PUCRS

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

MAXIMILIAN NICHOLAS SCHMIDT ZORZETTI

A REFERENCE CONCEPTUAL MODEL FOR AGILE SOFTWARE DEVELOPMENT, LEAN STARTUP, AND USER-CENTERED DESIGN

Porto Alegre
2021

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

A REFERENCE CONCEPTUAL MODEL FOR AGILE SOFTWARE DEVELOPMENT, LEAN STARTUP, AND USER-CENTERED DESIGN

MAXIMILIAN NICHOLAS SCHMIDT ZORZETTI

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 dos Santos Marczak, PhD

Porto Alegre
2022

Ficha Catalográfica

Z88r Zorzetti, Maximilian Nicholas Schmidt

A Reference Conceptual Model for Agile Software Development, Lean Startup, and User-Centered Design / Maximilian Nicholas Schmidt Zorzetti. – 2021.
121.
Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientadora: Profa. Dra. Sabrina dos Santos Marczak.

1. Agile Software Development. 2. Lean Startup. 3. User-Centered Design. 4. Conceptual Model. 5. Software Engineering. 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

Maximilian Nicholas Schmidt Zorzetti

A Reference Conceptual Model for Agile Software Development, Lean Startup, and User-Centered Design

This Master 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.

Sanctioned on November 30, 2021.

COMMITTEE MEMBERS:

Prof. Dr. Tiago Silva da Silva (ICT/UNIFESP)

Prof. Dr. Afonso Henrique Corrêa Sales (PPGCC/PUCRS)

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

To anyone who has inspired me to strive for the best, which is to say, everyone.

“The way I see it, our fates appear to be intertwined.”
(Solaire of Astora)

ACKNOWLEDGMENTS

First, I'd like to acknowledge that this research is sponsored by Dell Brazil using incentives of the Brazilian Informatics Law (Law no. 8.2.48, year 1991). Sabrina Marczak thanks the financial support by CNPq (grant no. 307177/2018-1).

A huge thanks to my family for the absolutely gigantic mountain of relentless support. This work would've been close to impossible without it. Yes, I can finally take the dogs for a walk outside now and, no, I'm not staying up *that* late again.

Much love and thanks to my friends from the various facets of my life. Thanks to Matheus Viana for assisting me with styling, visuals, and laughter. Thanks to Felipe Nunes for inspiring me to sharpen up more and perhaps dabble in the occult. Thanks to Victor Diehl for creating wondrous worlds for me to escape into. Thanks to my gaming pals for helping me put off my academic chores and procrastinate as much as inhumanly possible. Thanks to my cat Fifi for being fluffy and not biting too hard. And thank you for reading this, seriously.

Finally, I'd like to thank the MunDDos research group for enabling everything in this research project and being great companions during graduate school. Thanks to Ingrid Manfrim for doing a lot of the heavy lifting and being a sweetheart. Thanks to Cassiano Moralles for his kind words and for being a thoughtful colleague. Thanks to Larissa Salerno for putting up with me all these years. Thanks to Cássio Trindade for being a good teacher, colleague, and friend throughout my stay at the university. Thanks to Michele Machado for hyping me up! Special thanks to Matheus Vaccaro, for kickstarting this whole thing with me and for being a class act. Thanks to Eliana Pereira for the many late night calls on modeling and general academia. Thanks to Ricardo Bastos, who impressively stands tall as a beacon of positivity at all times (how does he do it?). Last but definitely not least, a huge thanks to Sabrina Marczak for extensive advice on academia, professionalism, and life.

UM MODELO CONCEITUAL DE REFERÊNCIA PARA AGILE SOFTWARE DEVELOPMENT, LEAN STARTUP, E DESIGN CENTRADO NO USUÁRIO

RESUMO

[Contexto] O uso extenso de Desenvolvimento de Software Ágil nas duas últimas decadas revelaram suas fraquezas, com alguns estudos alegando-o como insuficiente para lidar com questões de negócio e satisfazer as necessidades do usuário. Casos recentes da indústria mostram que combinar Desenvolvimento de Software Ágil com Lean Startup e Design Centrado no Usuário é uma forma efetiva de amenizar essas défices, visto que esses dois tem o intuito de agregar valor para os *stakeholders* do negócio e de fomentar a empatia pelos usuários, respectivamente. No entanto, não existe uma fundamentação teórica sobre uma integração desses três pilares. [Objetivo] Este estudo cria tal fundamentação teórica através de um modelo conceitual que ilustra o que os pilares têm em comum e o que os diferencia uns dos outros. [Método] Foi feita uma Revisão Sistemática da Literatura para identificar um metamodelo que ilustrasse adequadamente como cada método complementa e suplementa os demais. Depois foi instanciado um modelo conceitual usando dados adquiridos a partir da análise de literatura chave dos três métodos e de um estudo de caso sobre dois times de desenvolvimento de software que usam uma abordagem que combina Extreme Programming, Lean Startup, e Design Centrado no Usuário. O modelo foi avaliado com os participantes do estudo de caso em uma workshop. [Resultados] O modelo identifica os princípios, os aspectos, e as práticas de cada pilar, expondo quais elementos são próprios de cada método e quais são compartilhados. No total, o modelo é composto por 12 princípios de Desenvolvimento de Software Ágil, 5 de Lean Startup, e 6 de Design Centrado no Usuário; que são derivados em 35 aspectos distintos, cada um com o suporte de um subconjunto de um total de 55 práticas. [Conclusão] Os achados ajudam a compreender como os métodos dão suporte uns aos outros, servindo como um alicerce para sustentar novas abordagens de desenvolvimento de software, justificando abordagens existentes, e habilitando a criação de recursos e instrumentos suplementares para essas abordagens.

Palavras-Chave: desenvolvimento de software ágil, Lean Startup, design centrado no usuário, modelo conceitual, engenharia de software.

A REFERENCE CONCEPTUAL MODEL FOR AGILE SOFTWARE DEVELOPMENT, LEAN STARTUP, AND USER-CENTERED DESIGN

ABSTRACT

[Context] The extensive use of Agile Software Development in the past two decades has unearthed its shortcomings, with some studies suggesting it as lackluster when tackling business-level issues and in addressing user needs. Recent industry cases show that combining Agile Software Development with Lean Startup and User-Centered Design is an effective way to remedy these failings, as the latter two focus on adding value to business stakeholders and fostering empathy towards the user, respectively. However, these three methods have considerable overlap, raising concerns on how each method stands apart from the others and making integration efforts difficult. A sound theoretical foundation of an integration of these methods could remedy such concerns. [Objective] This study develops such a foundation, by means of developing a conceptual model that illustrates what makes each method unique and what they have in common. [Method] We conducted a systematic literature review to select a metamodel that could best illustrate how each method complements and supplements one another. We then instantiated a conceptual model from it using data acquired from analyzing core literature from the three methods and from a case study on two software development teams that use a combined approach of Extreme Programming, Lean Startup, and User-Centered Design. Finally, we evaluated our model with the case study participants with a thorough workshop. [Results] The model identifies the principles, features, and practices of each method, showcasing which elements are unique to and shared between them. In total, the model comprises 12 principles for Agile Software Development, 5 for Lean Startup, and 6 for User-Centered Design; which are derived into 35 distinct features, each supported by a subset of a total of 55 practices. [Conclusion] Our findings provide insight on how these pillars support one another, serving as a foundation that will support new software development approaches, justify existing ones, and enable the development of supplementary instruments and resources to such approaches.

Keywords: agile software development, Lean Startup, user-centered design, conceptual model, software engineering.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ASD – Agile Software Development

BDD – Behavior-Driven Development

BML – Build-Measure-Learn

HCD – Human-Centered Design

IT – Information Technology

LS – Lean Startup

MVP – Minimum Viable Product

TDD – Test-Driven Development

UCD – User-Centered Design

XP – Extreme Programming

CONTENTS

1. INTRODUCTION

Agile Software Development (ASD) is extensively used by organizations today [79] as it is a great fit for the modern world, which is one of constant change and unpredictability [5]. While it continues to be a powerful and adaptive approach to software development, its use in the past two decades since its inception revealed some of its shortcomings, such as insufficient customer engagement [67] and difficulty in developing the right solution for the right problem [25]. Some authors state that Agile alone is not enough to tackle business-level issues [82] and that it is evolving into something greater, as current practices are not fast enough to support the desired feedback speed of product designers [15].

Although there are efforts to improve Agile itself [14], there are those who propose combining it with something that compensate its weaknesses. It has been suggested that integrating ASD with User-Centered Design (UCD) is highly beneficial [13], as it enables developers to be closer to the user throughout development, thus ensuring that their needs are met [18]. Others advocate the incorporation of the continuous experimentation practices of Lean Startup (LS) to identify relevant problems that deliver value when solved [23]. Thus, it stands to reason, and has been demonstrated as such, that a combination of ASD, LS, and UCD elements (hereinafter referred to as "combined approach") is a great way to drive development [19, 74, 85] and mitigate the risks inherent with the innovation that the software industry pursues in its ever-increasing business-disrupting goals [26].

Motivated by the perceived effectiveness of this development approach, our research group is working on assistance tools to help development teams and organizations who wish to transition to such an approach. To develop these resources, we need a sound basis for what an integration of the aforementioned "pillars" should look like. However, such a foundation is not immediately evident, as the pillars themselves do not have a single, widely accepted theoretical basis, which makes integration efforts difficult. Moreover, the difficulty in conceptualizing such an integration is exacerbated by obstacles such as pillars handling the same issues differently or even untangling a pillar from one another, which is the case for ASD and LS, as both have their roots in Lean manufacturing [84].

1.1 Research Goal

As of today, the combined approach stands on some "shaky" foundations: LS is a novel method that is not yet fully understood in software development, and integrating UCD with ASD is an on-going research effort. This thesis' main goal is to highlight how the pillars of the combined approach complement and supplement one another and to provide a foundation for future research endeavors on the combined approach. Given the context

of our research project (which has focused on modeling the combined approach in several forms) we decided on developing a conceptual model to better outline the similarities and differences between the pillars. In sum, the pragmatic goal of this thesis is to develop a conceptual model of an integration of ASD, LS, and UCD. Hence, we seek to answer the following research question:

RQ1. How can Agile Software Development, Lean Startup, and User-Centered Design be conceptualized in one integrated model?

Additionally, we use the following two research questions to guide our study:

RQ2. What concepts are shared between Agile Software Development, Lean Startup, and User-Centered Design?

RQ3. What concepts are unique to Agile Software Development, Lean Startup, and User-Centered Design?

1.2 Research Design

This study is of a qualitative nature. Corbin and Strauss [17] define qualitative research as one that seeks to understand experiences, behaviors, and feelings as they occur in an organizational context, in social movements, or as part of culture phenomena. We outline how each step of our study fuels the next and ultimately concludes into our conceptual model next. See Figure 1.1 for a visual summary.

Systematic Literature Review

We conducted a systematic literature review to outline existing conceptual models of ASD, LS, and UCD with the intent of finding a model that is broad enough or has the necessary mechanisms to be adapted into a conceptual model for the combined approach.

Conceptual Model Development

We established a metamodel derived from existing literature and partially populated it with concepts derived from authoritative sources of ASD, LS, and UCD, thus developing a preliminary version of the conceptual model grounded on literature.

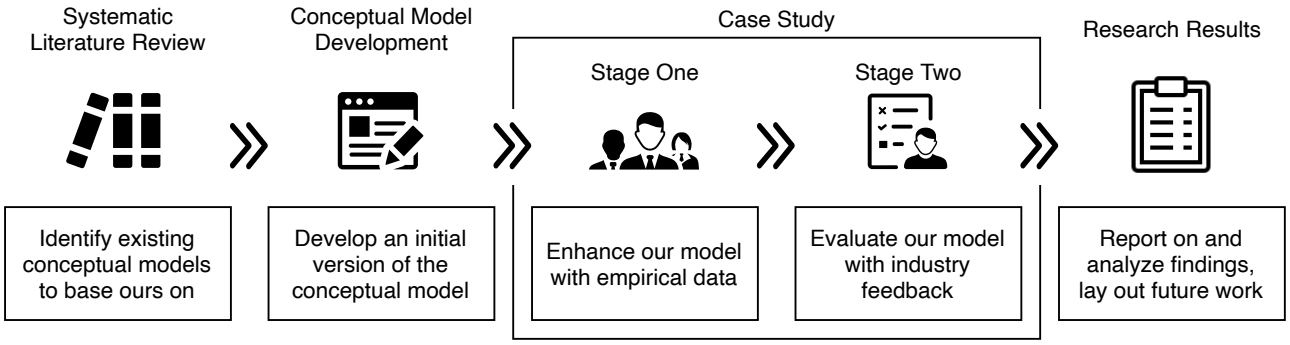


Figure 1.1: Research design

Case Study

Taking advantage of our ongoing research partnership with an organization that utilizes the combined approach, we conducted a case study to obtain the unfulfilled concepts of our preliminary model, thus obtaining all components needed to complete the model. Afterwards, we ran a survey and a focus group within the case study to evaluate both the concepts previously sourced from existing literature and from the case study itself.

Research Results

Finally, we report on the finalized conceptual model, which is ultimately grounded on literature and practice-sourced concepts and evaluated empirically. We discuss the model and its place in the larger agile movement.

1.3 Contribution

This thesis contributes to the literature with a literature- and practice-grounded and industry-evaluated conceptual model of a combined development approach of ASD, LS, and UCD; furthering existing work on the understanding of it [73]. The model itself can be used in several ways, such as by instantiating process models or verifying what concepts a given combined approach team uses; in addition to providing a solid knowledge framework to understanding the combined approach. As part of the research project this thesis is a part of, a future use of the conceptual model will be that of basing the development of supportive resources on it, as is the case of a combined approach adoption acceleration model proposed by Moralles et al. [47] and an evaluation model proposed by Machado et al. [44].

1.4 Publications

The following publications written by the author served as input for this thesis:

- *On the Mapping of Underlying Concepts of a Combined Use of Lean and User-Centered Design with Agile Development: The Case Study of the Transformation Process of an IT Company* [46]

  Authors: Cassiano Moralles, Matheus Vaccaro, **Maximilian Zorzetti**, Eliana Pereira, Cássio Trindade, Bruna Prauchner, Sabrina Marczak, and Ricardo Bastos.

  Brazilian Workshop on Agile Methods (WBMA), 2019. Best paper award. Qualis: B4.

  Our first step in this thesis was in identifying the underlying concepts involved in the use of ASD, LS, Lean Software Development, and UCD. We conducted multiple literature reviews to outline the concepts of each approach followed by an empirical study that sought to identify which concepts were considered useful by two software teams of a multinational IT organization. This particular study revealed that there are concepts from literature not yet considered in practice and also the other way around—there are practiced concepts not yet found in literature.

- *A Practice-informed Conceptual Model for a Combined Approach of Agile, User-Centered Design, and Lean Startup* [92]

  Authors: **Maximilian Zorzetti**, Ingrid Signoretti, Eliana Pereira, Larissa Salerno, Cassiano Moralles, Cássio Trindade, Michele Machado, Ricardo Bastos, and Sabrina Marczak.

  International Conference on Product-Focused Software Process Improvement (PROFES), 2020. Qualis: B1.

  In this study, we aimed to showcase an initial conceptual model based on the empirical study of two software development teams that use the combined approach. We performed a case study where we investigated their day-to-day work using daily observations, semi-structured interviews, and focus group sessions; and built a conceptual model of the activities, techniques, and work products that both teams used daily. We reflected on augmenting this model with concepts sourced from literature, an idea which eventually became the basis for this thesis.

- *An Empirical-informed Work Process Model for a Combined Approach of Agile, User-Centered Design, and Lean Startup* [91]

  This study reported on a process model that reflects how two software development teams that use the combined approach do their work. The process model was created following a case study where we investigated the participants' day-to-day work, which

resulted in a mapping of their activities, techniques, and work products [92]; which they used to visually represent their work process in a workshop. The result was a software development process model expressed in Business Process Model and Notation that encompasses ASD, LS, and UCD concepts; which can be used as a starting point for industry practitioners who want to adhere to such a development approach.

- *Improving Agile Software Development using User-Centered Design and Lean Startup* [93]

  Authors: **Maximilian Zorzetti**, Ingrid Signoretti, Larissa Salerno, Sabrina Marczak, and Ricardo Bastos.

  Information and Software Technology (IST), 2022. Qualis: A1.

  Following the more formalized and rigid viewpoint of the previous study, we took a step back to report on the combined approach as a whole. The study examines the case study in its entirety to present an overarching report of several of its aspects so as to fully characterize it. We describe how the combined approach promotes a problem-oriented mindset, encouraging team members to work together and engage with the entire development process, actively discovering stakeholders needs and how to fulfill them; while each of its pillars provide unique contributions to the development process.

Although not directly related to this thesis, the author has also written or significantly contributed to the following work on the combined approach:

- *On the Development of a Model to Support the Combined Use of Agile Software Development with User-Centered Design and Lean Startup* [47]

  Authors: Cassiano Moralles, **Maximilian Zorzetti**, Ingrid Signoretti, Eliana Pereira, Matheus Vaccaro, Bruna Prauchner, Larissa Salerno, Cássio Trindade, Sabrina Marczak, and Ricardo Bastos.

  European Conference on Systems, Software and Services Process Improvement (EuroSPI), 2020. Qualis: B2.

- *On the Understanding of Experimentation Usage in Light of Lean Startup in Software Development Context* [81]

  Authors: Bruna Vargas, Ingrid Signoretti, **Maximilian Zorzetti**, Sabrina Marczak, and Ricardo Bastos.

  Evaluation and Assessment in Software Engineering (EASE), 2020. Qualis: A3.

- *Success and Failure Factors for Adopting a Combined Approach: A Case Study of Two Software Development Teams* [76]

  Authors: Ingrid Signoretti, **Maximilian Zorzetti**, Larissa Salerno, Cassiano Moralles, Eliana Pereira, Cássio Trindade, Sabrina Marczak, and Ricardo Bastos.

International Conference on Product-Focused Software Process Improvement (PROFES), 2020. Qualis: B1.

- *Maturity Models for Agile, Lean Startup, and User-Centered Design in Software Engineering: A Combined Systematic Literature Mapping* [94]

  Authors: **Maximilian Zorzetti**, Matheus Vaccaro, Cassiano Moralles, Bruna Prauchner, Ingrid Signoretti, Eliana Pereira, Larissa Salerno, Ricardo Bastos, and Sabrina Marczak.

  International Conference on Enterprise Information Systems (ICEIS), 2020. Qualis: A3.

- *Adopting Agile Software Development Combined with User-Centered Design and Lean Startup: A Systematic Literature Review on Maturity Models* [90]

  Authors: **Maximilian Zorzetti**, Cassiano Moralles, Larissa Salerno, Eliana Pereira, Sabrina Marczak, and Ricardo Bastos.

  International Conference on Enterprise Information Systems (ICEIS) (Revised Selected Papers), 2021. Qualis: A3.

Additionally, the following study is currently in development:

- *Processes for Agile Software Development, User-Centered Design, and Lean Startup: A Systematic Literature Review*

  Authors: **Maximilian Zorzetti**, Ingrid Signoretti, Larissa Salerno, Cassiano Moralles, Sabrina Marczak, and Ricardo Bastos.

  To be submitted to the journal of Information and Software Technology (IST).

  We conducted a systematic literature review of studies that present an integrated approach of ASD, LS, and UCD; mapping this subject area and analyzing the development approaches we found in regards to their context of use and the benefits and challenges of their adoption.

1.5 Thesis Structure

This document is structured as follows. Chapter 2 presents the background that supports this research. Chapter 3 presents a systematic literature review to kickstart the research process. Chapter 4 defines the building blocks of the conceptual model. Chapter 5 reports on a case study used to develop and evaluate the model. Chapter 6 presents the conceptual model itself. Chapter 7 discusses the conceptual model. Finally, Chapter 8 concludes this thesis, examines limitations, and considers future work.

2. BACKGROUND

2.1 Agile Software Development

Traditionally, software development was conducted in a methodical and rigid manner akin to other engineering disciplines [77]. As software evolved, so did the demand for it, which became increasingly prone to change in order to match the unpredictability [5] caused by constant advancements in technology. The traditional development method (i.e., waterfall [62]) became unsuitable to this changeful environment and failed in several ways (e.g., cost and time overruns), wasting billions of dollars in project cancellations alone in the USA during the early 1990s and possibly trillions in opportunity costs [78].

As change became a constant during software development, several development methods (e.g., Extreme Programming (XP) [7], Feature-Driven Design [59], and Scrum [68]) were created in response to accommodate such changes, dismissing the traditional "frozen design" model and embracing an iterative model instead. In 2001, the Agile Manifesto [8] streamlined the wisdom found in these approaches into the following values:

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> **Individuals and interactions** over processes and tools
>
> **Working software** over comprehensive documentation
>
> **Customer collaboration** over contract negotiation
>
> **Responding to change** over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more.
>
> *Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas*
>
> *© 2001, the above authors. This declaration may be freely copied in any form, but only in its entirety through this notice.*

The manifesto defined the foundations for the Agile movement, along with a set of 12 principles that materialize the aforementioned values. Thus, agility is defined as the embodiment of these values, such as efficient adaptability to change, team structures and attitudes that facilitate communication, empowered self-organizing teams, emphasis on rapid delivery of working products, and adoption of the customer as part of the team [62]. As such, agile methodologies are software development methods that incorporate such things

into their processes. Common practices among agile methodologies are short development cycles, daily meetings to synchronize developer knowledge, and meetings to take a step back and analyze the development process so far to look for possible improvements.

ASD promotes several advantages over traditional development and has achieved widespread use as of today [79], in spite of difficulties in adopting it in a large enterprise context [58]. Remarkably, its core values have shown to be beneficial even outside the software development domain [70].

2.2 Lean Startup

Lean Startup is an entrepreneurship method that focuses on iteratively developing a viable business plan for startup companies—"human institution[s] designed to create a new product or service under conditions of extreme uncertainty" [63]. It provides tools to identify what should and should not be developed, thus reducing development "waste". The focus on waste reduction is a core value of its ancestor, Lean manufacturing, which aims to reduce any kind of waste found in the production process, be it in the product's design phase, in stocking warehouses, in task management among workers, or really in any step of the product's life cycle that delays it from getting to the consumer's hands [84]

While LS is not concerned with production per se, it instead focuses on learning what should be produced at all. The method consists in acquiring customer feedback on an idea and using it to strategize the startup's next moves. To avoid development waste on a given idea, it promotes the use of minimum viable products (MVP): products with a small feature set that aim to sell the vision of the final product and attract early adopters [9]. MVPs are tested directly in the market, so as to acquire meaningful, "valid" data from real consumers. The method achieves this through its primary activity, the build-measure-learn (BML) cycle (see Figure 2.1), in which experiments (e.g., MVPs) are built to measure how customers respond to an idea (e.g., a product), enabling the startup to learn what works on the market and confidently persevere on the idea or pivot to another one entirely [63]. LS defines the following five principles for successful entrepreneurship:

1. Entrepreneurs are everywhere;
2. Entrepreneurship is management;
3. Validated learning;
4. Innovation accounting; and
5. Build-measure-learn.

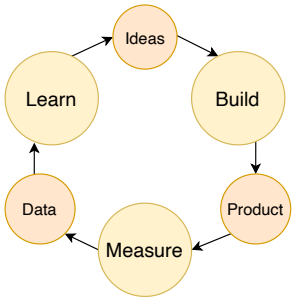


Figure 2.1: The build-measure-learn cycle
Adapted from Ries [63]

Principles 3 to 5 are encapsulated in the process just described. *Entrepreneurship is management* states that a startup organization is not to be managed like a regular organization would, as the metrics used in one are “incompatible” with the chaotic environment startups find themselves in [65]. *Entrepreneurs are everywhere* asserts that not only startups deal with environments of extreme uncertainty—existing organizations of any age or size can become tangled in one, especially if pursuing innovation [64].

The method has been reported as a way for software companies to innovate [22], although it demands technological capabilities (e.g., continuous deployment) and organizational support (e.g., culture) to enable its continuous experimentation approach [43]. Although not specifically a software development method, it follows the idea of lightweight processes (via the build-measure-learn cycle) and studies have reported on it being a great driving force when developing software [23, 86].

2.3 User-Centered Design

User-Centered Design is an umbrella term for design philosophies, processes, and practices that focus on centering the user at the heart of the design space with the goal of achieving user satisfaction through products that fulfill their needs and goals [66]. The intensity of user involvement varies, from consultation of their needs and participation in usability testing, to having the user actively participate in the design process [1]. Nevertheless, solutions devised using UCD are generally regarded as having improved usefulness, usability, and user satisfaction [83], as it enables developers to understand their anticipated users’ real needs [66]. Don Norman, founding figure of UCD [53], defines four UCD principles [50]:

- Understand and address the core problems;
- Be people-centered;
- Use an activity-centered systems approach; and
- Use rapid iterations of prototyping and testing.

The term originated in the 1980s [53] and has since evolved into Human-Centered Design (HCD) [29], which is mostly the same[1], as the rebranding was made with the explicit intent of humanizing users[2], rather than due to a change of values or similar concerns. Nevertheless, UCD has spawned several design processes, such as Design Thinking [11, 12] and the Double Diamond [10], the last of which encompasses two working spaces—one for exploring an issue more widely or deeply (divergent thinking) and one for taking focused action based on the learning from the previous phase (convergent thinking). Regardless of processes specifics, UCD generally suggests a number of design principles, such as an earlier focus on the user during development, as well as incorporating tasks and measurements into an iterative design process [66].

Under the name of HCD, the approach has principles and activities defined as an ISO standard "...concerned with ways in which both hardware and software components of interactive systems can enhance human-system interaction" [31]. The standard defines four main activities that take place during design: understand and specify the context of use; specify the user requirements; produce design solutions; and evaluate the design. Figure 2.2 shows the interdependence of HCD activities. Although the standard explicitly states that it does prescribe a proper design process nor does it describe all necessary activities to ensure effective design, it highlights the several benefits brought by using human-centered methods, such as increased user productivity and organizational efficiency, improved user experience, reduced training and support costs, better sustainability, reduced discomfort and stress, and the increase of competitive edge.

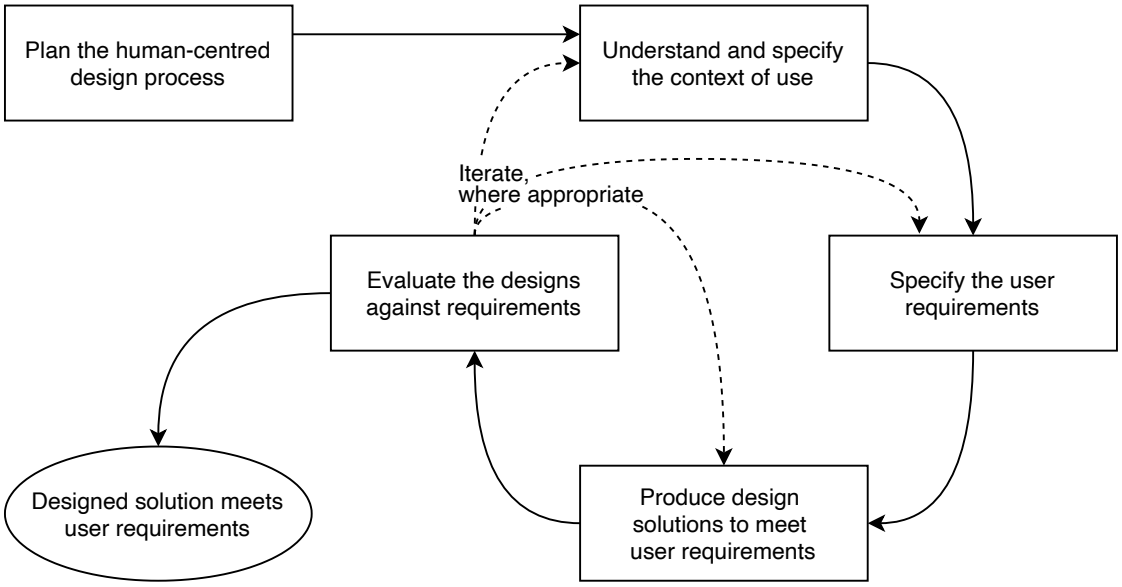


Figure 2.2: Interdependence of HCD activities
Adapted from ISO 9241-210 [31]

[1]UCD originally defined four phases to designing: observation, ideation, prototyping, and testing [52]. IDEO's HCD guide defines only three: inspiration, ideation, and implementation [29].

[2]Nowadays, Don Norman advocates the term People-Centered Design [51].

2.4 A Combined Approach of Agile Software Development, Lean Startup, and User-Centered Design

As Agile was extensively used in the past two decades, its shortcomings became more apparent [67]. Some authors argue that ASD alone is not enough to tackle business-level issues [82] and that it provides insufficient customer involvement [6]. Combining agile methods with other approaches to development has been suggested as a way to fix these issues [82], and combining them specifically with LS and UCD has shown great promise as it seemingly increases the engagement of business stakeholders and end users while also enabling rapid experimentation, among other benefits [74, 81].

The three pillars deal with different aspects of product development, namely design (UCD), product management (LS), and development itself (ASD). Notably, combining any of the three pillars can be quite challenging despite their many similarities and ways they can complement one another. For instance, even though agile methods in general have been criticized as being lackluster on the process of devising an initial design [54], merging them with UCD practices can be difficult due to the former's focus on getting working software quickly and the latter's needs for upfront design planning. These incompatibilities can span to a more philosophical and principled level as well, such as with the generalist views present in ASD in contrast to the specialist views of UCD [66], or LS's quantitative methods [63] and UCD's qualitative approach [52].

An example of a development approach that combines the three pillars is Pivotal Labs[3], which proposes principles and ceremonies based on the three pillars, with XP as its agile method. Pivotal Labs' main goal is to assist teams in building software products that deliver meaningful value for users and their business, offering a framework and starting point for any team to discuss its needs and define its own path towards software development. To do so, it suggests the adoption of a cross-functional team composed of three main roles: Product Designer, Product Manager, and Software Engineer; the latter specialized in an Anchor role, responsible for bridging the engineers with users and business stakeholders. It also proposes that the team finds its own work "rhythm" by collectively deciding on which ceremonies and workflows derived from the three pillars to adopt, revisiting them constantly. Pivotal Labs does not provide a prescriptive set of work practices (which would go against the Agile philosophy); rather, it suggests the following principles: that teams should learn by doing, work in a co-located fashion in order to facilitate coordination and fast feedback loops among team members, promote constant collaboration, take the leadership and own the product development cycle, and find their own sustainable pace having in mind that product development is a "marathon" and not a sprint.

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

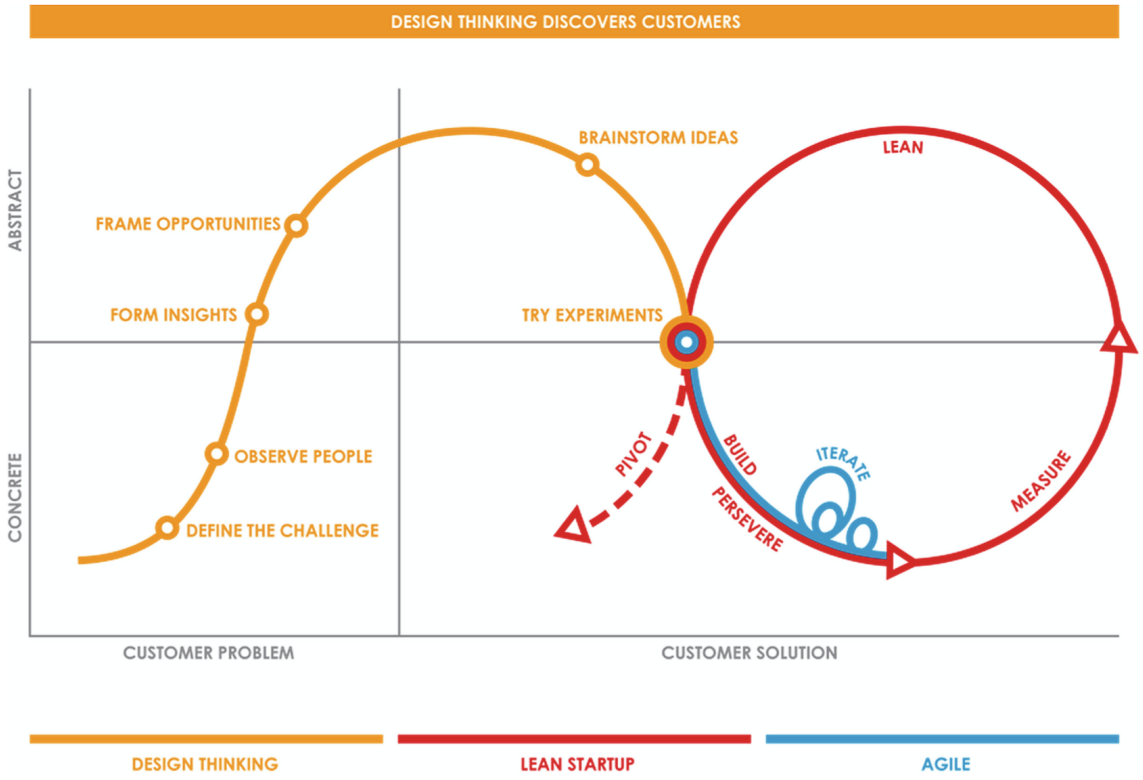


Figure 2.3: Discovery by Design™
Reproduced from the work of Grossman-Kahn and Rosensweig [26]

Another example of the referred approach is fashion retailer Nordstrom's Discovery by Design™ [26]. The creation of this multidisciplinary innovation approach was undertaken in an iterative and "organic" fashion by a dedicated innovation team by combining ASD with Lean Manufacturing, LS, and Design Thinking. Spearheaded by an employee passionate about ASD and Lean Manufacturing principles, the innovation team fully embraced the principles of the Agile Manifesto [8] and started to prototype new ideas, but soon found that their efforts failed to gain traction inside Nordstrom, as Agile and Lean Manufacturing lacked the tools to validate ideas before handing them over to the business. By adopting the LS method afterwards, the team turned itself into a successful "discovery vehicle" that rapidly tested and validated new ideas by means of measuring customer demand, albeit with their products seeming a little too much "business-centric". Given their positive experiences with interacting with end users during this iteration of their development method, they eventually decided to incorporate Design Thinking [29] into it in addition to adopting a more risk-tolerant attitude, completing their method into what they called Discovery by Design™. The team with its "iterative mindset, relentless focus on the needs of the customer, and bias towards rapid experimentation, prototyping and testing" [26] emerged as a successful dynamic capability [80] for Nordstrom. The Nordstrom case is a good example of how the combined approach is successful in supporting software teams in keeping up with market and user needs, providing a solid source of evidence to its success.

2.5 Conceptual Modeling

A conceptual model is an abstraction expressed in a semi-formal or formal language that uses artifices such as hierarchy to facilitate human understanding of complex phenomena [38]. Modeling is an inherently subjective activity, regardless if the phenomena being modeled has a real-world equivalent, thus, it is acceptable to allow distinct models of the same thing [38]. As a consequence, constructs in modeling languages influence the interpretation of phenomena and eventually the modeler's perspective itself [38]. Models also allow for quick interpretation if the audience is already familiar with them in some fashion, typically by already knowing their language/notation or even their metamodel. A metamodel is a higher-abstraction model that defines the semantics of other models and serves as a modeling framework—essentially, a metamodel is "the model of a model".

Modeling languages and/or notations vary greatly in their purpose and form. For instance, Business Process Modeling and Notation (BPMN) [55] is a notation created with the express purpose of being easily and readily understandable by all business users, assisting in bridging the gap between process modeling and process implementation. BPMN arose from the consolidation of best practices and ideas from several other business notations, ultimately providing three possible frames of reference for business: choreography diagrams, collaboration diagrams, and process diagrams.

Another widespread notation is the Unified Modeling Language (UML) [56]. UML was designed as a general-purpose language capable of modeling both software systems and business processes, providing a total of fourteen diagram types—one half specializing in modeling static structures, the other in dynamic behaviors. For the purposes of this thesis, we highlight the class diagram [56], which focuses on depicting the static structure of a system by means of describing its elements by detailing their attributes, their operations, and how they relate to one another. We illustrate its core mechanisms used in this thesis next.

Consider Figure 2.4, which depicts a simple class diagram with two classes: *Idea* and *Theory*. The named arrow indicates an association between the two concepts. Further describing this association are the notations *1...** (one or more) and *0...** (zero or more), which refer to the association's *multiplicity*—the cardinality of the instances of each class in the association. Taking these mechanisms into account, the diagram can be read as "An *Idea* is part of zero or more *Theories*, and a *Theory* has at least one *Idea* that is part of it". While UML is much more complex, these few mechanisms are sufficient for this thesis.

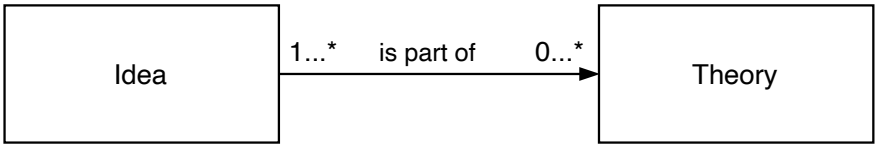


Figure 2.4: Example UML class diagram

3. SYSTEMATIC LITERATURE REVIEW

This chapter describes a systematic literature review on conceptual models for ASD, LS, UCD, and their intersections. The expected result of this review is to find an adequate modeling schema to model the pillars in a way that our research questions can be satisfied. We performed this review following the guidelines on systematic literature reviews in software engineering by Kitchenham and Charters [33].

This chapter is organized as follows. Section 3.1 defines the research question answered by the systematic review. Section 3.2 details the search string and databases used. Section 3.3 details study selection criteria. Section 3.4 details a quality assessment procedure applied on selected studies. And Section 3.5 presents the review's findings.

3.1 Research Question

The goal of this literature review is to answer the following research question:

SR1. What are the conceptual models available for Agile Software Development, Lean Startup, User-Centered Design, and their intersections?

3.2 Search Process

As suggested by the guidelines, we use the PICO criteria to guide the formulation of our search string. We decided to exclude the *Intervention* and *Comparison* criteria as the former restricts our search too much (as there are several approaches to modeling) and the latter is not relevant to the purposes of this systematic literature review.

Population: ASD, LS, and UCD.

Outcomes: Conceptual models, dictionaries, glossaries, or mind maps.

For database selection, we use the following criteria:

- Databases that include journal articles, conference, and workshop papers related to software engineering;
- Databases with an advanced search mechanism that allows filtering of the results by keywords that address the research questions; and

- Databases that provide access to full papers written in English.

Based on these criteria, we used the following databases to retrieve studies from: ACM Digital Library, IEEExplore, Science Direct, Scopus, and Springer Database. The search string (Equation 3.1) was adapted for use in each database based on the search functionality offered by each given database. Each search string consists of two parts—S1 and S2—defined as follows:

- S1 is a logical disjunction of keywords related to Agile Software Development, Lean Startup, and User-Centered Design: Agile Software Development; Lean Startup, Continuous Experimentation, and Experiment-driven Software Development; User-Centered Design and Human-Centered Design
- S2 is a logical disjunction of keywords related to conceptualization, namely: conceptual model, dictionary, glossary, or mind map;

Equation 3.1 – Search criteria boolean expression.

$$S1 \text{ AND } S2 \tag{3.1}$$

See Appendix B for the search strings as written in each database.

3.3 Study Selection

Studies were selected by applying the following inclusion and exclusion criteria:

Inclusion criteria: (I1) the study presents a conceptualization for Agile Software Development, Lean Startup, UCD, or their intersections; (I2) the study is written in English; (I3) the study is fully written in electronic format; (I4) the study was retrieved from a conference, workshop, or journal.

Exclusion criteria: (E1) the study does not pertain to software engineering; (E2) the study is an extended abstract or editorial paper; (E3) the study is duplicated.

We performed two rounds of selection. The first consisted of analyzing paper metadata (title, abstract, and keywords) while the second consisted of the full reading of the papers accepted in the previous round.

3.4 Quality Assessment

We used a set of quality criteria to assess the methodological quality of the studies selected for review and to serve as the final selection filter. We adapted criteria from the work of Guyatt et al. [27] as they cover thoroughness, trustworthiness, and significance of the studies [30]. The criteria are based on four quality assessment questions:

C1. Is the research objective clearly defined?

C2. Is the research context well addressed?

C3. Are the findings clearly stated?

C4. Based on the findings, how valuable is the research?

We graded the selected studies on each criterion using a 3-point ordinal scale instead of a dichotomous one to obtain a more accurate assessment [30]. We doubled the weight of criterion **C4** as it was our main mechanism for deciding whether the schema presented in each study was adequate for our modeling purposes. We only considered and analyzed studies with scores higher than 0.8 (inclusive). See Appendix B for the complete quality assessment data.

3.5 Findings

Table 3.1 shows the number of selected studies in the selection and quality assessment stages of the search. We describe the studies selected next.

Table 3.1: Studies selected in the systematic search

Database	**Retrieved**	**Round 1**		**Round 2**		**Qlty. Ast.**	
		Excl.	**Incl.**	**Excl.**	**Incl.**	**Excl.**	**Incl.**
ACM Digital Library	158	131	27	23	4	4	0
IEEEXplore	132	122	10	10	0	—	—
Science Direct	30	30	0	—	—	—	—
Scopus	260	242	18	14	4	3	1
Springer Database	1363	1314	49	36	13	12	1
Total	**1943**	**1839**	**104**	**82**	**21**	**19**	**2**

3.5.1 **SR1.** Existing Conceptual Models

Conceptual Framework of Agility

Conboy and Fitzgerald [16] try and define what exactly is agility, since several agile methods proclaim to be agile whilst sometimes making contradictory use of the Agile Manifesto [8] by focusing too much on some principles at the expense of others. To that end, the authors study the roots of ASD and 30 years of software development literature, untangling the concepts of flexibility, leanness, and agility. The authors settle on the following description for agility:

> *The continual readiness of an entity to rapidly or inherently, proactively or reactively, embrace change, through high quality, simplistic, economical components and relationships with its environment.*

Conboy and Fitzgerald conclude the study by developing a conceptual framework of the agility process (see Figure 3.1), which maps four aspects of agility: drivers, strategy, capabilities, and providers. Each aspect is fulfilled by a series of concepts which describe agility in the given aspect—such as the importance of *simplicity* as a capability.

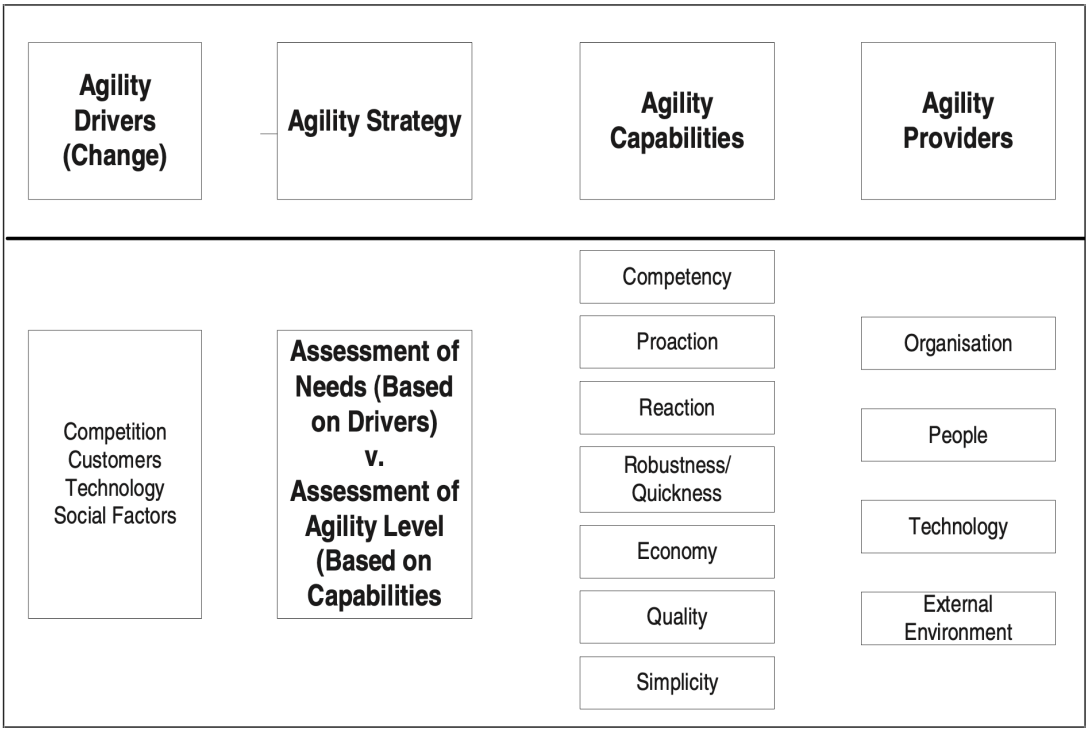


Figure 3.1: Conceptual framework of agility [16]

Agile Conceptual Model

Kiv et al. [34] propose a generic goal-oriented conceptual metamodel to assist in tailoring agile methods. Their work focuses on the original goals of the Agile Manifesto [8], instead of similar tailoring approaches that target business goals. The metamodel is based on UML and contemplates two levels of agile tailoring: a tactical level, in which teams choose practices to integrate into their process, and an operational level, in which “agile practices are implemented” [34]. Each level consists of a number of concepts, as seen in Figure 3.2. Of note is the *Agile Feature* concept, a “distinctive characteristic of [a] principle”, which serves to narrow the gap between principles and practices.

To fulfill each class of concepts, the authors map the Agile Manifesto [8] to the tactical-only portion of the metamodel, assigning principles to specific values and using the work of Laanti et al. [41] as the basis for the agile features of each principle; and Scrum [68] and XP [7] to the operational level, finalizing their conceptual model. The authors proceed to translate their model to the i* framework [88] to further their efforts in agile methods tailoring.

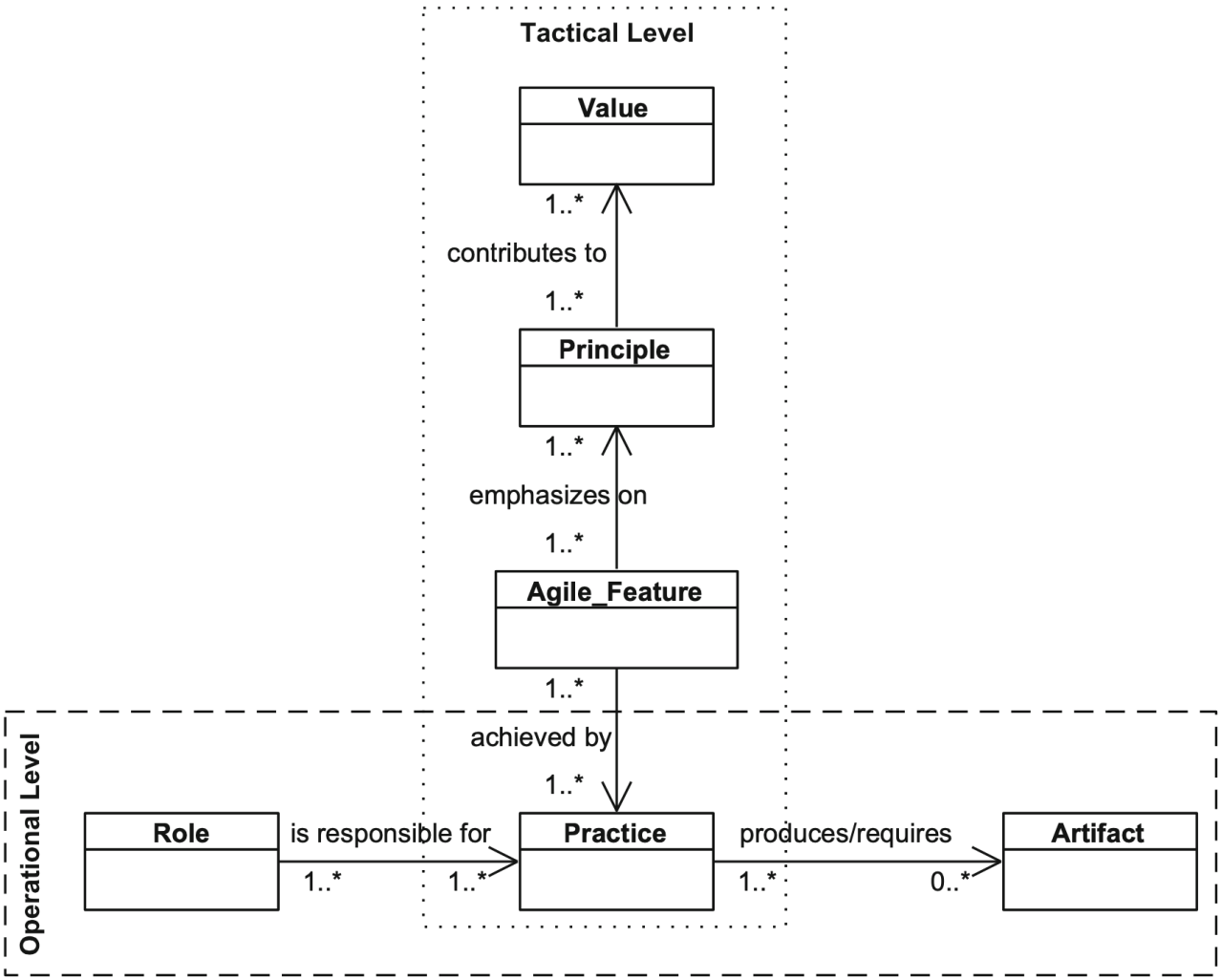


Figure 3.2: Goal-oriented agile metamodel [34]

4. DEVELOPING THE CONCEPTUAL MODEL

This chapter is organized as follows. Section 4.1 outlines the chosen metamodel and modeling strategy. Section 4.2 details the modeling effort for LS. Section 4.3 details the modeling effort for UCD. And Section 4.4 presents definitions for the key concept of *Feature*.

4.1 Metamodel

We chose the metamodel developed by Kiv et al. [34] as we found it to be the most adequate for fulfilling our research objectives: by modeling the pillars using their metamodel, we will be able to discover how each pillar complements and supplements one another by looking at similar and unique instances of the *Agile_Feature* class.

To better suit our research context, we made a few changes to their metamodel, as seen in Figure 4.1. We removed the *Value* class due to LS and UCD lacking explicit definitions for values and renamed the *Agile_Feature* class into the more generic *Feature*. We only consider the tactical level of the metamodel in order to greatly reduce the scope of our work while not significantly diminishing its contribution, leaving the modeling of the operational level open for future research endeavors.

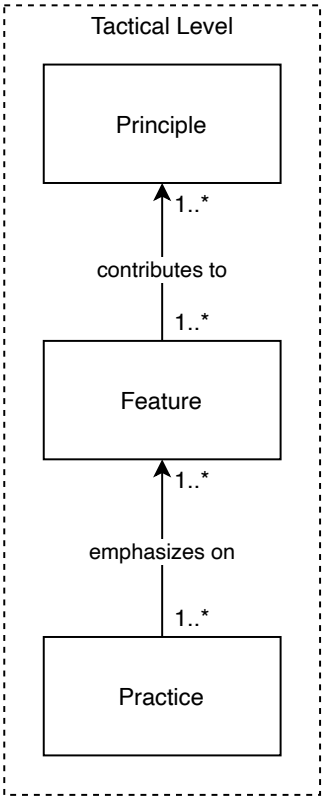


Figure 4.1: Adapted metamodel

The metamodel's three core classes are defined as follows:

- **Principle:** establishes a mindset for solid software engineering practices [62].
- **Feature**: a distinctive characteristic of a principle, as defined by Laanti et al. [41]. Features narrow the gap between principles and practices; and are the class of concepts we use to bridge the three pillars—by identifying the features present in multiple pillars.
- **Practice**: a concrete activity and practical technique that is used to develop and manage software projects in a manner consistent with principles [72].

Given this adapted metamodel and the existing work of Kiv et al. on Agile [34, 35], we outline the necessary steps to conceive our conceptual model:

C1. Model the ASD pillar;

C2. Model the LS pillar;

C3. Model the UCD pillar; and

C4. Integrate the ASD, LS, and UCD models using identical instances of the *Feature* class as junction points.

We can skip **C1** as the ASD pillar has already been modeled by Kiv et al. (see Table 4.1). To implement **C2** and **C3**, we need to perform the following steps for each:

S1. Determine a set of *Principles* for the respective pillar;

S2. Derive and map a set of *Features* from the *Principles*;

S3. Determine a set of *Practices* for the respective pillar; and

S4. Map each *Practice* to appropriate *Features*.

Steps **S1** and **S2** for LS and UCD are detailed in the following sections. Although reported in a sequential fashion, the analysis of LS and UCD principles (**S2**) was executed simultaneously, so as to identify similar concepts between these two pillars and avoid the creation of extraneous *Features*. We tried to limit our feature derivation process using the original features devised by Laanti et al. [41], creating new features only when the existing set lacked a concept expressed by a given principle.

We conducted a case study to assess these derived features and to perform steps **S3** and **S4**. We analyzed the day-to-day work process of two high-performance development teams that use the combined approach to obtain the practices to include in our model (**S3**). We also asked the teams to validate our set of features (**S2**) and map their practices to them (**S4**). Chapter 5 details the case study and resulting findings.

Table 4.1: ASD features per principle [34]

Principle	**Feature**
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done	Motivated individuals Good environment Support Trust
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation	Efficiency (for conveying information) Communication
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely	Sustainability People
The best architectures, requirements, and designs emerge from self-organizing teams	Self-organization
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly	Built-in improvement of efficiency and behavior
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software	Customer satisfaction Continuous delivery Value Early deliveries
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale	Frequent deliveries
Working software is the primary measure of progress	Measure progress via deliverables
Simplicity—the art of maximizing the amount of work not done—is essential	Simplicity Optimize work
Business people and developers must work together daily throughout the project	Collaboration
Welcome changing requirements, even late in development	Adaptability Competitiveness Customer benefit
Continuous attention to technical excellence and good design enhances agility	Focus on technical excellence

4.2 Lean Startup Principles and Features

We use the founding book on Lean Startup [63] as our source for definitions for its principles and to understand the approach overall. The principles are written as follows:

LP1 *Entrepreneurs are everywhere. You don't have to work in a garage to be in a startup. The concept of entrepreneurship includes anyone who works within my definition of a startup: a human institution designed to create new products and services under conditions of extreme uncertainty. That means entrepreneurs are everywhere and the Lean Startup approach can work in any size company, even a very large enterprise, in any sector or industry.*

LP2 *Entrepreneurship is management. A startup is an institution, not just a product, and so it requires a new kind of management specifically geared to its context of extreme uncertainty. In fact, as I will argue later, I believe "entrepreneur" should be considered a job title in all modern companies that depend on innovation for their future growth.*

LP3 *Validated learning. Startups exist not just to make stuff, make money, or even serve customers. They exist to learn how to build a sustainable business. This learning can be validated scientifically by running frequent experiments that allow entrepreneurs to test each element of their vision.*

LP4 *Innovation accounting. To improve entrepreneurial outcomes and hold innovators accountable, we need to focus on the boring stuff: how to measure progress, how to set up milestones, and how to prioritize work. This requires a new kind of accounting designed for startups—and the people who hold them accountable.*

LP5 *Build-Measure-Learn. 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.*

After analyzing the principles several times and deriving features from each, we obtained the mappings seen on Table 4.2.

4.3 User-Centered Design Principles and Features

Due to the "umbrella" nature of UCD, several authors have defined principles for it [31, 51]. We chose the principles as defined by ISO9241-210[1] [31], as it provides a more in-depth look at each principle in addition to arguably being the most authoritative source.

[1]The principles defined by ISO ISO9241-210 are available online as part of the free document preview of ISO9241-210:2010 provided by ISO and will not be completely shown here due to legal concerns.

Table 4.2: Lean Startup features per principle

Italicized features are new features not present in the original work of Laanti et al. [41]

Principle	**Feature**
Entrepreneurs are everywhere	People
Entrepreneurship is management	Adaptability *Business development* Competitiveness
Validated Learning	Customer satisfaction Frequent deliveries *Reduce uncertainty* Sustainability Value
Build-Measure-Learn	Adaptability Built-in improvement of efficiency and behavior Continuous delivery Early deliveries Efficiency (for conveying information) *Iterative* Optimize work Simplicity *User feedback*
Innovation Accounting	Communication *Focus on innovation* *Measure empirically*

UP1 The design is based upon an explicit understanding of users, tasks and environments.

UP2 Users are involved throughout design and development.

UP3 The design is driven and refined by user-centred evaluation.

UP4 The process is iterative.

UP5 The design addresses the whole user experience.

UP6 The design team includes multidisciplinary skills and perspectives.

After analyzing the principles several times and deriving features from each, we obtained the mappings seen on Table 4.3.

Table 4.3: User-Centered Design features per principle

Italicized features are new features not present in the original work of Laanti et al. [41]

Principle	**Feature**
The design is based upon an explicit understanding of users, tasks and environments	Customer satisfaction *Design based on context of use*
Users are involved throughout design and development	Support *User involvement*
The design is driven and refined by user-centered evaluation	Customer satisfaction *Measure empirically* *Reduce uncertainty* *User feedback*
The process is iterative	Adaptability Customer benefit *Iterative* *Reduce uncertainty*
The design addresses the whole user experience	Customer satisfaction *Focus on user experience* Value
The design team includes multidisciplinary skills and perspectives	Collaboration *Multi-disciplinary team* *Multi-perspective team* People Support

4.4 Feature Definitions

As the derivation process finished and the complete list of features stabilized, we set out to define each feature, as they were not given a proper definition in the work of Laanti et al. [41]. Similar to the initial derivation process, we re-analyzed the principles that each feature contributed to and derived definitions for each. The features and their definitions were reviewed by three senior researchers. The definitions are as follows:

- **Adaptability:** Embrace change at any point of development, should it stem from failed ideas to newfound paths of success, by making regular use of processes, tools, or other mechanisms that help you to seek and prepare for it.

- **Built-in improvement of efficiency and behavior:** Have concrete mechanisms to self-evaluate and achieve meaningful change, all in service of increasing value.

- **Business development:** Strive to accelerate the generation of value by searching for meaningful and sustainable solutions.

- **Business satisfaction:** Fulfill business expectations by constantly providing clear, meaningful data that justifies your decisions on how to best generate value.
- **Collaboration:** Ensure collaboration between all related parties to foster creativity and improve decision-making.
- **Communication:** Encourage communication and ensure easy access to all relevant and important information.
- **Competitiveness:** Act fast, generate value, and constantly seek for new opportunities.
- **Continuous delivery:** Ensure a continuous stream of value and feedback by regularly delivering solutions.
- **Customer benefit:** Err on the side of customer and constantly adapt to their needs.
- **Customer satisfaction:** Make sure you are constantly providing value to the customer with regular deliverables based on meaningful data.
- **Design based on context of use:** Design while taking into account the actual context of the end user.
- **Early deliveries:** Deliver early to gain critical feedback.
- **Efficiency (for conveying information):** Prefer richer communication channels, such as face-to-face.
- **Focus on innovation:** Strive to innovate and seek business opportunities.
- **Focus on technical excellence:** Strive to keep up on technological trends and make use of best practices at all times.
- **Focus on user experience**: Strive to craft an excellent user experience that addresses all of the user's needs.
- **Frequent deliveries:** Strive to craft an excellent user experience that addresses all of the user's needs.
- **Good environment:** Foster a safe, fulfilling, and healthy work environment.
- **Iterative:** Iterate on your work to refine it and enable its adaptation to something else should the need arise.
- **Measure empirically:** Use meaningful data to fuel your development effort, which can stem from user feedback and other real-world scenarios.
- **Measure progress via deliverables:** Measure progress by delivering valuable work.

- **Motivated individuals:** Motivate individuals by fostering individual growth and a good work environment.
- **Multi-disciplinary team:** Have individuals with different skill sets on the team.
- **Multi-perspective team:** Have people with access to different viewpoints (e.g., engineering, business) on the team.
- **Optimize work:** Have foresight and use techniques to avoid unnecessary work.
- **People:** Focus on individuals and have development be led by people instead of the other way around.
- **Reduce uncertainty:** Ground your decision-making on sensible data.
- **Self-organization:** Enable the development team to make most of the decision-making.
- **Simplicity:** Have foresight and use techniques to avoid overly complex work.
- **Support:** Enable individuals by supporting them with whatever they deem necessary and fostering their individual growth.
- **Sustainability:** Maintain a healthy work pace in addition to fueling decision-making with sensible data.
- **Trust:** Enable individuals by giving them the necessary trust and autonomy to make their own development decisions.
- **User feedback:** Actively gather and incorporate user feedback on your decision-making process.
- **User involvement:** Involve users directly on the development process.
- **Value:** Deliver value and have its pursuit guide development.

In the next chapter, we detail a case study used to fuel modeling steps **S3** (determining a set of *Practices* for each respective pillar) and **S4** (mapping each *Practice* to appropriate *Features*).

5. CASE STUDY

We conducted a single-case study [87] with two software development teams from a multinational company named ORG (name omitted for confidentiality reasons). As both teams worked closely together, forming the focal point of ORG's adoption of the combined approach in Brazil and sharing the same physical work environment, we considered them as a single unit of analysis. This study was primarily carried out to write a different thesis on the characteristics of the combined approach as a whole [73] with the assistance of this thesis' author. We analyze part of the data obtained in the original case study under a new light (Section 5.2) and continue the study a year later with a new stage of data collection procedures so as to gather the required resources to build our conceptual model (Section 5.3). In sum, we used the first stage of the case study to determine the practices of our conceptual model and the second stage to determine the features, the principle-feature mappings, and the feature-practice mappings of our conceptual model.

5.1 Case Setting

Our case takes place at ORG, an IT organization based in the USA with development sites spread across India and Brazil. True to its multinational nature, ORG boasts over 7,000 employees and is responsible for about 1,200 internal software products. Its IT department started an agile transformation in 2015 and moved to the combined use of ASD, LS, and UCD principles in late 2017. Before doing so, ORG improved their software products by means of a roadmap of software projects based on an annual budget negotiated between business departments. High-level business features were determined by business personnel to later be transformed into software requirements by IT staff. Project deadlines were strict and defined by quarter, i.e., every four months software project teams delivered software features to new or existing products to the organization's internal customers.

With the introduction of an agile transformation in 2015, project teams used Scrum as the guiding development framework—although some participants of this study reported that the strict quarterly deadlines made it waterfall-like. In 2017, the organization hires Pivotal Software Inc. consulting to support their transformation to a Pivotal Labs-like approach. Pivotal Labs proposes a "team rhythm" that encompasses development, product management, and design; areas that are fulfilled with principles and ceremonies from XP, LS, and UCD, respectively. Pivotal Labs' main goal is to help teams build software that delivers significant value for users and their business. It is both a framework and a starting point for any teams willing to discuss their needs and define their own path towards software development, including roles, practices, and work products.

Table 5.1: Profile of the case study participants

ID	Team	Role	IT Work Exp. (yr.)	Company Exp. (yr.)
P1	A	Product Designer	27	10
P2	A	Product Manager	21	6
P3	A	Product Manager	16	7.5
P4	A	Software Engineer	21	8
P5	A	Software Engineer	20	11
P6	A	Software Engineer	6	1
P7	A	Software Engineer	5.5	4
P8	B	Product Designer	5	4
P9	B	Product Manager	23	10.5
P10	B	Product Manager	19	0.5
P11	B	Software Engineer	15	11
P12	B	Software Engineer	10	4
P13	B	Software Engineer	7	7
P14	B	Software Engineer	5	5

5.1.1 The Teams

We observed *in loco* two software development teams (see Table 5.1) from ORG's Brazilian financial department. Both teams were built as a catalyst to prove the worth and spread the use of Pivotal Labs throughout the company and have been rated as high-performance and proficient in its use. To achieve this, some members underwent an immersive Pivotal Labs hands-on training at the company headquarters over the supervision of Pivotal Software Inc. consulting personnel before coming back to Brazil to teach the others. The teams have reported on several perceived benefits on using their new work process [74, 81]. Both teams are composed of two Product Managers, one Product Designer, and four Software Engineers.

Team A develops a software product that generates and manages data about ORG projects related to equipment and service delivery. This software manages general project information, such as personnel assignment and time spent on tasks. The application also calculates the associated costs of services offered by ORG products sold and provides this information to internal ORG consumers, while also generating profit data for each project, which is consumed (along with the rest of the data) by the accounting department. Team A is tasked with integrating all existing operations of their product into a single application that fulfills user needs and business expectations.

Team B handles a software product that consumes data from other ORG applications (including Team A's) to calculate the cost of Brazilian-built equipment. Their software generates reports for internal accounting (e.g., inventory reports). The team also works on

automating the validation process for the data coming from each source. Team B has to research current product processes to automate them into the application.

These teams worked for 9 months in a dedicated lab that follows 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 brainstorming and information sharing, and a meeting room that turns into an entertainment space for leisure time). The lab is located on PUCRS's campus grounds and was specifically built for ORG teams as an experimental learning environment. All participants signed a consent form (see Appendix C).

5.2 First Stage

In this stage of the case study, we analyzed the teams' day-to-day work in-depth to develop a complete work process model, so as to determine the practices that our conceptual model should encompass. We describe the data collection procedures, data analysis method, and results next.

5.2.1 Data Collection

We observed both teams for a 9-month period, executing several data collection procedures throughout it. Initial perceptions of the teams were collected using typical case study instruments. A *questionnaire* was used to identify the profile of each participant, while *daily observation sessions* were used to shadow team members and attend team ceremonies to learn about their approach to software development and the responsibilities of each of their roles. Two rounds of *semi-structured interviews* were used to specifically gather the team members' perceptions on the combined approach, on role changes, on interactions between roles, and on the impact of changes on their work routine. Sporadic interviews were used to follow up on unclear aspects of their day-to-day work unveiled in the observation sessions (e.g., alternate naming conventions). All interviews were voice recorded and transcribed for analysis.

We conducted six *focus group sessions* to discuss the activities, techniques, roles, and work products of the pillars of the combined approach as perceived by the teams in order to map their work process. We conducted two sessions for each pillar and each lasted 1.5 hours on average. We then conducted a *workshop* with both teams. At the start of the workshop, we handed them our mapping and asked each team (apart from the other) to visually represent their work process model on a whiteboard. After both teams stabilized on a process model, we had them discuss and explain together their work process in its

entirety while we took notes and recorded their discussion. We then converted their free-form drawings of the work process model into BPMN [55] using Bizagi Modeler[1] to guarantee syntactic validity and draw.io[2] for styling. We chose BPMN as it was similar to the free-form notation used by the teams. We also highlight that perceptions and opinions were similar between the different participants, with no significant divergences found between them.

5.2.2 Data Analysis

We analyzed data following the content analysis procedure by Krippendorff [37], organized into the following steps: organization and pre-analysis, reading and categorization, and recording the results. Using Atlas.TI[3], we first read the dataset, extracted text excerpts and marked them as codes. These codes were revisited and grouped into larger codes, forming categories—concepts to be included in our mapping efforts (see Figure 5.1). The identified concepts were then listed and organized in a document to be presented for the teams during the workshop, and revised afterwards to update our understanding of their work processes. We constantly reviewed our coding and mapping schemes with two seniors researchers to mitigate any limitations or bias in our analysis. Both senior researchers also reviewed the questionnaire and interview scripts and supported the piloting of these instruments for face and content validity with an external researcher with previous experience working with agile teams in the industry.

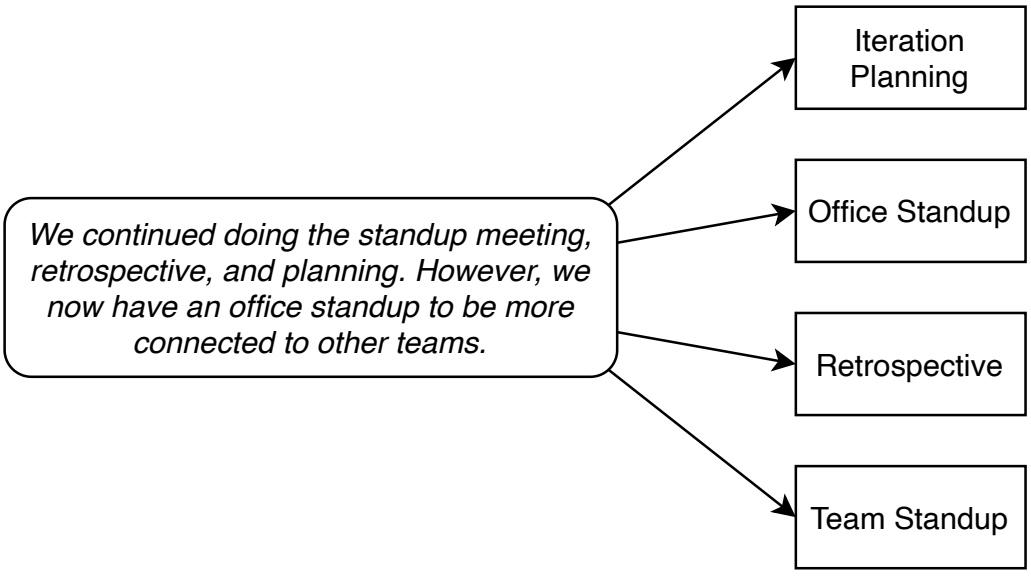


Figure 5.1: Example of data analysis procedure

[1] https://www.bizagi.com/en/platform/modeler
[2] https://drawio-app.com/
[3] atlasti.com

5.2.3 First Stage Results

Roles

The participants reported on the three roles that make up their teams, as per Pivotal Labs guidelines:

- ***Product Designer***, a facilitator that enables the team in communicating with the user, typically by conducting interviews and promoting the use of techniques to understand and foster empathy towards the user (e.g., a journey map);
- ***Product Manager***, a person that provides the business vision of things, helping the team address business needs and establish assumptions to experiment on; and
- ***Software Engineer***, a software developer responsible for implementing solutions and the environment in which they are developed, in addition to participating in decision making and other activities, such as user interviews. A specialization of it, the *Anchor*, is an engineer that resolves technical and non-technical issues by serving as a bridge from the engineers to the user and business stakeholders. Each team has one Anchor, and the Software Engineer that takes on this role does not necessarily have to be the most experienced one on the team.

Process Model

The teams defined three sequential phases (see Figure 5.2) for the entirety of the development process, in order: *Scoping*, which aims to discover the scope of their work; *Discovery and Framing*, which consists in refining the problem to solve and then determining the right solution to solve it; and *Iteration*, in which the chosen solution is repeatedly developed upon and properly implemented. We describe the phases in detail next.

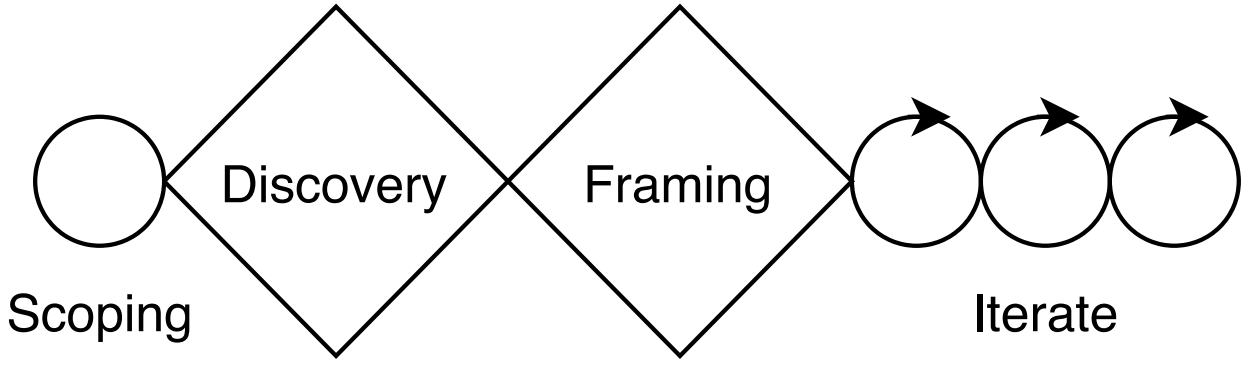


Figure 5.2: Process model overview

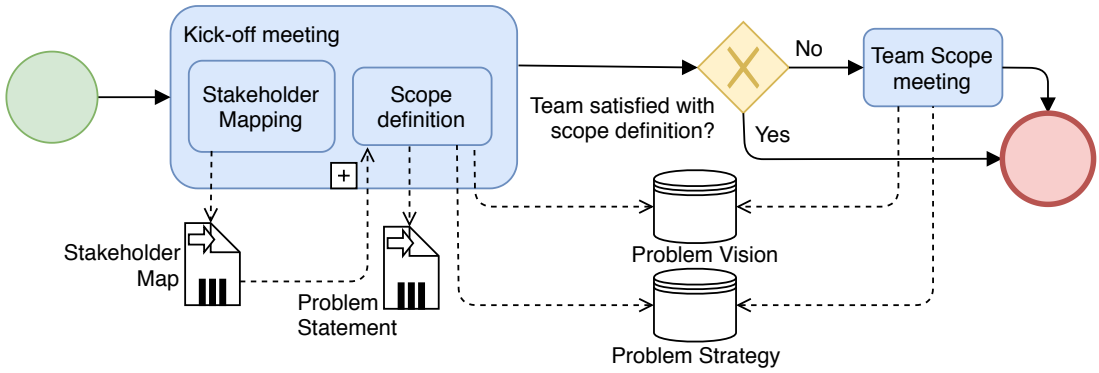


Figure 5.3: Scoping phase

Scoping Phase The teams put their efforts on the start of a project towards trying to understand what the problem to solve is—"We have a meeting with stakeholders to understand what will be the problem that we will work with, understanding our work scope and problem, trying to understand what it all entails." All team members are involved in this phase, in addition to stakeholders actively participating. They start with a kick-off meeting in which stakeholders discuss their needs, with the development teams mapping each and every stakeholder alongside their demands. In this meeting, project resources, such as specialized tools or extra personnel, are determined and secured; in addition to the execution of a brief brainstorm session that seeks to outline their approach to developing a solution, setting the pace of the project. Hopefully, they end up with a defined overview of the work scope—in case they do not, the teams host a meeting in which each team member shares their opinions so far on the current scope through a Hopes and Fears technique, revealing the teams' expectations and doubts about the project. At the end of this phase, teams could have four work products already: a stakeholder mapping, a product/problem statement, a product/problem vision, and a product/problem strategy (some teams begin working on new products and others on problems for already established products, hence the product/problem distinction). Figure 5.3 presents the *Scoping* phase.

Discovery and Framing Phase Both teams reported the use of a variant of the well-known "Double Diamond" model by the Design Council[4]. In the *Discovery and Framing* phase, as it is called, the Product Designer role is essential as it becomes more active than usual by constantly conducting interviews, researching, and generally acting as a facilitator that guides the team to the proper understanding of the user's needs. Nevertheless, the rest of the team is also involved in all the activities of this phase. The teams highlight the use of the build-measure-learn loop—"We are constantly building something: a problem understanding, a possible solution, an MVP, etc. We create assumptions for each build process to measure the effectiveness of the deliverable, and we learn in each delivery if we are treading the right path." The teams say the loop is present during the whole process of product development.

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

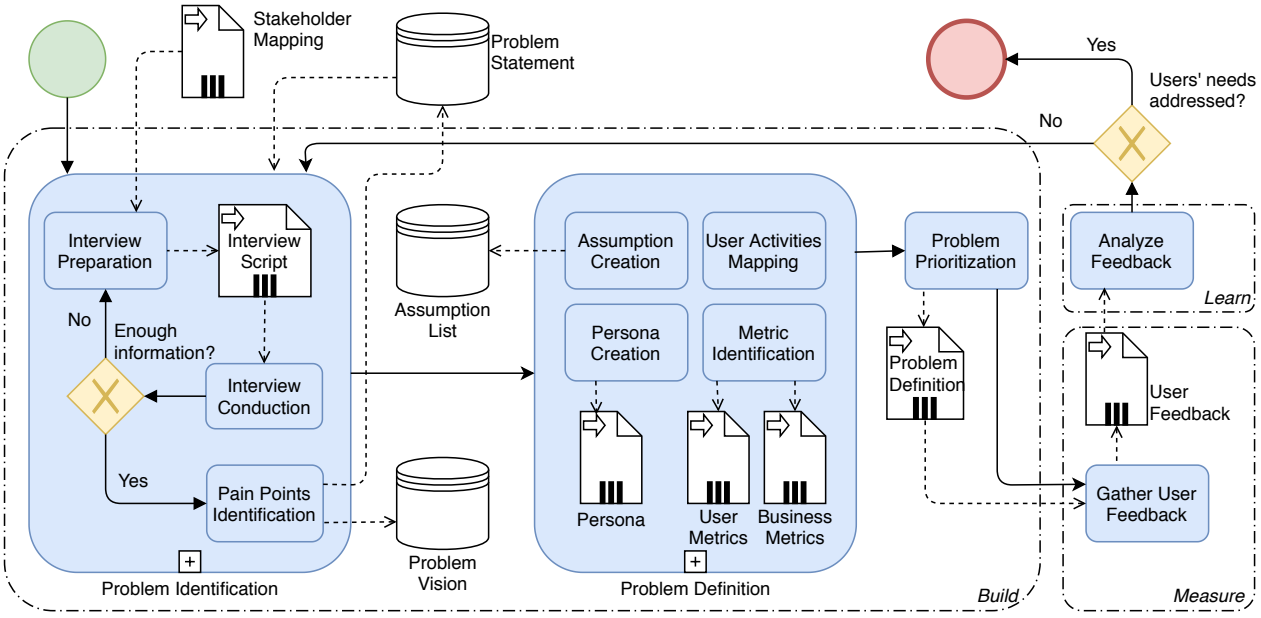


Figure 5.4: Discovery stage

The first stage of this phase, *Discovery* (see Figure 5.4), is focused on understanding the problem at hand on a deeper level, finding its possible root causes and prioritizing them by probability of being the correct cause—"We have the [root] problem prioritization that was defined and identified by the team. The discovery stage enables the whole team to understand the problem meticulously." The *Framing* stage (see Figure 5.5), in turn, is associated with the possible solutions for the problem discussed in the *Discovery* stage: "Our goal in the *Framing* stage is to identify multiple solutions for the problem we defined. At the the end of it, we are able to have a proper solution definition."

The two stages deal with different outcomes: the objective of *Discovery* is for all team members to further understand what is the problem that they are working to solve; while *Framing* has the aim of yielding a defined solution for the problem—"We must explore the Double Diamond concepts of convergence and divergence. We will be ready to advance to the *Framing* phase if we already converged on a specific problem to work on. And we will be ready to implement our solution if we already converged on a defined solution."

The *Discovery* stage has three main activities to achieve its goal: Problem Identification, Problem Prioritization, and Problem Definition. The Problem Identification activity is comprised of a set of activities and is part of the build step of the build-measure-learn cycle. It starts with the creation of an interview script using Interview Preparation, which uses the previously developed stakeholder mapping, along with affinity mapping, topic mapping, and brainstorming techniques. At least one representative from each role must participate in the following Interview Conduction, which is typically led by a Product Designer. After analyzing interview data, the teams evaluate if they need to conduct more interviews—"If we conduct an interview and learn new insights about the problem, we must do more sessions of Inter-

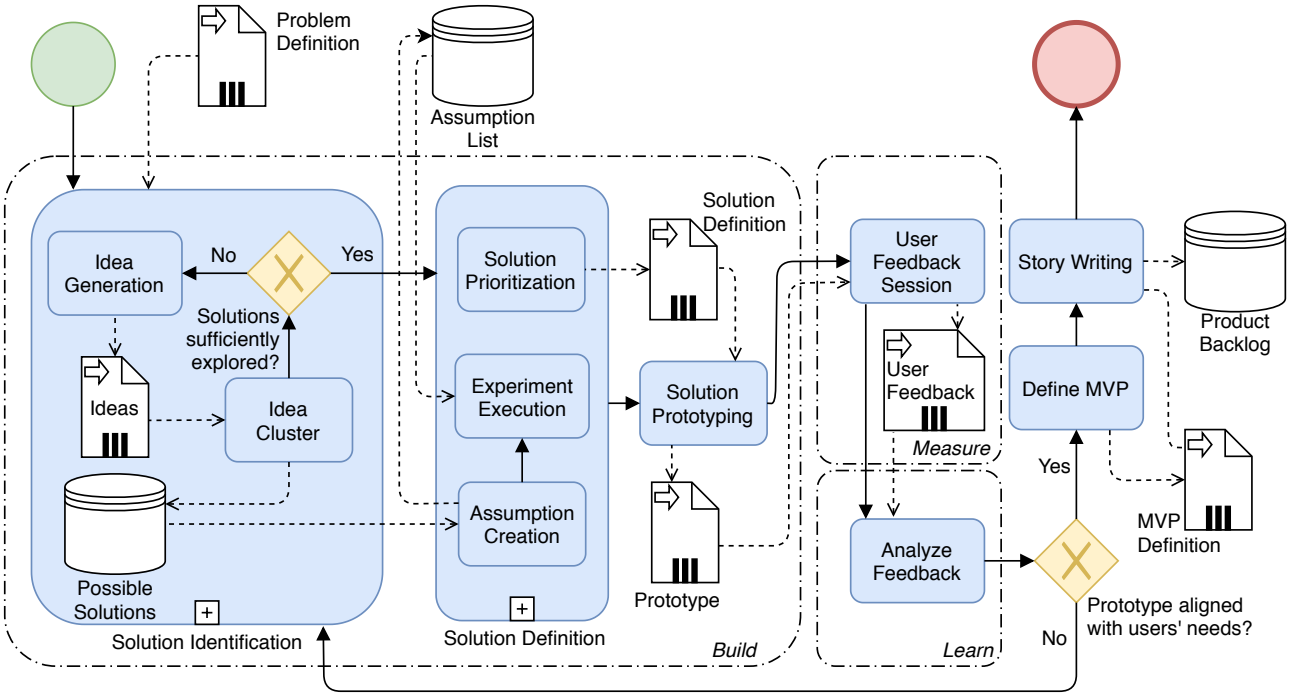


Figure 5.5: Framing stage

view Preparation using affinity mapping and brainstorming techniques to then perform new interviews, until we decide that we gathered enough information."

The teams report that after acquiring interview data, they use the Pain Points Identification activity (with the assistance of the service blueprint and user flow techniques) to map their user's daily work routine so as to locate their pain points—"We identify and answer our own questions about the process the users deal with, discovering their difficulties."

After identifying pain points, a refined problem statement and a refined problem vision arise. Teams can now do the Problem Definition activity, which starts with a Persona Creation activity by using user flow, service blueprint, and journey mapping techniques. Next comes the Assumption Creation activity, which generates an assumption list to be validated through experiments in the *Framing* stage or the *Iteration* phase. Teams also use User Activities Mapping to map the user's process using an ethnographic research technique.

Finally, they define the proper problem to solve through the Problem Prioritization activity by using a two by two matrix and now-near-next techniques. They later Gather User Feedback to validate their findings and see if they are addressing the user's needs by solving this problem—"We are always getting user feedback to refine and re-prioritize problems." If they find that the problem is correct and helps the user when solved, they move to the *Framing* stage, or restart the build-measure-learn cycle looking for a new problem definition.

The *Framing* stage (see Figure 5.5), also guided by the build-measure-learn loop, has two major activities: Solution Identification and Solution Definition. The Solution Identification activity starts by means of Idea Generation—"We meet to discuss what are the possibilities to solving the problem." The use of two by two matrices and brainstorming

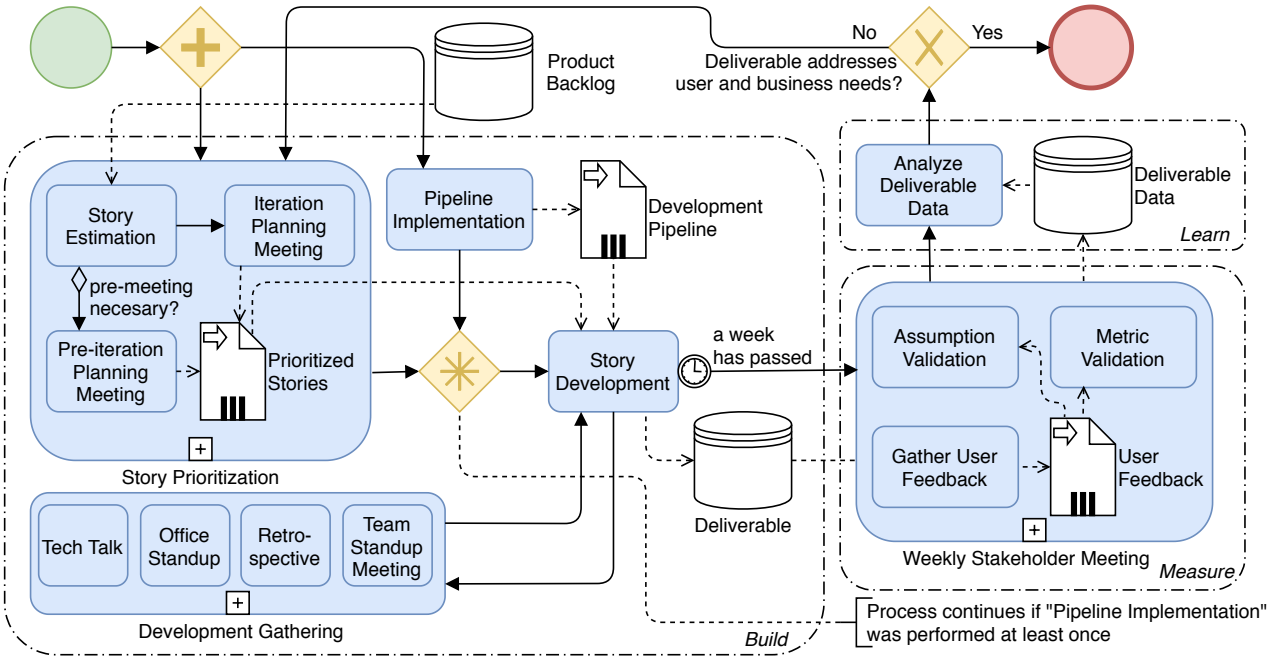


Figure 5.6: Iteration phase

generate an exhaustive idea list, which is organized by means of the Idea Cluster activity using the affinity mapping technique. Afterwards, they have a list of possible solutions which are turned into assumptions to validate later. To perform the Solution Definition activity, teams run experiments to validate assumptions and gather data on solutions, in addition to directly prioritizing them by using two by two matrices, now-near-next, and how might we techniques—“ ‘How might we’ mixed with some brainstorming allows us to identify how we can solve the defined problem.”

After prioritizing solutions, they can create prototypes with wireframe and mockup techniques and gather feedback on them with users. If they are confident enough with their findings they can move forward and Define an MVP, or do the whole build-measure-learn cycle to explore more possible solutions if need be. After defining an MVP, the teams finally move to the Iteration phase.

Iteration Phase The teams estimate the user stories on their product backlog using planning poker prior to prioritizing them in an Iteration Planning Meeting, which can be preceded by an optional Pre-Iteration Planning Meeting for general discussion on user stories and pre-prioritization. Meanwhile, they set up their whole development pipeline for continuous integration and delivery purposes. After prioritizing stories, story development begins (i.e., coding), focusing on small releases. The teams use several software development techniques (unit testing, pair programming, code reviews, and behavior-driven development) and make use of different types of meetings for process improvement and information sharing and synchronization purposes. Of note is the Office Standup, in which occurs inter-team knowledge sharing of their daily work, and Tech Talks, occasional meetings in which members presents new technology or practices to add to their repertoire.

The teams meet with stakeholders weekly—“It is a moment for the team to share their work and get feedback. We also use it to define our next steps.” They use stakeholder feedback data to validate their assumptions and other metrics, enabling the teams to learn more about their own deliverable and make the call on whether the current deliverable needs more polish or new features and if they should start a new iteration to continue working on it. The Iteration phase is executed repeatedly until all business and user needs are addressed.

Techniques

Throughout their development process, the teams use an ever-increasing set of techniques in an opportunistic fashion. Both teams continuously seek self-improvement, and as such are constantly studying recent trends and looking for new techniques to add to their toolbox. That said, we mapped all of the techniques they currently use (see Table 5.2).

Table 5.2: Techniques used by the teams

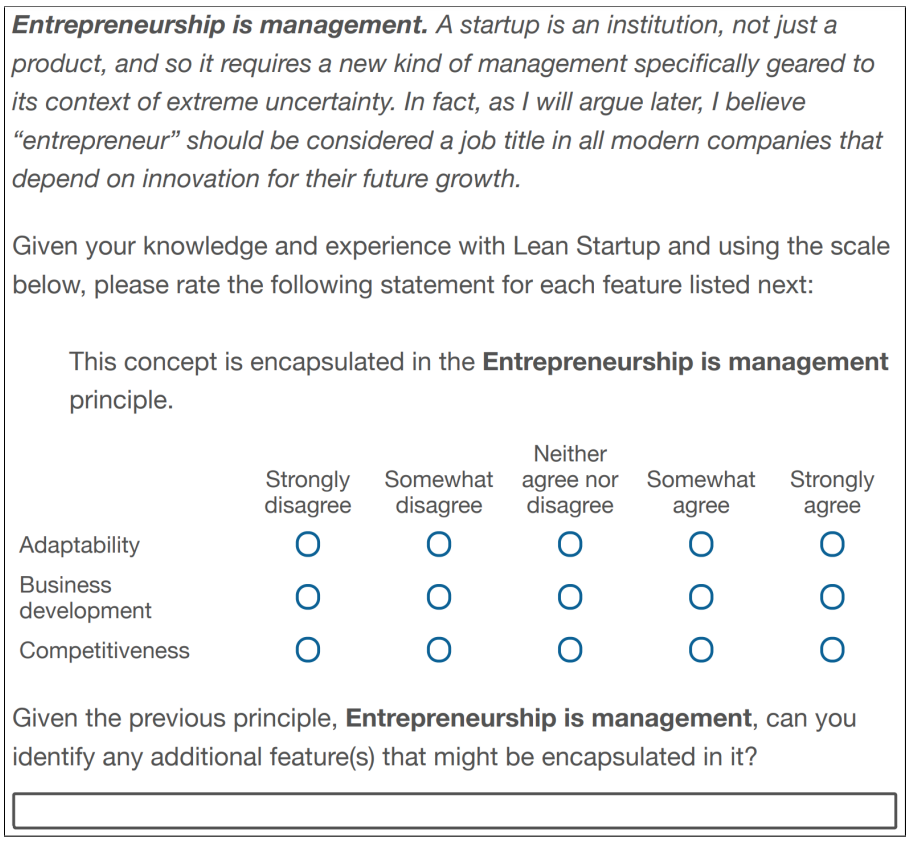
Pillar	**Techniques**
ASD (XP)	Behavior-Driven Development Code Review Continuous Integration Pair Programming Planning Poker Refactoring Small Releases Spike Test-Driven Development Unit Testing
LS	Continuous Delivery Customer Archetype Lean Canvas [45] Leap of Faith Minimum Viable Product
UCD	Affinity Map Brainstorm Design Studio Discovery & Framing Ethnographic Research Hope and Fears How Might We Interview Journey Map Mockup Persona Prototype Service Blueprint Two by Two Matrix User Flow Wireframe

5.3 Second Stage

A year after the original study, we contacted the two case study teams to evaluate the features we had derived so far (see Chapter 4) and to collect the missing building blocks of our conceptual model—the mappings between principles, features, and practices. We report on data collection and analysis procedures and results next.

5.3.1 Data Collection and Analysis

We used a *questionnaire* (see Appendix D) to determine what features should be included in the conceptual model. Participants were tasked with rating their level of agreement with a series of features derived from LS and UCD principles (see Table 4.2 and Table 4.3) using a 5-point scale, and were also asked to come up with new features if they deemed necessary. The questionnaire was reviewed by two fellow researchers who also supported a pilot application of the instrument to evaluate face and content validity. Figure 5.7 depicts an excerpt from the questionnaire.



Entrepreneurship is management. *A startup is an institution, not just a product, and so it requires a new kind of management specifically geared to its context of extreme uncertainty. In fact, as I will argue later, I believe "entrepreneur" should be considered a job title in all modern companies that depend on innovation for their future growth.*

Given your knowledge and experience with Lean Startup and using the scale below, please rate the following statement for each feature listed next:

This concept is encapsulated in the **Entrepreneurship is management** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Adaptability	○	○	○	○	○
Business development	○	○	○	○	○
Competitiveness	○	○	○	○	○

Given the previous principle, **Entrepreneurship is management**, can you identify any additional feature(s) that might be encapsulated in it?

Figure 5.7: Excerpt from the questionnaire

A week after compiling the questionnaire results, we ran a second *workshop* (see Appendix E) to clear up some questionnaire findings and gather more data. We first asked the participants to decide whether some features (that the questionnaire showed the majority was not in agreement with) should be kept in the model. Afterwards, we showed the participants the features that were derived from multiple principles and had them decide a single principle for those features to represent in the model. We then showed the participants the features they had suggested themselves and asked them to determine whether they should be included in the model or not. Finally, we had the participants determine which practices (from their work process and from the work of Kiv et al. [34]) contributed to which features.

Due to availability constraints, the questionnaire was only answered by participants P1, P4, P5, P9, and P13 of the original fourteen; while the workshop was only attended by a Software Engineer, a Product Designer, and a Product Manager[5].

5.3.2 Second Stage Results

Following the questionnaire and the second stage workshop, the participants ultimately agreed on our list of 34 unique features spread across ASD (see Table 4.1), LS (see Table 4.2), and UCD (see Table 4.3) in addition to suggesting one new feature (*Business satisfaction*). Participants also chose the "main" principle for features that were derived from multiple principles (see Table 5.3).

Table 5.3: Features derived from multiple principles

Italicized principles were chosen by the participants to be represented by the respective feature

Pillar	**Feature**	**Principles**
Lean Startup	Adaptability	Entrepreneurship is management *Build-Measure-Learn*
User-Centered Design	Customer satisfaction	*The design is based upon an explicit understanding of users, tasks and environments* The design is driven and refined by user-centered evaluation The design addresses the whole user experience
	Reduce uncertainty	The design is driven and refined by user-centered evaluation *The process is iterative*
	Support	Users are involved throughout design and development *The design team includes multidisciplinary skills and perspectives*

[5]We refrain from identifying these participants by their given ID to preserve their anonymity.

We added to the technique list (see Table 5.2) some elements we extracted from the teams themselves (*Balanced Team*[6]), from the process model (*Office Standup*, *Retrospective*, *Stakeholder Meeting*, *Team Standup Meeting*, and *Tech Talks*), from a suggestion given by the participants in the second stage workshop (*Survey*), and from the work of Kiv et al. [35] (XP and Scrum practices) to form the list of practices to be included in our model (see Table 5.4). We also omitted Discovery & Framing from this list as participants reported that it was more of an aggregate of practices rather than a practice in of itself.

Table 5.4: List of practices

ASD Practices	**LS Practices**	**UCD Practices**
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Ethnographic Research
Burndown Chart	MVP	Hopes and Fears
Code Review		How Might We
Collective Ownership		Interview
Continuous Integration		Journey Map
Daily Meeting		Mockup
Definition of Done		Persona
Definition of Ready		Prototype
Frequent Release		Service Blueprint
Iteration		Two by Two Matrix
Iterative Development		User Flow
Iteration Planning Meeting		Wireframe
Office Standup		
Pair Programming		
Planning Poker		
Point Estimates		
Relative Estimates		
Refactoring		
Retrospective		
Sign Up		
Simple Design		
Small Releases		
Spike		
Sustainable Pace		
Task Board		
TDD		
Tech Talks		
Three Questions		
Timebox		
Unit Testing		
User Stories		
Velocity		

[6]A multidisciplinary product team that considers design, development, and product management issues, usually due to being composed of Product Designers, Product Managers, and Software Engineers.

6. CONCEPTUAL MODEL

6.1 Conceptualizing an integration of Agile Software Development, Lean Startup, and User-Centered Design

To answer **RQ1**, we built a conceptual model of ASD, LS, and UCD following a metamodel adapted from the work Kiv et al. [34] (see Figure 4.1) using the principles from the Agile Manifesto [8] as the basis for ASD, the principles from the Lean Startup book [63] for LS, and the principles from ISO 9241-210 [31] for UCD. From a total of 23 principles, we derived 11 unique features for LS and UCD in addition to the original 23 derived by Laanti et al. for ASD [41]. The case study participants also suggested a single LS feature, for a total of 35 features. We then mapped these features to 55 practices sourced from Scrum and XP [35] and the case study of two development teams that use the combined approach, having the participants perform the mapping. We modeled each pillar individually using all of these elements (see Appendix A for the individual models of ASD, LS, and UCD expressed in table form). Afterwards, we combined all three models into one by using repeat *Feature* elements as junction points (i.e., by "overlapping" repeat elements). Figure 6.1 shows the conceptual model. Due to its sheer complexity, practices were split off the diagram and into Table 6.1 to improve its readability.

A quick glance at the diagram reveals how some principles are more overloaded than others. For instance, while *Users are involved throughout design and development* is connected to a single feature, *Build-Measure-Learn* is connected to a staggering 8. This is not surprising, as the BML cycle is the core of the LS method. Still, adding this to the fact that ASD encompasses 23 features using 12 principles while LS encompasses 21 features by only using 5 principles, we can clearly see how more atomic the ASD principles are.

As for practices, we highlight how most of them benefit from practices from multiple pillars, even if a given feature is not directly associated with a principle from said pillars. For instance, *Measure empirically* is a feature exclusive to LS and UCD, and yet it benefits from the ASD practice of *Spike*, suggesting to how the pillars passively strengthen one another. Curiously, the UCD features of *Multi-disciplinary team* and *Multi-perspective team* do not receive any benefit from UCD practices, but only from ASD practices instead.

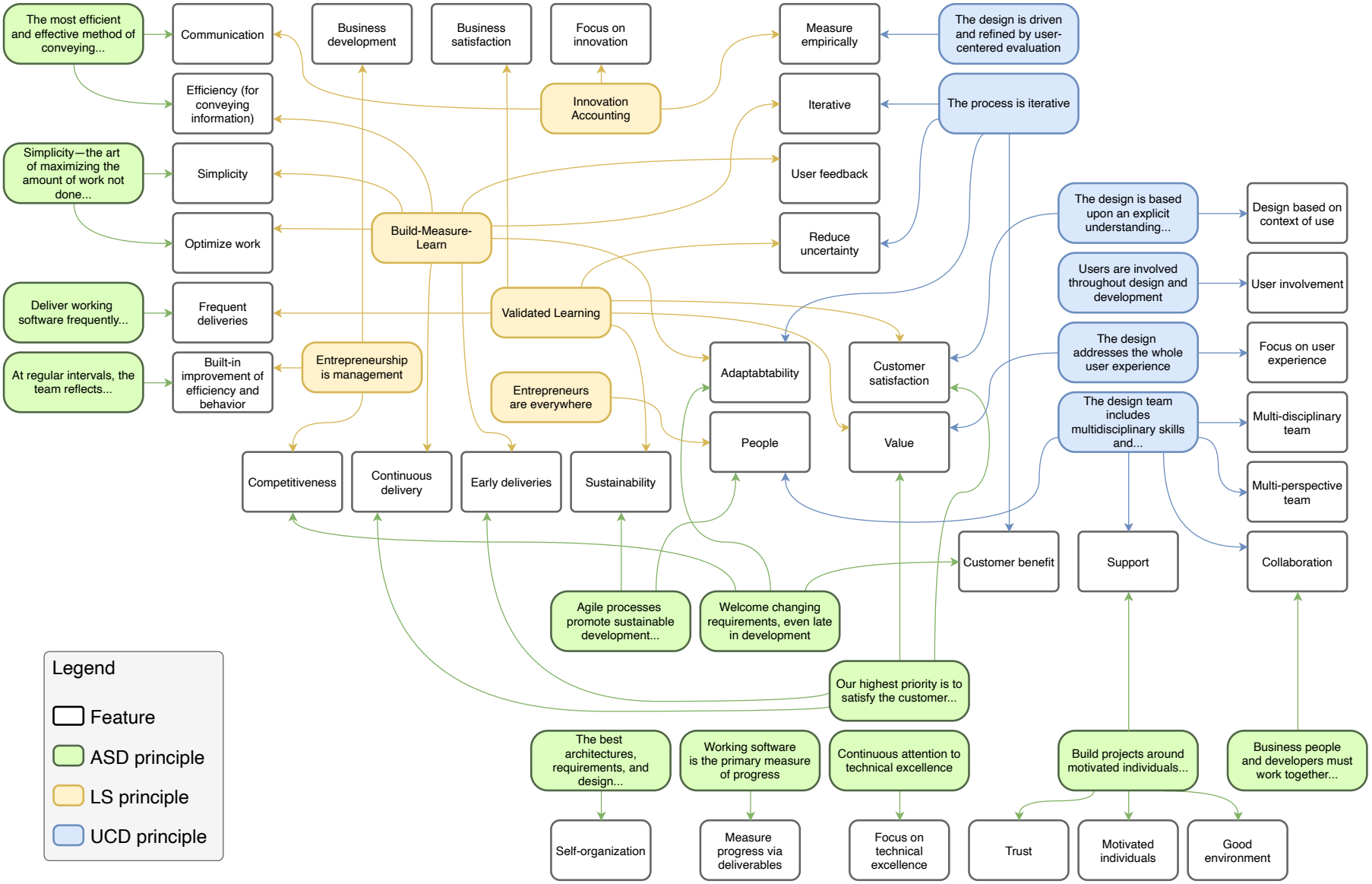


Figure 6.1: A conceptual model of ASD, LS, and UCD

Table 6.1: Mapping of features and practices

Feature	Practices
Adaptability	**ASD:** Backlog, Iteration, Iterative Development, Simple Design, Small Releases, Spike, User Stories. **LS:** MVP. **UCD:** Persona, Prototype.
Built-in improvement of efficiency and behavior	**ASD:** Continuous Integration, Daily Meeting, Retrospective, Three Questions.
Business development	**ASD:** Backlog Grooming, Frequent Release, Spike, Stakeholder Meeting. **LS:** Continuous Delivery, Customer Archetype, Lean Canvas, Leap of Faith, MVP. **UCD:** Brainstorm, How Might We, Persona, Prototype.
Business satisfaction	**ASD:** Backlog Grooming, Burndown Chart, Frequent Release, Point Estimates, Relative Estimates, Stakeholder Meeting, Sustainable Pace, User Stories. **LS:** Continuous Delivery. **UCD:** Survey.
Collaboration	**ASD:** Backlog Grooming, Balanced Team, Collective Ownership, Daily Meeting, Office Standup, Pair Programming, Planning Poker, Sign Up, Stakeholder Meeting, Three Questions. **UCD:** Design Studio, Hopes and Fears, Two by Two Matrix.
Communication	**ASD:** Backlog Grooming, Collective Ownership, Daily Meeting, Iteration Planning Meeting, Office Standup, Pair Programming, Planning Poker, Retrospective, Stakeholder Meeting, Tech Talks, Three Questions. **LS:** Lean Canvas. **UCD:** Brainstorm, Hopes and Fears, Mockup, Persona, Two by Two Matrix, User Flow, Wireframe.
Competitiveness	**ASD:** Continuous Integration, Frequent Release, Iteration, Iterative Development, Simple Design, Small Releases, Velocity. **LS:** Continuous Delivery, Lean Canvas, Leap of Faith, MVP. **UCD:** How Might We, Two by Two Matrix.
Continuous delivery	**ASD:** BDD, Code Review, Continuous Integration, Iteration, Iterative Development, Small Releases, TDD, Timebox, Unit Testing. **LS:** Continuous Delivery, MVP. **UCD:** Two by Two Matrix, Survey.
Customer benefit	**ASD:** BDD, Planning Poker, Stakeholder Meeting. **LS:** Continuous Delivery, MVP. **UCD:** Two by Two Matrix.
Customer satisfaction	**ASD:** BDD, Definition of Done, Definition of Ready, Planning Poker, TDD, Unit Testing, User Stories. **UCD:** Interview, Mockup, Persona, Two by Two Matrix, Wireframe.

Feature	Practices
Design based on context of use	**ASD:** Balanced Team. **LS:** MVP. **UCD:** Affinity Map, Design Studio, Ethnographic Research, Interview, Journey Map, Prototype, Service Blueprint.
Early deliveries	**ASD:** Refactoring, Simple Design, Small Releases. **LS:** MVP. **UCD:** Design Studio, How Might We, Mockup, Persona, Prototype.
Efficiency (for conveying information)	**ASD:** Backlog, Backlog Grooming, Continuous Integration, Pair Programming, Planning Poker, Stakeholder Meeting, Tech Talks, User Stories. **LS:** Lean Canvas. **UCD:** Mockup, Persona, Two by Two Matrix, User Flow, Wireframe.
Focus on innovation	**ASD:** Tech Talks. **LS:** Leap of Faith, MVP. **UCD:** Brainstorm, Design Studio, How Might We, Prototype.
Focus on technical excellence	**ASD:** Code Review, Collective Ownership, Continuous Integration, Iteration, Iterative Development, Pair Programming, Refactoring, TDD, Tech Talks, Unit Testing.
Focus on user experience	**ASD:** BDD, User Stories. **UCD:** Affinity Map, Design Studio, Ethnographic Research, Interview, Journey Map, Mockup, Persona, Prototype, Service Blueprint, User Flow, Wireframe.
Frequent deliveries	**ASD:** BDD, Continuous Integration, Frequent Release, Iteration, Iterative Development, Refactoring, Simple Design, Sign Up, Small Releases, Timebox. **LS:** Continuous Delivery.
Good environment	**ASD:** Collective Ownership, Daily Meeting, Office Standup, Retrospective, Sustainable Pace, Tech Talks, Three Questions. **UCD:** Hopes and Fears.
Iterative	**ASD:** Continuous Integration, Daily Meeting, Frequent Release, Iteration, Iterative Development, Iteration Planning Meeting, Retrospective, Sustainable Pace, TDD, Timebox. **LS:** Continuous Delivery, MVP. **UCD:** Prototype.
Measure empirically	**ASD:** Spike. **LS:** Continuous Delivery, MVP. **UCD:** Mockup, Prototype.
Measure progress via deliverables	**ASD:** Backlog, Burndown Chart, Point Estimates, Relative Estimates, Sustainable Pace, Velocity. **LS:** Continuous Delivery.
Motivated individuals	**ASD:** Balanced Team, Daily Meeting, Office Standup, Retrospective, Sign Up, Tech Talks, Three Questions. **LS:** Continuous Delivery. **UCD:** Hopes and Fears.
Multi-disciplinary team	**ASD:** Balanced Team, Tech Talks.
Multi-perspective team	**ASD:** Balanced Team, Stakeholder Meeting.

Feature	Practices
Optimize work	**ASD:** Continuous Integration, Pair Programming, Refactoring, Sign Up, Spike, TDD. **LS:** MVP. **UCD:** Prototype, Two by Two Matrix.
People	**ASD:** Balanced Team, Daily Meeting, Office Standup, Pair Programming, Retrospective, Sign Up, Stakeholder Meeting, Tech Talks, Three Questions. **UCD:** Design Studio, Hopes and Fears, Interview, Survey.
Reduce uncertainty	**ASD:** BDD, Iterative Development, Spike, Stakeholder Meeting, TDD, Unit Testing. **LS:** Continuous Delivery, Leap of Faith, MVP. **UCD:** Ethnographic Research, Interview, Journey Map, Prototype, Service Blueprint, Two by Two Matrix.
Self-organization	**ASD:** Balanced Team, Collective Ownership, Daily Meeting, Iteration Planning Meeting, Pair Programming, Planning Poker, Retrospective, Sign Up, Sustainable Pace, Tech Talks, Timebox.
Simplicity	**ASD:** BDD, Continuous Integration, Frequent Release, Iteration, Iterative Development, Pair Programming, Refactoring, Simple Design, TDD. **LS:** Continuous Delivery, MVP.
Support	**ASD:** Balanced Team, BDD, Collective Ownership, Daily Meeting, Office Standup, Pair Programming, Stakeholder Meeting, Tech Talks, Three Questions. **UCD:** Design Studio, Hopes and Fears, How Might We.
Sustainability	**ASD:** BDD, Burndown Chart, Code Review, Continuous Integration, Frequent Release, Iteration, Iterative Development, Point Estimates, Relative Estimates, Small Releases, Sustainable Pace, Task Board, TDD, Timebox, Velocity. **LS:** Continuous Delivery, Lean Canvas.
Trust	**ASD:** Collective Ownership, Daily Meeting, Pair Programming, Retrospective, Sign Up, Stakeholder Meeting. **UCD:** Hopes and Fears.
User feedback	**ASD:** Continuous Delivery, MVP. **LS:** Interview, Mockup, Prototype, Survey, Wireframe.
User involvement	**LS:** MVP. **UCD:** Design Studio, Interview, Journey Map, Mockup, Prototype, Two by Two Matrix, Wireframe.
Value	**ASD:** Backlog, BDD, User Stories. **LS:** Continuous Delivery, Lean Canvas, MVP. **UCD:** How Might We.

6.2 Concepts shared between and unique to Agile Software Development, Lean Startup, and User-Centered Design

We developed a Venn diagram of the features to answer **RQ2** and **RQ3** (see Figure 6.2). By observing the diagram, from the outset we can see that only 4 features are shared between all three pillars, in addition to ASD and LS having considerable overlap, likely due to their roots in Lean manufacturing.

Of note is that the features shared exclusively between LS and UCD—*Iterative*, *Measure empirically*, *Reduce uncertainty*, and *User feedback*—have a subtext of experimentation, which is indeed a core characteristic of these two approaches. What seems to tell each other apart is the subject of their experiments: LS deals with business ideas while UCD handles the product itself.

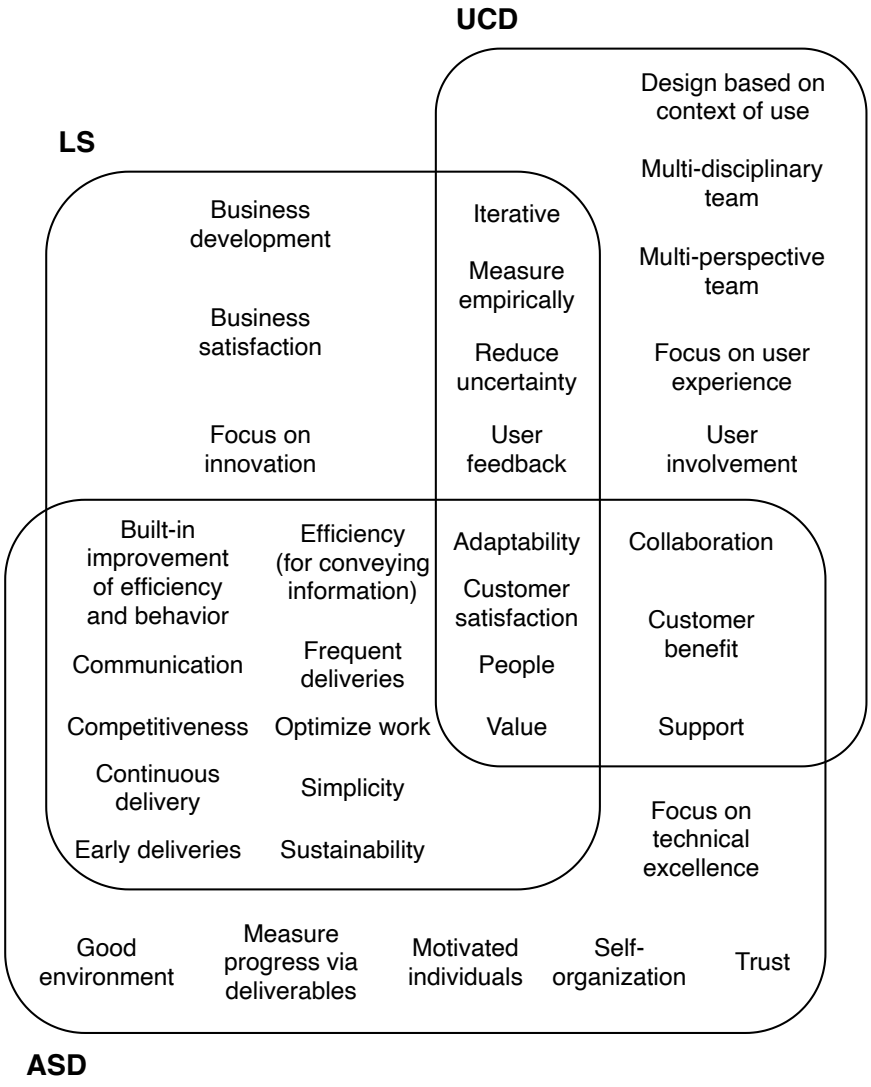


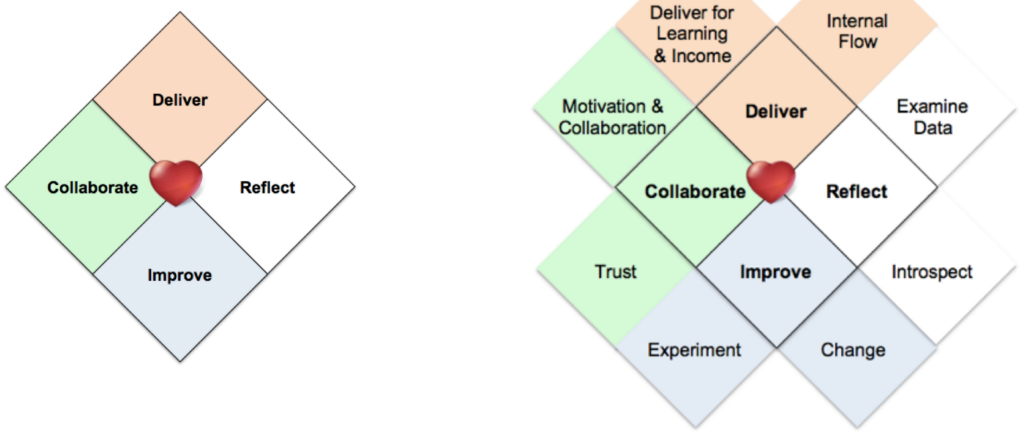
Figure 6.2: Venn diagram of ASD, LS, and UCD features

7. DISCUSSION

This chapter discusses our findings in light of the research questions. Section 7.1 refers to **RQ1**. Section 7.2 and Section 7.3 refer to **RQ2**.

7.1 Conceptualizing an integration of Agile Software Development, Lean Startup, and User-Centered Design

Establishing the right level of abstraction is crucial to modeling: define it too low and one loses sight of the bigger picture, define it too high and one has trouble going back down. This goes hand-in-hand with some of the problems ASD faces today—Agile Manifesto author Alistair Cockburn argues that agile has become overdecorated as of today [14] due to the increasing minutiae of agile methods. Similarly, the combined approach can be tough to understand as a whole due to all the apparatuses that accompany the three pillars that it encompasses. In response to this overwhelmingness problem, Cockburn created the heart of agile [14], an attempt to guide ASD back to its simpler roots—harking back to the original high-level values and principles of the Agile Manifesto [8]—defining four simple concepts that arguably support modern ASD (see Figure 7.1a), which can be expanded into lower levels of abstraction with smaller, more concrete concepts (see Figure 7.1b).



(a) The four core concepts of the heart of agile

(b) The heart of agile, expanded by one level

Figure 7.1: The heart of agile [14]

Our metamodel adapted from the work of Kiv et al. [34] provides similar semantics and level of abstraction to that of the expanded heart depicted in Figure 7.1b by means of the *Feature* class of concepts. Expanding the heart by one more level exposes concepts which are akin to the *Practice* level of our metamodel, such as *collaboration cards* [14].

Table 7.1: Features shared between ASD, LS, and UCD

ASD, LS, and UCD Features	ASD and LS Features	ASD and UCD Features	LS and UCD Features
Adaptability	Built-in improvement of efficiency and behavior	Collaboration	Iterative
Customer satisfaction	Communication	Customer benefit	Measure empirically
People	Competitiveness	Support	Reduce uncertainty
Value	Continuous delivery		User feedback
	Early deliveries		
	Efficiency (for conveying information)		
	Frequent deliveries		
	Optimize work		
	Simplicity		
	Sustainability		

We find that this approach is a great fit to describing the combined approach, as it allows our model to tackle the philosophical concerns of the pillars—allowing a better understanding of it overall—and operational concerns as well, which can, for instance, be used to determine the selection of techniques a team should use to make better use of the combined approach—enabling industry practitioners in the "learning stage", as Cockburn puts it [14].

Indeed, the semantics of the concepts used in our model are very similar to the ones used in Lean Software Development (LSD) [61]: *Principle*, *Tool* (analogue to our *Feature*), and *Concept* [46]. While the Poppendiecks' *Concept* is not as concrete as our *Practice*, we find that their levels of abstraction are largely comparable to ours. Much of the benefits behind LSD (arguably a merge of ASD and LS) stem directly from its simple principles, which get broken down into more concrete concepts as well. Arguably, mindset adjustment is the most beneficial change that LSD entails, similarly to what the heart of agile proposes and what our research group has found out about the combined approach [75]. It stands to reason then, that our model's focus on the *Feature* class is key to understanding the different advantages of the combined approach. Still, our model's toolbox of practices is also beneficial to software engineers looking to improve in a pinch.

7.2 Concepts shared between Agile Software Development, Lean Startup, and User-Centered Design

Table 7.1 outlines the concepts shared between the pillars. It is no surprise that ASD and LS share a great number of features, given that both have their origins in Lean

manufacturing [84]. The *Built-in improvement of efficiency and behavior* feature, for instance, is a direct implementation of the core Lean manufacturing principle of reducing waste. To see that there is, in fact, an overlap between the three pillars is no surprise either, given that both LS and UCD are generally described as agile-like.

True to LS's focus on acquiring "real data", the overlapping features of LS and UCD show how both approaches use essentially the same means to develop business and solutions, respectively: both approaches have a thorough focus on experimentation, which can be seen expressed directly by the *Reduce uncertainty* and *Measure empirically* features.

An interesting anecdote is that three of the four features shared between ASD, LS, and UCD—*Adaptability*, *People*, and *Value*—are eerily similar to the core values defined in the Agile Manifesto [8]. The manifesto reads as follows [8]:

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> **Individuals and interactions** over processes and tools
>
> **Working software** over comprehensive documentation
>
> **Customer collaboration** over contract negotiation
>
> **Responding to change** over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more.
>
> *Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas*
>
> *© 2001, the above authors. This declaration may be freely copied in any form, but only in its entirety through this notice.*

The three mentioned features can be directly mapped to the first, second, and fourth values written in the manifesto. Amusingly, the remaining value, *Customer collaboration over contract negotiation*, and the remaining feature, *Customer satisfaction*, both mention the customer: a somewhat imprecise term in software engineering, given it can refer to either business people or users. This ambiguity could have impacted the interpretation of the term by the participants of the case study—perhaps defining a proper definition for what the customer entails would have resulted in another *Feature* taking the place of *Customer satisfaction*, one that could be directly mapped to the third manifesto value, and thus exposing the large extent in which LS and UCD have been influenced by ASD.

Table 7.2: Features unique to ASD, LS, and UCD

ASD Features	**LS Features**	**UCD Features**
Focus on technical excellence	Business development	Design based on context of use
Good environment	Business satisfaction	Focus on user experience
Measure progress via deliverables	Focus on innovation	Multi-disciplinary team
Motivated individuals		Multi-perspective team
Self-organization		User involvement
Trust		

7.3 Concepts unique to Agile Software Development, Lean Startup, and User-Centered Design

Table 7.2 outlines the concepts unique to each pillars. Edison, Wang, and Abrahamsson [22] describe agile methods as being able to prescribe how to develop software faster, but inaccurate when determining what product should be built in the first place, as they lack focus on design [20] and so on. This can be seen directly on our own model, as we find features such as *Built-in improvement of efficiency of behavior* and *Frequent deliveries* associated with ASD principles, while features such as *Business development* and *Design based on context of use* are not.

In a symmetrical fashion, all three approaches have a feature that implies that the pillar focuses on something: ASD has *Focus on technical excellence*, LS has *Focus on innovation*, and UCD has *Focus on user experience*. Although tempting to say that this is the reason the combined approach works so well [73, 74, 81], we cannot assert this as perhaps the reason for its success is found in the aspects shared between the pillars (e.g., *Adaptability*), though these “focus” features do help differentiate one pillar from another.

8. CONCLUSION

A combined development approach of the three aforementioned pillars is a great way to mitigate the risks inherent with the innovation that the industry pursues. As industry practitioners and academics alike stitch together existing processes from the pillars, there still remains a knowledge gap on how the underlying concepts of these processes complement one another. Additionally, there is a lack of sound, widely accepted theoretical basis for the pillars: literature reports only on several self-contained processes instead. To fill in this knowledge gap, this thesis described the development of a conceptual model of an integration of ASD, LS, and UCD, showcasing how each of the three pillars that make up the combined approach can supplement and complement one another.

To develop the conceptual model, we first conducted a systematic literature review (see Chapter 3) to discover existing modeling schemas for abstractions of ASD, LS, and UCD. After a series of selection filters, we ended up with two studies on agile modelling [16, 34]. We decided to use the metamodel present in the work of Kiv et al. [34] due to its novel *Agile Feature* concept, which can serve as a bridge between ASD, LS, and UCD.

We adapted their metamodel into a simplified version (see Figure 4.1) spanning three classes of concepts: *Principle*, *Feature*, and *Practice*. We proceeded to fulfill the *Principle* and *Feature* classes by analyzing authoritative sources of LS [63] and UCD [31] (see Chapter 4), extracting principles from them and deriving a series of features for each (we skipped doing so for ASD as it was already covered in the work of Kiv et al. [34]), for a total of 23 principles and 35 features.

We then conducted a case study (see Chapter 5) with two product-oriented teams that use a development approach that combines ASD (XP), LS, and UCD to fulfill the *Practice* class of concepts and to confirm all of the model's mappings. The case study was conducted in two stages. The first stage consisted of the use of observations, semi-structured interviews, and focus group sessions, followed by Krippendorff's content analysis procedure [37]; which culminated in a report of the operational aspects of the combined approach (e.g., roles and work process in BPMN [55]), which we used to obtain the practices used by the teams. The second stage consisted of a survey and a workshop with the study's participants to determine and confirm all of the relationships between the model's concepts (e.g., which practices contribute to each feature), in addition to obtaining novel features the participants might suggest. With the end of the case study, the conceptual model was completed and reported on Chapter 6, with a total of 23 principles, 35 features, and 55 practices.

The findings of this study assist in understanding the overarching ideas behind the combined approach, and the conceptual model itself can be used to develop new resources, processes, and research related to the combined use of ASD, LS, and UCD.

8.1 Limitations

This section outlines limitations in each step of the study.

8.1.1 Systematic Literature Review

As with any systematic literature review, most threats to validity concern study selection bias and inaccuracy during data extraction. Unfortunately, our literature review (see Chapter 3) could only be conducted by a single researcher, and thus typical validity threat-reducing procedures (e.g., researchers validating the work of one another) could not be performed. Additionally, no snowball search of any kind was executed.

However, we argue that such limitations do not pose much threat to the validity of this study, as we found the ultimate outcome of the literature review (i.e., the discovery of the work of Kiv et al. [34, 35]) to be comprehensively adequate to our modeling needs.

8.1.2 Feature Analysis

The feature analysis (see Chapter 4) was done manually, and thus is prone to human error. Only a single researcher performed the analysis: even though the analysis itself was conducted carefully, this lack of redundancy can lead to inaccurate results. The analysis itself followed the same simple text reading approach of Laanti et al. [41], and, as such, the lack of a well-defined method is also cause for concern. In any case, we had the features evaluated both by senior researchers and by the case study participants to strengthen their validity.

Additionally, the meaning of “customer” in software development is murky at best. While originally it might have represented the developers’ contractor, UCD provides a closer contact with end-users, making “customer” be taken to represent either users or business stakeholders depending on the situation at hand. The lack of clear distinctions for this word introduces flaws in the meaning of the model where it appears, such as in the *Customer benefit Feature*. A future revision of the conceptual model should address this issue.

8.1.3 Case Study

Our case study (see Chapter 5) is limited in several aspects (e.g., generalization concerns [87]) as with any empirical study. A limitation specific to ours is that we did not have explicit contact with the teams' customers due to our research contract with ORG, as they were located outside of Brazil. However, we were allowed to use any customer and user data collected throughout any of our data collection procedures with full awareness of these individuals. Even with the explicit absence of direct contact, we managed to observe several activities in which users and customers were involved, thus allowing us to realize how cooperative and in line with the transformation they were.

As for the second stage of the case study, we have to disclose that the reduced number of participants was due to some of them leaving the organization during the one-year research hiatus and due to the COVID-19 pandemic, which led to less overall availability of the participants for research endeavors as the organization adapted to a remote work model. Additionally, we had to ask the participants to only consider their experience working at the PUCRS's lab, as the teams had since then been dissolved and reintegrated into other environments in which the use of Pivotal Labs was not as accentuated as the one they were in previously. Although this is a justifiable cause for concern, we argue that participants were able to accurately discern between their PUCRS and "regular" ORG experiences given the distinct work habits and unique physical space they employed at our campus.

We made use of several techniques to improve the quality of our study, and describe them in regards to specific quality tests [87] next. We do not address internal validity concerns as our study is of an exploratory nature [87].

Construct Validity

As a premise to construct validity, we defined our subject of change by asking each participant how they were used to developing software. This was essential as it provided us with a baseline to better understand what changed with the adoption of the combined approach. We had multiple researchers conduct the data collection and analysis procedures to reduce bias, and also used multiple methods and sources of data to triangulate our findings, in addition to using member checks (i.e., confirming our interpretations with study participants). We also had senior researchers validating draft case study reports with two keen study participants. Alongside the use of multiple data sources, both are tactics commonly used to address construct validity [87]. We made sure our research work was transparent as possible to the teams, to avoid any misguided suspicions that we might have been harming the teams in some way, such as by conducting performance evaluations on behalf of ORG. Our research team coexisted daily with the teams throughout the extensive time period they

were stationed at our campus. Indeed, some members of our research group even played ping-pong with the teams outside work hours. While this could have influenced our analysis somewhat, we think that the long exposure between participants and researchers led to increased trust, opening the teams to more honest dialogues about the combined approach and their situation.

External Validity

The study was conducted in a single organization, posing a threat to its external validity. We sampled two teams actively developing two different products to mitigate this, although we analyzed them both as a single cohesive unit. We also highlight that both teams had the support of one another when using the combined approach, having been stationed in the same environment for this exact purpose. This fact could have made the use of the combined approach an easier and more fortunate endeavor—all negative statements reported by the teams were not made towards the combined approach itself, but instead to contextual organizational issues that interfered with it [76]. Nevertheless, the two teams had a substantial level of empowerment, granting them the autonomy to work as they saw fit in the two distinct software products they were each working on. We believe this makes our study more generalizable to software development teams inserted in different contexts, however we can not assertively claim so as several factors need to be considered during the adoption of a development approach, from team maturity and organizational vision, to specifics of unique instances of the combined approach.

Reliability

Although case studies are seldom reproducible, we made sure to document the entire study (the entirety of the first stage of the study is available on Signoretti's master's thesis [73]). We also made use of the aforementioned triangulation efforts to make our data more consistent and dependable.

8.2 Future Work

This conceptual model opens up the development of a myriad of resources or research to support the combined approach, providing a theoretical basis for both philosophical and operational aspects of the approach. Still, one could look to extend the model to implement the entire operational level of the metamodel devised by Kiv et al. [34], so as to better support, for instance, transformation processes to the combined approach. To improve the conceptual model itself, one can take a look at who the "customer" really is

in software development, so as to iron out any semantic inconsistencies in the model, as explained in the limitations section (see Section 8.1).

As for our research group, we have two projects that will make use of the conceptual model, with both of them being already under way. One is an “acceleration model” as defined in the work of Moralles et al. [47], which quantitatively gauges the activities in which teams that use the combined approach should improve in order to properly complete their projects. The other is defined in the work of Machado et al. [44], a health check instrument created after Spotify’s Squad Health Check Model [36], which allows teams that use the combined approach to qualitatively evaluate themselves by analyzing a select set of the features defined in our conceptual model.

For a more straightforward study, we highlight how the conceptual model could already be of use by guiding teams to using the combined approach by means of assisting them in internalizing principles via practice adoption. An ideal research endeavor for this would be a case study of a development team using the conceptual model to this purpose, so as to reveal the lengths to which practices can help teams in internalizing principles.

REFERENCES

[1] Abras, C.; Maloney-krichmar, D.; Preece, J. “User-Centered Design”. In: Berkshire Encyclopedia of Human-Computer Interaction, 2004, pp. 1–14.

[2] Ahmad, M. O.; Lenarduzzi, V.; Oivo, M.; Taibi, D. “Lessons Learned on Communication Channels and Practices in Agile Software Development”. In: Proceedings of the Federated Conf. on Computer Science and Information Systems, 2018, pp. 929–938.

[3] Bajwa, S. S. “Pivoting in Software Startups”. In: *Fundamentals of Software Startups: Essential Eng. and Biz. Aspects*, Nguyen-Duc, A.; Münch, J.; Prikladnicki, R.; Wang, X.; Abrahamsson, P. (Editors), Cham, Switzerland: Springer, 2020, chap. 3, pp. 27–43.

[4] Bajwa, S. S.; Wang, X.; Duc, A. N.; Abrahamsson, P. “How Do Software Startups Pivot? Empirical Results from a Multiple Case Study”. In: Proceedings of the International Conference on Software Business, 2016, pp. 169–176.

[5] Barry, E. J.; Mukhopadhyay, T.; Slaughter, S. A. “Software Project Duration and Effort: An Empirical Study”, *Info. Tech. and Management*, vol. 3–1, Jan. 2002, pp. 113–136.

[6] Bastarrica, M. C.; Espinoza, G.; Marín, J. “Implementing Agile Practices: The Experience of TSol”. In: Proceedings of the International Symposium on Empirical Software Engineering and Measurement, 2018, pp. 38:1–38:10.

[7] Beck, K. “Embracing Change with Extreme Programming”, *Computer*, vol. 32–10, Oct. 1999, pp. 70–77.

[8] Beck, K.; Beedle, M.; van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; Kern, J.; Marick, B.; Martin, R. C.; Mellor, S.; Schwaber, K.; Sutherland, J.; Thomas, D. “Manifesto for Agile Software Development”. Source: http://www.agilemanifesto.org/, Retrieved on 2020-Feb-22.

[9] Blank, S. “Perfection By Subtraction – The Minimum Feature Set”. Source: https://steveblank.com/2010/03/04/perfection-by-subtraction-the-minimum-feature-set, Retrieved on 2020-Feb-22.

[10] British Design Council. “Eleven Lessons: Managing Design in Eleven Global Brands: A Study of the Design Process”, Technical Report, 2007, 144p.

[11] Brown, T. “Design Thinking”, *Harvard Business Review*, vol. 86, Jun. 2008.

[12] Brown, T. “Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation”. New York, NY, USA: HarperCollins, 2009, 272p.

[13] Caballero, L.; Moreno, A.; Seffah, A. “How Agile Developers Integrate User-Centered Design Into Their Processes: A Literature Review”, *International Journal of Software Engineering and Knowledge Engineering*, vol. 26–8, Oct. 2016, pp. 1175–1201.

[14] Cockburn, A. “The Heart of Agile”, Technical Report, 2016, 4p.

[15] Cockburn, A. “Cutting-Edge Agile: Opening Statement”, *Cutter Business Technology Journal*, vol. 32–3, Aug. 2019.

[16] Conboy, K.; Fitzgerald, B. “Toward a Conceptual Framework of Agile Methods”. In: Proceedings of the International Conference on Extreme Programming and Agile Methods, 2004, pp. 105–116.

[17] Corbin, J.; Strauss, A. “Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory”. Thousand Oaks, USA: SAGE, 2014, 456p.

[18] de Paula, D. F. O.; 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: Posters’ Extended Abstracts of the International Conference on Human-Computer Interaction, 2016, pp. 30–34.

[19] Dobrigkeit, F.; de Paula, D.; Uflacker, M. “InnoDev: A Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup”. Cham, Switzerland: Springer, 2019, chap. 11, pp. 199–227.

[20] Dybå, T.; Dingsøyr, T. “Empirical Studies of Agile Software Development: A Systematic Review”, *Information and Software Technology*, vol. 50–9, Aug. 2008, pp. 833–859.

[21] Edison, H.; Smørsgård, N. M.; Wang, X.; Abrahamsson, P. “Lean Internal Startups for Software Product Innovation in Large Companies: Enablers and Inhibitors”, *Journal of Systems and Software*, vol. 135, Jan. 2018, pp. 69–87.

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

[23] Fagerholm, F.; Sanchez, A. G.; Mäenpää, H.; Münch, J. “The RIGHT Model for Continuous Experimentation”, *Journal of Systems and Software*, vol. 123, Jan 2017, pp. 292–305.

[24] Gleasure, R.; Feller, J.; O’Flaherty, B. “Human-Centred Design: Existing Approaches and a Future Research Agenda”. In: Proceedings of the European Conference on Information Systems, 2012, pp. 105.

[25] 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.

[26] Grossman-Kahn, B.; Rosensweig, R. “Skip the Silver Bullet: Driving Innovation through Small Bets and Diverse Practices”. In: Proceedings of the Design Management Institute International Research Conference, 2012, pp. 815.

[27] Guyatt, G.; Rennie, D.; Meade, M.; Cook, D. “Users’ Guide to the Medical Literature: Essentials of Evidence-Based Clinical Practice”. New York, NY, USA: McGraw-Hill, 2008, 2nd ed., vol. 270, 567p.

[28] Hohl, P.; Klünder, J.; van Bennekum, A.; Lockard, R.; Gifford, J.; Münch, J.; Stupperich, M.; Schneider, K. “Back to the Future: Origins and Directions of the “Agile Manifesto” – Views of the Originators”, *Software Eng. Research and Dev.*, vol. 6, Nov. 2018, pp. 15.

[29] IDEO. “The Field Guide to Human-Centered Design: Design Kit”. Palo Alto, CA, USA: IDEO, 2015, 192p.

[30] Inayat, I.; Salim, S. S.; Marczak, S.; Daneva, M.; Shamshirband, S. “A Systematic Literature Review on Agile Requirements Engineering Practices and Challenges”, *Computers in Human Behavior*, vol. 51, Oct. 2015, pp. 915–929.

[31] International Organization for Standardization. “Ergonomics of Human-System Interaction — Part 210: Human-Centred Design for Interactive Systems”, Standard, Geneva, Switzerland, 2019, 33p.

[32] Itoh, Y.; Hirose, Y.; Takahashi, H.; Kurosu, M. “A New User-Centered Design Process for Creating New Value and Future”. In: Proceedings of the International Conference on Human-Computer Interaction, 2007, pp. 108–116.

[33] Kitchenham, B. A.; Charters, S. “Guidelines for performing Systematic Literature Reviews in Software Engineering”, Technical Report, Keele University and Durham University, 2007, 44p.

[34] Kiv, S.; Heng, S.; Kolp, M.; Wautelet, Y. “An Intentional Perspective on Partial Agile Adoption”. In: Proceedings of the International Conference on Software Technologies, 2017, pp. 116–127.

[35] Kiv, S.; Heng, S.; Wautelet, Y.; Kolp, M. “Towards a Goal-Oriented Framework for Partial Agile Adoption”. In: Revised Selected Papers of the International Conference on Software Technologies, 2017, pp. 69–90.

[36] Kniberg, H. “Squad Health Check model – visualizing what to improve”. Source: https://engineering.atspotify.com/2014/09/16/squad-health-check-model/, Retrieved on 2021-Jan-28.

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

[38] Krogstie, J.; Sølvberg, A. “Information systems engineering – Conceptual modeling in a quality perspective”. Trondheim, Norway: Kompendiumforlaget, 2003, 388p.

[39] Kropp, E.; Koischwitz, K. “Experiences with User-Centered Design and Agile Requirements Engineering in Fixed-Price Projects”. In: Proceedings of the International Workshop on Usability- and Accessibility-Focused Req. Eng., 2014, pp. 47–61.

[40] Laanti, M. “Characteristics and Principles of Scaled Agile”. In: Proceedings of the International Conference on Agile Software Development, 2014, pp. 9–20.

[41] Laanti, M.; Similä, J.; Abrahamsson, P. “Definitions of Agile Software Development and Agility”. In: Proceedings of the European Conference on Systems, Software and Services Process Improvement, 2013, pp. 247–258.

[42] Lenarduzzi, V.; Lunesu, M. I. “The Lean Strategies: A Tale of User-Driven Innovation Processes”. In: Proceedings of the International Conference on Agile Software Development, 2018, pp. 1–6.

[43] Lindgren, E.; Münch, J. “Raising the Odds of Success: The Current State of Experimentation in Product Development”, *Information and Software Technology*, vol. 77, Sep. 2016, pp. 80–91.

[44] Machado, M.; Zorzetti, M.; Bastos, R.; Marczak, S. “Definindo um Modelo de Health Check para o Uso Combinado de Ágil, Design-Centrado no Usuário e Lean Startup no Desenvolvimento de Software”. In: Proceedings of the Brazilian Symposium on Software Quality, 2021, pp. 1–10, in review.

[45] Maurya, A. “Running Lean: Iterate from Plan A to a Plan That Works”. Sebastopol, CA, USA: O’Reilly Media, 2012, 240p.

[46] Moralles, C.; Vaccaro, M.; Zorzetti, M.; Pereira, E.; Trindade, C.; Prauchner, B.; Marczak, S.; Bastos, R. “On the Mapping of Underlying Concepts of a Combined Use of Lean and User-Centered Design with Agile Development: The Case Study of the Transformation Process of an IT Company”. In: Proceedings of the Brazilian Workshop on Agile Methods, 2019, pp. 25–40.

[47] Moralles, C.; Zorzetti, M.; Signoretti, I.; Pereira, E.; Vaccaro, M.; Prauchner, B.; Salerno, L.; Trindade, C.; Marczak, S.; Bastos, R. M. “On the Development of a Model to Support the Combined Use of Agile Software Development with User-centered Design and Lean Startup”. In: Proceedings of the European Conference on Systems, Software and Services Process Improvement, 2020, pp. 220–231.

[48] Nguyen-Duc, A.; Seppänen, P.; Abrahamsson, P. “Hunter-Gatherer Cycle: A Conceptual Model of the Evolution of Software Startups”. In: Proceedings of the International Conference on Software and System Process, 2015, pp. 199–203.

[49] Noll, J.; Razzak, M. A.; Bass, J. M.; Beecham, S. “A Study of the Scrum Master’s Role”. In: Proceedings of the International Conference on Product-Focused Software Process Improvement, 2017, pp. 307–323.

[50] Norman, D. “The Four Fundamental Principles of Human-Centered Design and Application”. Source: https://jnd.org/the-four-fundamental-principles-ofhuman-centered-design/, Retrieved on 2020-Feb-22.

[51] Norman, D. “People-Centered (Not Tech-Driven) Design*”. Source: https://jnd.org/people-centered-not-tech-driven-design/, Retrieved on 2020-Feb-22.

[52] Norman, D. A. “The Design of Everyday Things”. New York, NY, USA: Basic Books, Inc., 2002, 240p.

[53] Norman, D. A.; Draper, S. W. “User Centered System Design; New Perspectives on Human-Computer Interaction”. Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc., 1986, 544p.

[54] Obendorf, H.; Finck, M. “Scenario-Based Usability Engineering Techniques in Agile Development Processes”. In: Extended Abstracts on Human Factors in Computing Systems, 2008, pp. 2159–2166.

[55] Object Management Group. “Business Process Model and Notation (BPMN), Version 2.0”, Standard, Needham, MA, USA, 2010, 538p.

[56] Object Management Group. “Unified Modeling Language, v2.5.1”, Standard, Needham, MA, USA, 2017, 796p.

[57] Olsson, H. H.; Bosch, J. “Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation”. In: Proceedings of the International Conference on Software Business, 2015, pp. 154–166.

[58] Paasivaara, M.; Behm, B.; Lassenius, C.; Hallikainen, M. “Large-Scale Agile Transformation at Ericsson: a Case Study”, *Empirical Software Engineering*, vol. 23–5, Jan. 2018, pp. 2550–2596.

[59] Palmer, S. R.; Felsing, M. “A Practical Guide to Feature-Driven Development”. London, United Kingdom: Pearson Education, 2001, 298p.

[60] Pant, V.; Yu, E. S. K. “Conceptual Modeling to Support the “Larger Goal” Pivot – An Example from Netflix”. In: Proceedings of the Conference on the Practice of Enterprise Modeling, 2018, pp. 394–403.

[61] Poppendieck, M.; Poppendieck, T. “Lean Software Development: An Agile Toolkit”. Boston, USA: Addison-Wesley, 2003, 240p.

[62] Pressman, R. “Software Engineering: A Practitioner’s Approach”. New York, NY, USA: McGraw-Hill, 2009, 976p.

[63] Ries, E. “The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses”. USA: Crown Business, 2011, 336p.

[64] Ries, E. “Open Innovation in DC”. Online, Source: http://www.startuplessonslearned.com/2011/06/open-innovation-in-dc.html, Retrieved on 2020-Feb-22.

[65] Ries, E. “Why We Need to Teach MBA’s About Modern Entrepreneurship (and What Harvard Business School is Doing About It)”. Online, Source: http://www.startuplessonslearned.com/2011/01/why-we-need-to-teach-mbas-about-modern.html, Retrieved on 2020-Feb-22.

[66] Salah, D.; Paige, R. F.; Cairns, P. A. “Patterns for Integrating Agile Development Processes and User Centred Design”. In: Proceedings of the European Conference on Pattern Languages of Programs, 2015, pp. 1–10.

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

[68] Schwaber, K.; Sutherland, J. “The Scrum Guide”. Westminster, CO, USA: Scrum Alliance, 2011, 21p.

[69] Séguin, N.; Tremblay, G.; Bagane, H. “Agile Principles as Software Engineering Principles: An Analysis”. In: Proceedings of the International Conference on Agile Processes in Software Engineering and Extreme Programming, 2012, pp. 1–15.

[70] Serrador, P.; Pinto, J. K. “Does Agile work? — A Quantitative Analysis of Agile Project Success”, *International Journal of Project Mgmt.*, vol. 33–5, Jul. 2015, pp. 1040–1051.

[71] Sharp, J. H.; Ryan, S. D. “A Preliminary Conceptual Model for Exploring Global Agile Teams”. In: Proceedings of the International Conference on Agile Software Development, 2008, pp. 147–160.

[72] 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.

[73] Signoretti, I. “Characterizing the Combined Use of Agile, User-Centered Design and Lean Startup”, Master’s Thesis, School of Technology – Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, RS, Brazil, 2020, 125p.

[74] Signoretti, I.; Marczak, S.; Salerno, L.; de Lara, A.; 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: Proceedings of the International Symposium on Empirical Software Engineering and Measurement, 2019, pp. 1–6.

[75] Signoretti, I.; Salerno, L.; Marczak, S.; Bastos, R. “Combining User-Centered Design and Lean Startup with Agile Software Development: A Case Study of Two Agile Teams”. In: Proceedings of the Int’l Conf. on Agile Software Development, 2020, pp. 39–55.

[76] Signoretti, I.; Zorzetti, M.; Salerno, L.; Moralles, C.; Pereira, E.; Trindade, C.; Marczak, S.; Bastos, R. “Success and Failure Factors for Adopting a Combined Approach: A Case Study of Two Software Development Teams”. In: Proceedings of the International Conference on Product-Focused Software Process Improvement, 2020, pp. 125–141.

[77] Spector, A.; Gifford, D. “A Computer Science Perspective of Bridge Design”, *Communications of the ACM*, vol. 29–4, Apr. 1986, pp. 267–283.

[78] Standish Group. “The Chaos Report”, Technical Report, 1994, 8p.

[79] Stavru, S. “A Critical Examination of Recent Industrial Surveys on Agile Method Usage”, *Journal of Systems and Software*, vol. 94, Aug. 2014, pp. 87–97.

[80] Teece, D. J. “Capturing Value from Knowledge Assets: The New Economy, Markets for Know-How, and Intangible Assets”, *California Management Review*, vol. 40–3, Apr. 1998, pp. 55–79.

[81] 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: Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, 2020, pp. 330–335.

[82] Vilkki, K. “When Agile Is Not Enough”. In: Proceedings of the International Conferece on Lean Enterprise Software and Systems, 2010, pp. 44–47.

[83] Vredenburg, K.; Mao, J.-Y.; Smith, P. W.; Carey, T. “A Survey of User-Centered Design Practice”. In: Proceedings of the International Conference on Human Factors in Computing Systems, 2002, pp. 471–478.

[84] Womack, J. P.; Jones, D. T.; Roos, D.; of Technology, M. I. “The Machine That Changed the World: The Story of Lean Production”. New York, USA: HarperCollins, 1991, 323p.

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

[86] 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.

[87] Yin, R. K. “Case Study Research and Applications: Design and Methods”. Thousand Oaks, USA: SAGE, 2003, 350p.

[88] Yu, E.; Giorgini, P.; Maiden, N.; Mylopoulos, J. “Social Modeling for Requirements Engineering”. Cambridge, MA, USA: MIT Press, 2011, 752p.

[89] Zamudio, L.; Aguilar, J. A.; Barba, C. T.; Misra, S. “A Requirements Engineering Techniques Review in Agile Software Development Methods”. In: Proceedings of the Int’l Conference on Computational Science and Its Applications, 2017, pp. 683–698.

[90] Zorzetti, M.; Moralles, C.; Salerno, L.; Pereira, E.; Marczak, S.; Bastos, R. M. “Adopting Agile Software Development Combined with User-Centered Design and Lean Startup: A Systematic Literature Review on Maturity Models”. In: Proceedings of the International Conference on Enterprise Information Systems, 2020, pp. 517–541.

[91] Zorzetti, M.; Signoretti, I.; Pereira, E.; Machado, M.; Salerno, L.; Moralles, C.; Bastos, R.; Marczak, S. “An Empirical-informed Work Process Model for a Combined Approach of Agile, User-Centered Design, and Lean Startup”. In: Proceedings of the Brazilian Symposium on Software Quality, 2020, pp. 1–10.

[92] Zorzetti, M.; Signoretti, I.; Pereira, E.; Salerno, L.; Moralles, C.; Trindade, C.; Machado, M.; Bastos, R.; Marczak, S. “A Practice-Informed Conceptual Model for a Combined Approach of Agile, User-Centered Design, and Lean Startup”. In: Proceedings of the Int’l Conf. on Product-Focused Software Process Improvement, 2020, pp. 142–150.

[93] Zorzetti, M.; Signoretti, I.; Salerno, L.; Marczak, S.; Bastos, R. “Improving Agile Software Development using User-Centered Design and Lean Startup”, *Information and Software Technology*, vol. 141, Jan. 2022, pp. 106718.

[94] Zorzetti, M.; Vaccaro, M.; Moralles, C.; Prauchner, B.; Signoretti, I.; Pereira, E.; Salerno, L.; Bastos, R.; Marczak, S. “Maturity Models for Agile, Lean Startup, and User-Centered Design in Software Engineering: A Combined Systematic Literature Mapping”. In: Proceedings of the Int’l Conf. on Enterprise Information Systems, 2020, pp. 145–156.

APPENDIX A – INDIVIDUAL CONCEPTUAL MODELS IN TABLE FORM

Table A.1: Mapping of Agile Software Development principles, features, and practices

Principle	**Feature**	**Practices**
Build projects around motivated	Motivated individuals	**ASD:** Balanced Team, Daily Meeting, Office Standup, Retrospective, Sign Up, Tech Talks, Three Questions. **LS:** Continuous Delivery. **UCD:** Hopes and Fears.
individuals. Give them the	Good environment	**ASD:** Collective Ownership, Daily Meeting, Office Standup, Retrospective, Sustainable Pace, Tech Talks, Three Questions. **UCD:** Hopes and Fears.
environment and support they need, and trust them to get	Support	**ASD:** Balanced Team, BDD, Collective Ownership, Daily Meeting, Office Standup, Pair Programming, Stakeholder Meeting, Tech Talks, Three Questions. **UCD:** Design Studio, Hopes and Fears, How Might We.
the job done	Trust	**ASD:** Collective Ownership, Daily Meeting, Pair Programming, Retrospective, Sign Up, Stakeholder Meeting. **UCD:** Hopes and Fears.
The most efficient and effective method of conveying	Efficiency (for conveying information)	**ASD:** Backlog, Backlog Grooming, Continuous Integration, Pair Programming, Planning Poker, Stakeholder Meeting, Tech Talks, User Stories. **LS:** Lean Canvas. **UCD:** Mockup, Persona, Two by Two Matrix, User Flow, Wireframe.
information to and within a development team is face-to-face	Communication	**ASD:** Backlog Grooming, Collective Ownership, Daily Meeting, Iteration Planning Meeting, Office Standup, Pair Programming, Planning Poker, Retrospective, Stakeholder Meeting, Tech Talks, Three Questions. **LS:** Lean Canvas. **UCD:** Brainstorm,
conversation		Hopes and Fears, Mockup, Persona, Two by Two Matrix, User Flow, Wireframe.

Principle	Feature	Practices
Agile processes promote sustainable development. The sponsors,	Sustainability	**ASD:** BDD, Burndown Chart, Code Review, Continuous Integration, Frequent Release, Iteration, Iterative Development, Point Estimates, Relative Estimates, Small Releases, Sustainable Pace, Task Board, TDD, Timebox, Velocity. **LS:** Continuous Delivery, Lean Canvas.
developers, and users should be able to maintain a constant pace indefinitely	People	**ASD:** Balanced Team, Daily Meeting, Office Standup, Pair Programming, Retrospective, Sign Up, Stakeholder Meeting, Tech Talks, Three Questions. **UCD:** Design Studio, Hopes and Fears, Interview, Survey.
The best architectures, requirements, and designs emerge from self-organizing teams	Self-organization	**ASD:** Balanced Team, Collective Ownership, Daily Meeting, Iteration Planning Meeting, Pair Programming, Planning Poker, Retrospective, Sign Up, Sustainable Pace, Tech Talks, Timebox.
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly	Built-in improvement of efficiency and behavior	**ASD:** Continuous Integration, Daily Meeting, Retrospective, Three Questions.

Principle	Feature	Practices
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software	Customer satisfaction	**ASD:** BDD, Definition of Done, Definition of Ready, Planning Poker, TDD, Unit Testing, User Stories. **UCD:** Interview, Mockup, Persona, Two by Two Matrix, Wireframe.
	Continuous delivery	**ASD:** BDD, Code Review, Continuous Integration, Iteration, Iterative Development, Small Releases, TDD, Timebox, Unit Testing. **LS:** Continuous Delivery, MVP. **UCD:** Two by Two Matrix, Survey.
	Value	**ASD:** Backlog, BDD, User Stories. **LS:** Continuous Delivery, Lean Canvas, MVP. **UCD:** How Might We.
	Early deliveries	**ASD:** Refactoring, Simple Design, Small Releases. **LS:** MVP. **UCD:** Design Studio, How Might We, Mockup, Persona, Prototype.
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale	Frequent deliveries	**ASD:** BDD, Continuous Integration, Frequent Release, Iteration, Iterative Development, Refactoring, Simple Design, Sign Up, Small Releases, Timebox. **LS:** Continuous Delivery.
Working software is the primary measure of progress	Measure progress via deliverables	**ASD:** Backlog, Burndown Chart, Point Estimates, Relative Estimates, Sustainable Pace, Velocity. **LS:** Continuous Delivery.
Simplicity—the art of maximizing the amount of work not done—is essential	Simplicity	**ASD:** BDD, Continuous Integration, Frequent Release, Iteration, Iterative Development, Pair Programming, Refactoring, Simple Design, TDD. **LS:** Continuous Delivery, MVP.
	Optimize work	**ASD:** Continuous Integration, Pair Programming, Refactoring, Sign Up, Spike, TDD. **LS:** MVP. **UCD:** Prototype, Two by Two Matrix.

Principle	Feature	Practices
Business people and developers must work together daily throughout the project	Collaboration	**ASD:** Backlog Grooming, Balanced Team, Collective Ownership, Daily Meeting, Office Standup, Pair Programming, Planning Poker, Sign Up, Stakeholder Meeting, Three Questions. **UCD:** Design Studio, Hopes and Fears, Two by Two Matrix.
Welcome changing requirements, even late in development	Adaptability	**ASD:** Backlog, Iteration, Iterative Development, Simple Design, Small Releases, Spike, User Stories. **LS:** MVP. **UCD:** Persona, Prototype.
	Competitiveness	**ASD:** Continuous Integration, Frequent Release, Iteration, Iterative Development, Simple Design, Small Releases, Velocity. **LS:** Continuous Delivery, Lean Canvas, Leap of Faith, MVP. **UCD:** How Might We, Two by Two Matrix.
	Customer benefit	**ASD:** BDD, Planning Poker, Stakeholder Meeting. **LS:** Continuous Delivery, MVP. **UCD:** Two by Two Matrix.
Continuous attention to technical excellence and good design enhances agility	Focus on technical excellence	**ASD:** Code Review, Collective Ownership, Continuous Integration, Iteration, Iterative Development, Pair Programming, Refactoring, TDD, Tech Talks, Unit Testing.

Table A.2: Mapping of Lean Startup principles, features, and practices

Principle	**Feature**	**Practices**
Entrepreneurs are everywhere	People	**ASD:** Balanced Team, Daily Meeting, Office Standup, Pair Programming, Retrospective, Sign Up, Stakeholder Meeting, Tech Talks, Three Questions. **UCD:** Design Studio, Hopes and Fears, Interview, Survey.
Entrepreneurship is management	Business development	**ASD:** Backlog Grooming, Frequent Release, Spike, Stakeholder Meeting. **LS:** Continuous Delivery, Customer Archetype, Lean Canvas, Leap of Faith, MVP. **UCD:** Brainstorm, How Might We, Persona, Prototype.
	Competitiveness	**ASD:** Continuous Integration, Frequent Release, Iteration, Iterative Development, Simple Design, Small Releases, Velocity. **LS:** Continuous Delivery, Lean Canvas, Leap of Faith, MVP. **UCD:** How Might We, Two by Two Matrix.
Validated Learning	Business satisfaction	**ASD:** Backlog Grooming, Burndown Chart, Frequent Release, Point Estimates, Relative Estimates, Stakeholder Meeting, Sustainable Pace, User Stories. **LS:** Continuous Delivery. **UCD:** Survey.
	Customer satisfaction	**ASD:** BDD, Definition of Done, Definition of Ready, Planning Poker, TDD, Unit Testing, User Stories. **LS:** Continuous Delivery, MVP. **UCD:** Two by Two Matrix.
	Frequent deliveries	**ASD:** BDD, Continuous Integration, Frequent Release, Iteration, Iterative Development, Refactoring, Simple Design, Sign Up, Small Releases, Timebox. **LS:** Continuous Delivery.
	Reduce uncertainty	**ASD:** BDD, Iterative Development, Spike, Stakeholder Meeting, TDD, Unit Testing. **LS:** Continuous Delivery, Leap of Faith, MVP. **UCD:** Ethnographic Research, Interview, Journey Map, Prototype, Service Blueprint, Two by Two Matrix.
	Sustainability	**ASD:** BDD, Burndown Chart, Code Review, Continuous Integration, Frequent Release, Iteration, Iterative Development, Point Estimates, Relative Estimates, Small Releases, Sustainable Pace, Task Board, TDD, Timebox, Velocity. **LS:** Continuous Delivery, Lean Canvas.

Principle	Feature	Practices
	Value	**ASD:** Backlog, BDD, User Stories. **LS:** Continuous Delivery, Lean Canvas, MVP. **UCD:** How Might We.
	Adaptability	**ASD:** Backlog, Iteration, Iterative Development, Simple Design, Small Releases, Spike, User Stories. **LS:** MVP. **UCD:** Persona, Prototype.
	Built-in improvement of efficiency and behavior	**ASD:** Continuous Integration, Daily Meeting, Retrospective, Three Questions.
Build-Measure-Learn	Continuous delivery	**ASD:** BDD, Code Review, Continuous Integration, Iteration, Iterative Development, Small Releases, TDD, Timebox, Unit Testing. **LS:** Continuous Delivery, MVP. **UCD:** Two by Two Matrix, Survey
	Early deliveries	**ASD:** Refactoring, Simple Design, Small Releases. **LS:** MVP. **UCD:** Design Studio, How Might We, Mockup, Persona, Prototype.
	Efficiency (for conveying information)	**ASD:** Backlog, Backlog Grooming, Continuous Integration, Pair Programming, Planning Poker, Stakeholder Meeting, Tech Talks, User Stories. **LS:** Lean Canvas. **UCD:** Mockup, Persona, Two by Two Matrix, User Flow, Wireframe.
	Iterative	**ASD:** Continuous Integration, Daily Meeting, Frequent Release, Iteration, Iterative Development, Iteration Planning Meeting, Retrospective, Sustainable Pace, TDD, Timebox. **LS:** Continuous Delivery, MVP. **UCD:** Prototype.
	Optimize work	**ASD:** Continuous Integration, Pair Programming, Refactoring, Sign Up, Spike, TDD. **LS:** MVP. **UCD:** Prototype, Two by Two Matrix.
	Simplicity	**ASD:** BDD, Continuous Integration, Frequent Release, Iteration, Iterative Development, Pair Programming, Refactoring, Simple Design, TDD. **LS:** Continuous Delivery, MVP.
	User feedback	**LS:** Continuous Delivery, MVP. **UCD:** Interview, Mockup, Prototype, Survey, Wireframe.

Principle	Feature	Practices
Innovation Accounting	Communication	**ASD:** Backlog Grooming, Collective Ownership, Daily Meeting, Iteration Planning Meeting, Office Standup, Pair Programming, Planning Poker, Retrospective, Stakeholder Meeting, Tech Talks, Three Questions. **LS:** Lean Canvas. **UCD:** Brainstorm, Hopes and Fears, Mockup, Persona, Two by Two Matrix, User Flow, Wireframe.
	Focus on innovation	**ASD:** Tech Talks. **LS:** Leap of Faith, MVP. **UCD:** Brainstorm, Design Studio How Might We, Prototype.
	Measure empirically	**ASD:** Spike. **LS:** Continuous Delivery, MVP. **UCD:** Mockup, Prototype.

Table A.3: Mapping of User-Centered Design principles, features, and practices

Principle	Feature	Practices
The design is based upon an explicit understanding of users, tasks and environments	Customer satisfaction	**ASD:** BDD, Definition of Done, Definition of Ready, Planning Poker, TDD, Unit Testing, User Stories. **UCD:** Interview, Mockup, Persona, Two by Two Matrix, Wireframe.
	Design based on context of use	**ASD:** Balanced Team. **LS:** MVP. **UCD:** Affinity Map, Design Studio, Ethnographic Research, Interview, Journey Map, Prototype, Service Blueprint.
Users are involved throughout design and development	User involvement	**LS:** MVP. **UCD:** Design Studio, Interview, Journey Map, Mockup, Prototype, Two by Two Matrix, Wireframe.
The design is driven and refined by user-centered evaluation	Measure empirically	**ASD:** Spike. **LS:** Continuous Delivery, MVP. **UCD:** Mockup, Prototype.
	User feedback	**ASD:** Continuous Delivery, MVP. **LS:** Interview, Mockup, Prototype, Survey, Wireframe.
The process is iterative	Adaptability	**ASD:** Backlog, Iteration, Iterative Development, Simple Design, Small Releases, Spike, User Stories. **LS:** MVP. **UCD:** Persona, Prototype.
	Customer benefit	**ASD:** BDD, Planning Poker, Stakeholder Meeting. **LS:** Continuous Delivery, MVP. **UCD:** Two by Two Matrix.
	Iterative	**ASD:** Continuous Integration, Daily Meeting, Frequent Release, Iteration, Iterative Development, Iteration Planning Meeting, Retrospective, Sustainable Pace, TDD, Timebox. **LS:** Continuous Delivery, MVP. **UCD:** Prototype.
	Reduce uncertainty	**ASD:** BDD, Iterative Development, Spike, Stakeholder Meeting, TDD, Unit Testing. **LS:** Continuous Delivery, Leap of Faith, MVP. **UCD:** Ethnographic Research, Interview, Journey Map, Prototype, Service Blueprint, Two by Two Matrix.
The design addresses the whole user experience	Focus on user experience	**ASD:** BDD, User Stories. **UCD:** Affinity Map, Design Studio, Ethnographic Research, Interview, Journey Map, Mockup, Persona, Prototype, Service Blueprint, User Flow, Wireframe.
	Value	**ASD:** Backlog, BDD, User Stories. **LS:** Continuous Delivery, Lean Canvas, MVP. **UCD:** How Might We.

Principle	Feature	Practices
The design team includes multidisciplinary skills and perspectives	Collaboration	**ASD:** Backlog Grooming, Balanced Team, Collective Ownership, Daily Meeting, Office Standup, Pair Programming, Planning Poker, Sign Up, Stakeholder Meeting, Three Questions. **UCD:** Design Studio, Hopes and Fears, Two by Two Matrix.
	Multi-disciplinary team	**ASD:** Balanced Team, Tech Talks.
	Multi-perspective team	**ASD:** Balanced Team, Stakeholder Meeting.
	People	**ASD:** Balanced Team, Daily Meeting, Office Standup, Pair Programming, Retrospective, Sign Up, Stakeholder Meeting, Tech Talks, Three Questions. **UCD:** Design Studio, Hopes and Fears, Interview, Survey.
	Support	**ASD:** Balanced Team, BDD, Collective Ownership, Daily Meeting, Office Standup, Pair Programming, Stakeholder Meeting, Tech Talks, Three Questions. **UCD:** Design Studio, Hopes and Fears, How Might We.

APPENDIX B – EXTRA SYSTEMATIC LITERATURE REVIEW DATA

B.1 ACM Search Strings

The search was expanded to include The ACM Guide to Computing Literature.

Agile Software Development
[Abstract: agile] AND [[Abstract: "conceptual model"] OR [Abstract: dictionary] OR [Abstract: glossary] OR [Abstract: "mind map"]]

Lean Startup
[[Abstract: lean startup] OR [Abstract: "continuous experimentation"] OR [Abstract: "experiment-driven"]] AND [[Abstract: "conceptual model"] OR [Abstract: dictionary] OR [Abstract: glossary] OR [Abstract: "mind map"]]

User-Centered Design
[[Abstract: "user-centered design"] OR [Abstract: "human-centered design"]] AND [[Abstract: "conceptual model"] OR [Abstract: dictionary] OR [Abstract: glossary] OR [Abstract: "mind map"]]

B.2 Science Direct Search Strings

The search was restricted to the Computer Science subject area.

Agile Software Development
agile AND ("conceptual model" OR "mind map" OR "glossary" OR "dictionary")

Lean Startup
("lean startup" OR "continuous experimentation" OR "experiment-driven") AND ("conceptual model" OR "mind map" OR "glossary" OR "dictionary")

User-Centered Design
("user-centered design" OR "human-centered design") AND ("conceptual model" OR "mind map" OR "glossary" OR "dictionary")

B.3 Scopus Search Strings

Agile Software Development
TITLE-ABS-KEY ("agile") AND (TITLE-ABS-KEY ("conceptual model") OR TITLE-ABS-KEY ("dictionary") OR TITLE-ABS-KEY ("glossary") OR TITLE-ABS-KEY ("mind map")) AND (LIMIT-TO (SUBJAREA , "COMP"))

Lean Startup
(TITLE-ABS-KEY ("lean startup") OR TITLE-ABS-KEY ("continuous experimentation")

OR TITLE-ABS-KEY ("experiment-driven")) AND (TITLE-ABS-KEY ("conceptual model") OR TITLE-ABS-KEY ("mind map") OR TITLE-ABS-KEY ("glossary") OR TITLE-ABS-KEY ("dictionary")) AND (LIMIT-TO (SUBJAREA , "COMP"))

User-Centered Design

(TITLE-ABS-KEY ("user-centered design") OR TITLE-ABS-KEY ("human-centered design")) AND (TITLE-ABS-KEY ("conceptual model") OR TITLE-ABS-KEY ("mind map") OR TITLE-ABS-KEY ("glossary") OR TITLE-ABS-KEY ("dictionary")) AND (LIMIT-TO (SUBJAREA , "COMP"))

B.4 IEEE Search Strings

Agile Software Development

(("All Metadata":agile AND ("All Metadata":"conceptual model" OR "All Metadata":dictionary OR "All Metadata":"mind map" OR "All Metadata":glossary)))

Lean Startup

(("All Metadata":"lean startup" OR "All Metadata":"continuous experimentation" OR "All Metadata":"experiment-driven") AND ("All Metadata":"conceptual model" OR "All Metadata":dictionary OR "All Metadata":"mind map" OR "All Metadata":glossary))

User-Centered Design

(("All Metadata":"user-cent*red design" OR "All Metadata":"human-cent*red design") AND ("All Metadata":"conceptual model" OR "All Metadata":dictionary OR "All Metadata":"definition" OR "All Metadata":glossary))

B.5 Springer Search Strings

The search was restricted to the Software Engineering subject area.

Agile Software Development

"agile" and ("conceptual model" or "mind map" or "dictionary" or "glossary")

Lean Startup

("lean startup" or "continuous experimentation" or "experiment-driven") AND ("conceptual model" or "glossary" or "dictionary" or "mind map")

User-Centered Design

("user-centered design" OR "user-centred design" OR "human-centered design" OR "human-centred design") AND ("conceptual model" OR "glossary" OR "dictionary" OR "mind map")

Table B.1: Quality scores of the selected studies

Study	C1	C2	C3	C4	Weighted Average
[16]	1	1	1	0.5	0.8
[32]	0.5	0	1	0	0.3
[71]	1	1	1	0	0.6
[24]	1	1	1	0	0.6
[69]	1	0.5	1	0.5	0.6
[39]	1	0.5	1	0	0.5
[40]	1	1	1	0	0.6
[48]	1	1	0.5	0	0.5
[22]	1	1	1	0	0.6
[57]	1	1	0.5	0	0.5
[4]	1	1	0.5	0	0.5
[34]	1	1	1	1	1
[89]	1	1	1	0	0.6
[49]	1	1	0.5	0	0.5
[2]	1	0.5	1	0	0.5
[42]	1	1	1	0	0.6
[21]	1	1	1	0	0.6
[28]	1	1	1	0	0.6
[60]	0.5	0.5	1	0	0.3
[46]	1	1	0.5	0.5	0.7
[3]	1	1	1	0	0.6

The criteria are based on four quality assessment questions:

C1. Is the research objective clearly defined?

C2. Is the research context well addressed?

C3. Are the findings clearly stated?

C4. Based on the findings, how valuable is the research?

Criterion **C4** is worth double. A score of 0.8 (exclusive) was the cut-off threshold.

APPENDIX C – 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 R. N. Moralles (Ph.D Candidate), Ingrid Signoretti, Maximilian Zorzetti, Matheus Vaccaro and Cássio Trindade (Master Students), Bruna Prauchner and Larissa Salerno (Bachelors Students)

APPENDIX D – QUESTIONNAIRE ON LEAN STARTUP AND USER-CENTERED DESIGN

Introduction

We appreciate having you back for one more survey!
Your contribution is, as always, of great importance to us.

Now let's get down to business: we are currently studying the principles behind Agile Software Development, Lean Startup, and User-Centered Design. The purpose of this questionnaire and the follow-up workshop is to "distill" these principles into concepts that accurately represent them entirely —we call these concepts **features**. For example, consider the following principle from the Agile Manifesto:

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

This principle can be distilled down to four features: motivated individuals, good environment, support, and trust. We actually already have all of these mapped for Agile thanks to the work of another researcher, so we'll only be asking you questions about Lean Startup and User-Centered Design *(phew!)*. What we will ask of you is pretty straightforward, and shouldn't take more than 30 minutes.

Before we start, please answer these profiling questions so we can triangulate your data with our previous findings.

How long have you worked in IT?

How long have you worked at the company?

What is your educational degree? (major and university, if applicable)

Were you trained by Pivotal consultants from the USA?

- ○ Yes
- ○ No

Which role were you assigned to when you were stationed in the PUCRS development lab?

- ○ Product Designer
- ○ Product Manager
- ○ Software Engineer

Instructions

Here's a rundown of what we will ask of you, in order:

1. **Read** the textual description of a principle.
2. **Rate** how well a list of features represent the given principle.
3. **Identify** new features for the given principle, if you find that it should be represented by additional features other than the ones we have identified.

When identifying new features, you can use features you've seen in other principles or create brand new ones: just write down the concepts you think are missing in whatever way you see fit. This part is especially useful to us, so please give it some thought (and feel free to hop back in forth between questions if creativity strikes you suddenly). In case you think of new features after completing this questionnaire, you can send them to maximilian.zorzetti@acad.pucr.br afterwards.

We'll ask of you those three things for each principle of Lean Startup and User-Centered Design. Just so you know, the texts of the User-Centered

Design principles are considerably longer and more complex than the Lean Startup ones, making them more tiresome to analyze.

We'll start with Lean Startup principles to ease you in.

Lean Startup

All of the following principles were extracted directly from the book **The Lean Startup** (2011), written by Eric Ries.

Please read the following principle carefully:

Entrepreneurship is management. *A startup is an institution, not just a product, and so it requires a new kind of management specifically geared to its context of extreme uncertainty. In fact, as I will argue later, I believe "entrepreneur" should be considered a job title in all modern companies that depend on innovation for their future growth.*

Given your knowledge and experience with Lean Startup and using the scale below, please rate the following statement for each feature listed next:

> This concept is encapsulated in the **Entrepreneurship is management** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Adaptability	O	O	O	O	O
Business development	O	O	O	O	O
Competitiveness	O	O	O	O	O

Given the previous principle, **Entrepreneurship is management**, can you identify any additional feature(s) that might be encapsulated in it?

Please read the following principle carefully:

Validated Learning. *Startups exist not just to make stuff, make money, or even serve customers. They exist to learn how to build a sustainable business. This learning can be validated scientifically by running frequent experiments that allow entrepreneurs to test each element of their vision.*

Given your knowledge and experience with Lean Startup and using the scale below, please rate the following statement for each feature listed next:

This concept is encapsulated in the **Validated Learning** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Sustainability	○	○	○	○	○
Reduce uncertainty	○	○	○	○	○
Customer satisfaction	○	○	○	○	○
Frequent deliveries	○	○	○	○	○
Value	○	○	○	○	○

Given the previous principle, **Validated Learning**, can you identify any additional feature(s) that might be encapsulated in it?

Please read the following principle carefully:

Entrepreneurs are everywhere. *You don't have to work in a garage to be in a startup. The concept of entrepreneurship includes anyone who works within my definition of a startup: a human institution designed to create new products and services under conditions of extreme uncertainty. That means entrepreneurs are everywhere and the Lean Startup approach can work in any size company, even a very large enterprise, in any sector or industry.*

Given your knowledge and experience with Lean Startup and using the scale below, please rate the following statement for each feature listed next:

This concept is encapsulated in the **Entrepreneurs are everywhere** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
People	○	○	○	○	○

Given the previous principle, **Entrepreneurs are everywhere**, can you identify any additional feature(s) that might be encapsulated in it?

Please read the following principle carefully:

Build-Measure-Learn. *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.*

Given your knowledge and experience with Lean Startup and using the scale below, please rate the following statement for each feature listed next:

This concept is encapsulated in the **Build-Measure-Learn** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Adaptability	○	○	○	○	○
Built-in improvement of efficiency and behavior	○	○	○	○	○
Continuous delivery	○	○	○	○	○
Early deliveries	○	○	○	○	○
Efficiency (for conveying information)	○	○	○	○	○
Iterative	○	○	○	○	○

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Optimize work	○	○	○	○	○
Simplicity	○	○	○	○	○
User feedback	○	○	○	○	○

Given the previous principle, **Build-Measure-Learn**, can you identify any additional feature(s) that might be encapsulated in it?

Please read the following principle carefully:

Innovation Accounting. *To improve entrepreneurial outcomes and hold innovators accountable, we need to focus on the boring stuff: how to measure progress, how to set up milestones, and how to prioritize work. This requires a new kind of accounting designed for startups—and the people who hold them accountable.*

Given your knowledge and experience with Lean Startup and using the scale below, please rate the following statement for each feature listed next:

This concept is encapsulated in the **Innovation Accounting** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Communication	○	○	○	○	○
Focus on innovation	○	○	○	○	○
Measure empirically	○	○	○	○	○

Given the previous principle, **Innovation Accounting**, can you identify any additional feature(s) that might be encapsulated in it?

Next, we'll analyze User-Centered Design principles.

User-Centered Design

All of the following principles were extracted (with minor alterations) from **ISO 9241-210**.

Please read the following principle carefully:

The design is based upon an explicit understanding of users, tasks and environments. [redacted]

Given your knowledge and experience with User-Centered Design and using the scale below, please rate the following statement for each feature listed next:

> This concept is encapsulated in the **The design is based upon an explicit understanding of users, tasks and environments** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Customer satisfaction	○	○	○	○	○
Design based on context of use	○	○	○	○	○

Given the previous principle, **The design is based upon an explicit understanding of users, tasks and environments**, can you identify any additional feature(s) that might be encapsulated in it?

Please read the following principle carefully:

Users are involved throughout design and development.

Given your knowledge and experience with User-Centered Design and using the scale below, please rate the following statement for each feature listed next:

This concept is encapsulated in the **Users are involved throughout design and development** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Support	○	○	○	○	○
User involvement	○	○	○	○	○

Given the previous principle, **Users are involved throughout design and development**, can you identify any additional feature(s) that might be encapsulated in it?

Please read the following principle carefully:

The design is driven and refined by user-centred evaluation. [redacted]

Given your knowledge and experience with User-Centered Design and using the scale below, please rate the following statement for each feature listed next:

This concept is encapsulated in the **The design is driven and refined by user-centred evaluation** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Customer satisfaction	○	○	○	○	○
Measure empirically	○	○	○	○	○
Reduce uncertainty	○	○	○	○	○
User feedback	○	○	○	○	○

Given the previous principle, **The design is driven and refined by user-centred evaluation**, can you identify any additional feature(s) that might be encapsulated in it?

Please read the following principle carefully:

The process is iterative. [redacted]

[redacted]

Given your knowledge and experience with User-Centered Design and using the scale below, please rate the following statement for each feature listed next:

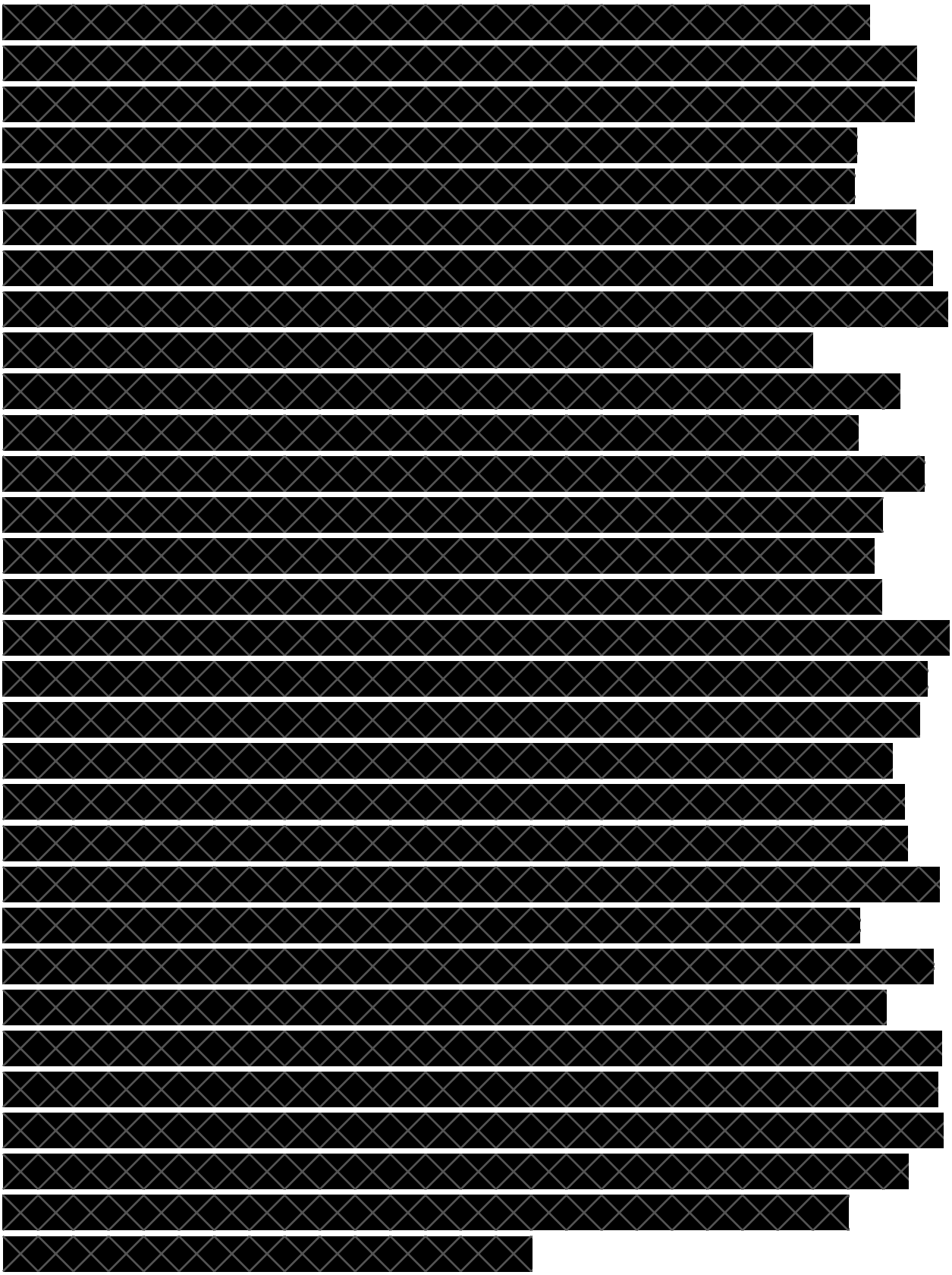
This concept is encapsulated in the **The process is iterative** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Adaptability	○	○	○	○	○
Customer benefit	○	○	○	○	○
Iterative	○	○	○	○	○
Reduce uncertainty	○	○	○	○	○

Given the previous principle, **The process is iterative**, can you identify any additional feature(s) that might be encapsulated in it?

Please read the following principle carefully:

The design addresses the whole user experience. [redacted]



Given your knowledge and experience with User-Centered Design and using the scale below, please rate the following statement for each feature listed next:

This concept is encapsulated in the ***The design addresses the whole user experience*** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Customer satisfaction	○	○	○	○	○
Focus on user experience	○	○	○	○	○
Value	○	○	○	○	○

Given the previous principle, **The design addresses the whole user experience**, can you identify any additional feature(s) that might be encapsulated in it?

Please read the following principle carefully:

The design team includes multidisciplinary skills and perspectives.

- [redacted]
- [redacted]

[redacted]

Given your knowledge and experience with User-Centered Design and using the scale below, please rate the following statement for each feature listed next:

> This concept is encapsulated in the **The design team includes multidisciplinary skills and perspectives** principle.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Collaboration	○	○	○	○	○
Multi-disciplinary team	○	○	○	○	○
Multi-perspective team	○	○	○	○	○
People	○	○	○	○	○
Support	○	○	○	○	○

Given the previous principle, **The design team includes multidisciplinary skills and perspectives**, can you identify any additional feature(s) that might be encapsulated in it?

APPENDIX E – KEYNOTE OF WORKSHOP ON FEATURES AND PRACTICES

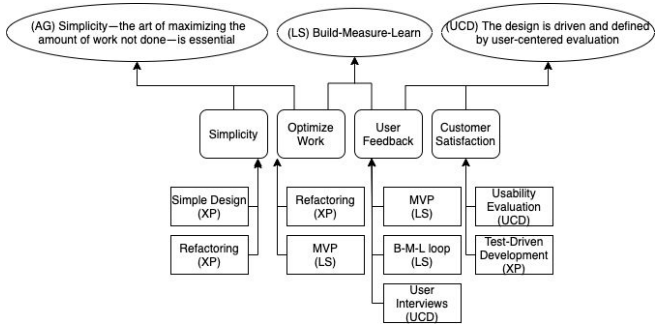
Workshop on Lean Startup and UCD features

Introduction

Team

Maximilian N. S. Zorzetti
MSc student

Goal



5

Schedule

14:00 – Start of workshop

30 min – Features

1h15m – Practices

5-minute break

15:55 – Wrap up
16:00 – End of workshop

Features

6

7

Features to include

Consider the **Entrepreneurship is management** principle of Lean Startup.

1. Should it be represented by the **Competitiveness** feature?

8

Features to include

Consider the **Validated Learning** principle of Lean Startup.

1. Should it be represented by the **Sustainability** feature?

9

Features to include

Consider the **Validated Learning** principle of Lean Startup.

1. Should it be represented by the **Frequent deliveries** feature?

10

Features to include

Consider the **Validated Learning** principle of Lean Startup.

1. Should it be represented by the **Value** feature?

11

Features to include

Consider the **Entrepreneurs are everywhere** principle of Lean Startup.

1. Should it be represented by the **People** feature?

12

Features to include

Consider the **Build-Measure-Learn** principle of Lean Startup.

1. Should it be represented by the **Built-in improvement of efficiency and behavior** feature?

13

Features to include

Consider the **Build-Measure-Learn** principle of Lean Startup.

1. Should it be represented by the **Optimize work** feature?

14

Features to include

Consider the **Innovation Accounting** principle of Lean Startup.

1. Should it be represented by the **Focus on innovation** feature?

15

Features to include

Consider the **Innovation Accounting** principle of Lean Startup.

1. Should it be represented by the **Measure empirically** feature?

16

Features to include

Consider the **The design is driven and refined by user-centred evaluation** principle of User-Centered Design.

1. Should it be represented by the **Measure empirically** feature?

17

Features in multiple principles

Consider the **Adaptability** feature for Lean Startup.

1. Which of the following principles is it more prominent in?
    - **Entrepreneurship is management**
    - **Build-Measure-Learn**

18

Features in multiple principles

Consider the **Customer satisfaction** feature for User-Centered Design.

1. Which of the following principles is it more prominent in?
    a. **The design is based upon an explicit understanding of users, tasks and environment**
    b. **The design is driven and refined by user-centred evaluation**
    c. **The design addresses the whole user experience**

19

Features in multiple principles

Consider the **Reduce uncertainty** feature for User-Centered Design.

1. Which of the following principles is it more prominent in?
    a. **The design is driven and refined by user-centred evaluation**
    b. **The process is iterative**

20

Features in multiple principles

Consider the **Support** feature for User-Centered Design.

1. Which of the following principles is it more prominent in?
    a. **Users are involved throughout design and development**
    b. **The design team includes multidisciplinary skills and perspectives**

21

Suggested features

Consider the **Validated Learning** principle of Lean Startup, represented by the features below:

Sustainability, **Reduce uncertainty**, **Customer satisfaction**, **Frequent deliveries**, and **Value**

1. Should any of the following features be used to represent this principle? If so, which ones?
    a. **Business satisfaction**

Practices

23

Mapping practices to features

Consider the **Adaptability** feature. What practices contribute to it?

Continuous Integration
Frequent Release
Iteration
Iterative Development
MVP
Simple Design
Small Releases
Spike

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

24

Mapping practices to features

Consider the **Built-in improvement of efficiency and behavior** feature. What practices contribute to it?

Daily Meeting
Retrospective
Three Questions

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

25

Mapping practices to features

Consider the **Business development** feature. What practices contribute to it?

Backlog Grooming
Brainstorm
Continuous Delivery
Customer Archetype
Discovery & Framing
Frequent Release
How Might We
Lean Canvas
Leap of Faith
Persona
Prototype
Spike

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

26

Mapping practices to features

Consider the **Collaboration** feature. What practices contribute to it?

Backlog Grooming
Collective Ownership
Daily Meeting
Design Studio
Hopes and Fears
Pair Programming
Planning Poker
Sign Up
Three Questions
Two by Two Matrix

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

27

Mapping practices to features

Consider the **Communication** feature. What practices contribute to it?

Backlog Grooming
Daily Meeting
Hopes and Fears
Iteration Planning Meeting
Lean Canvas
Mockup
Office Standup
Pair Programming
Persona
Planning Poker
Retrospective
Tech Talks
Three Questions
Two by Two Matrix
User Flow
Wireframe

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

28

Mapping practices to features

Consider the **Competitiveness** feature. What practices contribute to it?

Continuous Delivery
Continuous Integration
Discovery & Framing
Frequent Release
How Might We
Iteration
Iterative Development
Lean Canvas
Leap of Faith
MVP
Simple Design
Small Releases
Two by Two Matrix

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

29

Mapping practices to features

Consider the **Continuous delivery** feature. What practices contribute to it?

BDD
Code Review
Continuous Delivery
Continuous Integration
Iteration
Iterative Development
MVP
Small Releases
TDD
Timebox
Unit Testing

Agile	Lean Startup	UCD
Backlog	~~Continuous Delivery~~	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
~~BDD~~	Leap of Faith	Dirty Map
Burndown Chart	~~MVP~~	Discovery & Framing
~~Code Review~~	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
~~Continuous Integration~~	Simple Design	How Might We
Daily Meeting	~~Small Releases~~	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
~~Iteration~~	~~TDD~~	Prototype
~~Iterative Development~~	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	~~Timebox~~	User Flow
Pair Programming	~~Unit Testing~~	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

30

Mapping practices to features

Consider the **Customer benefit** feature. What practices contribute to it?

Backlog
Backlog Grooming
BDD
Planning Poker
Two by Two Matrix

Agile	Lean Startup	UCD
~~Backlog~~	Continuous Delivery	Affinity Map
~~Backlog Grooming~~	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
~~BDD~~	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	~~Two by Two Matrix~~
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
~~Planning Poker~~	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

31

Mapping practices to features

Consider the **Customer satisfaction** feature. What practices contribute to it?

Backlog
Backlog Grooming
BDD
Definition of Done
Definition of Ready
Journey Map
Mockup
Planning Poker
TDD
User Stories
Wireframe

Agile	Lean Startup	UCD
~~Backlog~~	Continuous Delivery	Affinity Map
~~Backlog Grooming~~	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
~~BDD~~	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
~~Definition of Done~~	Spike	Journey Map
~~Definition of Ready~~	Sustainable Pace	~~Mockup~~
Frequent Release	Task Board	Persona
Iteration	~~TDD~~	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	~~Wireframe~~
~~Planning Poker~~	~~User Stories~~	
Point/Relative Estimates	Velocity	
Refactoring		

32

Mapping practices to features

Consider the **Design based on context of use** feature. What practices contribute to it?

Affinity Map
Discovery & Framing
Ethnographic Research
Interview
Journey Map
Service Blueprint

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	~~Affinity Map~~
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	~~Discovery & Framing~~
Code Review	Retrospective	~~Ethnographic Research~~
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	~~Interview~~
Definition of Done	Spike	~~Journey Map~~
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	~~Service Blueprint~~
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

33 Mapping practices to features

Consider the **Early deliveries** feature. What practices contribute to it?

MVP
Refactoring
Simple Design
Small Releases

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

34 Mapping practices to features

Consider the **Efficiency (for conveying information)** feature. What practices contribute to it?

Backlog
Backlog Grooming
Lean Canvas
Mockup
Persona
Planning Poker
Tech Talks
Two by Two Matrix
User Flow
User Stories
Wireframe

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

35 Mapping practices to features

Consider the **Focus on innovation** feature. What practices contribute to it?

Brainstorm
Design Studio
Discovery & Framing
How Might We
Leap of Faith
MVP
Prototype

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

36 Mapping practices to features

Consider the **Focus on technical excellence** feature. What practices contribute to it?

Code Review
Collective Ownership
Continuous Integration
Iteration
Iterative Development
Pair Programming
Refactoring
TDD
Tech Talks
Unit Testing

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

37

Mapping practices to features

Consider the **Focus on user experience** feature. What practices contribute to it?

Affinity Map
Ethnographic Research
Interview
Journey Map
Mockup
Persona
Prototype
Service Blueprint
User Flow
User Stories
Wireframe

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

38

Mapping practices to features

Consider the **Frequent Deliveries** feature. What practices contribute to it?

BDD
Continuous Delivery
Continuous Integration
Frequent Release
Iteration
Iterative Development
Refactoring
Simple Design
Timebox

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

39

Mapping practices to features

Consider the **Good environment** feature. What practices contribute to it?

Collective Ownership
Daily Meeting
Three Questions
Sustainable Pace
Hopes and Fears
Office Standup

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

Break

be back by 15:25, please!

41

Mapping practices to features

Consider the **Iterative** feature. What practices contribute to it?

Continuous Delivery
Continuous Integration
Daily Meeting
Frequent Release
Iteration
Iteration Planning Meeting
Iterative Development
Prototype
Retrospective
Sustainable Pace
Timebox

Agile	Lean Startup	UCD
Backlog	~~Continuous Delivery~~	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	~~Retrospective~~	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
~~Continuous Integration~~	Simple Design	How Might We
~~Daily Meeting~~	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	~~Sustainable Pace~~	Mockup
~~Frequent Release~~	Task Board	Persona
~~Iteration~~	TDD	~~Prototype~~
~~Iterative Development~~	Tech Talks	Service Blueprint
~~Iteration Planning Meeting~~	Three Questions	Two by Two Matrix
Office Standup	~~Timebox~~	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

42

Mapping practices to features

Consider the **Measure empirically** feature. What practices contribute to it?

Continuous Delivery
Mockup
MVP
Prototype
Spike

Agile	Lean Startup	UCD
Backlog	~~Continuous Delivery~~	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	~~MVP~~	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	~~Spike~~	Journey Map
Definition of Ready	Sustainable Pace	~~Mockup~~
Frequent Release	Task Board	Persona
Iteration	TDD	~~Prototype~~
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

43

Mapping practices to features

Consider the **Measure progress via deliverables** feature. What practices contribute to it?

Burndown Chart
Continuous Delivery
Point/Relative Estimates
Velocity

Agile	Lean Startup	UCD
Backlog	~~Continuous Delivery~~	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
~~Burndown Chart~~	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
~~Point/Relative Estimates~~	~~Velocity~~	
Refactoring		

44

Mapping practices to features

Consider the **Motivated individuals** feature. What practices contribute to it?

Continuous Delivery
Daily Meeting
Hopes and Fears
Office Standup
Sign Up
Tech Talks
Three Questions

Agile	Lean Startup	UCD
Backlog	~~Continuous Delivery~~	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	~~Sign Up~~	~~Hopes and Fears~~
Continuous Integration	Simple Design	How Might We
~~Daily Meeting~~	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	~~Tech Talks~~	Service Blueprint
Iteration Planning Meeting	~~Three Questions~~	Two by Two Matrix
~~Office Standup~~	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

45

Mapping practices to features

Consider the **Multi-disciplinary team** feature. What practices contribute to it?

Balanced Team

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

46

Mapping practices to features

Consider the **Multi-perspective team** feature. What practices contribute to it?

Balanced Team

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

47

Mapping practices to features

Consider the **Optimize work** feature. What practices contribute to it?

MVP
Refactoring
Spike
TDD

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

48

Mapping practices to features

Consider the **People** feature. What practices contribute to it?

Balanced Team
Daily Meeting
Design Studio
Hopes and Fears
Office Standup
Pair Programming
Sign Up
Three Questions

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

49 Mapping practices to features

Consider the **Reduce uncertainty** feature. What practices contribute to it?

BDD
Continuous Delivery
Discovery & Framing
Ethnographic Research
Interview
Iterative Development
Journey Map
Leap of Faith
MVP
Service Blueprint
Spike

Agile	Lean Startup	UCD
Backlog	~~Continuous Delivery~~	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
~~BDD~~	~~Leap of Faith~~	Dirty Map
Burndown Chart	~~MVP~~	~~Discovery & Framing~~
Code Review	Retrospective	~~Ethnographic Research~~
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	~~Interview~~
Definition of Done	~~Spike~~	~~Journey Map~~
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
~~Iterative Development~~	Tech Talks	~~Service Blueprint~~
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

50 Mapping practices to features

Consider the **Self-organization** feature. What practices contribute to it?

Balanced Team
Collective Ownership
Daily Meeting
Pair Programming
Planning Poker
Sign Up
Tech Talks

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
~~Balanced Team~~	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
~~Collective Ownership~~	~~Sign Up~~	Hopes and Fears
Continuous Integration	Simple Design	How Might We
~~Daily Meeting~~	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	~~Tech Talks~~	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
~~Pair Programming~~	Unit Testing	Wireframe
~~Planning Poker~~	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

51 Mapping practices to features

Consider the **Simplicity** feature. What practices contribute to it?

BDD
Continuous Delivery
Continuous Integration
Frequent Release
Iteration
Iterative Development
MVP
Simple Design

Agile	Lean Startup	UCD
Backlog	~~Continuous Delivery~~	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
~~BDD~~	Leap of Faith	Dirty Map
Burndown Chart	~~MVP~~	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
~~Continuous Integration~~	~~Simple Design~~	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
~~Frequent Release~~	Task Board	Persona
~~Iteration~~	TDD	Prototype
~~Iterative Development~~	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

52 Mapping practices to features

Consider the **Support** feature. What practices contribute to it?

Balanced Team
BDD
Design Studio
Hopes and Fears
How Might We
Pair Programming
Tech Talks

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
~~Balanced Team~~	Lean Canvas	~~Design Studio~~
~~BDD~~	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	~~Hopes and Fears~~
Continuous Integration	Simple Design	~~How Might We~~
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	~~Tech Talks~~	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
~~Pair Programming~~	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

53 Mapping practices to features

Consider the **Sustainability** feature. What practices contribute to it?

BDD
Burndown Chart
Code Review
Continuous Delivery
Continuous Integration
Discovery & Framing
Frequent Release
Iteration
Iterative Development
Lean Canvas
Point/Relative Estimates
Small Releases
Sustainable Pace
Taskboard
TDD
Timebox
Velocity

Agile	Lean Startup	UCD
Backlog	~~Continuous Delivery~~	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	~~Lean Canvas~~	Design Studio
~~BDD~~	Leap of Faith	Dirty Map
~~Burndown Chart~~	MVP	~~Discovery & Framing~~
~~Code Review~~	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
~~Continuous Integration~~	Simple Design	How Might We
Daily Meeting	~~Small Releases~~	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	~~Sustainable Pace~~	Mockup
~~Frequent Release~~	~~Task Board~~	Persona
~~Iteration~~	~~TDD~~	Prototype
~~Iterative Development~~	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	~~Timebox~~	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
~~Point/Relative Estimates~~	~~Velocity~~	
Refactoring		

54 Mapping practices to features

Consider the **Trust** feature. What practices contribute to it?

Collective Ownership
Hopes and Fears
Sign Up

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
~~Collective Ownership~~	~~Sign Up~~	~~Hopes and Fears~~
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

55 Mapping practices to features

Consider the **User feedback** feature. What practices contribute to it?

Continuous Delivery
Interview
Mockup
MVP
Prototype
Wireframe

Agile	Lean Startup	UCD
Backlog	~~Continuous Delivery~~	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
BDD	Leap of Faith	Dirty Map
Burndown Chart	~~MVP~~	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	~~Interview~~
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	~~Mockup~~
Frequent Release	Task Board	Persona
Iteration	TDD	~~Prototype~~
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	~~Wireframe~~
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

56 Mapping practices to features

Consider the **User involvement** feature. What practices contribute to it?

Design Studio
Discovery & Framing
Interview
Journey Map
Mockup
Prototype
Wireframe

Agile	Lean Startup	UCD
Backlog	Continuous Delivery	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	~~Design Studio~~
BDD	Leap of Faith	Dirty Map
Burndown Chart	MVP	~~Discovery & Framing~~
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	~~Interview~~
Definition of Done	Spike	~~Journey Map~~
Definition of Ready	Sustainable Pace	~~Mockup~~
Frequent Release	Task Board	Persona
Iteration	TDD	~~Prototype~~
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	~~Wireframe~~
Planning Poker	User Stories	
Point/Relative Estimates	Velocity	
Refactoring		

57

Mapping practices to features

Consider the **Value** feature. What practices contribute to it?

Backlog
BDD
Continuous Delivery
Discovery & Framing
How Might We
Lean Canvas
User Stories

Agile	Lean Startup	UCD
~~Backlog~~	~~Continuous Delivery~~	Affinity Map
Backlog Grooming	Customer Archetype	Brainstorm
Balanced Team	~~Lean Canvas~~	Design Studio
~~BDD~~	Leap of Faith	Dirty Map
Burndown Chart	MVP	~~Discovery & Framing~~
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	~~How Might We~~
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	Sustainable Pace	Mockup
Frequent Release	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
Planning Poker	~~User Stories~~	
Point/Relative Estimates	Velocity	
Refactoring		

58

Mapping practices to features

Consider the **Business satisfaction** feature. What practices contribute to it?

Backlog Grooming
BDD
Burndown Chart
Continuous Delivery
Frequent Release
Planning Poker
Point/Relative Estimates
Sustainable Pace
User Stories

Agile	Lean Startup	UCD
Backlog	~~Continuous Delivery~~	Affinity Map
~~Backlog Grooming~~	Customer Archetype	Brainstorm
Balanced Team	Lean Canvas	Design Studio
~~BDD~~	Leap of Faith	Dirty Map
~~Burndown Chart~~	MVP	Discovery & Framing
Code Review	Retrospective	Ethnographic Research
Collective Ownership	Sign Up	Hopes and Fears
Continuous Integration	Simple Design	How Might We
Daily Meeting	Small Releases	Interview
Definition of Done	Spike	Journey Map
Definition of Ready	~~Sustainable Pace~~	Mockup
~~Frequent Release~~	Task Board	Persona
Iteration	TDD	Prototype
Iterative Development	Tech Talks	Service Blueprint
Iteration Planning Meeting	Three Questions	Two by Two Matrix
Office Standup	Timebox	User Flow
Pair Programming	Unit Testing	Wireframe
~~Planning Poker~~	~~User Stories~~	
~~Point/Relative Estimates~~	Velocity	
Refactoring		

Wrap up

59

That's all, thank you!

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br