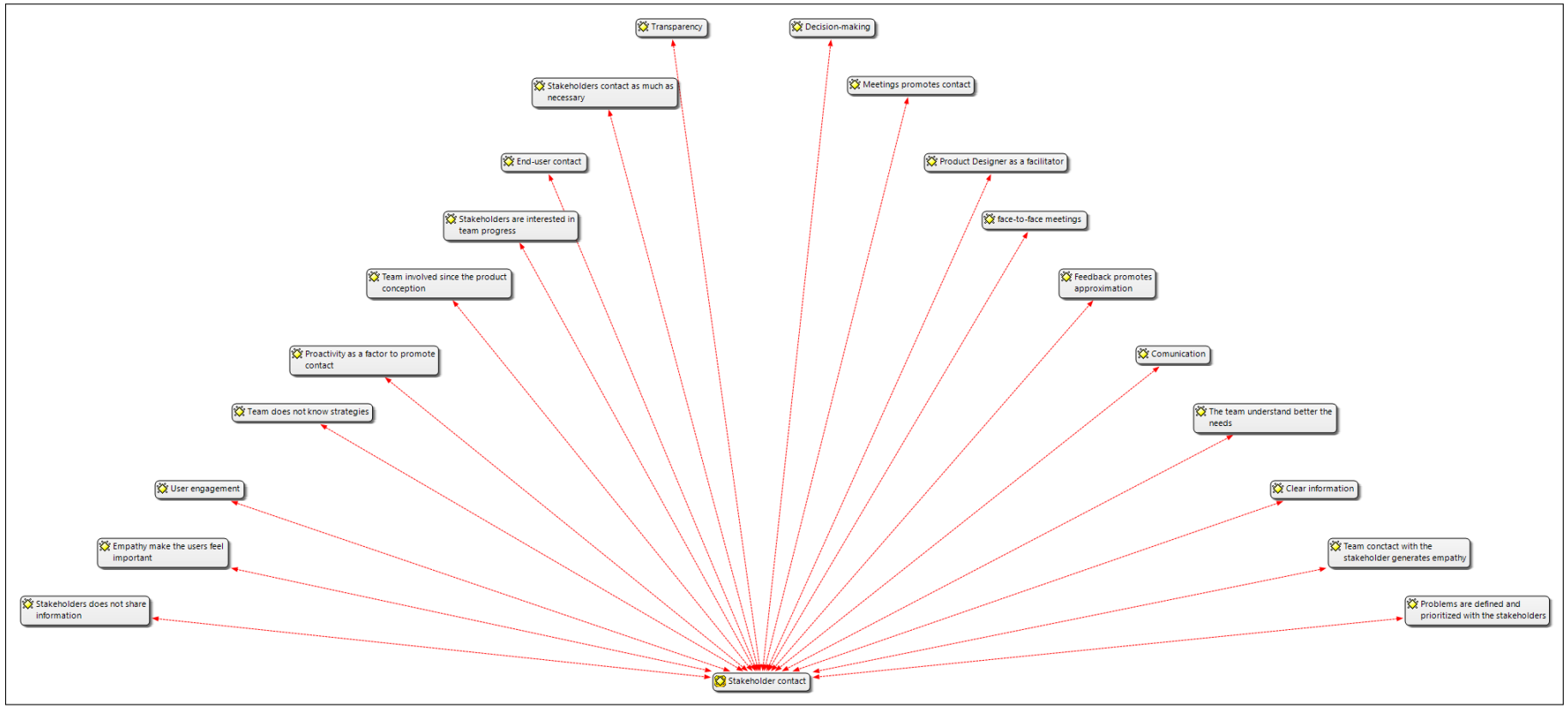
This code analysis corresponds to the eighth and tenth question of the interview script in Appendix D



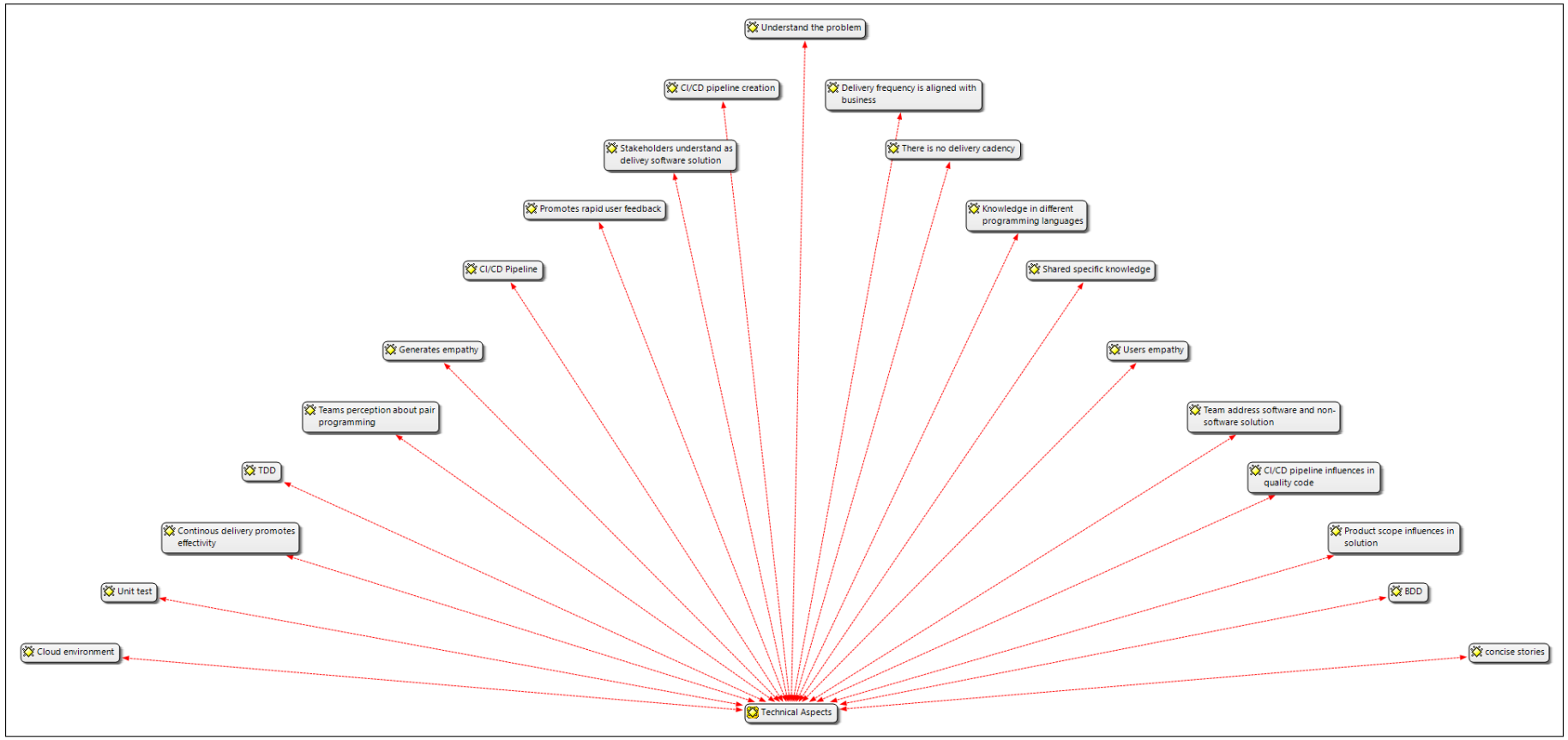
This code analysis corresponds to the first question of the interview script in Appendix E



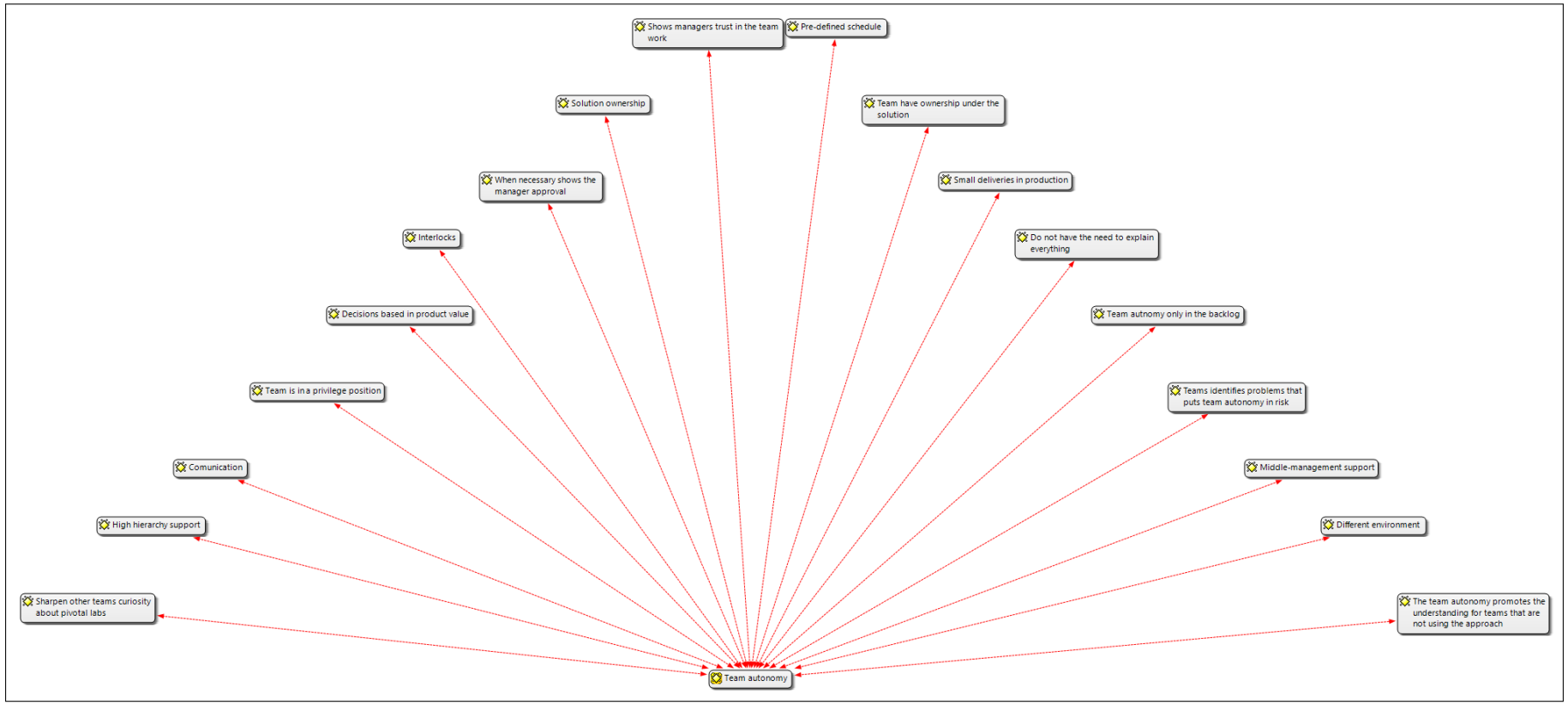
This code analysis corresponds to the second question of the interview script in Appendix E



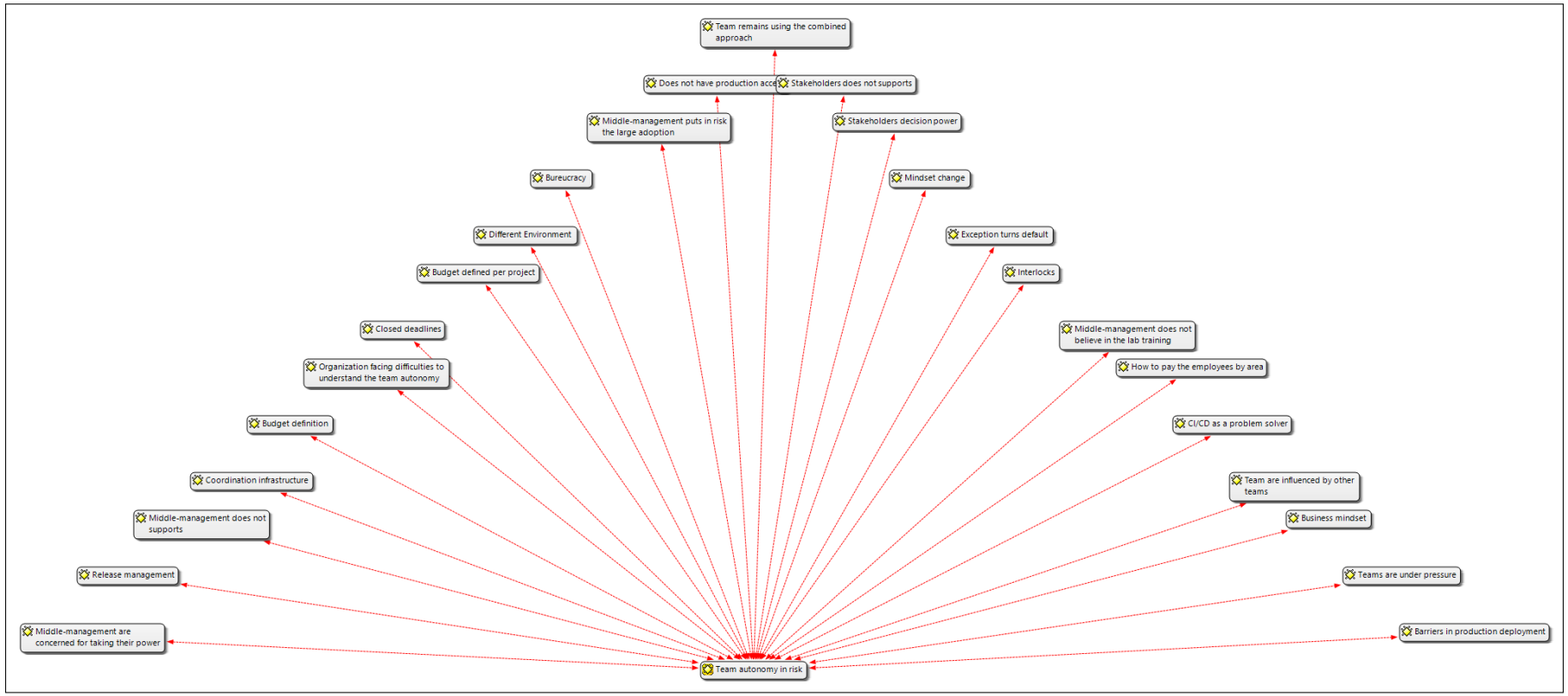
This code analysis corresponds to the third question of the interview script in Appendix E



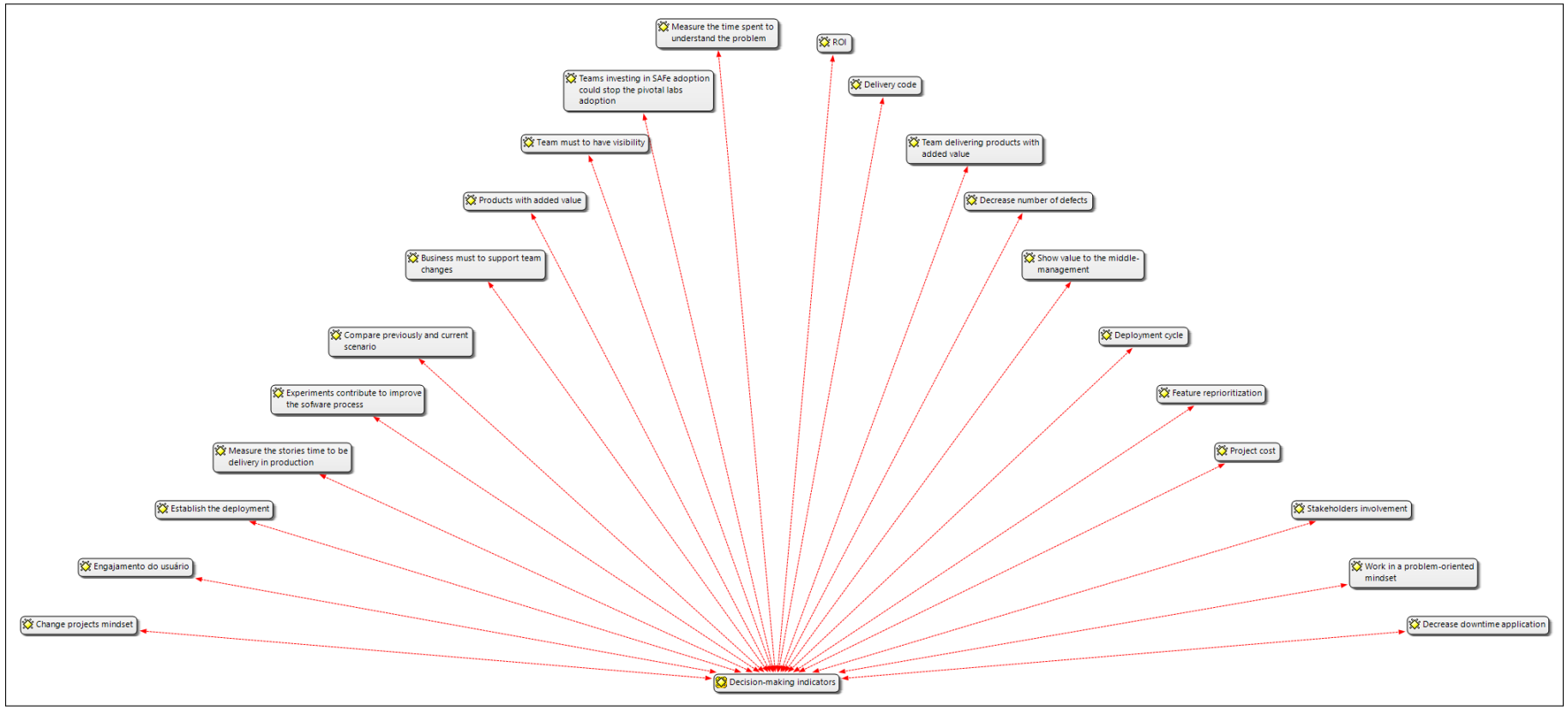
This code analysis corresponds to the fourth question of the interview script in Appendix E



This code analysis corresponds to the fifth question of the interview script in Appendix E

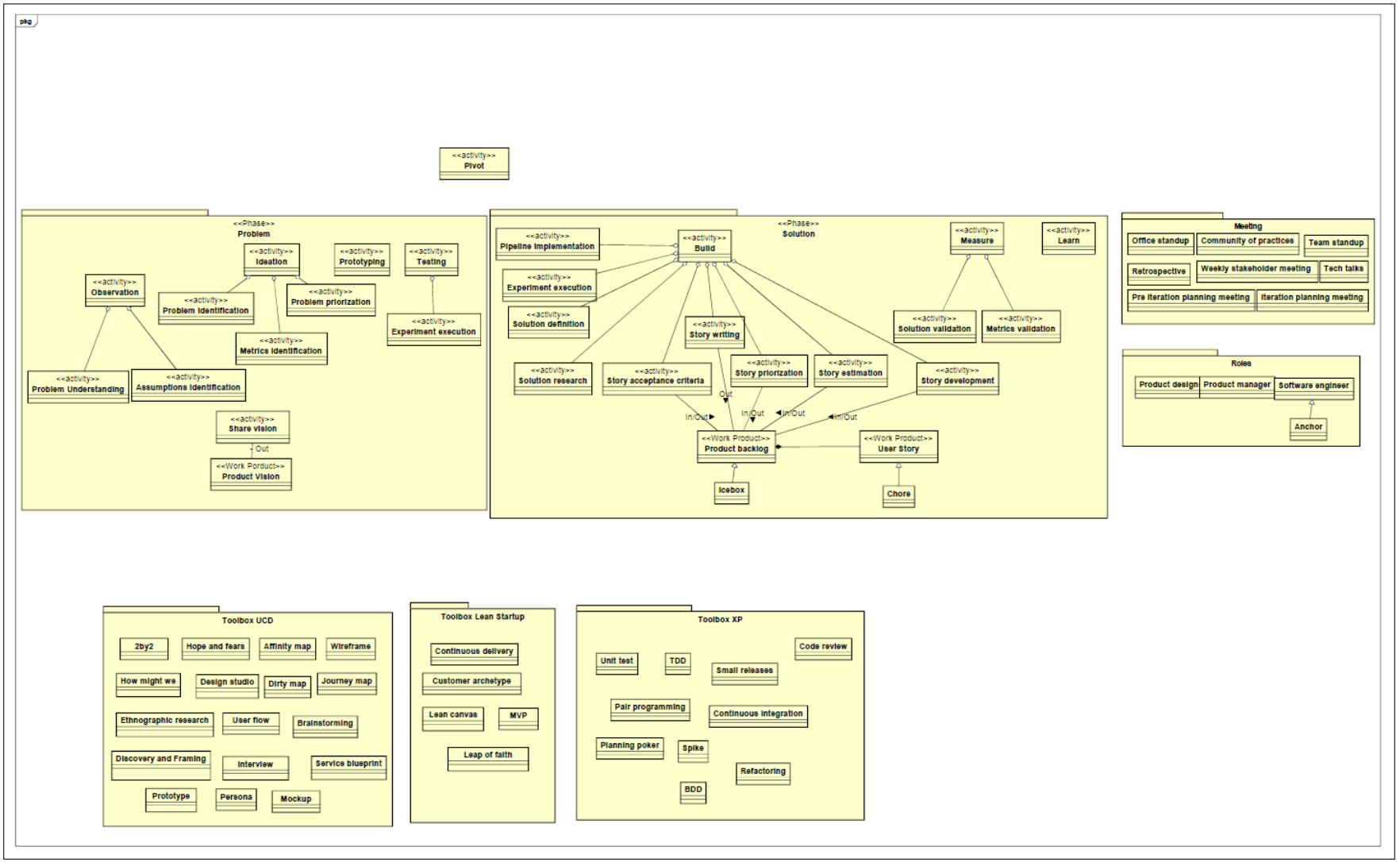


This code analysis corresponds to the sixth question of the interview script in Appendix E



This code analysis corresponds to the seventh question of the interview script in Appendix E

APPENDIX H – CONCEPTUAL MODEL INSTANCE



APPENDIX I – WORKFLOW ELEMENTS

Role

- Software Engineer
- Product Manager
- Product Designer

Phase

- Scoping
- Discovery and Framing
- Build-Measure-Learn
- Iteration

Activities

Scoping:

- Kick-off Meeting
- Scope Definition
- Stakeholder Meeting
- Team Meeting

Discovery and Framming:

- Build
  - Problem identification
  - Interview preparation
  - Interview conduction
  - Pain points identification
  - Problem definition
  - Persona creation
  - Assumption Creation
  - User activities mapping

  - Metric identification
  - Problem prioritization
  - Solution identification
  - Ideas generation
  - Ideas cluster
  - Solution prioritization
  - Solution prototyping
  - Solution definition
  - Story writing
  - MVP definition
  - Experiments execution
- Measure
  - User feedback
- Learn

Iteration:

- Build
  - CI/CD pipeline implementation
  - Story estimation
  - Story prioritization
  - Story development
- Measure
  - User Validation
  - Assumption Validation
  - Metric Validation
- Learn

Meetings

- Retrospective
- Weekly stakeholder meeting

- Iteration planning meeting
- Pre iteration planning meeting
- Team Standup
- Office Standup
- User interview
- Tech talks

Outcomes

- Product statement
- Problem strategy
- Problem vision
- Stakeholder map
- Interview script
- Problem statement refined
- Problem vision refined
- Persona
- Assumption list
- Business metric
- User metric
- User feedback
- Ideas
- Possible solutions
- Solution
- Prototype
  - Mockups
  - Wireframes
- Product Backlog

- User Story
- MVP Definition
- Prioritized stories
- Daily status
- Data
- Deliver

Values

- Feedback

Mindset

- Empathy
- Pro-activity
- Shared understanding
- User feedback

Techniques/Practices

- Affinity technique
- Topic mapping
- Brainstorming
- 2by2
- Interviews techniques
- Service blueprint
- User flow
- Journey map
- Ethnographic research
- We know, We’re right
- Now-near-next

- Design studio
- Continuous integration
- Continuous delivery
- Planning poker
- Pair programming
- Unit test
- BDD
- Code review
- Small releases