

Investigating Problem Definition and End-User Involvement in Agile Projects that Use Lean Inceptions

Bruna Ferreira,
Marcos Kalinowski
Pontifical Catholic University of Rio
de Janeiro (PUC-Rio)
Rio de Janeiro, Rio de Janeiro
{bferreira,kalinowski}@inf.puc-rio.br

Marcos V. C. Gomes,
Marcelo C. Marques
Petrobras
Rio de Janeiro, Rio de Janeiro
{mvcg,marcelo.marques}@petrobras.com.br

Hélio Lopes,
Simone D. J. Barbosa
Pontifical Catholic University of Rio
de Janeiro (PUC-Rio)
Rio de Janeiro, Rio de Janeiro
{lopes,simone}@inf.puc-rio.br

ABSTRACT

Agile methods focus on high quality, frequent deliveries, and customer and development team communication. Despite emphasizing customer involvement, in practice end users commonly do not participate in the early stages of these methods; instead, other business-related customers represent them in defining end-user needs. As a result, the produced software often does not satisfy the actual users' needs, and the User Experience (UX) suffers. User-centered design, Design Thinking, and Lean Startup have been combined with agile methods to support defining the problem focused on user needs and producing effective software. Lean Inception (LI) represents a workshop that combines Design Thinking and Lean Startup. However, the effectiveness of problem definition and end-user participation in LIs is little explored. This paper presents a case study addressing this gap, investigating the use of LIs to define the problem and how the participation of end users in this context contributes to the resulting UX. We conducted ten semi-structured interviews with customers and end-users of an agile software project using LI. Our analysis suggests that, while LI supports the identification of business goals, more details about user needs and software requirements are needed to support designers and developers in understanding the problem and evaluating its feasibility. We also identified a mischaracterization of user journeys and personas in LIs due to the short time dedicated to their creation, hampering the discussion of relevant UX issues. Finally, we observed that, when end users were present in the early stages, they helped the development team to design software that fits into their routine, improving the UX and increasing its adoption. We qualitatively discuss these issues and gains of directly involving end users and not "proxy" stakeholders in LI's.

CCS CONCEPTS

• **Software and its engineering** → *Software creation and management; Designing software; Software design engineering;*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SBQS '21, November 8-11, 2021, Vitoria, ES, Brazil

© 2021 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM... \$15.00

<https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx>

KEYWORDS

agile, design thinking, user centred design, user experience, lean inception

ACM Reference Format:

Bruna Ferreira., Marcos Kalinowski, Marcos V. C. Gomes., Marcelo C. Marques, and Hélio Lopes., Simone D. J. Barbosa. 2021. Investigating Problem Definition and End-User Involvement in Agile Projects that Use Lean Inceptions. In *SBQS '21: 35th Brazilian Symposium on Software Quality*. ACM, New York, NY, USA, 10 pages. <https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx>

1 INTRODUCTION

Agile methods and practices are widely employed in the software industry. They aim to develop software with high quality, promote constant customer and team communication and support continuous deliveries and fast actions to changes [2]. Despite their benefits, there is still little participation of end-users in agile methods and low effectiveness of the product in their routines [23]. Agile methods promote communication with customers. Usually, the customers are "proxies" of end users and represent them on validations of deliveries. However, they do not know the actual problems that end users (will) face in their routine and they may not be directly affected by software shortcomings. Even though agile is an interactive approach and it performs user tests, in practice, it does not focus on designing the user experience (UX) based on users' needs [2]. Also, agile does not guarantee that the correct software will be developed [29].

Due to their complexity and the demand for velocity, user research is not always appropriately performed in agile processes [22]. Zaina et al. [28] present challenges in integrating UX information in agile projects: (i) most of the information integrated into the software considers the users' *interactions* with the product, but not their *needs and goals*; (ii) questions about UX are handled verbally and not represented in artifacts or documentation, making the flow of information more complex; (iii) usually, questions about UX are based on the team's experience and not on UX-specific models, theory, or tools. Incorporating user feedback in agile development processes is not trivial, because of the fast deliveries, and usually, because the team collects feedback from few users [17]. Additionally, the feedback is collected only at the final delivery. Difficulties in using UX methods lead software engineers into adapting artifacts familiar to them to support UX activities [28]. And the lack of clear procedures makes the team rationalize their own experience [27].

Integrating User-Centered Design (UCD) and Design Thinking (DT) with agile methodologies improves user participation and

adoption of UX design activities in the development process [8]. Also, Lean Startup supports understanding business strategy and generating MVPs (Minimum Viable Products) [8, 23]. DT and UCD have been integrated into agile projects [7, 23, 26, 28] so that the product is more focused on users' needs. User-centered approaches focus on designing and evaluating interactive systems for human use and have devised several techniques and methods to develop usable systems [21]. Adopting user-centered approaches aims to improve the quality of usability and UX for the end-user. Design Thinking is considered a user-centered approach. However, end users usually do not participate in early phases of the DT process as they do in other UCD approaches [17]. Also, DT involves combining three points of view: business goals, customer needs, and technical viability [10]. As such, user-related issues are not the center of discussions [17]. One of the challenges when adopting UCD or DT combined with agile projects is the lack of detailed documentation on integrating these approaches and which roles should participate in each stage [29]. In this paper, we report a practical experience in a cooperation project between academia and a large publicly-held company that operates in the oil, gas, and energy industry in Brazil with the use of Lean Inception (LI) [4] workshops in the context of agile development.

LI is a workshop that combines Lean Startup and DT methodologies and aims to generate a product MVP based on the alignment of customer expectations, users, and the development team. However, there are few studies about LI. We choose LI because the software industry uses it and its creators provide material and examples of usage [4] to support its adoption. As the book describing LI [4] does not explicitly define how each role (business, users, technical) should be involved in each activity in the workshop, we have included end users in all activities of the LI. In this context, we conducted a case study where we used semi-structured interviews to investigate the following issues: RQ1 – What are the lessons learned when using LI to understand the problem domain? and RQ2 – Did the users who participated in the development process have a better perception of UX with the final product? The interviewees are ten stakeholders of the project: five customers and five end users –two of end users participated in all activities, one in tests and two did not participate in any activity.

As a result, we identified that LIs seem to be more suitable for identifying business goals and a coarse grained list of features and it does not focus on UX issues. In this way, it may be feasible to shorten the workshop to focus on less detailed business information. This would allow reducing the time and help to focus on identifying the main goals and some features of the project. Also, these can guide the following stages of modeling the problem and converging to solutions. Activities related to UX design and analysis of software requirements (such as personas and user journeys) need detailed information. The development team can carry out other phases with users after the LI workshop to understand users' needs and identify software requirements. Regarding users' participation in the process, we identified that users who participated in the process had a better perception of how to fit the product into their work routine and saw more value in the developed software.

This paper is organized as follows: In section 2, we present concepts related to Lean Startup, UCD, and DT, and work that combines these methodologies with agile practices. In section 3, we explain

the case in which we ran LI workshops. In section 4, we present the research questions and data collection and analysis procedures. section 5 presents the results, followed by the discussion of the results in section 6. Finally, in section 7 we make some concluding remarks and point to future work.

2 COMBINING UCD, DESIGN THINKING AND LEAN STARTUP WITH AGILE METHODS

As previously mentioned, Agile development processes have been integrated with other methodologies such as UCD, DT, and Lean Startup because, despite their benefits, there are still some gaps in their application. There is still little end-user participation in agile projects and little effectiveness of the product in the users' routines [23]. Software development processes focus on developing functional products that anticipate system requirements [6]. In this context, DT and UCD aim to support the design of the product based on the identification of user needs, and Lean Startup supports the understanding of the business and its strategies [8].

User-Centered Design focus on the design, development, and evaluation of interactive systems for human use and have developed several techniques and methods, which can be applied to develop usable systems [17]. According to ISO 13407 [11, 20], UCD consists of four activities: (i) understanding and specifying the context of use, (ii) specifying user and organizational requirements, (iii) developing design solutions, and (iv) evaluating the design according to the requirements. Adopting user-centered approaches can improve usability, user experience, and support in meeting user needs.

DT is a design-driven methodology with origins in the business world to improve their processes [3]. DT was popularized by IDEO and IBM and is viewed as a user-centered approach that considers the point of view of several stakeholders in its process [17]. Professionals in the software industry widely adopt it. However, there are few examples in the literature on the use of its techniques in the software industry, and there are few empirical studies to validate its application [3]. Despite being considered a user-centered approach, DT does not place the end user as a stakeholder at the beginning of the process, as it occurs in the co-creation workshop [14, 17]. It is essential to include end users in the development process from the beginning, as they know better than anyone their work context [17]. Therefore, it is essential to define and evaluate user participation in DT processes.

The Lean Startup Methodology is based on the Lean philosophy and concepts of startups. The Lean concept emerged in the automotive industry, focusing on maximizing value and minimizing waste in production processes [25]. Lean is based on five principles: (i) creating value for the user, (ii) eliminating waste, (iii) optimizing value flows, (iv) empowering people, and (v) continuous improvement [18]. The startup is a human institution designed to create new products in conditions of extreme uncertainty [16]. Lean Startup focuses on creating entrepreneurial hypotheses embedded in a first version of the business [9]. These hypotheses are tested on a set of *minimum viable products* (MVPs) [9].

2.1 Related Work

Some works have proposed methods integrating UCD, DT, and Lean Startup. They aim to facilitate the understanding of the problem,

align expectations between product stakeholders and the development team, and bring users closer to the agile development process. Dobrigkeit et al. [7] proposed the Innodex method, integrating Lean Startup and Design Thinking into Scrum to develop innovative customer-oriented products. They evaluated it through interviews, which revealed that participants perceived the method as easy to understand and as adding value to the process. Their work presents the techniques used and the activities carried out in each step but it does not present guides or examples of how to carry out each step of the method in projects in the industry.

Signoretti et al. [23] presented a case study in which they investigated how UCD and Lean Startup are used in the work routine of two agile teams in the software industry. They identified that these methodologies resulted in changes to the teams' mindset: As they started to work focusing on the problem to be solved, the teams shared a vision about the product, had more significant interactions with stakeholders, and gained a better awareness of each member's responsibility. Those methodologies also included activities to improve user participation.

Lean UX is a method that combines the principles of Lean Startup, DT, and Agile. It focuses on solving real problems using user-centered approaches to avoid waste in the process and to improve UX [19]. By adopting Lean UX, the development team builds MVPs based on real user needs [1]. At every sprint, tests are performed with users to validate the generated MVPs [15]. Aarlien and Colomo-Palacios [1], in a review of the literature on Lean UX, discussed its benefits and challenges. As benefits, they identified that Lean UX is more adjusted to the market and encourages team collaboration. The challenges they identified were: maintaining clear communication is not easy – the more time to pass on knowledge, the more the context is lost –; and the lack of design documentation can lead to communication failures.

Converge [26] is a model based on DT, Lean Startup, and agile practices. Based on the D.school's methodology, Ximenes et al. [26] proposed an iterative process consisting of the phases empathize, define the problem, ideate, prototype, and test. They evaluated it through a case study in the context of a startup. They concluded that the model was well accepted by the testing and development team and helped the team to understand the users' mental model and needs. The testers of the final product considered that combining the methodologies was productive for the process.

In this paper, we investigate the use of the Lean Inception (LI) workshop. This workshop [4] combines practices from Lean Startup and Design Thinking methodologies. LI has several steps to align expectations and direct the development team and stakeholders towards the MVP. The name Lean Inception is derived from the RUP (Rational Unified Process)[13]: its first stage is called Inception and includes analysis of the project's goals, architecture, and planning. LI has some characteristics of DT, including personas and user journeys. Also, LI is based on Lean Startup, adopting the idea of converging to the MVP. In this way, the name Lean Inception means that the Inception must be short to be carried out (maximum one week) and enough to outline the MVP [4].

2.1.1 Example of Lean Inception Application. Teixeira et al. [24] present an example of using LI in Agile Projects, as part of the Lean

R&D approach [12]. In this project, the team developed a dashboard to support the identification of odors emitted by a refinery. The first phase carried out was the LI. They detailed requirements through low-fidelity and high-fidelity prototyping. The prototypes supported the implementation of the dashboard. At the end of the process, the prototypes were evaluated with customers. Participants through the process are customers and the development team participated. The end-users participated at the end of the development, where the team presented demonstrations of the functional dashboard. Due to lack of user participation in detailing meetings and LI, there was a great effort to identify how to fit the dashboard into the users' routine at the end of the project. Since users did not participate in the LI, they did not know the purpose of the dashboard, its benefits, and how to use it. Besides, there was resistance to start using the tool. As a result, the authors identified that it is necessary to encourage greater user participation in the process and improving contextualizing the problem domain to make LI more productive and focused. In our work, we applied these lessons learned in LI to evaluate their effectiveness.

3 APPLYING LEAN INCEPTION IN A PROJECT OF SIMULATION AND OPTIMIZATION

This section presents the stages and participants of an agile project of simulation and optimization. Regarding the nomenclature used in the paper, we use *development process* to encompass design, implementation, and testing activities with users; *customer* for participants who know the processes but will not be end users of the product; and *users* for the actual end users. In this project, the goal was to develop a set of tools to support using the digital simulator of an oil refinery. Two tools were developed: (1) a collaborative system to support the process of integrating the information needed for the optimization process and (2) a tool whose goals were to develop an algorithm to automate the optimization process in order to make the process less costly for the leading optimizers and to maximize the financial gains.

The project was divided into two-week sprints, with daily meetings, planning, and sprint review meetings. In total, 24 sprints were performed. After the LI, the process was performed remotely due to the COVID-19 pandemic. To manage the project, we used the Azure DevOps tool, in which the backlog structure was composed of epics, features, user stories, and tasks. In this project, a virtual room was created named checkpoint where the development team had constant access to the project's stakeholders (including end users of the software) to resolve doubts and impediments to the activities. In the following, we describe all the steps taken in the development project before and after the LI (Figure 1).

Contextualization workshop: As Teixeira et al. [24] recommended to include an early meeting to improve the understanding of the problem domain and make the LI more productive, we added a *contextualization workshop* to the process. In the previous project, there was no contextualization. Therefore, at the end of the LI, an existing system was presented. It was the same as the one being conceived. It was necessary to review all the content produced to understand what features would be developed.

The contextualization workshop lasted seven hours. The customers presented, to the development team, business and technical

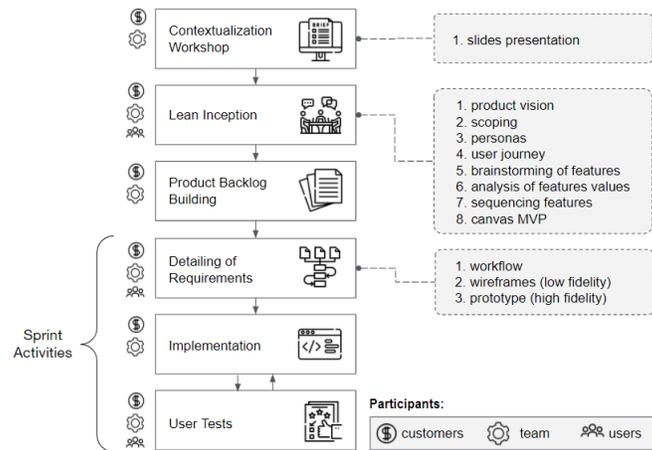


Figure 1: Stages of the project

concepts related to the domain. In the end, the team's scrum master made a brief presentation about the LI, where they explained the LI activities as preparation for the LI workshop. **Running the Lean Inception:** After the contextualization workshop, the LI workshop took place, during three days. The workshop participants were customers who represented the business goals, end users called Lead Optimizers (OTLs), and the software development team (Developers, UX and UI Designers, Researchers, Scrum Master, and Product Owner). The Lean Inception (LI) steps were applied based on the guidelines presented in [4]. The LI's facilitator was not a researcher nor one of the paper authors. He is certified at Caroli.org and has extensive experience in agile projects and in conducting LIs.

Before starting the workshop activities, the participants defined the business goals that guided the workshop. During the LI, we carried out the following activities: description of the product view, delimitation of the product scope (what the product is, what it is not, what it does, and what it does not), creation of personas to identify the users of the product and your needs, user journey, feature brainstorming, feature value analysis, sequencer to organize and prioritize product features, and the construction of the Canvas MVP to record the final LI result. As a final result, three features were prioritized as essential to meet the business goals: collaborative system, optimizer, and reliability portal.

Product Backlog Building: After the LI, we performed two Product Backlog Building (PBB) sessions. In these, the three features of the MVP were detailed for the identification of user stories. The sessions lasted two hours each and were held remotely using the Microsoft Teams tool. Participants in this stage were representatives of the customer, the product owner, the scrum master, and the development team.

In the first stage of building the backlog, the team and the customers defined user needs and problems to be addressed by each feature. We listed benefits and expected results and the necessary steps for the construction of each feature. From this information, the product owner directed the creation of user stories, in the format: AS a <user role>, I WANT <functionality> FOR <problem/goal>.

Detailing Requirements: This step was performed to support the developers and UX/UI designers in understanding the features. The first activity was to elaborate a workflow to understand the tasks performed by the users at work and who performed each task. We made notes of information needed to perform each task and other systems accessed to obtain this information in workflows.

Workflows supported the interaction design and provided a basis for identifying what information should be in the low-fidelity prototypes (wireframes). They also aimed to help the development team to understand what information should be stored in a database and what other databases would need to be accessed. Low-fidelity workflows and prototypes were developed collaboratively with the development team, UX/UI designers, users, and customers. After elaborating on low-fidelity prototypes, the designer created high-fidelity prototypes of the final user interface.

Software Implementation and User Tests: This phase refers to the software implementation. The first feature developed was a collaborative information system between users of the refinery, the second feature was an optimizer, and the third feature prioritized was a reliability portal. The features are the result of the first MVP created in the LI workshop.

To develop the first feature, we used Microsoft® SharePoint. However, after the first tests with users, the team observed that the tool had limitations and would not meet the users' needs. Also, the team had little experience with the tool, requiring learning and consuming an excessive amount of time, which could cause delays in the project. Then, we decided to build the second version of the collaborative system using React. Despite demanding more work, it would bring greater flexibility to code the user interface and improve the user experience and usability.

The team developed the interface for the second feature using Excel spreadsheets, so that the team could test the feature in less time. The feature was developed and tested quickly and brought more significant value to users and the company. Regarding the third feature, we opted to disregard it because the problem to be addressed was complex and involved understanding many concepts unknown to the development team.

3.1 Participants throughout the development process

Around 19 people participated in this project: customers, end users, and the development team composed of researchers, UX designers, developers, product owner, scrum master, and coordinators. Unlike the project reported by Teixeira. et al. [24], in our work end users participated in some phases of the LI, design, validation meetings, and tests with users. Their participation helped the team to identify the usefulness of the software from the beginning of the project. Also, they showed how the software would fit into their routine. In the results section, we discuss how end-user participation throughout the process influenced their experience with the developed tools.

4 METHOD

This case study aims to investigate the integration of LI with other phases of the process, mainly on issues related to the participation of users in the process and the understanding of the problem domain. In this context, two main research questions will be addressed: RQ1. What are the lessons learned when using LI to understand the problem domain? and RQ2. Users who participated in the development process had a better perception of UX with the final product? The main research questions comprise the following subquestions:

RQ1: "RQ1.1 What are the LI's contributions to the definition of the problem?" and "RQ1.2 How was the integration of LI with other stages of the development process?" and "RQ1.3 What elements of the problem domain emerged during the LI?".

RQ2: "RQ2.1 How was the user participation in the project?" and "RQ2.2 How was the users' experience with the final product?".

We conducted ten individual semi-structured interviews to collect data to answer the research questions, with five customers and five users. We conducted a pilot test with the interview scripts before data collection. Since the focus was on evaluating the problem definition and the end users' involvement, we interviewed customers who followed the entire project and users who used the final product and were available. Customers followed the progress of the project and were responsible for keeping the project in line with the business goals. End users contributed to understanding details of the processes they followed in their routine, identifying other actors involved, defining the problems and opportunities for improvement in these processes, and evaluating the final product. In all interviews, the Miro tool¹ was used to facilitate the conversation. During the interview, the customer participants filled in boxes related to the phases of the process carried out in the project and discussed the answers with the interviewer. In total, each participant completed five tables with the following themes: Contextualization, LI, Detailing of Requirements, Monitoring, and Development Meetings. In addition to the tables, the participants talked about each activity performed at the LI workshop.

To evaluate the experience with the produced software, we interviewed 5 users with experience in the leader optimizer function ranging from 3 to 4 years. Two users (P3 and P5) participated in the LI, design, and testing. One user (P2) participated only in the tests.

¹<https://miro.com>

Two other users (P4 and P1) got to know the tools through training and did not participate in any stage of the development process. During the interviews, users answered questions about the utility, ease of use, and attractiveness of the tools. Also, they answered questions about which tools were most useful, improvements in the tools, what they did not use in the tools, and what should remain as is.

5 RESULTS

This section reports the results of the interviews, organized by each research question. In our analysis, we used some procedures of Grounded Theory [5], especially open coding, *i.e.*, the process of adding meaning to data. After the open coding, another researcher reviewed the codes during a meeting. The next step was axial coding, where we grouped codes into categories. Two other researchers reviewed the identified categories. The results of the qualitative analysis were used to answer the research questions. The artifacts used in the study are available at <https://doi.org/10.5281/zenodo.5512643>.

5.1 RQ1.1. What are the LI's contributions to the definition of the problem?

The activities carried out at the LI helped the development team and stakeholders converge on the MVPs and their features. Nevertheless, those results were not effectively integrated into the development and UX/UI design activities because these required more detailed information. Next, we describe an overview about activities of LI.

Product Vision: this activity supported discussions among participants and helped to identify information about product weaknesses before development. However, an earlier step was missing to define which gap the product should fill so the activity would be better targeted. In addition, information about the problem and the solution must be filled in this activity. There was no practical exploration of the problem before thinking about solutions. The lack of guidance made it difficult to fill in the product vision template because the process the product should support was complex, and each participant focused on a different problem, making it difficult to converge towards a solution and consuming more time: "*when it starts with the product vision, an important stage may have been skipped, which was to consolidate which gap the product aims to fill*" - [P2, customer] and "*(...) there may have been something missing in defining the problem clearly*" - [P2, customer].

Scoping: The second activity for defining the scope (what the product: is/is not / does/does not) generated doubts. The participants had difficulty in differentiating what to fill in the items in the template: "*it did not work, we did not know what exactly to write on this board*" - [P1, customer]; "*how to define what you do or do not do without defining features?*" - [P2, customer]; and "*improving is/is not / does/does not, use another less confusing way or better exemplify what comes in each category*" - [P3, customer].

Personas: in LI, the Personas technique aims to identify the actors who would interact with the product and their needs. The personas created in LI workshops do not contain information to support the UX design directly. At this phase, the product was in the goal setting activity, and the personas were not adequately detailed for this moment: "*I did not find it very useful, I think I could have invested more time. The personas detailed information that did not*

contribute to the definition of the product" - [P3, customer]. Personas would be used more effectively if built at later stages of the project to identify features that could improve the user's experience with it. However, one of the end users cited that using personas helped to understand the roles of the users in the process: "Personas helped to target where each person fit in" - [P3, user].

User Journeys: it aims to represent how the user wants the product to fit into their work process. One of the benefits of a user journey cited by the participants was that it helped to identify user interactions with the process: "it worked well, they were easy to define. Essential to define interactions with the process" - [P5, customer]. In addition, it was important to help make an abstract idea more concrete: "useful, because you need to materialize the use of the product to get a sense if we are not to forget an important stage of the idea" - [P2, customer]. The journey also helped to identify other users who are essential to the process. The drawback was that, during the LI, we identified a new type of user, who had not participated in the process and did not discuss their needs: "In this part, we define the importance of the participation of the PP [new user] and the process analyst [new user] in the development and use in the tool. It was useful. It helped to see how dependent on the PP is the OTL [current user]" - [P3, customer].

Brainstorming of Features: participants identified a large amount of functionality that was not feasible for the established deadline of the project: "more features were identified than were important or what was feasible to do in the given time" - [P2, customer] and "improve the planning to know what is possible within the established deadline" - [P3, customer]. Similar to the product vision activity, an earlier planning stage was missing so that the workshop could start with an initial objective and be better directed.

User Journeys x Brainstorming of Features: Analyzing the relationship between journey and brainstorming in this project, we identified opposite perceptions of two stakeholders: "maybe brainstorming should come before the journey, the functionality must come from the product itself. The journey must come from what it [the product] does or does not do" - [P2, customer] and "it worked well: having the journey well defined, it was not difficult to define the functionalities" - [P5, customer]. One of the stakeholders thought that the functionality brainstorming should come before the journey, and another thought it should come later. In this context, we can identify different kinds of information: less detailed information related to objectives to guide the creation of the journeys; the journey itself, designed to understand the user's routine; and information to extract insights and requirements from the journey (e.g., functional requirements).

Analysis of Features Values: This activity was very divergent because it is strongly based on the personal opinions of the workshop participants: "assigning scores generally differs, it may not be a fair thing to average. Very personal opinion, the average may not represent the best option." - [P5, customer]. The features were not yet detailed at this stage, and the participants did not have enough knowledge to define the technical feasibility. In addition, the technical team did not have sufficient knowledge of the domain to judge the value for users and customers.

Sequencing Features: this activity was considered a good way to present and organize the features. However, as the features were not ordered according to priority, it was noticed that the second

feature developed brought more significant value to the stakeholders (both customers and end users): "it was in the wrong order but not due to the method. We should have started with the optimizer to generate the greatest benefit and indicate the highest priority information for the collaborative tool." - [P2, customer] and "it is an important step. It gave a good template for prioritizing activities" - [P3, customer].

Canvas MVP: it was considered a good activity to keep the focus and check if there were still gaps at the end of LI: "canvas is one of most important steps because it helps to improve the focus on what is going to be done" - [P3, customer]. However, other participants stated that the canvas created was too generic and was not used to help in the steps after: "It didn't work: I don't remember to put any relevant information that was used later" - [P1, customer].

5.2 RQ1.2. How was the integration of LI with other stages of the development process?

Next, we describe four categories related to steps of the process: Preparing for the Lean Inception, Problem definition, Detailing Requirements, Implementation.

Preparing for the Lean Inception: The contextualization workshop aims to make LI more productive and focused. However, we realized that the structure of the workshop was not appropriate for this. The presentations had many details out of context: "the meeting was lacking (...): in what we wanted to know" - [P2, customer].

Problem definition: the description of routines and needs in LI was essential to define the problem. It helped to identify difficulties and weaknesses: "process helped define the problem to be solved because it allowed the right time for people involved to describe the routine and needs. Describing routine helps to see difficulties and weaknesses" - [P3, customer]. However, our project had a complex domain that required more time for the development team to understand the problem: "Lean inception is good to define what to do and not how to do it. However, in the case of [the project name], it should have been explored how to do" - [P1, customer] and "it was difficult to get the problem understood, many items in the domain made the problem difficult" - [P2, customer]. Another point discussed in interviews was that the LI is very conceptual. It makes it difficult to understand the problem: "Ideation is very conceptual, the understanding of each one leads to misunderstandings, one person understands A, other understands B" - [P2, customer]. In short, the LI helped to expose the problem, but it was not enough for the team to understand the details needed to carry out the project.

During the LI, essential participants were identified to support the definition of the problem. These participants could be identified before contributing to the LI: "the number of people was good; there could be more people involved, but it is in the lean inception that we find that out – the discovery that communication with PP was important [Example]" - [P3, customer].

Detailing Requirements: We detailed the requirements for a sufficient understanding of the problem by the development team. Some difficulties emerged during this phase due to the complexity of the domain. It took a long time to understand the requirements. As a consequence, there was a delay in starting the development and producing results: "The project was very complex, it took a long time to explain the problem (...) it takes too long to produce a result, the

result may not be what I expected" - [P2, customer]. In the detailing stage, we used two main activities to improve the understanding of the problem: building workflows and low fidelity prototypes (wireframes). Workflows were useful to guide deliveries and help keep features in mind: *"Things started to go after we adopted the workflow method as a delivery guide"* - [P2, customer] and *"without workflows and wireframes, you might find that you forgot some part of the process only at the end of the sprint"* - [P3, customer]. A suggestion to improve the workflow elaboration activity was to present examples before the activity started to facilitate the targeting of the participants: *"bring an example of a simple process flow before starting"* - [P2, customer].

Workflows generated basic knowledge for the construction of wireframes: *"it went well. I did not know the tool, but it served as a basis to develop a graphical interface"* - [P4, customer]. In addition, some essential elements were identified for building workflows that help in understanding the problem domain: *"I missed a workflow linked in time, considering artifacts, actions, which records, what systems you need, who, how, what, why"* - [P2, customer]. The following elements were identified: workflow linked in time, artifacts generated/consumed at each stage, actions (what is performed), which records to be performed, other systems needed to the workflow, who performed each action, and why and how the actions are carried out.

The creation of wireframes helped to capture ideas and facilitated the explanation of terms and procedures for the development team: *"wireframes were good, they managed to capture the ideas and write terms that the development team would understand, and the calculation of sequences"* - [P4, customer]

The use of workflows and wireframes was important to materialize the ideas, visualize the concepts, and help in the discussions, because the information was still abstract after the LI. These activities facilitated the transformation of information into user requirements and *de facto* software requirements: *"It is important to materialize the idea through prototype and workflows, even those who proposed it begin to see what is being proposed is weird. It helps to improve the idea"* - [P2, customer] and *"Wireframes were important to visualize the concept, it facilitated the visualization of the concept. It facilitates the discussion"* - [P4, customer]. For the design in general, it was suggested to organize it in stages, merging design activities with development: *"break the design into several steps, interleave more the design with development. Create a full screen to test hypotheses and check if the technology was enough. Use learning to speed up later steps"* - [P1, customer].

Implementation: The software implementation did not present any significant technical difficulties. The biggest obstacles identified occurred due to difficulties in passing knowledge to the development team and decisions made in previous phases: *"the problem was not implementation, what made it difficult was that it took us a while to know in more detail what it was to implement: better define the user's needs, work process, business objectives and give conditions who will develop software can develop their work"* - [P1, customer]. One of the decisions that impacted software development was the choice of technologies because it limited the user interface development. After the implementation of the first version, which lasted one month and two weeks (3 sprints), the tool was presented to the first user, who was unable to use it. At that point, the tool would

hamper this user's work routine. It was then decided to redevelop the tool using another technology.

The technology chosen initially was not suitable for the intended purpose: *"(...) the interface was not very good, due to the poor suitability of the tool for the intended purpose"* - [P1, customer] and *"(...) the first impression was not good; in fact, at first, it seemed very primitive. (...) The tables and numbers formats were very strange (...)"* - [P5, user]

Another point reported was that the second feature developed to support optimization generated more value to the users' work and the company than the first feature. The first feature generated greater difficulties in the development and understanding of the problem. In this context, if the prioritization of the features had been carried out differently, the final product would have generated greater value in less time: *"(...) we worked 1.5 in [tool]. When we started the optimization algorithm, it was done faster and brought more gains."* - [P2, customer]. In addition to the customers, all end users interviewed agreed with this point.

5.3 RQ1.3. What elements of the problem domain emerged during LI?

Elements of the problem domain: In the initial phase of LI, we identified some problems, expected benefits, functionalities, and comparisons with existing solutions. The details of the problem has not yet been defined; however, the template deals with elements to understand the problem and elements for the solution. There is no separation between problem and solution definition to facilitate the understanding and guiding of the workshop. In the second stage of the LI, we started to define the scope in terms of product characteristics and functionalities. In this phase, participants had difficulties to identify functionalities because the problem remained undefined.

UX Issues: In next stages, the personas were not detailed. UX issues were not dealt with in personas and user journeys. Using these activities after the LI can generate more information to improve the UX. However, these activities help to make the problem less abstract during LI.

Level of Detailing Information: During the LI, participants had several doubts about the level of detail of the information at each stage. Goals, user needs, and software requirements were mixed at the various stages. Tracing between the information was lost, information was confusing, and at the end of the LI, the participants thought that the problem was understood. However, in the development stage, difficulties and lack of understanding arose. Analyzing the LI activities and the information produced throughout the process, we observe that the LI contributes to understanding the business goals, which has a higher level of abstraction and requires less time for identification. User needs and software requirements are not dealt with satisfactorily during the LI and require more details in later stages.

We identified that it is essential to structure the steps of eliciting and the documentation according to the level of detail of the information. Also, organizing it in levels according to the participants involved in the meetings can facilitate understanding the parts of the problem; for instance, exploring more general information such as business goals for customers and more details for the development team and user participation: *"(...) different levels of details to*

facilitate understanding by others. Easier to document and facilitate access for people Levels: more detailed for those who are going to develop, a less detailed level that assists in the discussions of conception and the presentation of ideas" - [P1, customer].

5.4 RQ2.1. How was the user participation in the project?

This section analyzes the users' participation in the process to answer the research question RQ2.1. There were three main activities in which five users participated: LI, detailing requirements, and final tests of the tool. Two of the five users participated in LI: they collaborated with all the activities for defining features and partially participated in the detailing of requirements.

Users helped mainly in the activity of elaborating workflows. Some items they helped to identify were: opportunities for integration with other systems from which they accessed data manually, other actors involved in the process, and necessary information as input for the tools. Three of the five users participated in the testing phase and helped identify usability problems that would impact their work before deploying the software. Two other users interviewed did not participate in any stage of the process. They had difficulties with the initial configuration of the system. However, they recognized the value generated by the tool in their routines and had no usability problems that impacted their work.

5.5 RQ2.2. How was the users' experience with the final product?

This section will report the experience of the five users interviewed with the software developed. We will describe an overview and the experience regarding three factors: ease of use, usefulness, and attractiveness.

Ease of Use: Regarding the ease of use, the users had doubts about integrating the optimizer and the collaborative system tools in the first use: *"It generated some doubts about the interaction between site, spreadsheet, and files. Could be more integrated" - [P1, user].* After the first use, the tool was considered easy to use.

During the initial tests, users considered the software complicated to understand: *"during the tests. It was not delivered; it was more complicated to understand. Compromised ease of use" - [P2, user].* After the first corrections, its use became easier. Also, they cited issues related to the organization of the user interface: *"When formatted and placed addresses, placed minimum and maximum, check address button made it easier to use" - [P2, user].*

Users who participated in the whole process had no difficulties in the initial integration of the tools: *"As I already knew the process, it was not difficult, the initial configuration was intuitive." - [P3, user].* However, some structural problems were identified that prejudice the usability in the collaborative system: *"even migrating (...), structural errors continued: unstable database, slowness, data formatting, cut data, tank ordination" - [P5, user].* By contrast, the optimizer, which was developed with greater participation from some users, was considerably more stable: *(...) was more friendly, stable, and didn't have many surprises about what would happen. - [P5, users].*

Usefulness: Regarding the usefulness, the tools were considered useful for some users since the first use: *"useful since the first use." - [P1, user]* and *"quite useful. Because it sped up the generation of*

data from optimization conditions regarding the execution of the optimization" - [P3, user]. Users who participated in the first tests of the collaborative system mentioned that, in the first use, the usefulness was questionable: *"it caused more work than it brought benefits because it caused additional work: Excel, data connection - technical problems - tests" - [P5, user].* This same user mentioned that the optimizer was useful. However, in the first few uses, the collaborative system continued to present data security problems that were resolved later.

All users interviewed mentioned that they would recommend the use of the optimizer to other users. It is a tool that saves time and makes work easier: *"Before, I could only be an OTL; today I can do other activities" - [P1, user]* and *"It makes work easier and reduces time." - [P5, user].* Regarding the collaborative system, users cited that the tool helps the work, but it can be improved. Due to the obstacles faced in the development, some functionalities specified for the collaborative system had to be removed to avoid schedule delays. Thus, the tool did not fulfill its goal of supporting collaboration.

Attractiveness: Users who did not participate in the process consider the tool attractive in its first use: *"site well set up, easy to navigate. The spreadsheet allows editing, can change functionality. Improve spreadsheet interface for easier viewing." - [P1, user].* At this point, the tools had already undergone refinements as tests had been carried out with other users who participated in the process. Users who participated in the tests did not find the collaborative system attractive in their first contact with the tool: *"it was in the testing phase, it was not formatted yet" - [P2, user].* After the corrections, this user considered that the attractiveness of the tools had improved: *"attractiveness improved when formatted and separated the tabs, clicked on the buttons, and returned the variables" - [P2, user].* The P5 did not find the system attractive in first use due to usability problems mentioned in the utility section.

6 DISCUSSION

Below are some insights obtained to help answer the two main questions in the context of this project: problem definition (RQ1) and user participation (RQ2) in the development process..

6.1 RQ1. What are the lessons learned when using LI to understand the problem domain?

We divided the lessons learned when using the LI to support the problem into three moments: (1) before the LI, where we discuss points identified for preparing the LI; (2) during LI, where points related to the activities carried out in the Lean Inception workshop are presented; and (3) After LI, where we present lessons related to how the problem was defined and the details of the requirements in other stages of the process.

6.1.1 Before Lean Inception.

Perform a LI preparation to assist in the focus: Before the LI, it is important to have an initial preparation to direct the workshop. This finding is in line with [24]. The preparation refers to understanding the application domain before carrying out the LI; as the LI has a well-defined and focused schedule, it does not allow for long discussions. In this way, the team missed many details

necessary for designing the application and evaluating the viability of the features.

Development team support in directing contextualization: The development team must define the information that they consider important to understand the application domain and help direct the LI. Long presentations are not effective, generate fatigue, and do not bring the desired information. Having relevant initial information to understand the problem can help fill out the LI templates and avoid wasting time with long discussions.

6.1.2 During the Lean Inception.

Definition of actors instead of detailing personas in LI: At LI, it is essential to gather information about which actors will participate in the process and what roles they will play so that end users understand where they fit into the problem. Personas should be dealt with in more detail at other stages of the process to identify profiles that support the UX design.

User journeys help materialize the idea: Using journeys during the LI aids the participants in visualizing the users' routine steps. It makes the ideation process less abstract.

Identify features from the user journey stages: We propose to define the steps to achieve the goals after the LI. In this way, the steps of the user routine remain aligned with the goals of the journey.

LI aims to define what to do and not how to do it: The LI's activities aim to align expectations and converge toward the MVP. Thus, during the LI, technical discussions that do not contribute to understanding business goals should be avoided. Technical definition steps and UX definitions should be dealt with in more detail outside the LI.

6.1.3 After the Lean Inception.

Workflows aid in identifying elements to understanding the problem: Creating workflows with users aided to identify functionalities and details of the problem. In our work, workflows identified elements involving in the process: databases, tasks, other people.

Workflows aid in preparing wireframes: Understanding the functionalities and identifying the technical details help in understanding the problem for building wireframes.

Important to materialize the idea to facilitate understanding the problem: Wireframes, journeys, and workflow elaboration aid to materialize abstract concepts and ideas. Visualization of ideas and concepts allowed identify elements of the problem.

Not knowing the required user interactions can lead to errors in the choice of development technologies: Not understanding the user's needs and the actions they need to take with the software can generate errors in the choice of tools. This issue is in line with [24].

Choosing inappropriate technologies for the software development can cause usability problems: In the project, the choice of technologies limited the user's interaction with the software and made it difficult to organize the interface. Also, the usability of its first version has been compromised. The same situation occurred in [24].

Structure process activities and documentation according to the level of detail of information: Structuring the project's steps and the documentation according to the information details supports each participant's needs. Each participant should have

a moment to explain their needs: customers explain the business goals, users explain their needs and problems. Also, developers and designers collect information to make decisions regarding the user interface design and development issues: the LI mixed business goals, user needs, and technical requirements. The participants had difficulties in understanding when discussing each type of information. In this context, it becomes difficult to explain and understand the problem, prioritize the features and make decisions about the UX.

6.2 RQ2. Did users who participated in the development process have a better perception of UX with the final product?

In this project, some users participated at the beginning of the project. They knew how to integrate the software into their routine. In case presented in [24], the opposite occurred. Users did not participate in any stage of the project. After developing and delivering the software, it was necessary to hold several meetings with the user to understand its purpose and how he can use it. Also, there was resistance from project users in [24] to use the first versions of the tool. The problems reported in their work were related to scope problems; the problems identified in our work were usability issues. In this project, users who participated only in the testing phase identified usability problems before using the system in their routine. Thus, users who did not participate in all steps of the project had initial difficulties in configuring the application. However, they did not report any usability difficulties that impacted their work. In the following, we describe some lessons learned in these projects regarding the user's participation in the process:

- When users participate from the beginning of the process, objectives are defined with greater accuracy, as they are aligned with their needs. Also, they perceive more value in the product and support its use. By contrast, when users receive the product without having participated in its construction, they have greater difficulties and are more resistant to using it and incorporating it in their routine.
- Users' participation from the beginning of the process supports understanding where and how the product can fit into their routine and aids in identifying elements of the problem domain and obstacles to implementation.
- Conducting user tests reduces usability problems in the final versions of the system. Only the participation of users in LI does not guarantee that the system will have better usability. Further activities are needed.

7 CONCLUDING REMARKS

Regarding the definition of the problem, we observed that LIs mainly aid in identifying business goals and coarse grained features. Unlike requirements engineering methods, LI's guidelines do not define the granularity of the features or the complexity of the domains to which it can be applied. In this way, we defined the features at a macro level due to the limitations of the LI schedule. Consequently, within the LI, participants do not have details to prioritize, evaluate, and understand the features. Indeed, in our experience we observed that the prioritized feature generated minor value. The problem to be addressed in this project was complex and involved

understanding many concepts unknown to the development team. The example in Caroli [4] is related to an application that does not involve the understanding of complex work processes/routines. At the end of LI, the features are inputs for the design and development processes. However, they are not adapted to the user needs and the software requirements. In this context, post-LI activities were necessary to complement the understanding of the problem, as the LI was not sufficient to support it.

Another issue addressed was the participation of end-users to support the definition of the problem. They helped to identify several elements necessary to understand the problem and elaborate on the solution. The users: (i) indicated other people involved in the process; (ii) talked about databases and documents to obtain necessary data for the project; (iii) presented their routine activities to the team; and (iv) supported detailing workflows and wireframes. Such contributions helped to generate software with better acceptance and fewer usability problems, positively impacting the routine of end users. The results presented here reinforce existing evidence in the literature that end-user participation in the early phases of the project is essential to facilitate the definition of the problem and identify critical elements, as well as to report context-specific observations. Regarding the study limitations, all interview participants work for the same company and we carried out the study in the context of a single software project. As future work, we aim at proposing and evaluating in practice ways to facilitate the inclusion of users in agile processes and facilitate the communication of information between stakeholders (customers and end users) and the development team. Also, we will carry out studies in other contexts, including the vision of the development team.

REFERENCES

- [1] David Aarlien and Ricardo Colomo-Palacios. 2020. Lean ux: A systematic literature review. In *International Conference on Computational Science and Its Applications*. Springer, 500–510.
- [2] Ohoud Almughram and Sultan Alyahya. 2017. Coordination support for integrating user centered design in distributed agile projects. In *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)*. IEEE, 229–238.
- [3] Edna Dias Canedo, Ana Carolina Dos Santos Pergentino, Angelica Toffano Seidel Calazans, Frederico Viana Almeida, Pedro Henrique Teixeira Costa, and Fernanda Lima. 2020. Design Thinking Use in Agile Software Projects: Software Developers' Perception.. In *ICEIS (2)*. 217–224.
- [4] Paulo Caroli. 2018. *Lean Inception: Como Alinhar Pessoas e Construir o Produto Certo* (1st ed.). Editora Caroli.
- [5] Juliet Corbin and Anselm Strauss. 2007. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (3rd edition ed.). SAGE Publications, Inc, Los Angeles, Calif.
- [6] Luis Corral and Ilenia Fronza. 2018. Design thinking and agile practices for software engineering: an opportunity for innovation. In *Proceedings of the 19th Annual SIG Conference on Information Technology Education*. 26–31.
- [7] Franziska Dobrigkeit, Danielly de Paula, and Noel Carroll. 2020. InnoDev Workshop: A One Day Introduction to Combining Design Thinking, Lean Startup and Agile Software Development. In *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 1–10.
- [8] Franziska Dobrigkeit, Danielly de Paula, and Matthias Uflacker. 2019. InnoDev: a software development methodology integrating design thinking, scrum and lean startup. In *Design Thinking Research*. Springer, 199–227.
- [9] Antonio Ghezzi and Angelo Cavallo. 2020. Agile business model innovation in digital entrepreneurship: Lean startup approaches. *Journal of business research* 110 (2020), 519–537.
- [10] Carina S González González, Evelio González González, Vanesa Muñoz Cruz, and José Sigut Saavedra. 2010. Integrating the Design Thinking into the UCD's methodology. In *IEEE EDUCON 2010 Conference*. IEEE, 1477–1480.
- [11] ISO13407 ISO. 1999. 13407: Human-centred design processes for interactive systems. *Geneva: ISO* (1999).
- [12] Marcos Kalinowski, Hélio Lopes, Alex Furtado Teixeira, Gabriel da Silva Cardoso, André Kuramoto, Bruno Itagyba, Solon Tarso Batista, Juliana Alves Pereira, Thuener Silva, Jorge Alam Warrak, Marcelo Silva da Costa, Marinho Fischer, Cristiane Salgado, Bianca Rodrigues Teixeira, Jacques Chueke, Bruna Ferreira, Rodrigo Lima, Hugo Villamizar, André Brandão, Simone D. J. Barbosa, Marcus Poggi, Carlos Pelizaro, Deborah Lemes, Marcus Waltemberg, Odnei Lopes, and Willer Goulart. 2020. Lean R&D: An Agile Research and Development Approach for Digital Transformation. In *Product-Focused Software Process Improvement - 21st International Conference, PROFES 2020, Turin, Italy, November 25-27, 2020, Proceedings (Lecture Notes in Computer Science, Vol. 12562)*, Maurizio Morisio, Marco Torchiano, and Andreas Jedlitschka (Eds.). Springer, 106–124. https://doi.org/10.1007/978-3-030-64148-1_7
- [13] Philippe Kruchten. 2004. *The rational unified process: an introduction*. Addison-Wesley Professional.
- [14] André Liem and Elizabeth B-N Sanders. 2011. The impact of human-centred design workshops in strategic design projects. In *International Conference on Human Centered Design*. Springer, 110–119.
- [15] Lassi A Liikkanen, Harri Kilpiö, Lauri Svan, and Miko Hiltunen. 2014. Lean UX: the next generation of user-centered agile development?. In *Proceedings of the 8th nordic conference on human-computer interaction: Fun, fast, foundational*. 1095–1100.
- [16] Roland M Müller and Katja Thoring. 2012. Design thinking vs. lean startup: A comparison of two user-driven innovation strategies. *Leading through design* 151 (2012), 91–106.
- [17] Abiodun Afolayan Ogunyemi, David Lamas, Marta Kristín Lárusdóttir, and Fernando Loizides. 2019. A systematic mapping study of HCI practice research. *International Journal of Human-Computer Interaction* 35, 16 (2019), 1461–1486.
- [18] David Parsons, Rochelle Thorn, Milla Inkila, and Kathryn MacCallum. 2018. Using Trello to support agile and lean learning with Scrum and Kanban in teacher professional development. In *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. IEEE, 720–724.
- [19] Maria C Ramos-Vega, Victor M Palma-Morales, Diana Pérez-Marin, and Javier M Mogerza. 2021. Stimulating children's engagement with an educational serious videogame using Lean UX co-design. *Entertainment Computing* 38 (2021), 100405.
- [20] Elizabeth Salinas, Rony Cueva, and Freddy Paz. 2020. A Systematic Review of User-Centered Design Techniques. In *International Conference on Human-Computer Interaction*. Springer, 253–267.
- [21] Lydia Schneidewind, Stephan Hörold, Cindy Mayas, Heidi Krömker, Sascha Falke, and Tony Pucklitsch. 2012. How personas support requirements engineering. In *2012 First International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE)*. IEEE, 1–5.
- [22] Eva-Maria Schön, Jörg Thomaschewski, and María José Escalona. 2020. Lean User Research for Agile Organizations. *IEEE Access* 8 (2020), 129763–129773.
- [23] Ingrid Signoretti, Larissa Salerno, Sabrina Marczak, and Ricardo Bastos. 2020. Combining user-centered design and lean startup with agile software development: a case study of two agile teams. In *International Conference on Agile Software Development*. Springer, 39–55.
- [24] Bianca Teixeira., Bruna Ferreira., André Damasceno., Simone Barbosa., Cassia Novello., Hugo Villamizar., Marcos Kalinowski., Thuener Silva., Jacques Chueke., Hélio Lopes., André Kuramoto., Bruno Itagyba., Cristiane Salgado., Sidney Comandulli., Marinho Fischer., and Leonardo Fialho. 2021. Lessons Learned from a Lean R&D Project. In *Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 2: ICEIS, INSTICC, SciTePress*, 345–352. <https://doi.org/10.5220/0010459803450352>
- [25] John Tripp, Jeff Saltz, and Dan Turk. 2018. Thoughts on current and future research on agile and lean: Ensuring relevance and rigor. In *Proceedings of the 51st Hawaii International Conference on System Sciences*.
- [26] Bianca H Ximenes, Isadora N Alves, and Cristiano C Araújo. 2015. Software project management combining agile, lean startup and design thinking. In *Design, user experience, and usability: Design discourse*. Springer, 356–367.
- [27] Sezin Yaman, Fabian Fagerholm, Myriam Munezero, Tomi Männistö, and Tommi Mikkonen. 2020. Patterns of user involvement in experiment-driven software development. *Information and Software Technology* 120 (2020), 106244.
- [28] Luciana AM Zaina, Helen Sharp, and Leonor Barroca. 2021. UX information in the daily work of an agile team: A distributed cognition analysis. *International Journal of Human-Computer Studies* 147 (2021), 102574.
- [29] Maximilian Zorzetti, Ingrid Signoretti, Eliana Pereira, Larissa Salerno, Cassiano Moralles, Michele Machado, Ricardo Bastos, and Sabrina Marczak. 2020. An Empirical-informed Work Process Model for a Combined Approach of Agile, User-Centered Design, and Lean Startup. In *19th Brazilian Symposium on Software Quality*. 1–10.