**King Saud University**

**College of Computer and Information Sciences**

**Software Engineering department**

**UX-Estimator: A Framework to Facilitate the Integration of User Experience Practices into Agile**

SWE 595

Razan Mohammed Aldossari

443203966

443203966@student.ksu.edu.sa

Advisor

Dr. Manal M. Alhammad

Master Project proposal for the degree of  
MSc in Software Engineering

College of Computer and Information Sciences

King Saud University

Third Semester

[Date Hijri dd/mm/yyyy]

[Date Gregoriandd/mm/yyyy]

ABSTRACT (300)

*Context.* Companies increasingly use Agile methods to develop digital products and services faster and more effectively. Today’s users demand products that are not only easy to use but also have a high User Experience (UX). Using Agile methods alone cannot necessarily guarantee a good user experience; UX activities need to be combined with agile methods to fulfil that purpose. However, there seems to be a lack of time for conducting UX activities when integrating Agile and UX. To create a product with a great user experience within a tight schedule, it is essential to assess the efforts involved in conducting UX experiments and choose the appropriate method to achieve valuable results.

*Objective.* The objective of this study is to develop a tool called “UX-Estimator” to assist agile teams in determining the worthiness of conducting UX experiments. With this tool, agile teams can 1) estimate the effort for conducting UX experiments, 2) select the most appropriate UX method for discovering and validating user stories.

*Method.* We developed the “UX-Estimator” step by step, according to the Design Science Research Methodology. First, we conducted a Structured Literature Review (SLR) to determine the current state of the art regarding the integration of Agile and UX, specifically in estimating UX work. As a result of the SLR, there is potential to improve UX integration into agile by addressing one of the reported challenges in the literature, i.e., the difficulty of assisting the worthiness of conducting UX experiments. On this basis, we explored the factors influencing the effort required to conduct UX methods and understood how UX practitioners decide what UX methods to use for a given user story. Thus, we designed a framework that UX practitioners have validated (see section 1.1 for detail on UX-Estimator). The final step was to validate that UX-Estimator.

*Results.*………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………… ………………………………………………………………………………………………… …………………………………………………………………………………………………

*Conclusion.*…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………… ………………………………………………………………………………………………… …………………………………………………………………………………………………

Keywords

User Experience, Agile, Agile Software Development, Agile UX Integration, Practitioners.

**ACKNOWLEDGEMENTS (optional)**

(Sample Wording)

I would like to thank my committee chair, Dr. Smith, and my committee members, Dr. Jones, Dr. Morton, Dr. Anderson, and Professor Benner, for their guidance and support throughout the course of this research.

Table of Contents

[Chapter 1 9](#_Toc126833860)

[Introduction 9](#_Toc126833861)

[1.1 Problem Definition 10](#_Toc126833862)

[1.2 Objectives and Research Questions 13](#_Toc126833863)

[1.3 Relevance to the Department 13](#_Toc126833864)

[Chapter 2 14](#_Toc126833865)

[Background, Related Work and Plan 14](#_Toc126833866)

[2.1 Background 14](#_Toc126833867)

[2.1.1 Agile Software Development 14](#_Toc126833868)

[2.1.2 User Experience 14](#_Toc126833869)

[2.2 Related Work 15](#_Toc126833870)

[2.2.1 Integration of Agile and UX 15](#_Toc126833871)

[2.2.2 UX Methods in Agile 17](#_Toc126833872)

[2.2.3 Estimating UX work 17](#_Toc126833873)

[2.3 Plan 22](#_Toc126833874)

[2.3.1 Methodology 22](#_Toc126833875)

[2.3.2 Timeline 24](#_Toc126833876)

[Chapter 3 25](#_Toc126833877)

[Problem Investigation 25](#_Toc126833878)

[3.1 ……….. 25](#_Toc126833879)

[3.1.1 ………… 25](#_Toc126833880)

[3.2 ………………………… 25](#_Toc126833881)

[3.3 …………… 25](#_Toc126833882)

[3.4 …………… 25](#_Toc126833883)

[Chapter 4 26](#_Toc126833884)

[Treatment Design 26](#_Toc126833885)

[4.1 Building UX-Estimator framework 26](#_Toc126833886)

[4.1.1 RQ1 – Estimating the worthiness of conducting UX experiments 26](#_Toc126833887)

[4.1.2 RQ2 – Deciding which UX method to apply 33](#_Toc126833888)

[4.2 Building UX-Estimator tool 37](#_Toc126833889)

[4.2.1 Product Backlog 37](#_Toc126833890)

[4.2.2 Implementation 38](#_Toc126833891)

[4.2.2.1 Hardware Description 39](#_Toc126833892)

[4.2.2.2 Development Environment 39](#_Toc126833893)

[4.2.2.3 Development Process 40](#_Toc126833894)

[Chapter 5 49](#_Toc126833895)

[Treatment Validation 49](#_Toc126833896)

[5.1 …………… 49](#_Toc126833897)

[5.1.1 …………… 49](#_Toc126833898)

[5.2 …………… 49](#_Toc126833899)

[Chapter 6 50](#_Toc126833900)

[Conclusions and Future Work 50](#_Toc126833901)

[6.1 Conclusions 50](#_Toc126833902)

[6.1.1 ……………….. 50](#_Toc126833903)

[6.2 Future Work 50](#_Toc126833904)

[References 51](#_Toc126833905)

[Appendices 56](#_Toc126833906)

[Appendix A - Interview Participants 56](#_Toc126833907)

**LIST OF ABBREVIATIONS**

UCD User-Centered Design

UX User Experience

**LIST OF FIGURES**

[Figure 1. A fishbone diagram illustrating the difficulty to assist the worthiness of conducting UX experiments. 11](#_Toc125418772)

[Figure 2. The Design Cycle by Wieringa [42] 21](file:////Users/user/Desktop/SWE%20595%20Razan.docx#_Toc125418773)

[Figure 3.Project Timeline 23](file:////Users/user/Desktop/SWE%20595%20Razan.docx#_Toc125418774)

[Figure 4.Factors that influence the amount of effort required for UX experiments 26](#_Toc125418775)

[Figure 5. Summary of Interview Findings 31](file:////Users/user/Desktop/SWE%20595%20Razan.docx#_Toc125418776)

[Figure 6. Summary of RQ results 34](file:////Users/user/Desktop/SWE%20595%20Razan.docx#_Toc125418777)

[Figure 8.Example user story 36](file:////Users/user/Desktop/SWE%20595%20Razan.docx#_Toc125418778)

**LIST OF TABLES**

[Table 1. Summary of the related work 20](#_Toc125389726)

[Table 2. UX-Estimator mapping table 35](#_Toc125389727)

# 

# **Introduction**

Agile and User Experience (UX) integration has received significant attention in the last 15 yearsdue to their economic success and business value [1]. Integrating both approaches, however, poses several challenges as there is inherent tension between the two areas. For example, Agile and UX share the same goal of developing software that meets the needs of users despite being from a different discipline that brings a different perspective on software development. Agile development focuses on delivering small sets of working software features in short iterations while UX focuses on user needs, interaction, and goals prior to development.

*[Here we’re talking about the step before the integration, this step is about how to decide whether to integrate UX into agile or not, and what are the factors that influence that decision.]*

There is no doubt that integrating UX into Agile has various benefits [2]. However, determining whether it is suitable for a given context requires identifying factors that could encourage or hinder the success of the integration. UX work gains the most motivation through delivering value to the business and the user as well as by producing successful products. Yet, UX work may be hindered by lack of economic and human resources, especially in early development stages or with limited resources [3].

*[Here we decided to integrate UX into Agile. There are some challenges, and we focus on solving one/two of these challenges and gave examples of other solutions]*

This study comes as a step further to facilitate this integration. It is still likely to face some challenges, and this study attempts to address one of them. Clearly, we cannot address all of the challenges with a single solution; therefore, this is a partial solution of the challenges. Moreover, some challenges were discussed in the literature, where some studies have proposed gamification frameworks to motivate and engage agile teams in conducting UX activities collaboratively and developing a user-centered mindset [4]. Other solutions include using user stories to integrate UX work into agile without sacrificing agility [5].

*[Here we’re talking about the challenges that we’re currently addressing]*

One common challenge in Agile and UX integration is lack of time for UX work [6]. Executing UX methods can minimize development changes, thereby resulting in a significant time and cost savings [7]. Developing a product with a great user experience within a tight schedule, requires assessing the effort associated with conducting UX experiments and deciding the best UX method to perform to achieve useful results.

The aim of this study is to facilitate the integration of UX into Agile by addressing one of its main challenges – Lack of time to carry out UX work. As the timeframe in Agile can be quite limited, it is difficult to conduct activities such as user research or design planning, where they require a lot of time. Additionally, agile teams may lack experience with UX, making learning how to use these methods difficult within the limited timeframe.

To potentially solve these problems, an aid in the form of a tool is needed to help in optimizing the effort required for conducting UX methods for specific user stories and identifying the most appropriate UX method to discover and validate these user stories. Without assistive tools, UX work might be neglected or dismissed since its benefits are not directly visible or tangible unlike agile practices, where increments are released at the end of each sprint. To achieve this aim, an assistive tool will be created to help agile teams 1) to estimate the effort for conducting UX experiments and 2) to select the most appropriate UX method to discover and validate user stories.

In this phase, the study assumes that the team decided to integrate UX into their agile process. As a result of this study, agile teams will be able to prioritize which features are worth the time and attention of the team to achieve UX experiment. In addition, they will be able to determine which UX methods to be used to achieve the desired outcome. Thereby improving the quality of results produced by agile teams and save a significant amount of time and effort.

This report explains the phases of developing the proposed solution in detail. The rest of this report is organized as follows: the remaining portion of this chapter presents the definition of the problem and the goal of this study. Chapter 2 gives the background information on Agile, UX. It also presents a literature review of some related work in the integration of these two processes. Chapter 3 provides an overview of the proposed framework and the development approach. Chapter 4 illustrates the implementation details and the results. Finally, Chapter 5 concludes this study with a summary of the study’s contribution, limitations, and potential future work directions.

## **Problem Definition**

Integrating UX into Agile presents several challenges, according to a study by Curcio et al. [6], these challenges include a lack of time for conducting initial design and user testing, conflicts between UX designers and developers, a lack of overall vision for the UX project, difficulties in determining priorities, and inadequate documentation. One of the main challenges in the study was the lack of time to carry out UX work within Agile process [8]. That is, Agile's time-boxed nature creates scheduling challenges when conducting UX activities such as design planning, user research, and workflow analysis, where they require a lot of time [9]. UX processes typically rely on extensive up-front design, which can be incompatible with agile processes, as the development cycle is essentially based on small incremental releases [10].

As such, it is imperative that the amount of time spent on conducting UX activities be aligned with Agile's time-boxed nature. Since there can be a significant amount of effort spent on conducting comprehensive UX methods to discover and validate features. It is possible that the time spent on these activities may exceed the value provided by those features [11]. Despite the fact that UX work is highly time constrained, UX methods often require agile teams to take the time to learn how to use them, especially agile teams with no or less experienced UX practitioners [12]. Furthermore, many Agile projects lack UX practitioners [13]. Also, UX practitioners often work on multiple projects at the same time, which is challenging since they need to fulfill multiple roles on a project, and sometimes their responsibilities are unclear [14], [15].

~~According to "The UX Book: Agile UX Design for a Quality User Experience" [16], there is a difference between UX methods and techniques. A method is a way to complete whole or part of an activity or sub-activity within the lifecycle, while a technique is a specific practice used within those activities. Usage Research would be defined as a UX method, while user interviews and observing users would be examples of UX techniques. In this study, however, the terms "UX method" and "UX technique" are being used to mean the same thing.~~

While agile teams need to incorporate UX methods throughout the development process, they may not have the time nor the experience to assess the effort involved in conducting UX experiments. They may also lack the knowledge required in deciding which UX method is appropriate for a particular user story given the time they have for UX work and the characteristics of the user story.

The literature analysis revealed that this is essentially a two-part problem illustrated in Figure 1 are as follows:

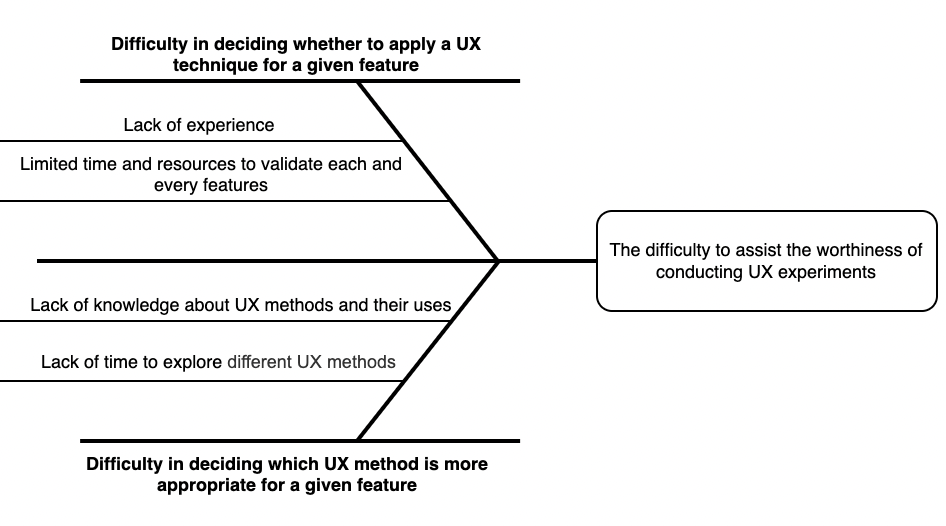
1. Difficulty in deciding whether to apply a UX method for a given feature. This is caused by lack of experience towards UX [17] as well as limited time and resources to validate each and every feature [18].
2. Difficulty in deciding which UX method is more appropriate for a given feature. This is caused by lack of knowledge about UX methods and their uses as well as lack of time to explore different UX methods [12].

Figure 1. A fishbone diagram illustrating the difficulty to assist the worthiness of conducting UX experiments.

The previously mentioned issues contribute to the key problem which this study aims to solve: The difficulty to assist the worthiness of conducting UX experiments. When Agile teams focus on conducting UX methods to discover and validate features that may not have significant value, they will likely end up spending an excessive amount of time and resources that may outweigh the benefits of these features [19]. Even if Agile teams knew or decided that a particular feature is worth conducting a UX experiment, they may not be able to make informed decisions about which UX methods to apply to achieve what they aim for or to learn something useful [20]. If they chose an unsuitable method, the activity ends up being simply a waste of time and efforts [8].

A clear understanding of the effort required to conduct UX experiments for each feature will help agile teams decide on which user stories they should spend their product discovery and UX hours. This minimizes waste of resources and time. Besides, selecting the right UX method helps agile teams to validate their assumptions in a more effective manner, such that they seek the results that can inform their decisions on how a feature can be improved or even rolled out [\*]. Therefore, it is necessary to design a framework that assists agile teams in determining the worthiness of conducting a UX experiment for each user story. *[I think we will talk about the parts of the framework here (what these parts achieve without mentioning the formula and table since we are in the problem def.), what do you think?]*

To help agile teams to effectively utilize this framework, a Trello Power-Up will implemented based on this framework. This Power-Up is the result of a detailed examining the literature and interviewing with UX experts who are working in agile teams*.*

## **Objectives and Research Questions**

The main aim of this study is to assist agile teams in conducting UX methods in an optimal manner. This is done by optimizing the effort required for conducting UX methods for specific user stories and identifying the most appropriate UX method to discover and validate these user stories. To achieve that, we designed a framework that can assist agile teams in determining the worthiness of conducting a UX experiment for each user story. Thus, this study aims to answer the following research questions:

**RQ1:** How can agile teams decide which user stories are worth validating from a UX perspective?

The aim of RQ1 is to explore the factors influencing the amount of effort required to conduct UX methods. Based on the findings of that, we designed formula *UXScore* that will allow agile teams to determine a score independently, thereby allowing them to determine how much effort needs to be invested into UX methods without continually involving experts.

**RQ2:** How can agile team decide which UX method is more appropriate for a particular user story?

The aim of RQ2 is to identify the amount of effort regularly spent on each UX method. *[I will change the aim of RQ2]* Then a mapping table will be designed showing what *UXScore* values suggest which methods are appropriate.

## **Relevance to the Department**

In this project, we intend to apply software engineering practices. The initial step was to apply requirements engineering techniques to gather requirements by interviewing UX practitioners (elicitation) in order to understand their work process within an agile team. Based the reviews (analysis), we came up with two artifacts which are a formula and a mapping table that represents the framework, which was then validated with UX professionals (requirements validation). In our next step, we will develop the tool (development) that represents the framework we are designing to help make agile team's job (software development teams) in estimating the effort of UX work easier and faster, and finally we will validate the tool with agile teams of different maturity level.

# 

# **Background, Related Work and Plan**

In this chapter, necessary background to fully understand following sections is provided, with some previous work from literature that relates to the proposed framework.

## **Background**

## **Agile Software Development**

Agile software development is an iterative and incremental approach that emphasizes collaboration to achieve quick and cost-effective results [21]. In addition, it allows for rapid response to changing environment and customer expectations, which subsequently reduces project risks [22]. Agile projects are delivered incrementally in sprints, with small chunks of product features being released at each sprint [23]. The approach focuses on delivering a valuable product for customers with minimal planning and documentation [24]. Agile software development became known through the Agile Manifesto [21]. The Agile Manifesto laid out four key values and twelve principles that can be applied to all agile software development approaches. The key values of the Agile Manifesto are (1) the concentration on individuals and interactions over processes and tools, (2) the delivery of working software over focusing on comprehensive documentation, (3) regular customer collaboration over contract negotiation and (4) responding to changes instead of following plans [21]. The most widely known agile frameworks are Scrum, Extreme Programming, Feature Driven Development and Rational Unified Process.

## **User Experience**

The term User Experience (UX) was introduced by Donald Norman in 1995 in his paper on Human Interface at Apple Computer [25]. User Experience (UX) is the experience the product creates for the people who use it in the real world [26]. ISO defines user experience as a person's perceptions and reactions resulting from using or the anticipation of using a product, system, or service [27]. The goal of UX is to enhance satisfaction and loyalty among a customer by providing them with ease of use, utility, and pleasure in their interaction with a product [28]. For software products to succeed in today's competitive markets, a good user experience is crucial [29].

In order to deliver a good user experience, it is essential to focus on UX activities on an ongoing basis, everyone's efforts will be more effective and valuable. Different UX methods can be applied during every step of the product development process to ensure that the effort is aligned with user needs, and it is more beneficial when the methods used are in place in early development phases and in each iteration to drastically reduce usability problems for the final version of the software [30].

User-centered design (UCD) is a design process that focuses on user needs and requirements, and it was initially defined by the ISO 13407 [31]. It is a general process it includes user-centered activities throughout the entire development lifecycle without specifying the exact methods to develop it. On the other hand, “Usability” is defined by ISO 9241-11 [32] as "the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". Similarly, Nielsen defines Usability as a quality attribute quality attribute composed of five components as learnability, efficiency, memorability, low error rate, and satisfaction [33]. The Institute of Electrical and Electronics Engineers (IEEE) defines ‘Usability’ as “the ease with which a user can learn to operate, prepare inputs for and interpret outputs of a system or component” [34]. ~~In this study, the terms ‘UX’, ‘UCD’ and ‘Usability’ are used interchangeably to mean the same thing.~~

## **Related Work**

## **Integration of Agile and UX**

Several systematic reviews have been carried out to study the integration of Agile and UX [35]–[37]. Silva et al. [35] outlined some key aspects concerning the integration of Agile and UX. The first aspect was Little Design Up Front, where the author suggested conducting UX-related activities before the official kickoff of the project. Prototyping was suggested as one way to improve the communication between developers and UX designers during the early stages of development. User testing has also been suggested for usability evaluation, where the team should conduct user testing within each sprint, aiming to refine the design for the next iteration. In order to maintain the Big Picture, One Sprint Ahead was suggested in which UX designers work one sprint ahead of the developers to ensure that the design is aligned with the business goals and users’ needs.

The integration of UX and Agile software development has been addressed extensively in the literature, leading to the proposal of new methods and techniques for merging the two domains, as well as discussions of the challenges associated with such integration. Recent research by Argumanis et al. [38] stated that the challenges of integrating UX activities into Scrum lie in the lack of importance assigned to UX activities, the inadequate communication between UX designers and developers, insufficient allocation of resources to upfront activities in Scrum and customers attempt to represent final users without understanding their real needs. Research by Curcio et al. [6] demonstrated that integrating UX into agile development poses several challenges, including the lack of time for upfront design and testing with real users, the power struggle between UX designers and developers, the lack of a vision for the entire UX project, the difficulty of prioritizing activities and the lack of documentation. In contrast, Da Silva et al. [1] stated that both communities have recognized that UX activities must be integrated into agile development, implying that these processes cannot be treated separately. Still, more work is needed on tools that facilitate agility and UX integration.

There have been some attempts to address these challenges; for instance, a study by Alhammad [4] addressed the issue of developers' mindsets towards user experience by presenting a gamified framework to motivate and engage agile teams in conducting UX activities collaboratively and develop a user-centered mindset. Kuusinen [39] addressed the synchronization of tasks between developers and UX specialist by identifying three types of cooperation: minimal, PO-UXS, and Developer-UXS cooperation. In projects with the minimal cooperation type, the UXS works mainly apart from the other team members, while with the PO–UXS and Developer–UXS cooperation types, the UXS works mostly with the PO and developers, respectively. The Developer–UXS cooperation was the most desirable cooperation type among the participant projects. Ananjeva et al. [5] proposed that user stories and Deliberation concerns “mutual communication that involves weighing and reflecting on preferences, values and interests regarding matters of common concern” can be used to integrate UX work into agile without sacrificing agility, which addresses one of the most challenging aspects of integrating UX, which is building a common ground between software developers and UX designers.

Persson et al. [40] conducted a case study on how Agile and UX processes are integrated to maintain agility through mutual adjustment of software development and UX design. The authors used three perspectives on agility that include assimilation, separation, and integration. Assimilation relies on coordinating by direct supervision and standardization of work processes, while Separation relies on coordination by standardization of skills and outputs. Integration lies between Assimilation and Separation, and it is taken from a across-entities perspective. They analyzed the company’s integration approaches of upfront design and work in parallel with Conboy’s taxonomy of Agility [41]. The results of this analysis showed how Agility differs for the two roles (Software development and UX design) in relation with these integration approaches (Agile and UX). Through the process of mutual adjustment, the Agility for UX designers and software developers are different, yet complementary. This integration contrasts with assimilation, which potentially makes their Agility mutually indistinguishable, and with separation, which makes their Agility different and mutually competing.

**[figures summarize the different ways of integrating ux with agile (chart etc) [see de Silva]**

## **UX Methods in Agile**

Several studies have suggested how UX work can be integrated into agile development. Bruun et al. [42] presented an in-depth case study to understand the responsibilities associated with UX professionals in an agile company that successfully integrated UX activities. It was found that UX practitioners have a very broad set of responsibilities including sales and business development, both of which place a customer focus over a user-centered focus. It is therefore possible that UX practitioners may not have enough time to conduct all the UX experiment that is needed, especially when it comes to an experiment that requires a lot of time.

Salvador et al. [30] conducted a systematic review with the focus on when, and which, usability methods are applied in Agile. The authors found that usability methods were mostly used during the implementation phase, and that the most commonly used methods included fast prototyping, individual inquiry, formal tests, and heuristic evaluations. However, there is no clear indication when UCD should be considered in an agile process; nor is it clear which particular method would be more suitable at different stages of the process. This gap was addressed by Caballero et al. [7] by gaining a greater understanding of how UCD techniques are used in each agile method and at what stages of the process they are applied. Researchers observed that the most common techniques used were prototypes, user stories, scenarios, usability testing, personas, and contextual inquiry, which were applied at the following stages of the agile process: big design upfront, minimal design upfront, implicit interaction design, and working ahead of development at least one iteration. Although, most projects reported agile teams working in a parallel track to the main agile process in which the interaction design and the requirements are developed.

Despite this, there is no clear indication how these techniques were been selected for a particular feature, as due to the agile time-box nature, it is difficult to apply set of UX methods for all these features; especially, some of these techniques are quite complicated and time-consuming [8].

## **Estimating UX work**

There are ways to guide Agile teams through conducting Heuristic Evaluations (HE) which is one of the UX methods. For instance, Abulfaraj and Steele [43] demonstrated how novice evaluators can use HE in the most effective manner. The authors developed a step-by-step protocol called Coherent Heuristic Evaluation (CoHE) to facilitate the use of HE for novice evaluators, thereby improving their performance and the quality of results they produce. To develop such a protocol, the authors interviewed usability experts and then analyzed the interview data to come up with coherent insights, focusing on the difficulties experienced by usability experts when they began doing HE and the mistakes novices were making. In addition, how to overcome these challenges, as well as how to improve HE as a method and focused on the overall organization of the HE process. Based on the analysis, they developed a step-by-step protocol. While we strive to guide agile teams that are novice UX practitioners in applying UX practices.

Relevant literature has revealed some factors influencing UX in term of effort. For example, a study by Gothelf and Seiden [44] described in their book "Lean UX" a prioritization matrix for determining how much risk there is and how much value we believe this idea will generate based on assumptions. The higher the risk and the more perceived value involved, the validity of these assumptions is of higher priority that means, the prioritization matrix focuses on what is more important and therefore to be researched first. A recent study by Lang et al. [19] presented the Discovery Effort Worthiness (DEW) Index, which assists companies and product owners in determining how much effort they should spend on discovering and validating features using Design Thinking methods. Similarly, DEW Index applies the same dimensions discussed by Gothelf and Seiden but describes the risk more precisely by asking how well understood is the actual demand of the user (clarity of the requirement) but focuses on the level of effort and what that effort should look like.

Moreover, Lang et al. [19] index is designed to determine the effort for performing "design thinking" techniques. Alternatively, our proposed solution is designed to determine the effort for conducting "UX" work. Also, their target role is the Product Owner (PO), whereas we target the entire team, not just the PO, since agile emphasizes autonomy and collaboration. Besides, the whole team is involved in estimating story points for user stories, so why should this be any different? In addition, Gothelf and Seiden [44] deigned the HPC (hypothesis prioritization canvas) which is a simple tool for determining which hypotheses (features) are worth testing. However, Gothelf's approach falls short on two key points which include 1) it fails to help in prioritizing in terms of effort, and 2) it does not provide any guidance on "how". According to Gothelf, the presence of UX experts in agile teams is essential for the success of integration, and if the agile team lacks a UX expert, the integration might fail. However, this statement can be argued as the study can actually achieve a successful integration by conducting the UX experiments the way this study recommended.

Therefore, our solution aims to fill these gaps by Gothelf and Lang et al. by 1) focusing on UX (rather than design thinking) and helping agile teams to 2) assess the effort needed for UX work so they can prioritize which features requires UX experiments, and 3) determining the best UX methodto achieve useful results. Table 1 provides a summary of the related work discussed above.

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref.** | **Year** | **Objectives** | **Findings** |
| [35] | 2011 | To examine how usability problems are handled in Agile projects. | The study identified the following key aspects for integrating Agile and UCD: little upfront design, prototyping, user stories, user testing, inspection evaluation, and one sprint ahead. |
| [38] | 2020 | To identify the challenges associated with integrating Scrum by examining recent case studies described in the literature. | The study revealed that UX activities are poorly prioritized in Scrum, inadequate communication between UX designers and developers, lack of resources allocation to upfront activities and customers attempt to represent final users without understanding their actual needs. |
| [6] | 2019 | To explore how the recent studies are integrating Agile with UX and identify the challenges associated with integrating these two processes. | The study revealed that integrating UX into agile development poses several challenges, including the lack of time for upfront design and testing with real users, the power struggle between UX designers and developers, the lack of a vision for the entire UX project, the difficulty of prioritizing activities and the lack of documentation. |
| [1] | 2018 | To provide a brief overview of the Agile UX Design and to point out unaddressed gaps, challenges, and future trends. | The study revealed that both communities have recognized that UX activities need to be integrated into agile development, implying that these processes cannot be treated separately. |
| [4] | 2020 | To improve the process of integrating UX practices into agile by addressing the issue of developers' mindsets towards user experience. | The study proposed a gamified framework to motivate and engage agile teams in conducting UX activities collaboratively and develop a user-centered mindset. |
| [39] | 2015 | To clarify which UX-related tasks can be handled by developers and which require a UX Specialist. | The study identified three descriptive types of cooperation, namely minimal, PO–UXS, and Developer–UXS cooperation. |
| **Ref.** | **Year** | **Objectives** | **Findings** |
| [5] | 2020 | To determine how to improve the integration of UX work with agile software development. | The study revealed that user stories and deliberation proved useful in integrating UX work with agile software development without sacrificing their agility. |
| [40] | 2022 | To understand how Agile and UX processes can be integrated to maintain agility through mutual adjustment of software development and UX design. | The study showed that Agility differs for Software development and UX design when they are integrating Agile and UX processes. Through mutual adjustment, the Agility for UX designers and software developers are different, yet complementary. |
| [42] | 2018 | To understand the responsibilities associated with UX professionals in an agile company. | The study found that UX practitioners have a very broad set of responsibilities including sales and business development, which focus more on customer than user. |
| [30] | 2014 | To identify which usability methods are used in Agile methods and when they are used. | The study revealed that usability methods were mostly used during the implementation phase, particularly fast prototyping, heuristic evaluations, individual inquiry and formal tests. |
| [7] | 2016 | To understand how UCD techniques are applied to each agile method and at what stages of the process they are applied. | The study revealed that prototypes, user stories, scenarios, usability testing and personas were the most common techniques applied at the following stages of the agile process: big design upfront, minimal design upfront, implicit interaction design, and working ahead of development at least one iteration. |
| [43] | 2020 | To facilitate the use of HE for novice evaluators, thereby improving their performance and the quality of results they produce. | The study developed a step-by- step protocol to guide novice evaluators through HE to improve the quality of their evaluations, thereby improving the usability of the systems they evaluate. |
| **Ref.** | **Year** | **Objectives** | **Findings** |
| [44] | 2021 | To prioritize which hypotheses to test based on their level of risk and the potential value. | The study presented a prioritization matrix for determining how much risk there is and how much value we believe this idea will generate based on assumptions, where the higher the risk and the greater the perceived value, the higher the importance of testing those hypotheses first. |
| [19] | 2021 | To develop a tool called Discovery Effort Worthiness (DEW) Index, which assists companies and product owners in determining how much effort should be spend on discovering and validating features using Design Thinking methods. | The study revealed that “Discovery Effort Worthiness Index” is a useful tool for companies and their product owners to determine how much effort they should spend on Design Thinking methods to discover and validate requirements. |

Table 1. Summary of the related work

## **Plan**

## **Methodology**

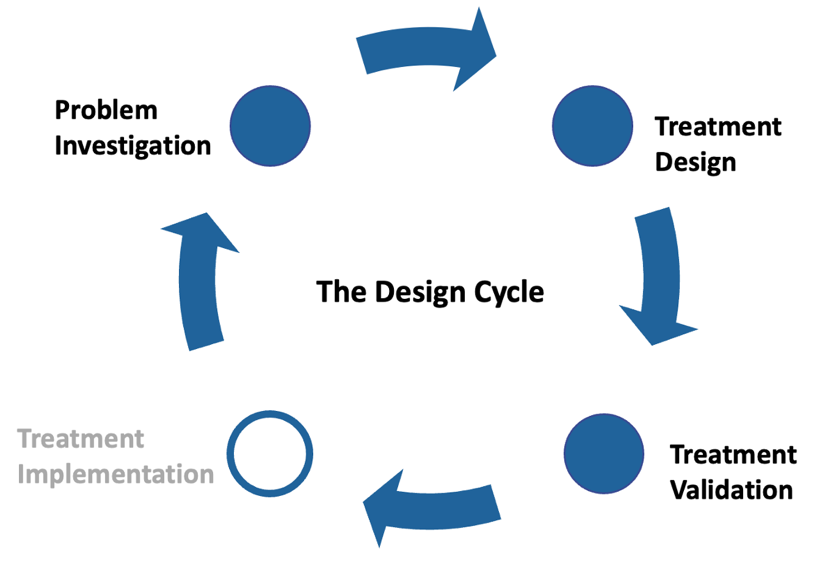
To ensure the validity of our framework, we employed Design Science Methodology (DSM) proposed by Wieringa [45]. DSM provides an iterative approach that allowed us to build, validate and implement the framework without having to stop its execution for testing. We chose this methodology due to its capability of creating an artefact with the intention of solving real-life problems, which was the case in our study targeting the difficulty to assess the worthiness of UX experiments.

Figure 2. The Design Cycle by Wieringa [45]

1. **Design problem**

A design problem is defined by Wieringa as “a problem to (re)design an artifact so that it better contributes to the achievement of some goal” [45]. Wieringa proposes a template for expressing a design problem:

* Improve <*a problem context*>
* by <*(re)designing an artifact*>
* that satisfies <*some requirements*>
* in order to <*help stakeholders achieve some goals*>

An artifact is a broad term that can be defined as “something created by people for some practical purpose” and the interaction between an artifact and the problem context is referred to as the treatment that is expected to solve a problem. Therefore, the problem context should be understood, and we need to study this interaction in order to evaluate if it solves the initial design problem. The template applied to our research context gives the following representation:

* Improve *the integration of UX into Agile*
* by *designing a framework and a tool*
* that *assess the worthiness of conducting UX experiments for each user story based on the influencing factors.*
* in order to *help agile teams to prioritize which user stories are worth investing their effort carrying out UX experiments and guide them in selecting which UX method is more appropriate.*

1. **The Design Cycle**

To conduct the research, we will follow the different steps described in the design cycle process proposed by Wieringa [45] (Figure 2). The design cycle is part of a larger problem-solving process, the engineering cycle. The design cycle aims to iterate into problem investigation, treatment design, treatment validation where the engineering cycle also includes a task for the transfer of an artifact to the real-world, called implementation.

* 1. **Problem Investigation**

The first step in the methodology involves identifying, describing, and explaining the problem that must be solved or improved as well as defining motivation and goals for the research. This step included asking questions such as: Why do agile teams want to integrate UX into agile? what are the challenges of agile and UX integration? How are these challenges caused or why do they occur? What will happen if these challenges remain unsolved? What will improve if we find a solution?

Following Wieringa's approach to problem investigation, we explored literature as a common technique for understanding the problem context. Therefore, the literature on the integration of agile and user experience was explored to answer the previous questions and assess the relevance and significance of the problem.

* 1. **Treatment Design**

The second step in the methodology involves designing an artifact to address the problems context of this study. Considering this is a pilot study, we decided to conduct a semi-structured interview with 10 UX professionals, which provided us with enough information to help us build the framework. The purpose of conducting interviews was to understand their work process as well as how they determine which features should go through UX experiments. Thereby to identify the main factors that were considered when determining the effort of a feature along with the weight that were suggested by the experts.

Based on the findings of that, we will design a formula *UXScore* that will allow agile teams to determine a score independently, thereby allowing them to determine how much effort needs to be invested into UX methods without continually involving experts. As for the UX methods that will be considered in the designed artifact, we will reveal the frequently used methods that were outlined by Norman [46] and discuss with the experts how much effort is regularly spent on each method. Accordingly, we will design a mapping table to show what *UXScore* values suggest which methods are appropriate.

Our solution requirements (formula & mapping table) will be validated by an agile team and comparing the results to experts' suggestion. This validation will be done to ensure the feasibility of these requirements by applying it to a real-life. On the basis of the validation results, we will build the tool.

* 1. **Treatment Validation**

The third step in the methodology involves justifying whether the designed treatment contributes to satisfying the stakeholders' goals, when applied in a real context [45]. In this study, the **designed framework will be implemented as a tool called UX-Estimator, and then UX-Estimator will be validated by an agile team** to answer two questions: 1) Did the tool help novice practitioners or agile teams to better estimate the effort associated with UX activities or not? 2) Did the tool guide them in deciding which UX method to use?

## **Timeline**

Timeline

Description automatically generated with low confidenceIn this section, we will cover the project plan. Figure 3 illustrates the timeline established for the project, starting with the proposal, and ending with the final phase. As part of the proposal phase, which began in September 2022, relevant studies were reviewed, the problem statement and project objectives were defined, and the methodology that will be followed in the implementation phase has been proposed.

Figure 3.Project Timeline

# 

# **Problem Investigation**

*The first step in the methodology involves identifying, describing, and explaining the problem that must be solved or improved as well as defining motivation and goals for the research. This step included asking questions such as: Why do agile teams want to integrate UX into agile? what are the challenges of agile and UX integration? How are these challenges caused or why do they occur? What will happen if these challenges remain unsolved? What will improve if we find a solution?*

*Following Wieringa's approach to problem investigation, we explored literature as a common technique for understanding the problem context. Therefore, the literature on the integration of agile and user experience was explored to answer the previous questions and assess the relevance and significance of the problem.*

## **………..**

## **…………**

## **…………………………**

## **……………**

## **……………**

# 

# **Treatment Design**

This chapter presents details of the designed framework and answering the research questions which are essential in constructing the framework. In addition, it provides details on the tool that has been developed to assist agile teams in estimating the worthiness of conducting UX experiments.

## **Building UX-Estimator framework**

## **RQ1 – Estimating the worthiness of conducting UX experiments**

To start building the framework, we need to answer RQ1– *How can agile teams decide which user stories are worth validating from a UX perspective?*that aims to address how agile teams decide which user stories are worth conducting UX experiments. To answer this question, we decided to conduct interviews with a number of UX practitioners to understand their work process and how they decide which features should go through UX experiments. The aim of this procedure was to identify what factors practitioners usually consider when conducting UX experiments.

A semi-structured interview process is conducted until data saturation is reached, meaning no new insights are provided. Guest et al. [47] suggest that 12 interviews are usually needed to reach saturation. Accordingly, we planned to interview 10-15 UX practitioners, with the possibility of adding more saturation was not reached.

We interviewed 15 UX practitioners working at different companies in Riyadh, Saudi Arabia. Their roles ranged from UX Researcher, UX Designer, UX/UI Designer, CX Designer, Consultant to Manager. The interviews, audio-recorded, were conducted either in-person or via conference call. An overview of the participants is provided in Appendix A - Interview Participants. The interviews were then transcribed verbatim. After all the interviews were completed, we listened to the recordings again and compared them with the data to ensure that nothing was missing.

In the first part of the interview, the practitioners were asked what factors they consider when conducting UX experiments. Addressing these factors is likely to help us in building an equation which is the basis for building the tool that will generate a score for each user story. Based on that score, the agile team can make a decision on whether to prioritize this user story for a UX experiment or not. In summary, we found three factors (Figure 4): *Potential Value*, *Clarity* and *Familiarity*. The factors are further described in detail below.

Figure 4.Factors that influence the amount of effort required for UX experiments

* 1. **Potential Value**

Among the participants we interviewed, nearly all of the participants mentioned that business and customer value are taken into consideration, but there were some tensions of whether the business value has higher impact than customer value or not. Several participants stated that they focus on business value, however one participant sees from his perspective that the customer value should consider more important than business value:

“It is business, so we focus on delivering features and services that return profit to the company”, yet, in his opinion, “If your product is not used by the users, will it have value? No, it will not. Therefore, from my perspective, we should focus on the users.”. Continuing, the participant said, “We should focus more on customer value than business value, as the services/features might not succeed as expected without knowing the users' needs”. (Saad)

“Another factor to consider is, which feature is important to the stakeholder (business)”. (Hamad)

One participant stated that the feature that has a business value is considered a high priority regardless of whether the user needs it or not:

“A feature that provides profit for the company should be considered a high priority regardless of whether the user needs it or not, as long as it furthers the company's business goal. I try to balance between both business and customer value.” (Moath)

However, several participants stated the effect of business and customer is subjective, where it depends on the company’s strategy:

“I consider customer value has a higher impact than business value. However, when the process is conducted in a way that ensures that both values are treated equally, by including UX researchers during the requirement and analysis phases will ensure that these values will be aligned.”. In addition, the participant said, “In practice it is subjective; it depends on the company’s strategy, whether the UX team is part of the product team (“business value is higher”) or are part of the development and operations team (“customer value is higher”), again it depends on the company’s strategy”. (Saif)

“We conduct usability testing to a critical feature, and the criticality depends on what sector the UX practitioner belongs to. In the case of a sector related to human safety, then we will focus on user value, whereas in a private sector we will focus on business value.”. (Hamad)

“As UX practitioners are part of the organization, their prioritization should not be driven by only their work, rather it should be based on the organization's strategic goals.” (Bandar)

There is an inherent tension between business value and customer value, which can be resolved by considering both values as potential value. Three participants emphasized that both values are related and can be aligned:

“In the research phase, I should know what the user's needs are, because in the end the user is going to use the system. I do not like designers who put themselves in the role of the user and ignore the business. If there is a lack of alignment between both business and customer value, I know there are some intersections and differences, but we should unite points of view.” (Sara)

“There is no difference between customer value and business value if we look at it as a big picture. If it does not add value to the customer, it will not add value to the business. Both values are related to each other, at the end either it is customer value or business value it is a value.” (Hamad)

“There will not be any tension between business and customer value when UX is incorporated into the company’s culture.” (Saad)

* 1. **Clarity**

A number of participants mentioned the clarity of the user story is one of the factors that should be considered, as *Clarity* in Software Requirements refers to the degree to which a requirement is unambiguous and understandable [48].

“We need to make sure that the user story is clear and detailed when we are receiving it, so that we are aligned with the other teams such as business on how the feature is likely to be”. (Hamad)

“We understand there may be some missing information in the user story, but as a UX designer, I prefer the user story to be full of information, so that I do not need to go back and forth with the owner of the feature.” (Sara)

One participant stated that the feature should be clear and detailed before moving to the design phase:

“We do not pay much attention to the clarity of the feature in the design phase since it can be handled during the requirements phase, when we set up with the client to get the feature, we perform some tests to learn what the users want.” (Saif)

Two participants noted that the requirements were not completely clear when they received them, but they expect that, as the rest of the work can be done by them:

“We just want to know what the user expects from the services/features, and we will handle the rest from our end and see what works best for our users.” (Nasser)

“In practice, the requirements are not completely clear, a UX practitioner wishes to have clear requirements so that they can be more creative and design something better.”, the participant said, adding that “businesses are not always contented or did not consider all the cases that might apply to the feature. In addition, he added that “most of the time, clients come to us with a huge idea without defining the requirements and expect a solution from us.”. (Moath)

A participant stated that to get a clear requirement, it is better to have wireframes rather than documents:

“We received up to 200 pages of documents describing requirements, which was chaos. We solved the problem by letting the product team design some wireframes and sketches instead of us reading large documents.”. (Anas)

One participant considered the complexity of the feature should be defined and clear as it has an impact in determining whether to conduct UX experiments:

“When the feature has a complex business model or when it depends upon or impacts another feature, we decide whether this feature is complex or not, and if yes, we will focus on it and consider it important”. In addition, the participant continued, “When the feature has impact on future features, I need to study this feature so I can build it in a flexible way in order to make some modification in future.”. (Moath)

* 1. **Familiarity**

Several participants stated having experience within the product domain also plays a role when deciding to conduct UX experiments:

“I am familiar with how users behave towards a particular feature, as I have done a similar feature in the past, so I do not need to do extensive research.”. (Anas)

“There are some UI patterns that are verified from UX guidelines, and others are verified from existing features for which we have already performed some testing, which in turn can be reused when implementing a new feature, so we can use them without conducting any further testing”. (Nasser)

“It is good to make sure the UI patterns cover part of the system design, as if we test them for a specific feature, we can reuse them in another feature; this ultimately gives us the opportunity to reduce effort by using proven components and avoiding testing them, except when they are utilized differently from what is expected.”. The participant continued, saying that, “In most cases we do not need to conduct user research, since we already have insight on how the user behave. We only need to conduct user research when the product team give us feedback on that feature.” (Saif)

One participant considered the maturity of the UX designer:

“UX designers can estimate the effort based on their experience, by saying I am familiar with, and I did something similar to it, or I may say I should do part A before part B and so on.” (Sara)

One participant stated that enhancing an existing feature is simpler than working on a new one, since it already has a track record:

“There is a difference between working on a user story that is an enhancement of an existing feature and a new feature. An enhancement user story will do better than a new feature, because it already has a success and failure rate, so the work will be based on feedback from the existing feature, but the new feature we do not yet know whether the feature is really needed or not.” (Nasser)

One participant stated that making changes to features that impact the existing behavior of the user is challenging, leading to a need for intensive UX experiments:

“It is important to consider whether this feature will change the existing behavior of the user. For example, a user might be accustomed to a certain behavior, and I as a designer will change it. So, the question here is whether this change will have a positive impact or not? In this case, I consider investing heavily in UX experiments.” (Nasser)

Several participants stated that introducing a new feature into the market requires a lot of effort:

“When a new feature enters the market and there are no competitors, and it did not exist before, it will require effort, as I have to understand the user's behavior.”. Continuing, the participant said, “If I already know how the user behaves, here I will start from the last point where we reached in the product and do further enhancement.” (Hamad)

“If the service is very newly introduced to the market, I will conduct a benchmark analysis of how the outsiders did it, I will make quick research on how this feature is used externally, and if this feature is available in our market, I will conduct a competitive analysis. Then I will see if this feature achieves the desired need or not, if not, I will take it to the market and do my best and take feedback on it.” (Saad)

“When adding a new feature, let us say *auctions of special numbers*, there are no familiar features that have done that before, we need to look for indirect studies instead to find out what are the problems that the user faces in an auction feature in general, not the exact same service we provide, which is special numbers. In this case, the users will buy, we will pay attention to how they buy.” (Lujain)

“The process of adding new features that have not been applied yet, and how we can apply them, does not go like regular features, because the new features have no customer base, therefore we will use surveys to find out our target audience. Since it is a new idea, we need to sketch a new experience, where we need do heavy work to understand what users are expecting from this feature.” (Bandar)

One participant stated that they should determine the complexity of the feature and whether it will be easily understood by its users.

“How complicated is the feature, which I mean the complicated of the process to make it simple to the user or not, for example, when we say that this feature may confuse the users in specific steps or other features, here I will try to see how I can simple it without making a problem to the user.” (Lujain)

Drones, time to market (Moath, Bandar and Hamad 36.40) ...…………...…….………… ……………………………………………………………………………………………………………………………………………………………………………………………………

Based on the practitioners’ responses, they determine the importance of conducting UX experiments by evaluating potential value, familiarity and clarity relationship, with potential value having a significant impact over familiarity and clarity, while familiarity has slightly higher impact than clarity. Resulting to conclusion the suggested weighting is 6:3:1 for potential value, familiarity and clarity.

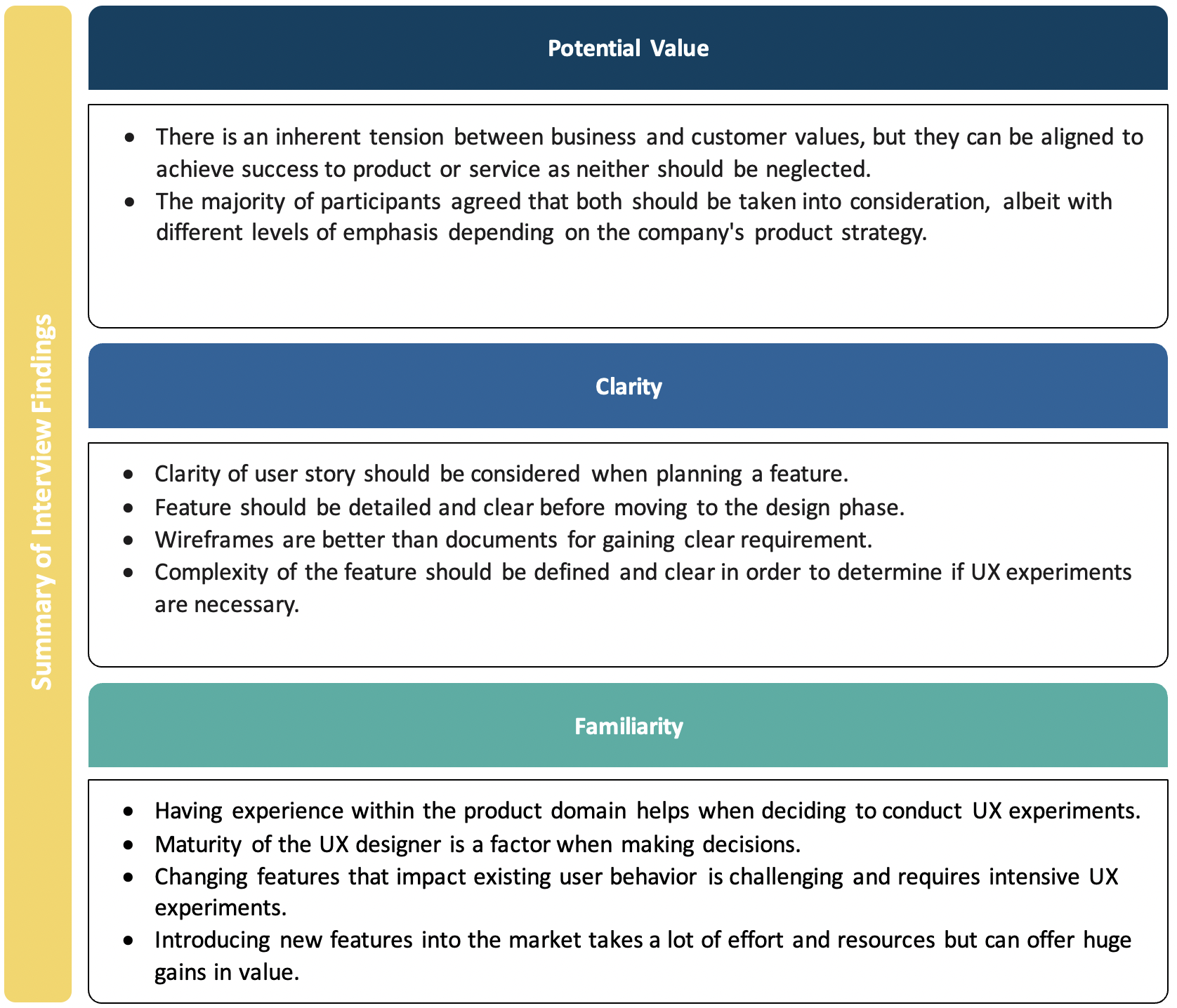
The weighting is done because a high level of potential value (customer and business value) leads to long-term profitability and growth, thus increasing the need for carrying out UX experiments. Besides, a high level of unfamiliarity can negatively impact whether a feature will fulfill the user's expectations. However, the feature can still be delivered with a minimum requirement so the UX team can observe how the user behaves towards this feature and get some feedback. Unclarity also has an impact on conducting UX experiments. However, when compared with potential value and unfamiliarity, it does not have a significant impact, because it is expected that the requirements are not completely understood. Figure 5 summarizes the interview findings with UX practitioners.

Figure 5. Summary of Interview Findings

Based on this weighting, *UXScore* is calculated by scoring each user story on a scale from 1 to 10, where 10 represents high potential value, high unfamiliarity and high unclarity respectively. To be able to do the calculation of *UXScore*, the familiarity and clarity ratings must be inverted since the effort for discovery increases with high unfamiliarity and high unclarity. Finally, due to the different weights (6 for potential value, 3 for unfamiliarity and 1 for unclarity), the score must be divided by 10 to get the required effort.

Equation 1. UXScore formula

The results of this equation will be used to decide which UX method using the mapping table will provide in the following.

## **RQ2 – Deciding which UX method to apply**

In order to design a mapping table, which is part of our framework, we need to answer **RQ2**–How can agile team decide which UX method is more appropriate for a particular user story?, which focuses on exploring what *UXScore* values suggest which methods are appropriate. To accomplish this, we returned to the participants to see how they decide what methods to use for a given user story, asked them to share some examples of the user stories they worked on, and rank the Potential Value, Clarity and Familiarity of the user story. *[they were excited more than first interview]*

A participant provided an example of a new feature that is considered to have a high value and that had never been implemented in an application that he knows of, and he does not know how users will respond to it; he suggested doing a *Field study* to find out if there are applications not known that may use this feature or perhaps a *Secondary study* to see how the feature is implemented in a different context to gain insight on how to start doing it. Furthermore, he suggests doing *High fidelity prototypes* and *In-person usability testing* to observe how users interact with the feature so that the feature can be improved to match their behavior.

Several participants suggested doing *Remote usability testing* when the user story has moderate value, and they want to be sure they understand their targeted users. Whereas *In-person usability testing* is recommended when the user story has significant value and there is no certainty about how users will react.

Another participant suggested conducting *In-person usability testing* when a feature is complex, meaning that it has multiple scenarios and various cases; this will help in gaining more insight and understanding of the users and facilitate easier communication.

Several participants stated that if a user story is clear and simple and does not hold a lot of value, a *low-fidelity prototype* and *design review* may be enough to define the design to the development team so it can be implemented and tested smoothly.

One participant stated that he prefers doing *High-fidelity prototypes* over *Low-fidelity prototypes*, even if the feature is not so crucial. He explained that this will be more beneficial when dealing with multiple stakeholders (Business, Development, and Quality Assurance) as it helps align all the teams by taking into account resources, time, and existing UI components. However, when time is limited, he will conduct high-fidelity prototypes of important features.

Four participants suggested doing *User interviews* when the user story has a high value and when they want to get more details about what the user expects from the user story. However, one participant stated that they may conduct *User interviews* even when the value of the user story is not high; if the value is high, they will conduct a detailed interview, otherwise, they will conduct a simple interview to understand what the user expects.

Another participant stated that he will conduct *User interviews* to see what the users expect the entry flow to be for the feature that would be implemented, and he will conduct *Usability testing* for users who do not usually use the product in order to validate the feature.

Four participants suggested doing *Competitive analysis* when the user story has a high value and when they want to get more details about what the user expects from the user story. However, one participant argued that *Competitive analysis* can lead to an overly conservative mindset, as it focuses mainly on existing approaches and technologies which may lead to overlooking some potential solutions or innovative ideas. He believed that instead of approaching user stories with an ‘if this is what our competitors are doing, then this must be the way forward’ attitude, He recommended testing new ideas and approaches, without being overly concerned about what the competitors might do. [Saif]

*Lujain does quick research to see if there is a familiar feature have been done before or not if not she looks for indirect research and Saif do formative research for core features.*

Nearly all of the participants stated that if a user story has a variety of stakeholders, a *Workshop* and *Design review* are considered to ensure the user story is finalized by all the parties.

Nasser stated that he conducts *Task analysis* when the user story is huge, or even when having a capacity of recourses.

Based on the practitioners’ responses, they………………………………………………. ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  | 5 |  | 6 | 7 |  | 8 | 9 | 10 |
| Set 1 | Set 2 | Set 3 | Set 4 | Set 5 | Set 6 | Set 7 | Set 8 |  | Set 9 | Set 10 | Set 11 |
| Design review | Design review | Design review | Design review | Design review | Design review | Design review | Design review |  | Workshop | Design review | Design review |
| Low fidelity prototype | Low fidelity prototype | Low fidelity prototype | Low fidelity prototype | Low fidelity prototype | Low fidelity prototype | User interview | User interview |  | Competitive analysis | User interview | User interview |
|  |  |  |  | Remote usability testing | Remote usability testing | Competitive analysis | Competitive analysis |  | High fidelity prototype | Field study | Field study |
|  |  |  |  | + | + | Low fidelity prototype | Low fidelity prototype |  | In-person usability testing | Low fidelity prototype | Low fidelity prototype |
|  |  |  |  | User interview | User interview | Remote usability testing | High fidelity prototype |  | + | High fidelity prototype | High fidelity prototype |
|  |  |  |  | Task analysis | High fidelity prototype | + | In-person usability testing |  | Field study | In-person usability testing | In-person usability testing |
|  |  |  |  |  | + | High fidelity prototype |  |  | Design review | Analytics review | Analytics review |
|  |  |  |  |  | Analytics review | + |  |  |  | + | + |
|  |  |  |  |  |  | Analytics review |  |  |  | Competitive analysis | Competitive analysis |

Table 2. UX-Estimator mapping table

## ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

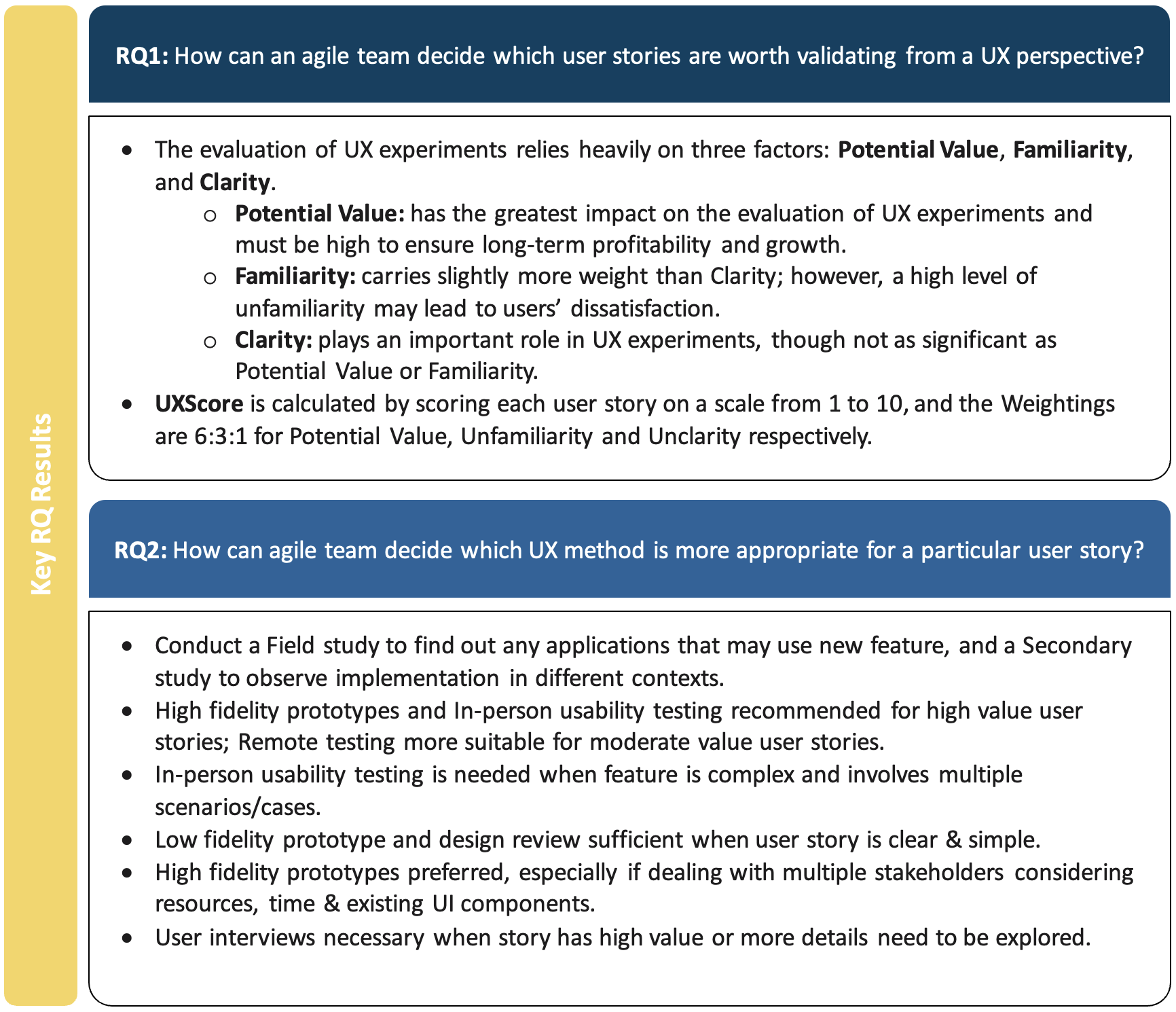


Figure 6. Summary of RQ results

## **Building UX-Estimator tool**

To validate the proposed framework, we developed a Trello Power-up called UX-Estimator that represents our framework. UX-Estimator will assist in conducting a pilot study to demonstrate its effectiveness in facilitating UX integration into Agile, helping agile teams prioritize which user stories are worth pursuing UX experiments and guide them in selecting which UX method is most appropriate.

Since this study has followed as a design science research method, and has been developed using an agile development method, the development process has been a series of iterations. That is, each iteration has been building on the previous one and each involved a set of goals needed to be accomplished.

## **Product Backlog**

In our product backlog, we have presented our functional and non-functional requirements into user stories to align with our agile approach. We have divided the development process into several sprints, with each sprint lasting two weeks. This allows us to quickly deliver working versions of the product and gather feedback from stakeholders, while also allowing for flexibility and adaptation as we progress through the development process. The number of sprints will vary depending on the scope of the project, but we typically aim for around 4-5 sprints in total. This allows us to have a manageable timeframe for each sprint, while also ensuring that we have enough time to fully develop and test the product before its release.

The following template represents our user stories:

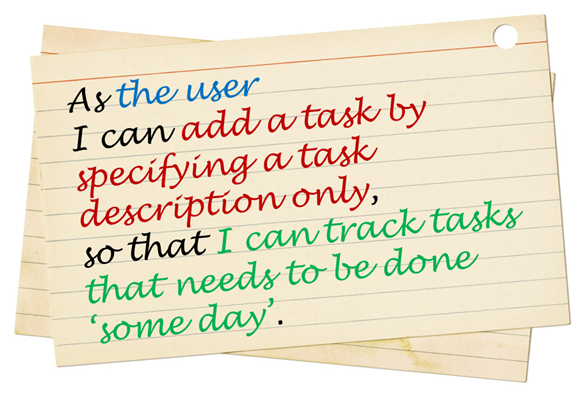
As a <*role*>, I want <*goal*> so that < *benefit* >

Figure 8.Example user story

The following are the initial product backlog.

1. As a user, I want to go through a wizard (step-by-step guide) when installing the power-up so that I understand how to use it.
2. As a user, I want to go directly to the board after completing the setup process for the power-up, so that I can begin using its features.
3. As a user, I want to receive an alert message indicating successful completion of the power-up setup, so that I am certain that everything is set correctly.
4. As a user, I want to assign scores for each dimension of a given user story in order to determine the worthiness score.
5. As a user, I want to modify the score of each dimension after setting them so that changing estimations are possible.
6. As a user, I want to view a list of suggested UX methods, so that I can pick one of them.

## **Implementation**

This section provides the implementation details in terms of hardware, development environment, development processes, and screenshots of the developed tool.

## **Hardware Description**

MacBook Pro personal computer (macOS Monterey) has been used to implement this project. The computer has the following specifications:

* Processor: Apple M1 Pro chip with 10‑core CPU.
* Random Access Memory: 16 GB.

## **Development Environment**

This tool (power-up) is implemented in Visual Studio Code (VS Code)[49], an integrated development environment (IDE) dedicated to building Web applications and cloud programs. The programming language used is JavaScript which is one of the most used programming languages for front-end development [50], it is used for developing Trello Power-ups. In this work, we used the following JavaScript library “jQuery” to simplify JavaScript as it takes a lot of common tasks that require many lines of JavaScript code to accomplish and wraps them into methods that you can call with a single line of code. Moreover, Netlify a web development hosting platform was used to build and deploy the tool. Below is the brief of the technologies ****that were used to develop UX-Estimator.

1. **Trello**

Trello [51] is a web-based project management application that supports Kanban, allowing users to track progress on their projects. Trello utilizes the concept of boards to represent projects, and within each board, there are lists of tasks and cards representing these tasks. Cards are moved between lists to indicate task completion. Trello provides functionality called Power-ups which are similar to plugins to extend the basic features of this collaboration platform and allow integration with other applications. Trello Power-Ups [52] provide features such as custom fields, calendar views, voting, and more. With Power-Ups, developers can create custom integrations with third-party services and build custom features for Trello boards. Developers can also use the Trello API to create powerful automation tools that can be used to automate tasks and processes within Trello.

1. **Netlify**

**Shape

Description automatically generated with low confidence**Netlify [53] is a web development platform that provides hosting, continuous deployment, and serverless functions for web applications and static websites. It offers developers a range of advantages, including faster deployment times, improved scalability, reduced costs associated with managing servers or infrastructure, and easy deployment without having to worry about setting up servers or configuring infrastructure. Netlify’s hosting platform can continuously integrate with a Git repositories; this means that the code in the connected Git repository will be used to create a website. When code is pushed to the Git repository, the website will be automatically built and deployed with the new changes.

1. **Git & GitHub**

Git [54] is a distributed version control system that tracks source code changes when developing software. It is intended to help programmers coordinate their work and is also used to track changes to any set of files. On the other hand, GitHub [\*] is a collaborative code hosting platform built on top of the Git version control system. It provides a wide range of features to help developers collaborate and work together. These features include code review, issue tracking, project management, and more. Additionally, GitHub allows developers to easily share their code with others and make changes without affecting the original code. This makes it easier for developers to collaborate on projects and keep track of changes made by other members of the team.

## **Development Process**

This section presents all of the sprints for our project along with the associated user stories and refinements. Each sprint will typically last for two weeks, during which time the development team will work on completing a set of specific tasks and objectives. We currently have a total of 5 sprints planned for this project, with each sprint will be carefully planned and organized to ensure that all tasks are completed on time and that the overall project stays on track. During each sprint, we will work on completing a set of user stories and any necessary refinements. This will include tasks such as designing, developing, testing, and deploying the features. By the end of each sprint, we aim to have a usable and potentially releasable product increment.

**Sprint 0:**

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….

**Sprint 1:**

In this first iteration, we established the local environment required to run a power-up hosted in the local environment. Next, we started building the power-up by implementing the wizard (step-by-step guide) as the first feature that will be displayed to the user upon installation of the power-up.

**Sprint Backlog:**

The proposed user stories for the first sprint of the power-up are as follows………………… ………………………………………………:

|  |  |  |
| --- | --- | --- |
| **Backlog item** | **Task** | **Estimated Effort** |
| **Feature 1- View** **step-by-step guide** | Outline the steps of the wizard by establish a sequence of steps which considers the data needed to initiate the power-up. |  |
| Create forms that represent each step of the wizard. And in each form create buttons, pick lists and other interactable elements to move through each step of the process in a controlled way. |  |
| Create a modal component that contains a wizard, and in this modal and create tabs to represent each step of the wizard. Within each tab, create a form that includes the necessary fields and buttons. |  |
| Apply functional testing:   * Test Case 1: Verify that the wizard is the first screen displayed after the power-up is installed. * Test Case 2: Verify that the wizard has multiple steps to guide the user through the setup process. * Test Case 3: Verify that each step in the wizard has clear instructions and visual aids to help the user understand what needs to be done. * Test Case 4: Verify that the user is not allowed to proceed to the next step until all required information is entered in the current step. |  |
| **Feature 2- Navigation creation** | Create a “Navigate” or “Go to Board” button on the final screen of the power-up setup wizard, so that the user can quickly and easily access the board that the user wishes to integrate the power-up with once set-up is completed. |  |
| Create a button in the last step of the wizard and assign a trigger to it that navigates the user to the board where power-up was installed. |  |
| Apply functional testing:   * Test Case 1: Verify that the user is taken directly to the board after completing the wizard. * Test Case 2: Verify that all power-up features are available and functional on the board. |  |
| **Feature 3- Alert component creation** | Create an alert message box that will appear after the user has completed their power-up setup. This message box should indicate the successful completion of the setup and be displayed on the project board where the setup was performed. |  |
| Create an alert component that triggers once the power-up setup has been successfully completed. |  |
| Apply functional testing:   * Test Case 1: Verify that the user receives an alert message after the setup process is completed. * Test Case 2: Verify that the alert message clearly indicates that the power-up setup is complete. * Test Case 3: Verify that the alert message provides information on what to do next or how to access the power-up features on the board. |  |

**Screenshots of the main features:**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Results of the Retrospective:**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

**Sprint Burndown chart:**

**Sprint 2:**

In the second iteration of the power-up development the goal is to implement the framework meaning the UX formula and the mapping table which are the core features of the power-up.

**Sprint Backlog:**

The proposed user stories for the first sprint of the power-up are as follows:

**User Story 1:** As a user, I want to assign scores for each dimension of a given user story in order to determine the worthiness score.

**User Story 2:** As a user, I want to modify the score of each dimension after setting them so that changing estimations are possible.

**User Story 3:** As a user, I want the ability to reset the worthiness score of a user story, so that I can adjust the estimation.

**UI Design of the User Stories:**

**User Story 1: Worthiness score form creation** – create button labeled “Set worthiness score”, when the button is clicked, a form will appear allowing the user can enter the score for each of the following dimensions: Potential value, Familiarity and Clarity). Once the user submits the score for each dimension, the worthiness score will be displayed with a color-coded indicator of its importance (red for high, orange for moderate, and green for low).

**User Story 2:** It does not require any design as it has form for setting the dimension already been designed in the previous sprint.

**User Story 3: Reset button creation** – create a “Reset” button next to the “worthiness score value”, so that allows the user to reset the worthiness score to “not estimate!”.

**Development of the User Stories:**

**User Story 1: Popup component and Card-buttons component creation –** create a Card-buttons component that, when clicked, triggers a popup with a form containing three distinct fields which represent potential value, familiarity, and clarity. In addition, each field will have 10 buttons numbered 1-10. When one of these buttons is clicked, the corresponding field will be updated.

**User Story 2: Enable to modify the dimensions scores** – display the current score of each dimension and enable the user to modify the scores by entering a new values. Also create a button labeled “Save” to enable the user to save the modified score.

**User Story 3: Card-buttons component creation** – Create a Card-buttons component labeled “Reset” that, when clicked, it will reset the worthiness score along with score of each dimension.

**Test of the User Stories:**

**User Story 1:**

* Test Case 1: Verify that a user is able to assign a score to each dimension of a user story.
* Test Case 2: Verify that the worthiness score is calculated correctly based on the scores assigned to each dimension.
* Test Case 3: Verify that the user is unable to submit the worthiness score calculation if any dimension score is missing.

**User Story 2**

* Test Case 1: Verify that a user is able to modify the score of a dimension after it has been set.
* Test Case 2: Verify that the worthiness score is updated correctly after the modification of a dimension score.

**User Story 3:**

* Test Case 1: Verify that a user is able to reset the worthiness score of a user story.
* Test Case 2: Verify that the worthiness score is reset to zero after the reset action.

**Sprint 3:**

The third iteration of the power-up development focuses on improving the user experience by providing more information and choices for users in regard to the dimensions and UX methods. The three user stories aim to allow users to gain a better understanding of the dimensions, view a list of suggested UX methods, and understand each method in order to make informed decisions.

**Sprint Backlog:**

**User Story 1:** As a user, I want to see description of each dimension, so that I can gain a better understanding of the dimensions.

**User Story 2:** As a user, I want to view a list of suggested UX methods, so that I can pick one of them.

**User Story 3:** As a user, I want to see description of each suggested UX methods, so that I decide which methods suits my needs.

**UI Design of the User Stories:**

**User Story 1: Info indicator creation** – create an info indicator next to each dimension in the Worthiness score form.

**User Story 2: UX methods list creation** – create a list of the recommended UX methods and display it in a list view.

**User Story 3: UX methods description creation -** create a text element for each UX method to display a description of the UX method.

**Development of the User Stories:**

**User Story 1: Info tooltip creation** – create an info tooltip next to each dimension in the Worthiness score form.

**User Story 2: UX methods** **checkbox list creation** – create a list of the recommended UX methods using checkboxes, and indicate the selection status by modifying the design of the checked item to green.

**User Story 3: UX methods description creation -** create a text element next to each UX method to display a description of the UX method.

**Test of the User Stories:**

**User Story 1:**

* Test Case 1: Verify that description is displayed for each dimension when the user clicks on the dimension info tooltip.
* Test Case 2: Verify that the description text accurately reflects the content of each dimension.
* Test Case 3: Verify that the description text is easily understandable by the average user.

**User Story 2:**

* Test Case 1: Verify that a list of suggested UX methods is displayed to the user.
* Test Case 2: Verify that the list includes all relevant UX methods.
* Test Case 3: Verify that the UX methods are listed in a clear and easily understandable manner.

**User Story 3:**

* Test Case 1: Verify that a description is displayed for each suggested UX method.
* Test Case 2: Verify that the description accurately reflects the content of each method.
* Test Case 3: Verify that the description text is easily understandable by the average user.

**Sprint 4:**

The fourth iteration of the power-up development focuses on improving efficiency and tracking of user stories in a project board. The sprint aims to achieve this by adding functionality to display the worthiness score, indicating which user stories have not been estimated, and tracking the progress of completed and pending UX methods across all user stories in the board.

**Sprint Backlog:**

**User Story 1:** As a user, I want to see the worthiness score of all the user stories in the board, so that I can prioritize then more efficiently.

**User Story 2:** As a user, I want to see “No, Estimate” text on all the user stories that have not been estimated, so that I know instantly which user stories are missing estimates.

**User Story 3:** As a user, I want to see the quantity of completed and pending UX methods across all user stories in the board, so I can keep track of the progress.

**UI Design of the User Stories:**

**User Story 1: Worthiness score label creation** – create a colored label that will be placed on the card cover to indicate the worthiness score using a color-coded indicator of its importance (red for high, orange for moderate, and green for low).

**User Story 2: No Estimate label creation** – create a label with the text “No Estimate” that will be placed on the card cover to present that the user stories that have not been estimated.

**User Story 3: Quantity label creation** – create a label that will be placed on the card cover to present the quantity of UX methods that have completed and are pending.

**Development of the User Stories:**

**User Story 1: Card-badges component creation (worthiness score)** – Create a Card-badges component that presents the worthiness score and the background color of the badge is dependent on color-coded indicator of its importance (red for high, orange for moderate, and green for low).

**User Story 2: Handle No Estimate case** – After creating the Card-badges component that represents user story 1, we will add a case that handles the user stories that have not been estimated, by displaying the worthiness score to “No Estimate”.

**User Story 3: Card-badges component creation (Quantity of UX method)** – Create a Card-badges component that presents the number of completed and pending UX methods for each user story.

**Test of the User Stories:**

**User Story 1:**

* Test Case 1: Verify that the worthiness score of each user story is displayed in the board.
* Test Case 2: Verify that the worthiness score is calculated accurately for each user story.

**User Story 2:**

* Test Case 1: Verify that the "No, Estimate" text is displayed for user stories that have not been estimated.
* Test Case 2: Verify that the text is displayed in a prominent location for easy visibility.

**User Story 3:**

* Test Case 1: Verify that the quantity of completed and pending UX methods is displayed for all user stories in the board.
* Test Case 2: Verify that the quantity of completed and pending UX methods is calculated accurately for each user story.

**Sprint 5:**

The Fifth and last iteration of the power-up development focuses on providing functionality for automatically moving completed user stories to a "Done list" and allowing users to select the "Done list" from their existing lists on the board. The goal is to provide visibility into the completion status of user stories and make the process more efficient.

**Sprint Backlog:**

**User Story 1:** As a user, I want to select the “Done list” from my existing lists on the board, so that I can automatically move the completed user story there.

**User Story 2:** As a user, I want the user story to be moved to the “Done list” automatically after all required UX methods are completed, so that I can know that this user story is completed.

**UI Design of the User Stories:**

**User Story 1:** ………………………………………………………………………………… …………………………………………………………………………………

**User Story 2:** ………………………………………………………………………………… ………………………………………………………………………………………………

**Development of the User Stories:**

**User Story 1: Custom Select Box creation** – create a custom select box to display existing lists of the selected board. Once the list is selected, it will be the assigned “Done list”.

**User Story 2: Auto move card creation** – create a trigger event that when the user completes all the required UX methods of a particular user story, the card associated with the user story will be moved to the “Done list”.

**Test of the User Stories:**

**User Story 1:**

* Test Case 1: Verify that the user is able to view all existing lists on the board.
* Test Case 2: Verify that the user can select the "Done list" from the existing lists.
* Test Case 3: Verify that once the "Done list" is selected, the user is able to move completed user stories there.
* **User Story 2:**
* Test Case 1: Verify that user story is marked as complete after all required UX methods are completed.
* Test Case 2: Verify that the user story is automatically moved to the "Done list" once it is marked as complete.
* Test Case 3: Verify that the user story is not moved to the "Done list" until all required UX methods are completed.

# 

# **Treatment Validation**

This chapter presents a validation study that was conducted with expert UX practitioners and an agile team to validate UX-Estimator power-up. This validation aims to answer these two questions: 1) Did the tool help novice practitioners or agile teams to better estimate the effort associated with UX activities or not? 2) Did the tool guide them in deciding which UX method to use? In addition, it discusses the threats to the validity of the validation study.

## **Planning**

## **……………**

## **……………**

# 

# **Conclusions and Future Work**

## **Conclusions**

## **………………..**

## **Future Work**

Checklist:

1. Product Backlog (name and description of user stories, prioritized, and hypotheses)
2. Iterations

Per each iteration:

* 1. Sprint Backlog (estimation, task, acceptance criteria for the different user stories)
  2. Design Models or any other intermediate results
  3. Screenshots of the main features
  4. Results of the Demo (stories approved, changes, …)
  5. Results of the Retrospective (Table with four columns: things that went well, things that went bad, things to improve, and action item to improve the process in the next sprint)
  6. Team velocity
  7. Sprint burndown chart

1. Final Project Burndown

# **References**

[1] T. S. da Silva, M. S. Silveira, F. Maurer, and F. F. Silveira, “The evolution of agile UXD,” *Inf Softw Technol*, vol. 102, pp. 1–5, 2018, doi: 10.1016/j.infsof.2018.04.008.

[2] T. S. da Silva, F. F. Silveira, M. S. Silveira, T. Hellmann, and F. Maurer, “A systematic mapping on agile UCD across the major agile and HCI conferences,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9159, pp. 86–100, 2015, doi: 10.1007/978-3-319-21413-9\_7/COVER.

[3] J. Choma, E. M. Guerra, A. Alvaro, R. Pereira, and L. Zaina, “Influences of UX factors in the Agile UX context of software startups,” *Inf Softw Technol*, vol. 152, p. 107041, Dec. 2022, doi: 10.1016/J.INFSOF.2022.107041.

[4] M. M. Alhammad, “A Gamified Framework to Integrate User Experience into Agile Software Development Process,” 2020. [Online]. Available: https://oa.upm.es/66064/

[5] A. Ananjeva, J. S. Persson, and A. Bruun, “Integrating UX work with agile development through user stories: An action research study in a small software company,” *Journal of Systems and Software*, vol. 170, 2020, doi: 10.1016/j.jss.2020.110785.

[6] K. Curcio, R. Santana, S. Reinehr, and A. Malucelli, “Usability in agile software development: A tertiary study,” *Comput Stand Interfaces*, vol. 64, pp. 61–77, 2019, doi: 10.1016/j.csi.2018.12.003.

[7] L. Caballero, A. M. Moreno, and A. Seffah, “How Agile Developers Integrate User-Centered Design into Their Processes: A Literature Review,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 26, no. 8, pp. 1175–1201, 2016, doi: 10.1142/S0218194016500418.

[8] Å. Cajander, M. Larusdottir, and J. L. Geiser, “UX professionals’ learning and usage of UX methods in agile,” *Inf Softw Technol*, vol. 151, p. 107005, 2022.

[9] N. Pillay and J. Wing, “Agile UX: Integrating good UX development practices in Agile,” in *2019 Conference on Information Communications Technology and Society (ICTAS)*, 2019, pp. 1–6. doi: 10.1109/ICTAS.2019.8703607.

[10] A. P. O. Bertholdo, F. Kon, and M. A. Gerosa, “Agile usability patterns for user-centered design final stages,” *18th International Conference on Human-Computer Interaction, HCI International 2016*, vol. 9731. Springer Verlag, Department of Computer Science, University of São Paulo, São Paulo, Brazil, pp. 433–444, 2016. doi: 10.1007/978-3-319-39510-4\_40.

[11] D. Lang, S. Spies, S. Trieflinger, and J. Münch, “Tailored design thinking approach-a shortcut for agile teams,” in *International Conference on Software Business*, 2021, pp. 37–49.

[12] Å. Cajander, M. Larusdottir, and J. L. Geiser, “UX professionals’ learning and usage of UX methods in agile,” *Inf Softw Technol*, vol. 151, p. 107005, 2022.

[13] K. Curcio, R. Santana, S. Reinehr, and A. Malucelli, “Usability in agile software development: A tertiary study,” *Comput Stand Interfaces*, vol. 64, pp. 61–77, 2019, doi: 10.1016/j.csi.2018.12.003.

[14] G. Jurca, T. D. Hellmann, and F. Maurer, “Integrating Agile and User-Centered Design: A Systematic Mapping and Review of Evaluation and Validation Studies of Agile-UX,” in *2014 Agile Conference*, 2014, pp. 24–32. doi: 10.1109/AGILE.2014.17.

[15] A. Bruun, M. K. Larusdottir, L. Nielsen, P. A. Nielsen, and J. S. Persson, “The role of UX professionals in agile development: A case study from industry,” in *10th Nordic Conference on Human-Computer Interaction, NordiCHI 2018*, 2018, pp. 352–363. doi: 10.1145/3240167.3240213.

[16] R. Hartson and P. Pyla, “The UX book: Agile UX design for a quality user experience,” *The UX Book: Agile UX Design for a Quality User Experience*, pp. 1–888, Jan. 2018, doi: 10.1016/C2013-0-19285-9.

[17] A. Gordillo, E. Barra, S. Aguirre, and J. Quemada, “The usefulness of usability and user experience evaluation methods on an e-Learning platform development from a developer’s perspective: A case study,” in *44th Annual Frontiers in Education Conference, FIE 2014*, 2014, vol. 2015-Febru, no. February, pp. 1–8. doi: 10.1109/FIE.2014.7044340.

[18] A. P. O. Bertholdo, T. S. Da Silva, C. De O. Melo, F. Kon, and M. S. Silveira, “Agile usability patterns for UCD early stages,” *3rd International Conference on Design, User Experience, and Usability: Theories, Methods, and Tools for Designing the User Experience, DUXU 2014, Held as Part of 16th International Conference on Human-Computer Interaction, HCI International 2014*, vol. 8517 LNCS, no. PART 1. Springer Verlag, IME/USP - Universidade de São Paulo, Brazil, pp. 33–44, 2014. doi: 10.1007/978-3-319-07668-3\_4.

[19] D. Lang, S. Spies, S. Trieflinger, and J. Münch, “Tailored design thinking approach-a shortcut for agile teams,” in *International Conference on Software Business*, 2021, pp. 37–49.

[20] M. Alhammad and A. Moreno, “Integrating User Experience into Agile -- An Experience Report on Lean UX and Scrum,” Apr. 2022, doi: 10.1145/3510456.3514156.

[21] K. Beck *et al.*, “Manifesto for agile software development,” 2001.

[22] H. K. Flora and S. v Chande, “A systematic study on agile software development methodologies and practices,” *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 3626–3637, 2014.

[23] B. Boehm and R. Turner, “Management challenges to implementing agile processes in traditional development organizations,” *IEEE Softw*, vol. 22, no. 5, pp. 30–39, 2005.

[24] H. Hajjdiab and A. S. Taleb, “Adopting agile software development: issues and challenges”.

[25] D. Norman, J. Miller, and A. Henderson, “What you see, some of what’s in the future, and how we go about doing it: HI at Apple Computer,” in *Conference companion on Human factors in computing systems*, 1995, p. 155.

[26] J. J. Garrett, *The Elements of User Experience: User-Centered Design for the Web and Beyond*, 2nd ed. USA: New Riders Publishing, 2010.

[27] A. G. Mirnig, A. Meschtscherjakov, D. Wurhofer, T. Meneweger, and M. Tscheligi, “A formal analysis of the ISO 9241-210 definition of user experience,” *Conference on Human Factors in Computing Systems - Proceedings*, vol. 18, pp. 437–446, Apr. 2015, doi: 10.1145/2702613.2732511.

[28] S. Kujala, V. Roto, K. Väänänen-Vainio-Mattila, E. Karapanos, and A. Sinnelä, “UX Curve: A method for evaluating long-term user experience,” *academic.oup.com*, 2011, doi: 10.1016/j.intcom.2011.06.005.

[29] H. B. Santoso and M. Schrepp, “Importance of user experience aspects for different software product categories,” in *International Conference on User Science and Engineering*, 2018, pp. 231–241.

[30] C. Salvador, A. Nakasone, and J. A. Pow-Sang, “A Systematic Review of Usability Techniques in Agile Methodologies,” in *Proceedings of the 7th Euro American Conference on Telematics and Information Systems*, 2014. doi: 10.1145/2590651.2590668.

[31] I. O. for Standardization, *Human-centred design processes for interactive systems*. International Organization for Standardization, 1999.

[32] I. 159/SC 4 E. of Human-System Interaction (Subcommittee), *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs).: Guidance on Usability*. International Organization for Standardization, 1998.

[33] J. Nielsen, *Usability engineering*. Morgan Kaufmann, 1994.

[34] “IEEE Standard for a Software Quality Metrics Methodology,” *IEEE Std 1061-1992*, pp. 1–96, 1993, doi: 10.1109/IEEESTD.1993.115124.

[35] T. S. da Silva, A. Martin, F. Maurer, and M. Silveira, “User-centered design and agile methods: A systematic review,” in *2011 Agile Conference, Agile 2011*, 2011, pp. 77–86. doi: 10.1109/AGILE.2011.24.

[36] O. Sohaib and K. Khan, “Integrating usability engineering and agile software development: A literature review,” in *2010 International Conference On Computer Design and Applications*, 2010, vol. 2, pp. V2-32-V2-38. doi: 10.1109/ICCDA.2010.5540916.

[37] D. Salah, R. F. Paige, and P. Cairns, “A systematic literature review for Agile development processes and user centred design integration,” in *18th International Conference on Evaluation and Assessment in Software Engineering, EASE 2014*, 2014. doi: 10.1145/2601248.2601276.

[38] D. Argumanis, A. Moquillaza, and F. Paz, “Challenges in Integrating SCRUM and the User-Centered Design Framework: A Systematic Review,” *Communications in Computer and Information Science*, vol. 1334, pp. 52–62, 2020, doi: 10.1007/978-3-030-66919-5\_6/COVER.

[39] K. Kuusinen, “Task allocation between ux specialists and developers in agile software development projects,” *15th IFIP TC 13 International Conference on Human–Computer Interaction, INTERACT 2015*, vol. 9298. Springer Verlag, Tampere University of Technology, Tampere, Finland, pp. 27–44, 2015. doi: 10.1007/978-3-319-22698-9\_3.

[40] J. S. Persson, A. Bruun, M. K. Lárusdóttir, and P. A. Nielsen, “Agile software development and UX design: A case study of integration by mutual adjustment,” *Inf Softw Technol*, vol. 152, p. 107059, Dec. 2022, doi: 10.1016/J.INFSOF.2022.107059.

[41] K. Conboy, “Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development,” *Info. Sys. Research*, vol. 20, no. 3, pp. 329–354, Sep. 2009, doi: 10.1287/isre.1090.0236.

[42] A. Bruun, M. Larusdottir, L. Nielsen, … P. N.-P. of the 10th, and undefined 2018, “The role of UX professionals in agile development: a case study from industry,” *dl.acm.org*, pp. 352–363, Sep. 2018, doi: 10.1145/3240167.3240213.

[43] A. Abulfaraj and A. Steele, “Coherent Heuristic Evaluation (CoHE): Toward Increasing the Effectiveness of Heuristic Evaluation for Novice Evaluators,” in *Design, User Experience, and Usability. Interaction Design*, 2020, pp. 3–20.

[44] J. Gothelf and J. Seiden, *Lean ux*. “ O’Reilly Media, Inc.,” 2021.

[45] R. J. Wieringa, “Design science methodology: For information systems and software engineering,” *Design Science Methodology: For Information Systems and Software Engineering*, pp. 1–332, Jan. 2014, doi: 10.1007/978-3-662-43839-8/COVER.

[46] “UX Research Cheat Sheet.” https://www.nngroup.com/articles/ux-research-cheat-sheet/ (accessed Oct. 15, 2022).

[47] G. Guest, A. Bunce, and L. Johnson, “How many interviews are enough? An experiment with data saturation and variability.,” *Field methods*, vol. 18, pp. 59–82, 2006, doi: 10.1177/1525822X05279903.

[48] O. Balci, “Quality assessment, verification, and validation of modeling and simulation applications,” in *Proceedings of the 2004 Winter Simulation Conference, 2004.*, 2004, vol. 1, p. 129. doi: 10.1109/WSC.2004.1371309.

[49] “Visual Studio Code - Code Editing. Redefined.” https://code.visualstudio.com/ (accessed Jan. 23, 2023).

[50] F. de A. Farzat, M. de O. Barros, and G. H. Travassos, “Evolving JavaScript Code to Reduce Load Time,” *IEEE Transactions on Software Engineering*, vol. 47, no. 8, pp. 1544–1558, 2021, doi: 10.1109/TSE.2019.2928293.

[51] “Trello | Atlassian.” https://www.atlassian.com/software/trello (accessed Jan. 23, 2023).

[52] “Get Started Building on Trello.” https://developer.atlassian.com/cloud/trello/ (accessed Jan. 23, 2023).

[53] “Develop and deploy websites and apps in record time | Netlify.” https://www.netlify.com/ (accessed Jan. 23, 2023).

[54] “GitHub: Let’s build from here · GitHub.” https://github.com/ (accessed Jan. 23, 2023).

[55] “Get Started Building on Trello.” https://developer.atlassian.com/cloud/trello/ (accessed Jan. 20, 2023).

# **Appendices**

## **Appendix A - Interview Participants**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Participant** | **Gender** | **Years of Experience** | **Current of Position** | **Industry/field** |
| P1 saif | Male | 6 Years | User Research Manager | Financial Services |
| P2 saad | Male | 3+phd | Digital User Experience Design Manager | Banking |
| P3 Bandar | Male | 7 Years | Customer Experience Director | Government Administration |
| P4 LUJAIN | Female | 7 Years | Loyalty Digital Experience Manager | Banking |
| P6 ANAS | Male | 6 Years | Head of Core products | Technology, Information and Internet |
| P5 HAMAD | Male | 6 Years | Senior UX/UI Designer | Banking |
| P6 Nasser | Male | 5 Years | UI/UX Team lead | Telecommunications |
| P7 Muath | Male |  |  |  |
| P8 | Male |  |  |  |
| P9 | Male |  |  |  |
| P10 sara | Female | 6 Years | Senior UX/UI Designer |  |

**Appendix B Title**

Some detail if any

**[FYI: the following are summaries of the selected papers]**

**Influences of UX factors in the Agile UX context of software startups**

Choma presented a conceptual framework named UX Factors in Software Startups (UXFiSS) that identifies the factors that promote UX work and those that make it challenging and organizes them into four dimensions: business & market, product & process, customers & users, and UX work & teams. As a result of analyzing the responses, they determined which factors influence UX work in four dimensions based on four variables: startup age, startup size, time professionals have been applying UX practices to startups, as well as the areas in which they have been employed. The responses were classified into four groups based on the roles respondents play in startups. The responses were then verified by the Kruskal-Wallis (KW) test and by the Dwass-Steele-Critchlow-Fligner (DSCF) post hoc test to determine whether statistically significant differences were observed between groups and which groups differed in comparisons of factors influencing UX work in pairwise comparisons. Researchers found that software startup practitioners tend to emphasize positive factors that facilitate UX work more than negative factors that hinder it. The most frequently cited positive factors included delivering value to the business and the user as well as producing successful products. As for the negative factors that hinder UX work, lack of economical and human resources is a significant hindrance, especially in early-stage startups.

**Tailored Design Thinking Approach - A Shortcut for Agile Teams**

Lang [14] developed a supportive tool called Discovery Effort Worthiness Index (DEW Index) that assists agile teams in determining how much effort they should spend on Design Thinking methods to discover and validate requirements. A case study was conducted with a scrum team, in which user stories were analyzed using Design Thinking before sprint planning, and then backlog items were analyzed in workshops with the assistance of two UX and Design Thinking experts to assign corresponding Design Thinking methods based on the input-output ratio. In order to determine what dimensions influenced selecting the right Design Thinking methods, the authors asked the team and UX and Design Thinking experts why they decided to choose a particular method and why they decided to spend the effort involved. These interviews with experts and team members revealed two key factors, which were called dimensions. The following two dimensions are Clarity and Potential Value. Clarity indicates how well the actual demand of the user is understood, while Potential Value indicates how much value does the feature deliver. Next, the team selected a set of user stories from the team's benchmark and rated them according to the identified dimensions following Scrum poker's relative rating principle. The ratings were based on a scale of 1-10, where 10 represents high and 1 represents low. In examining the results of the dimension rating and the recommended effort by UX and Design Thinking experts, a correlation was observed between the dimensions and the effort worth spending. Based on these results, the formula was developed that enables the product owner to determine the score independently. After the benchmark was set in the preceding step, the assigned Design Thinking methods were sorted according to their DEW Index so that the product owner could choose from the ranges once, several times or not at all, depending on the user story and DEW Index.

* **Coherent Heuristic Evaluation (CoHE): Toward Increasing the Effectiveness of Heuristic Evaluation for Novice Evaluators**

Our framework still needs to be validated in the context of agile teams. A similar approach has been taken by Abulfaraj [\*] demonstrating how novice evaluators can use these Heuristic Evaluations in the most effective manner, while we strive to guide agile teams that are novice UX practitioners to apply UX practices. Abulfaraj [\*] developed a step-by-step protocol called Coherent Heuristic Evaluation (CoHE) to facilitate the use of HE for novice evaluators, thereby improving their performance and the quality of results they produce. To develop such a protocol, it is necessary to identify what are the difficulties that novices face when applying HE, which is why the author interviewed usability experts since they were once novices and were familiar with these problems. Therefore, they have a good understanding of how to overcome these difficulties. The author then analyzed the interview data to come up with coherent insights, focusing on the difficulties experienced by usability experts when they began doing HE and the mistakes novices were making. In addition, how to overcome these challenges, as well as how to improve HE as a method and focused on the overall organization of the HE process. Based on the analysis, they developed a step-by-step protocol.

* **Task Allocation Between UX Specialists and Developers in Agile Software Development Projects**

Kuusinen [27] carried out a multi-case study to understand how UX work is conducted in agile projects, and how developers and UXSs cooperate and allocate tasks, thereby clarifying which UX-related tasks can or even should be handled by developers, and which require the special competence of a UXS. The author identified three descriptive types of cooperation, namely minimal, PO–UXS, and Developer–UXS cooperation. In projects with the minimal cooperation type, the UXS works mainly apart from the other team members, while with the PO–UXS and Developer–UXS cooperation types, the UXS works mostly with the PO and developers, respectively. The Developer–UXS cooperation was the most desirable cooperation type among the participant projects.

* **The Role of UX Professionals in Agile Development: A Case Study From Industry**

Bruun et al. [31] presented an in-depth case study to understand the responsibilities associated with UX professionals in an agile company that successfully integrated UX activities. The authors conducted semi-structured, qualitative interviews with employees (UX consultants, project managers, software developers and a software architect), and all of them considered UX activities crucial for project success. However, it became apparent that UX professionals have the leverage to reject implemented designs that do not meet requirements, as they are responsible for managing requirements. It was found in the study that UX practitioners have a very broad set of responsibilities including sales and business development, both of which place a customer focus over a user-centered focus. It is therefore possible that UX practitioners may not have enough time to conduct all the UX experiment that is needed, especially when it comes to an experiment that requires a lot of time.

* **Agile software development and UX design: A case study of integration by mutual adjustment**

Persson et al. [40] conducted a case study to how Agile and UX processes are integrated to maintain agility through mutual adjustment of software development and UX design. The authors used three perspectives on agility that include assimilation, separation, and integration. Assimilation relies on coordinating by direct supervision and standardization of work processes, while Separation relies on coordination by standardization of skills and outputs. Integration lies between Assimilation and Separation, and it is taken from a across-entities perspective. They analyzed the company’s integration approaches of upfront design and work in parallel with Conboy’s taxonomy of Agility [41]. The results of this analysis showed how Agility differs for the two roles (Software development and UX design) in relation with these integration approaches (Agile and UX). Through the process of mutual adjustment, the Agility for UX designers and software developers are different, yet complementary. This integration contrasts with assimilation, which potentially makes their Agility mutually indistinguishable, and with separation, which makes their Agility different and mutually competing.

* **Integrating UX work with agile development through user stories: An action research study in a small software company**
* **The evolution of agile UXD**

However, the requirements are not always become clear due to the fact the business not always contented or did not think about all the cases of the feature.”, also the participant mentioned “that most of the time the clients have a huge idea without defining the requirements and expect a solution from us”.

In addition, another participant stated, “In practice, the requirements are not entirely clear, meaning a UX practitioner needs clarity to be more creative and design something better.” The participant continued, saying that “businesses are not always satisfied or did not consider all the scenarios that might apply to the feature.”.

1. Saif
2. Moath
3. Sara
4. Hamad
5. Nasser
6. Saad
7. Anas
8. Bandar
9. Lujain

مكان الux في أي قطاع مررة يأثر على موضوع هل يسوونه بخذافيرها او لا مثالا

قال انس ان ux مع الماركتنق فمو قاعدين يطبقونها او يلقون لها بال

اذا كانت مع البرودكت يهتمون للفيتشرز الي لها فالو للبزنس اكثر

اذا مع الديف و الاوبريشن يهتمون بالفيتشرز الي يهم المستخدم

[Sarah said that I should finish the feature that has high business value with clear req then I start with the unfamiliar feature] even Hamad said that.

[SAAD and bandar 54.20] familiarity, do at least minimum req and Hamad has a good matrix (11 2, 35:30)

Part 2:

Anas

Saif

Moath

Lujain

Hamad

draft

Power-up Application in Trello is an additional function that allows the integration of Trello with other applications

The development process began with designing an interface for UX-Estimator tool ; to go about this, a series of mock-up diagrams were fabricated and reviewed. Feedback from the latter culminated in an iterative improvement process where intended UI options and design were continuously refined and reviewed, before a final design was realised.

Graphical user interface, text, application, chat or text message

Description automatically generated

1. **Design Stage:**

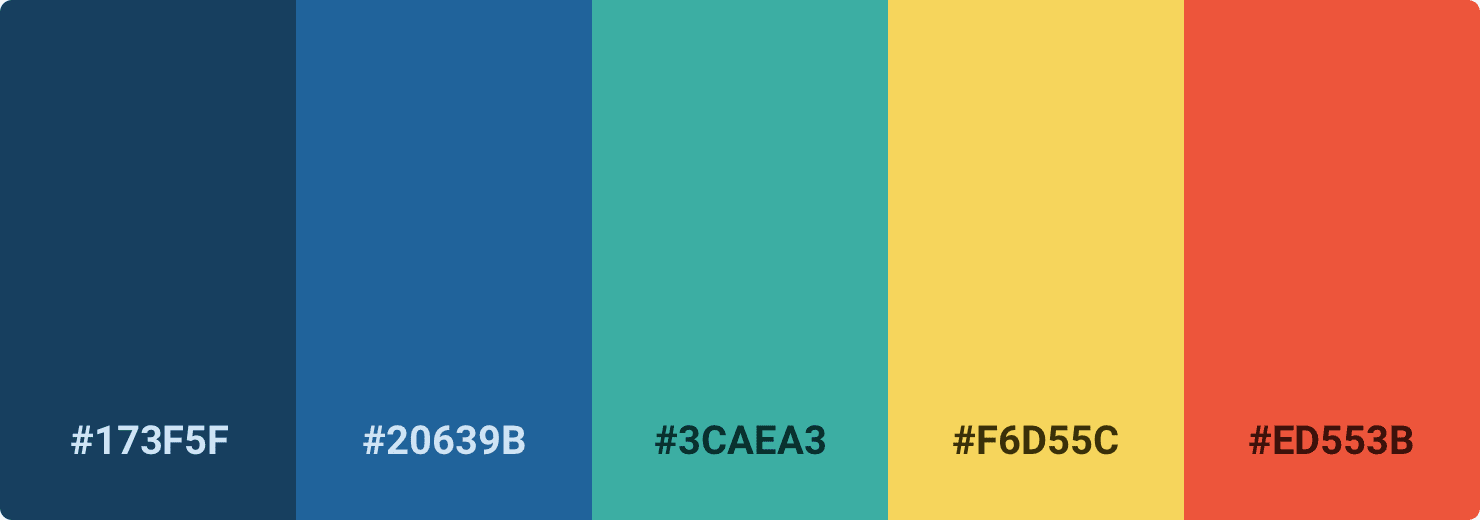
The first step in developing my Trello Power-up was to design it. We started by brainstorming ideas and researching existing power-ups to get an idea of what features we wanted to include. After that, we created a wireframe of the user interface and wrote out the user stories that would help guide the development process. We also created a list of tasks that needed to be completed in order to build the power-up. This gave us a clear roadmap for development and allowed us to stay organized throughout the process.

1. **Implementation Stage:**

[We need to mention that we worked though sprints]

Once we had our design finalized, it was time to start developing the power-up. We began by setting up our development environment and creating a repository for our code. From there, we started coding each feature one at a time, making sure to test as we went along. As we worked on each feature, we made sure that it was working properly before moving on to the next one. This allowed us to ensure that everything was working as expected before deploying it.

1. **Deployment Stage:**

………………………………………………….…………………….………………………………………………….…………………….………………………………………………….…………………….………………………………………………….…………………….………………………………………………….…………………….………………………………………………….…………………….………………………………………………….…………………….………………………………………………….…………………….………………………………………………….…………………….………………………………………………….…………………….………………………………………………….…………………….………………………………………………….…………………….………………………………………………….……………………. 

* There is an inherent tension between business and customer values, but they can be aligned to achieve success to product or service as neither should be neglected.
* The majority of participants agreed that both should be taken into consideration, albeit with different levels of emphasis depending on the company's product strategy.

**Potential Value**

**Summary of Interview Findings**

**Clarity**

**Familiarity**

* Clarity of user story should be considered when planning a feature.
* Feature should be detailed and clear before moving to the design phase.
* Wireframes are better than documents for gaining clear requirement.
* Complexity of the feature should be defined and clear in order to determine if UX experiments are necessary.
* Having experience within the product domain helps when deciding to conduct UX experiments.
* Maturity of the UX designer is a factor when making decisions.
* Changing features that impact existing user behavior is challenging and requires intensive UX experiments.
* Introducing new features into the market takes a lot of effort and resources but can offer huge gains in value.
* The tool was found to be helpful in selecting appropriate UX methods by the majority of participants.
* The methods suggested by the tool were found to be comparable to the methods that participants would use for their scenarios.
* The tool is a good starting point, but having an expert practitioner on hand for additional support and guidance would also be valuable.
* For novice practitioners, several participants found the tool to be a great way to start, while more experienced practitioners may reduce certain suggested methods as necessary.

**RQ2:** Does the UX-Estimator power-up guide novice practitioners and agile teams in selecting appropriate UX methods?

**Key RQ Results**

* The tool was found to be highly beneficial by the participants in estimating effort associated with UX activities.
* The tool was particularly helpful for novice practitioners who may not have extensive experience in estimating effort for UX activities.
* The tool provided accurate estimations which allowed for better project planning and decision-making.

**RQ1:** Does the UX-Estimator power-up effectively assist novice practitioners and agile teams in estimating effort associated with UX activities?