UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS INFORMÁTICOS

**A Gamified Framework to Integrate User Experience into Agile
Software Development Process**

Manal M. Alhammad

Bachelor's degree in Information Technology, King Saud University

Master's degree in Computing (Software Engineering), Imperial College London

Supervisor: Ana M. Moreno

PhD in Computer Science, Universidad Politécnica de Madrid

A thesis submitted for the degree of Doctor of Philosophy

Madrid 2020

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS INFORMÁTICOS

**A Gamified Framework to Integrate User Experience into Agile
Software Development Process**

Manal M. Alhammad

Bachelor's degree in Information Technology, King Saud University

Master's degree in Computing (Software Engineering), Imperial College London

Supervisor: Ana M. Moreno

PhD in Computer Science, Universidad Politécnica de Madrid

A thesis submitted for the degree of Doctor of Philosophy

Madrid 2020

Abstract

While the significance of integrating User eXperience Design (UXD) with agile development processes is underscored in the literature, there are yet challenges and issues that must be resolved to achieve a seamless integration process. Those challenges range from practical issues such as the lack of technical means in terms of supportive tools or artefacts, to more fundamental issues such as the complexity of building a shared understanding of the User eXperience (UX) vision among the entire agile team and organization. Those issues do not necessarily imply that the integration of agile and UX is infeasible, but rather point to areas in which further research is needed to improve the integration process.

In this respect, we saw an opportunity to contribute to improving the process of integrating UX practices into agile by addressing one of the reported challenges in the literature which is the lack of prioritizing UX aspects compared to functional aspects. Recognizing the importance of incorporating UXD into agile as a team culture, this thesis aims to design and validate a framework to help agile teams to develop a motivated mindset towards UX activities. Following Design Science Research (DSR), this doctoral work began by identifying, describing, and analyzing the challenge of focusing on functional features over UX features in agile teams. We essentially found that the prioritization of functional features over UX features embodies a set of issues and it represents a recurrent obstacle when integrating UX with agile. Among those issues are the lack of motivation towards UX work, resistance to change, and the curse of knowledge in which agile teams think they know better than the user.

To address this challenge, we proposed a gamified framework to integrate Lean UX into Scrum, called “GLUX” (Gamified Lean UX). The goal of our proposed framework is to motivate and engage agile teams to carry out UX activities collaboratively and develop a user-centered mindset using gamification. The choice of gamification as a motivational technique is primarily based on the hypothesis established in the literature that gamification has potential in influencing human behavior in general, and in promoting software engineering best practices in particular.

In validating our proposed framework, we conducted a pilot study to primarily examine the effectiveness of GLUX in terms of usability and utility in promoting and developing a motivated mindset in agile teams towards UX. Three agile teams consisting of software engineering graduate students participated in the study, simulating a real-world agile project. Our data collection occurred through student surveys, unstructured interviews, and the analysis of projects reports. Preliminary results suggest that GLUX can be effective in facilitating the integration process by engaging Scrum teams in carrying out UX activities throughout the development process. However, similar outcomes cannot be guaranteed if the GLUX framework was to be applied into a different context or when not implemented as specified.

Nonetheless, key observations have been reported in the form of lessons learned. We mainly found that the team’s collective mindset towards UX work can sometimes override the individual’s mindset, and that it can be influenced by a variety of factors such as the knowledge and experience in UX. We have also discussed some of the challenges that can emerge from managing the integration of agile and UX as one process such as the difficulty to incorporate UX activities and development work in short iterations. Finally, we highlight that gamification can be susceptible to the “novelty effect”, and that some game elements (e.g. challenges) can be motivating and de-motivating at the same time.

Resumen

Si bien en la literatura se subraya la importancia de integrar User eXperience (UX) con procesos de desarrollo ágiles, aún existen desafíos y problemas que deben resolverse para lograr un proceso de integración eficiente. Esos desafíos van desde cuestiones prácticas, como la falta de medios técnicos en términos de herramientas o artefactos de apoyo, hasta cuestiones más fundamentales como la complejidad de construir una comprensión compartida de la visión de la experiencia del usuario entre todo el equipo y la organización ágil. Estos problemas identifican áreas en las que se necesita más investigación para mejorar dicho proceso de integración.

Esta investigación aborda uno de estos desafíos, concretamente la necesidad de priorizar los aspectos de UX en comparación con los aspectos funcionales en el desarrollo ágil, lo que implica a su vez abordar, por ejemplo aspectos como la falta de motivación hacia el trabajo UX o la resistencia al cambio.

En este contexto, esta tesis doctoral tiene como objetivo diseñar un marco para ayudar a los equipos ágiles a considerar la UX como elemento guía del desarrollo ágil. Siguiendo Design Science Research (DSR) se ha propuesto un marco gamificado para integrar Lean UX en Scrum, llamado "GLUX" (Gamified Lean UX). El objetivo de GLUX es utilizar gamificación para motivar e involucrar a los equipos ágiles a realizar actividades de UX de manera colaborativa desarrollando una mentalidad de UX que guíe y motive el desarrollo ágil. La elección de la gamificación como técnica motivacional se basa principalmente en la hipótesis establecida en la literatura de que la gamificación tiene potencial para influir en el comportamiento humano en general y para promover las mejores prácticas de ingeniería de software en particular.

GLUX se ha aplicado en un estudio piloto con el fin de analizar su efectividad en términos de usabilidad y utilidad para promover y desarrollar una mentalidad UX en equipos ágiles. Tres equipos ágiles formados por estudiantes graduados de ingeniería de software participaron en el estudio, simulando un proyecto ágil. Se han recopilado datos a través de encuestas a los estudiantes, entrevistas no estructuradas y análisis de informes de proyectos. Los resultados sugieren que GLUX puede ser eficaz para facilitar el proceso de integración al involucrar a los equipos Scrum en la realización de actividades UX durante todo el proceso de desarrollo. Estos resultados, aunque alentadores son preliminares, requiriéndose replicación tanto en contextos similares como diferentes.

Además de estos resultados preliminares, se han identificado importantes lecciones aprendidas relacionadas con el uso de la gamificación en el desarrollo de software, como por ejemplo, el efecto de la mentalidad colectiva del equipo en relación con la mentalidad individual; el efecto de las iteraciones cortas que guían el trabajo ágil sobre la integración de UX; o el posible “efecto novedad” del que puede ser susceptible la gamificación haciendo algunos elementos del juego motivadores y desmotivadores al mismo tiempo.

Acknowledgements

I would like to first thank my supervisor Dr. Ana M. Moreno for her excellent and continuous guidance and support throughout this doctoral research. Her constructive feedback and invaluable assistance played a crucial role in the successful and timely completion of this thesis.

I wish to express my gratitude to King Saud University and the government of Saudi Arabia for offering the incredible opportunity to pursue higher education with a full scholarship.

My biggest thanks go to my lovely family. Special thanks to my mom Kaydah, from whom I learned patience, dedication, and commitment. To my dad Mubarak, who always inspires me to be a better person and a researcher.

To my partners in this journey, my husband Abdullah, and my son Faris, without whom this thesis would have been completed two years earlier, thank you for being amazing.

Table of Contents

1	Introduction.....	2
1.1	Motivation and Problem Context.....	2
1.2	Research Goal	4
1.3	Research Questions	5
1.4	Research Contributions	5
1.5	Thesis Structure	6
2	Research Methodology	9
2.1	Introduction.....	9
2.2	A Brief Overview of Design Science Research (DSR).....	9
2.3	The Design and Engineering Cycles	11
3	Problem Investigation	17
3.1	Introduction.....	17
3.2	Recent Developments in Agile and UX Integration	18
3.3	Challenges of Agile and UX Integration.....	19
3.4	The Challenge of Prioritizing UX Activities in the Agile Development Cycle.....	23
4	Treatment Design.....	28
4.1	Introduction.....	28
4.2	Background and Related Work	29
4.2.1	Scrum and Lean UX.....	29
4.2.2	Gamification	30
4.2.3	Gamification in Software Engineering.....	31
4.3	The Design Process of GLUX	35
4.3.1	Step 1. Define your objectives	36
4.3.2	Step 2. Delineate target behaviors.....	37
4.3.3	Step 3. Describe your players	39
4.3.4	Step 4. Devise your activity cycles	42
4.3.5	Step 5. Don't forget the fun	43
4.3.6	Step 6. Deploy the appropriate tools for the job	43
4.4	GLUX: A Gamified Framework to Integrate Lean UX Tactics into Scrum	46
4.4.1	Overview of GLUX	46
4.4.2	The Scope of GLUX	47
4.4.3	General Guidelines.....	47

4.4.4	GLUX in Practice.....	48
5	Treatment Validation	58
5.1	Introduction.....	58
5.2	Validation Goals and Research Questions	59
5.3	Validation Study Design	60
5.3.1	Client Selection and Overview of the Model Problem Context.....	60
5.3.2	Treatment Design.....	62
5.3.3	Measurement Design.....	67
5.3.4	Inference Design	71
5.3.5	Validity Threats.....	74
5.4	Validation Execution	77
5.4.1	Quick Summary of the Setting.....	77
5.4.2	Results.....	77
5.5	Discussion	96
5.5.1	Summary of Results	96
5.5.2	Lessons Learned.....	98
5.5.3	Improvements to GLUX	103
5.5.4	Design Theory	104
6	Conclusion	106
6.1	Research Contributions	106
6.2	Key Findings	107
6.3	Limitations	108
6.4	Directions for Future Research	109
6.5	Concluding Remarks.....	111
7	Bibliography	113
8	Appendices.....	119
	Appendix A – The Lean UX Canvas	119
	Appendix B – Badges Collection.....	121
	Appendix C – Questionnaire QNR1 and Responses.....	123
	Appendix D – The GLUX Guide	126
	Appendix E - Questionnaire QNR2 and Responses.....	139
	Appendix F - Transcript of The Retrospective Class Interview	149
	Appendix G - The GLUX Framework v2.0.....	155

List of Figures

Figure 1. Design science iterates over two main activities.	10
Figure 2. This doctoral research iterates over design and investigation, starting with investigation.....	11
Figure 3. The engineering and design cycles of the DSR framework (Wieringa, 2014).	12
Figure 4. Problem investigation is the first activity in the design cycle of this doctoral research.	17
Figure 5. A graphical representation of the seven themes of agile and UX integration challenges based on the work of Curcio et al. (2019).	21
Figure 6. A fishbone diagram illustrating the phenomenon of the developers' mindset towards UX in the integration of agile and UX. It represents a fish's skeleton with the key problem at its head and the influencing factors of the problem feeding into the spine.....	25
Figure 7. Treatment (artefact) design is the second activity in the design cycle of this doctoral research.	28
Figure 8. The 6D gamification process by (Werbach and Hunter, 2015b).	35
Figure 9. Example experiment story.	38
Figure 10. Engagement loops provide feedback and triggers to keep the players engaged in the gamified process.	42
Figure 11. Challenges are handled every other sprint.....	43
Figure 13. A general overview of the GLUX framework, integrating Scrum with Lean UX using gamification.	46
Figure 14. The five Lean UX tactics to integrate into the Scrum framework.....	48
Figure 15. The design studio is a collaborative design tactic that involves sketching and discussing the team's design ideas.....	50
Figure 16. Example of an experiment story containing the hypothesis to be validated, the validation tactic, the responsible member/s, and the estimated effort.	51
Figure 17. The three main techniques employed in the gamification strategy of GLUX.	52
Figure 18. The challenges technique of the GLUX framework.....	54
Figure 19. The levels technique offers five levels of competency that reflect the team's experience in integrating Lean UX with Scrum.	55
Figure 20. Achievements board is a useful technique to provide visibility of the team's collected points.	56
Figure 21. Achievements board can also be used to showcase the team's collected badges.	56
Figure 22. The validation of an artefact in the design cycle.	58
Figure 23. The validation model is used to represent the real-world context.	59
Figure 24. The badges collection designed and included in the GLUX guide (i.e. the client-specific version of the GLUX framework).	63
Figure 25. The gamification strategy of the GLUX guide uses a point system to move through its three levels.	65
Figure 26. The Lean UX achievements board of collected points and levels.....	65
Figure 27. The Lean UX achievements board of collected badges.....	66
Figure 28. The structure of the students' final project report.	70
Figure 29. The qualitative data analysis process adopted in this validation study.....	72
Figure 30. Nodes are created for each code in Nvivo12.	73

Figure 31. Relevant text segments are placed under the appropriate nodes.	74
Figure 32. Memos were used to capture the author's thoughts and interpretations of particular observations.	74
Figure 33. Participants perceived knowledge of Lean UX.	79
Figure 34. Participants opinions on the effect of Lean UX on the team's agility.....	81
Figure 35. Participants opinions on the effect of gamification on the team's agility.	82
Figure 36. A summary of the challenges of integrating Lean UX into agile.	83
Figure 37. Screenshot of the Trello board of team B showing how the team color labelled product backlog items to distinguish Lean UX tactics.	87
Figure 38. Participants' rating of the effectiveness of each gamification element of the GLUX framework.	88
Figure 39. Teams opinions on whether the GLUX framework improved their collaboration.	89
Figure 40. Participants opinions on the overall usability of the GLUX framework.	90
Figure 41. The most enjoyed gamification elements of the GLUX framework.	91
Figure 42. Participants agreement rates to the statement "I view UX as equally important as development".	92
Figure 43. Participants' willingness and intentions to apply Lean UX in their future projects.....	93
Figure 44. Participant agreement on the role of the GLUX framework on improving their motivation and mindset towards UX.	95
Figure 45. Summary of the general findings.....	97
Figure 46. Summary of RQ findings.....	98
Figure 47. Some Influencing factors found in this study that affect the experience of integrating Lean UX into agile.	100

List of Tables

Table 1. A summarized list of the challenges of agile and UX integration as reported by Curcio <i>et al.</i> (2019) & Da Silva <i>et al.</i> (2018)	19
Table 2. The lessons learned from our previous studies on gamification in SE education, agile, and SPI. 34	
Table 3. The gamification dynamics, mechanics, and components that compose GLUX's gamification strategy.....	45
Table 4. The customization actions required to apply the gamification strategy of GLUX.	52
Table 5. Research questions of the validation study.	60
Table 6. The structure of QNR2.....	68
Table 7. The list of "seed codes" and "emerged codes" used to guide the data analysis process.	73

Chapter 1

Introduction

1 Introduction

1.1 Motivation and Problem Context

Agile development processes, particularly Scrum, have become the standard for software development. Even though the theory of agile promises that usable systems with great User eXperience (UX) are produced when developed using agile methodologies (Baxter and Sommerville, 2011), agile development in theory may not be the same as in practice. Practice and theory may change, synergize, or even diverge over time, but they indeed influence each other. Larusdottir, Gulliksen and Cajander (2017) respond to such argument by presenting solid research to investigate to what degree usability and UX are addressed in agile development processes. Their conclusion highlights the immense importance of usability and great UX to the overall quality of software. However, they found that agile development methods, despite their popularity, do not guarantee to deliver such quality. Accordingly, the authors go through some challenges that could emerge from integrating agile and UX processes and suggest general guidelines for projects following agile methods, particularly Scrum, with a focus on UX.

During the agile manifesto 10th anniversary in 2011, Martin Fowler predicted that the integration of operations and the integration of UX would be the next frontier for agile. Due to the need for delivering highly usable software products for economic success and sustainable business in a fast-changing world, the integration of agile and UX has received significant attention, leading research in software development processes over the last 15 years and viewed as a promising yet undeveloped field (Da Silva *et al.*, 2018). However, there is an inherent tension between those two areas as integrating both approaches present several challenges. Both agile and UX put the user first and value a great user experience. But each approaches development from a different perspective with different work rhythms. While agile focuses on software development and project management intending to deliver small sets of valuable features in short iterations, UX focuses on the user's interaction, needs, and goals and call for considerable effort on user research and analysis before development. Consequently, endeavors to combine the two processes continue based on the premise that agile and UX share common principles such as iterative development, emphasis on the user, and team coherence. Research efforts mainly aim to incorporate UX practices while preserving the agility of the development process.

One of the first efforts to combine agile and UX was the work of Sy (2007). In her article, Sy describes a “staggered sprints” process for integrating agile and UX in which design activities take place one sprint ahead of development activities. Even though Sy advocated strong collaboration between design and development teams during both the design and development sprints, the model was misinterpreted as many teams created workflows in which developers and designers communicate by handoffs and documentation. Over the years, as illustrated by a number of systematic literature reviews, the field of agile has witnessed the rise of new strategies and techniques aiming at integrating UX principles and practices into agile (Salah, Paige and Cairns, 2014; Brhel *et al.*, 2015a; Curcio *et al.*, 2019).

While research in the field of agile and UX has reached such maturity that its importance is already established and arguments around this aspect are no longer necessary, there are yet challenges to be addressed and issues to be resolved. The open issues of the field range from practical challenges such as lack of technical tools to support the integration of UX practices into agile process, to more fundamental challenges such as the lack of shared UX vision and principles among members of the agile team (Curcio *et al.*, 2019). According to Da Silva *et al.* (2018), aspects related to integrating processes, practices, and

people have reached adequate maturity, while technological means in terms of supportive tools and artefacts still have a long way to go. Particularly, there is still a need to design and develop assistive artefacts that can support the integration activities and help teams to build a shared understanding. In this respect, we saw an opportunity to contribute to improving the process of integrating UX practices in agile by addressing one of the reported challenges in the literature.

The key problem which this thesis aims to investigate and address is related to the developers' mindset towards UX activities, in terms of the continuous prioritization of delivering fast and "working" features, without managing UX aspects with equal attention. As discussed later in Chapter 3, the problem is associated with what is commonly known as "UX debt" (Kaley, 2018). Like technical debt, UX debt occurs when increments are released on a rapid pace such that UX issues are not treated with equal significance as functional issues, resulting in teams building up UX deficiencies and forced to carry out additional rework. This situation represents a recurrent challenge when integrating UX with agile (Curcio et al., 2019). By analyzing and tracing the root causes and consequences of such a challenge and assessing its magnitude, we noticed that research on human factors in general, and on developers' mindset towards UX activities in particular, is yet scarce.

According to our analysis in Chapter 3, the tendency to prioritize functional quality over UX quality mostly emerge from four key issues. First, the term "working software" in the Agile Manifesto has been misinterpreted over time such that agile teams began to shift their focus to delivering functioning software on a frequent and rapid basis more than focusing on delivering valuable features to the end user. Second, due to the lack of knowledge and training on user experience design (UXD) and its value to the quality and success of the developing product, software engineers' motivation and interest towards UX work is perceived as minimal. Some even perceive UX work as a costly burden that can jeopardize the agility of the development process. Third, it is argued that the general resistance to change is linked to the resistance to integrate UX practices into agile. Getting developers to accept a new development process with an increased emphasis on UX practices can be challenging. The fourth and last issue is related to a phenomenon called "the curse of knowledge" in which agile teams may come to believe that they know better than the user in terms of what features to build and how it should be designed and delivered. Because of this, there has been a low acceptance of the results of user tests, and accordingly, users are involved less than necessary.

The problem with launching a suboptimal software product in terms of UX is that most users will give the product one try and then easily give up if the UX is substandard or simply does not satisfy their needs (Kaley, 2018). When UX issues are set to be fixed later, users may not even give the product another try if they had a bad experience in the first time. The value of empowering agile teams with UXD skills such that UX is considered everyone's responsibility is clearly established. Getting developers involved in the UXD process is even recommended by the Nielson Norman Group as one of the tactics to identify and resolve UX debt (Kaley, 2018). Yet, efforts in this respect (i.e. how to get software engineers to be involved in the UX process) remain hardly explored. It is argued that such a challenge is considered complex and difficult to manage because it involves human factors such as mindset and cultural change (Da Silva et al., 2018). In other words, reinforcing software engineers with UX skills requires a change in mindset and behavior towards UX. For this change to happen, it requires some unlearning of habits as much as learning new practices (Liikkanen et al., 2014).

In this context, gamification has received significant attention and popularity over the past few years as a powerful tool to encourage new behaviors and skills (Koivisto and Hamari, 2019). While many contemporary technologies such as artificial intelligence and robotics primarily seek to improve the well-being and advancement of humanity through enhancing the capabilities of machines, gamification seeks to improve the well-being and advancement of humanity through enhancing the capabilities of humans by becoming more motivated, engaged, and skilled. Gamification employs the philosophy and techniques of game design into non-game contexts to induce a target behavior in people and improve their motivation and engagement (Deterding et al., 2011). A well-designed gamification system gets people to do things they did not use to do and engaged in habit changing tasks.

Gamification has been employed in several sectors such as in education to improve the motivation and performance of students (Dicheva et al., 2015; Alhammad and Moreno, 2018a), in marketing to increase customer loyalty (Hofacker et al., 2016), and in healthcare to promote healthy habits such as exercise and hygiene (Pereira et al., 2014), to name a few. Gartner claims that digital businesses are the driving force of gamification's revival, and it predicts that by 2020, gamification combined with emerging technologies, will have a significant impact on the emergence of customer engagement platforms. So far, gamification has been the main component in the design of major applications such as Google's Local Guides, Todoist, and StackOverflow. In software engineering, gamification has been employed as an innovative strategy to enhance developers' interest and motivation in carrying out software engineering best practices (Pedreira et al., 2015; Khandelwal, Sripada and Reddy, 2017). Research in the field of gamification in SE argues that human factors are the main asset in software development in which the motivation and engagement of software engineers are key to the success of the software project (Koivisto and Hamari, 2019). Therefore, introducing gamification in the daily tasks of the development process can help software developers to become more engaged in their work which in turn reflects positively on the outcomes of software projects in terms of software quality and project performance (García et al., 2017). However, although gamification in SE seems promising, it is still at a very early stage with little evidence about its effect (Pedreira et al., 2015).

1.2 Research Goal

This doctoral research primarily seeks to design a framework that can contribute to addressing the challenge of the developers' mindset towards UX by employing gamification as a motivational technique. Ultimately, our goal is to improve the integration process of agile and UX.

Seeing that some challenges of the integration of agile and UX can be traced to the mindset of agile teams towards UX practices (which are arguably context-dependent), we approached the problem from the developers' perspective. To this end, we mainly aim to **design a framework that employs gamification as a technique to encourage and promote the integration of lightweight UX practices into the development process and facilitate seamless integration between agile and UX**. A distinctive contribution of the proposed framework is that it caters to the different characteristics and complexity of motivational factors of software engineers and agile teams.

In this research, we exploit the potential of gamification in promoting a user-centered mindset amongst the agile team. This is motivated by 1) the challenges associated with the developers' prioritization of functional features over UX issues, and 2) the need for more data on the effectiveness of gamification in software engineering. The existing evidence on the usefulness of gamification in software engineering supports the benefits of gamification in promoting SE best practices, but such results are not conclusive.

Therefore, this research: 1) seeks further evidence on the benefits and drawbacks of gamification in software engineering; and 2) contributes to the knowledge base of agile and UX integration, by investigating and addressing the challenge of the developers' mindset.

1.3 Research Questions

This thesis highlights the significance of agile teams' mindset towards integrating UX practices into the development process on the overall success of the integration process. Our central goal, as discussed in the previous section is to motivate agile teams towards carrying out UX activities throughout the development process. Therefore, the research of this thesis is centered around two research questions:

RQ1: How can gamification be employed to design an artefact that can improve developers' mindset towards UX so that agile teams adopt a more user-centered development process?

The aim of RQ1 is to outline the set of requirements and specifications essential to design an artefact that employs gamification to address the key problem (i.e. developers' mindset towards UX). Ultimately, the goal of RQ1 is to build an artefact that can improve the integration of agile and UX by encouraging agile teams to develop a motivated mindset towards UX.

This RQ involves understanding the problem context in terms of identifying the potential stakeholders' goals and the techniques that have been used before to solve such problem or a similar problem. Therefore, in Chapters 3 and 4, we define the problem context and analyze the goals and needs of potential stakeholders, and then outline the techniques and knowledge used to guide the design of the GLUX framework. This RQ also entails dealing with decisions regarding the UX practices or processes to integrate with agile, the different gamification techniques to incorporate, the design process to follow, and the most effective way to implement and introduce such artefact to the target problem context.

RQ2: To what extent can a gamified artefact improve the developers' mindset towards integrating UX practices into agile?

Prior academic studies have suggested that gamification can be effective to promote software engineering best practices as well as influence human behavior (Pedreira *et al.*, 2015). The goal of RQ2 is to explore the effects of employing gamification in the context of agile and UX integration on the mindset and attitude of developers. Particularly, can the designed artefact be effective in cultivating the mindset of agile teams and help the developers to build a user-centered mindset? Does it offer agile teams with a seamless and enjoyable method to integrate UX practices into agile?

In RQ2, we merely seek to examine the effectiveness of our proposed framework in terms of its utility and usability in facilitating the integration process. Unlike RQ1, the aim here is not to improve or change the mindset of developers, but to only verify if any change or improvement has occurred. We show and discuss in Chapter 5, how we used Technical Action Research (TAR) with a sample client represented by three agile teams of graduate students to validate the effectiveness of the proposed framework.

1.4 Research Contributions

In this section, we present the bodies of knowledge we believe the work presented in this thesis has contributed to:

- a. The integration of agile and UX,

b. Gamification applied into software engineering.

The elements of contribution made by our work stem from different components in this thesis that directly align with our research questions.

The key contribution includes the design and validation of a framework that employs gamification to address the challenge of agile teams' prioritization of functional features over UX features, which was established in the literature of agile and UX integration. The proposed framework is, to the best of our knowledge, the first gamified artefact that addresses the lack of motivation and engagement of agile teams towards integrating UX practices in the development process and promotes the collaboration of the entire team in carrying out UX tactics. It features:

1. A customizable and incremental gamification strategy to cater for different agile teams and different projects, and
2. Five lightweight Lean UX tactics to integrate into Scrum.

This thesis also contributes a pilot study to demonstrate the validity of the proposed framework in facilitating the integration of UX practices into agile and improving the agile teams' mindset towards UX work.

In addition, this doctoral research extends the knowledge boundaries of the field of gamification in software engineering. We published three systematic mapping studies to outline the state of practice of gamification in software engineering education, agile, and software process improvement (SPI). Our goal was to explore how gamification was put into practice in the field of software engineering, and to what extent it improves or impacts the motivation and engagement of software engineers to adopt best practices.

The first systematic mapping was titled "gamification in software engineering education" and was published in the Journal of Systems and Software (Q1) in March 2018. Our goal was to explore how gamification has been applied in software engineering education because the output of this doctoral research, the GLUX framework, would be firstly validated in an educational setting with graduate students from the software engineering program at La Universidad Politécnica de Madrid.

The second systematic mapping study was conducted in the field of gamification in agile and was published in the 19th International Conference on Agile Software Development (XP2018) that was held the week of May 21 to 25th in Porto, Portugal. Our aim was to get a snapshot of the state of the practice of gamification and agile software development, to discover what evidence there is about the benefits of this integration. The study is relevant in the sense that we are gamifying an agile framework. Therefore, our goal was to explore how and why agile processes were gamified.

The third systematic mapping was carried out to investigate the challenges of gamification in Software Process Improvement (SPI). The article was published in the journal of Software Evolution and Process in January 2020. The field of gamification in SPI is relevant because we share a similar goal with researchers of that field in terms of improving the development process, except that we are particularly interested in improving agile development processes by integrating Lean UX using gamification.

1.5 Thesis Structure

The research carried out as part of this thesis is structured as follows:

Chapter 2 describes the research methodology that guided the research of this thesis. In this chapter, we provide a brief overview of DSR, why it fits for this doctoral research, and then explain how it was applied in the context of our research. We also describe in this chapter the additional research methods employed for data collection and analysis, the design process of the GLUX framework, as well as the validation methodology.

Chapter 3 presents the first activity carried out as part of the design cycle of the DSR paradigm, i.e. problem investigation. In this chapter, we primarily present and examine the key problem which this thesis aims to address. Prior to that, we provide an overview of the literature of agile and UX integration and a summary of the current challenges in this field.

Chapter 4 delineates the second activity carried out as part of the design cycle of the DSR paradigm, i.e. treatment design. In this chapter, we detail the design process of the GLUX framework. The chapter starts with providing an overview of the techniques used develop the framework and the knowledge used from previous work to guide the design process. Then we demonstrate the design process step by step and explain the design decisions and choices made throughout the design process. Finally, we present the result of the design process in terms of the characteristics of the proposed framework and the guidelines to effectively apply the framework in the problem context.

Chapter 5 details the third activity carried out as part of the design cycle of the DSR paradigm, i.e. treatment validation. In this chapter, we primarily demonstrate the utility and usability of the GLUX framework by presenting a validation study that was conducted following a TAR approach. We begin by defining the objectives and research questions, explaining the design of the validation study, and describing the model context in terms of the characteristics of the framework's prototype and the sample client. We then discuss the threats to the validity of the validation study and the countermeasures taken to improve it. We finally report and discuss the results of the study and conclude with discussing the key findings and the lessons learned.

Chapter 6 concludes the thesis and summarizes the doctoral research. In this chapter, we present the key findings of the research and discuss the answers to our research questions. The chapter concludes by identifying the limitations of our study and outlining future research directions that could advance the ideas presented in this work.

Chapter 2

Research Methodology

2 Research Methodology

2.1 Introduction

The research of this doctoral study was guided by the Design Science Research (DSR) paradigm by Roel J. Wieringa for information systems (IS) and software engineering (Wieringa, 2014). To the best of our knowledge, the seminal work of Prof. Wieringa in DSR is the most recent and comprehensive source of DSR in software engineering. His approach is taught every year to master’s and PhD students of computer science, software engineering, and artificial intelligence. The material of his book on Design Science Methodology in Information Systems and Software Engineering is also taught in seminars, workshops, conferences to academics and industrial researchers. Throughout the work of this thesis, we also took into consideration the recommendations of Engström *et al.* (2020) on how to frame the contributions of a software engineering research in a systematic way using the lens of design science.

In this chapter, we briefly explain what DSR is, why it fits as a methodology for this research, and then go through how it was applied in the context of this doctoral research. Since DSR is considered a high-level *research strategy* that offers guidance to the whole research, the research approach was complemented with *research methods* that guide the work on a more detailed level. Accordingly, this chapter also describes in this chapter the additional research methods employed for data collection, data analysis, and validation of the artefact. Consequently, the goal of this chapter is to lay the foundation on which the work of this thesis was built and help the reader understand the overall rationale behind the research.

2.2 A Brief Overview of Design Science Research (DSR)

The past few years have seen a surge of interest in DSR as a research paradigm in information systems (Deng and Ji, 2018). While DSR is increasingly used in computer science, it is rarely referred to in software engineering research (Engström *et al.*, 2020). This low adoption of DSR in software engineering is puzzling as both DSR and software engineering share the mission of solving real-world problems. Engström *et al.* (2020) argue that because software engineering research is solution oriented that typically produces design knowledge, using design science paradigm to communicate and disseminate this knowledge is useful to elucidate certain aspects of research contributions. In their work, the authors analyzed 38 distinguished articles from the International Conference of Software Engineering (ICSE) published over a period of five years (2014–2018 inclusive) to investigate if and how design science can be used to communicate research contributions in software engineering. Even though none of the examined papers explicitly used the term of design science, the authors found that most of those papers follow DSR implicitly and can demonstrate their research contributions using the design science paradigm.

In efforts to promote design science for capturing design knowledge in software engineering research, Wieringa (2014) provides guidelines and a rigorous framework to conduct design science research in software engineering and information systems. By definition, design science is “the design and investigation of artefacts in context” (Wieringa, 2014). The key feature of DSR that distinguishes it from other relevant research methodologies (e.g. action research) is that DSR involves the design of *an artefact in a context*. The artefact, which is the key object of study, is created to solve a design problem in which the aim is to achieve an improvement in the problem context. Thus, design science problems are also known as *improvements problems*. It is important to note here that the artefact itself does not solve any problem, but rather it is the interaction between the artefact and the problem context that can contribute to solving or improving the problem.

According to Wieringa (2014), the researcher iterates over two main activities in DSR: the design of the artefact, and the investigation of the artefact in context (Figure 1). An artefact here can be a method, technique, framework, algorithm, or model to be used in software engineering or information systems. While the context can be software, an organization, people, services, methods, etc.

Figure 1. Design science iterates over two main activities.

In his DSR approach, Wieringa (2014) distinguishes between two types of research problems that correspond to the two main activities (i.e. design and investigation), namely design problems and knowledge questions. *Design problems* are practical problems that require the analysis of potential stakeholder goals and needs. The solution to a design problem is achieved through the design of an artefact that triggers a change in the problem domain which contributes to solving or improving the problem. The artefact is evaluated by its utility with respect to the stakeholder goals.

On the other hand, *knowledge questions* do not entail a change in the real-world but rather inquire knowledge about the problem environment. They can be considered as the goals or questions of the research project. For example, what do we want to know about the problem context or what do we want to know about the artefact we designed in the problem context? It is worth noting that the answer to a knowledge question is a proposition that seeks to describe the environment and explores the truth, regardless of whether the findings are compatible with the stakeholders' goals or what is expected. The empirical knowledge or the theories we develop based on the finding is accepted as is and it cannot be proved with absolute certainty based on the principle of fallibilism in research philosophy.

In DSR, the iteration over design problems and knowledge questions can start from either activity. One sequence can start with a design problem (design activity), and then move to answer knowledge questions about the artefact in context (investigation activity), which could generate new problem and knowledge that takes us back again to the design activity. Conversely, the research project may start with a knowledge question (investigation activity) that leads to new design problem such as the need to build a prototype (design activity) in order to answer such questions.

In the case of this doctoral research, DSR fits as a research methodology because it is aligned with the research goals. The DSR methodology supports a pragmatic approach that calls for the creation and evaluation of innovative artefacts that solve practical problems. As discussed in the Chapter 1, our goal entails the design of an artefact that can contribute to addressing some of the challenges of agile and UX integration (design), as well as the investigation of the effectiveness of such an artefact in improving such challenges (investigation).

The design problem we aim to address is motivated by two aspects: 1) the challenges of integrating agile and UX identified in the literature and 2) a knowledge question that seeks to explore what techniques or

solutions can be employed to address such challenges. Consequently, this doctoral study iterates over the *design* and *investigation* activities following a sequence that starts with investigation (Figure 2). We started with *investigating* the open challenges of agile and UX integration, exploring the techniques, if any, that were employed to address such challenges, and thoroughly analyzing the one challenge that caught our attention (Chapter 3). Then, using the new knowledge and design problems that emerged from the *investigation* activity, we embarked on the *design* of a solution prototype that focuses on a particular challenge in a particular context (Chapter 4). As a result, the design of such prototype took us back to the question-answering activity (*investigation*) to examine how the artefact prototype interacts with its context in terms of its effectiveness and usability (Chapter 5). In principal, the key difference between the first investigation activity and the second investigation activity is by the research goal of each activity. In the first investigation (i.e. as refer to it later as problem investigation), the goal is to investigate a problem context *before* an artefact is designed and applied. While in the second investigation activity, the goal is to assess the situation *after* the artefact has been designed and applied in a model context. It is important to note here that while a DSR research can iterate over design and investigation for several cycles, the scope of this doctoral research is limited to a cycle of investigation-design-investigation, so far.

Figure 2. This doctoral research iterates over design and investigation, starting with investigation.

2.3 The Design and Engineering Cycles

The DSR framework is composed of two cycles: the *engineering cycle* and the *design cycle* (Figure 3). The design cycle is comprised of three activities: *problem investigation*, *treatment design*, *treatment validation*. The engineering cycle is a larger cycle containing the design cycle within, except that the engineering cycle continues further to *treatment implementation* and *treatment evaluation* where the design artefact is transferred to the real world context, applied, used, and evaluated.

Figure 3. The engineering and design cycles of the DSR framework (Wieringa, 2014).

The design and engineering cycles of the DSR approach by Wieringa (2014) uses terminology that are overloaded. Treatment here refers to the interaction between the artefact and the problem context. Albeit commonly used among engineers, the term “solution” is avoided because an artefact may solve a problem partially, or not at all. An artefact is something created to address a practical problem. Finally, unlike how the term “implementation” is commonly used in software engineering as the creation of a running software program, in DSR it refers to the application of the treatment to the problem context.

Typically, a DSR researcher iterates through the design cycle for few iterations before transferring the artefact into the real-world problem context. Implementation into the market may be done after the research is finished but not as part of the research project (Wieringa, 2014). In the design cycle, particularly during the problem investigation activity, the researcher starts by examining the problem context including the significance of the problem, and the underlying causes and consequences. The goal of this activity is to explicate and formulate the problem precisely. In treatment design, the researcher develops an artefact that addresses the identified problem. This activity also requires outlining the desired properties or requirements of the proposed artefact and check it against the stakeholders’ goals and needs. Before transferring the developed artefact into the real-world context, it must be validated. In treatment validation, a prototype of the artefact is investigated with a model of the problem context. The aim of validation is to produce a design theory that structures the knowledge about the artefact and its interaction with the problem context. The design theory describes how the artefact improves the model problem context, and allows to predict if, and how, it would improve the original problem context.

Moving further through the engineering cycle, treatment implementation entails applying the artefact in the original problem context, i.e. the real-world. In this thesis, this step is out of the research scope as we are not investigating a real-world implementation of the framework but rather investigating a prototype of the artefact with a model of the real-world context. Implementation evaluation involves the assessment of how successful the treatment is in improving the problem, typically independently from researchers.

The main three activities of DSR carried out in this thesis follows one cycle through the design cycle which are described below in a logical order:

Problem investigation (Chapter 3): The first activity in this DSR-led research involved identifying, describing, and explaining the problem that must be solved or improved as well as defining the research motivation and goals. This step included asking questions such as: what are the challenges of agile and UX integration? How are they caused or why do they occur? What are the consequences of unresolved challenges? And what has been done so far to address those challenges?

Following the approach of Wieringa (2014) for problem investigation, we explored the scientific, professional, and technical literature as a common technique that can provide sufficient knowledge to understand the problem context. Consequently, to answer those questions, and assess the relevance and significance of the problem, the literature of agile and UX was analyzed, with a focus on the challenges of the integration. Luckily, the literature of agile and UX is massive and abundant with research studies that investigate the different forms of integrating agile and UX, as well as analyzes the challenges of the different approaches. Accordingly, the problem investigation process relied heavily and primarily on the work of Da Silva *et al.* (2018) and Curcio *et al.* (2019), which are the most recent and comprehensive peer-reviewed articles in the field. In Da Silva *et al.* (2018) the authors report the evolution and current state of agile UX, providing a brief and general overview of the field and highlighting the unaddressed gaps and open questions. On the other hand, Curcio *et al.* (2019) developed a more detailed and comprehensive tertiary study that critically analyzes the literature and identify the different strategies and forms of integrating usability and agile software development as well as the challenges of this integration.

As a result of analyzing both scholarly articles, and the subsequent primary papers, we basically found and confirmed that the research on human factors in the integration of agile and UX, such as teams and culture, is still scarce. Many challenges regarding this particular aspect remain unresolved. This had led us to narrow down and explicitly identify a key problem in the integration of agile and UX which this thesis aims to address.

The principal problem addressed by this research is the developers' mindset towards UX in the context of agile and UX integration. By learning and discovering the developers' mindset situation and its effects on the success of the integration of agile and UX, our goal was to prepare the design of an artefact (i.e. the GLUX framework) that can contribute to improving the issue. This activity also entailed analyzing the relevance and significance of the problem.

Treatment design (Chapter 4): In this activity, the design of the artefact that could address the problem takes place. In this work, it included defining the key techniques to be employed in the design of the artefact, as well as describing the design process the researcher followed to produce the artefact. We show in Chapter 4 how some design decisions were made and what are the characteristics of the developed artefact (i.e. the GLUX framework). We also justify how the characteristics of the artefact contribute to the goals and needs of the stakeholders.

The design process followed to build the GLUX framework is the 6D gamification framework (Werbach and Hunter, 2015b). The framework is built on top of the Scrum framework and employs two main techniques, namely, gamification and Lean UX. The designed framework is intended to be used by Scrum teams who wish to incorporate UX activities in the development process to achieve improved usability and UX of the product. Therefore, this activity did not only require specifying how the framework should be designed, but also included specifying how the framework relates with the stakeholders (i.e. scrum teams) and the problem (i.e. developers' mindset). Moreover, this activity involved analyzing the literature of

gamification in SE, SE education, agile, and SPI to discover how gamification was put into practice in software engineering, from gamification approaches and methodologies to gamification elements and tools and lessons learned.

Treatment validation (Chapter 5): This activity included validating if the designed artefact would improve the identified problem. The term validation in the context of DSR might differ than what is commonly known. In DSR, validation is usually done before implementation (i.e. transferring the artefact to the real-world problem context). It involves simulating the interaction between the artefact and a model of the problem context where the participants are asked to perform tasks in a simulated environment. The goal is to develop a design theory that structures the knowledge about the artefact from the interaction of a prototype of the artefact and a model of the problem context.

To validate the artefact of this thesis, we followed the Technical Action Research (TAR) approach by Wieringa (2014). TAR is a case-based research method used for validation in DSR. It is a special case of Action Research (AR), except that TAR is artefact driven that aims to examine an artefact in its context as well as to help a potential client. On the other hand, AR is problem driven, in which the researcher works with the client to solve a problem without the aim of testing a particular artefact. To this end, a model of the artefact was created and validated with graduate software engineering students who used the GLUX framework in a team agile project.

Applying TAR in DSR required the researcher to play three roles:

- Technical researcher: the researcher designs an artefact to address a class of problems. In our case, we designed a gamified framework that aims to address the problem of agile teams' focus on functionality over usability.
- Empirical researcher: the researcher answers validation knowledge questions about a prototype of the artefact in a model context by. In our case, we defined and answered knowledge questions about the interaction of a prototype the GLUX framework with agile teams of graduate students.
- Helper: the researcher creates and applies a client-specific version of the artefact to help a client with a problem. In our case, we customized the GLUX framework to the graduate students' situation to use it in their agile projects and help them apply UX practices and build a motivated mindset towards UX.

The design of our TAR study involved decisions about client selection, treatment design, measurement design (data collection methods), inference design (data analysis methods) and threats to validity, research execution, and finally reporting. The selected client is represented by 17 graduate students enrolled in the course of "Agile Methods: Agile Techniques and Agile Usability" as part of one-year master's program in Software Engineering at Universidad Politécnica de Madrid (UPM). A model of the GLUX framework was created which is depicted in the form of a step-by-step gamified guide to integrate Lean UX tactics into Scrum. Our data collection occurred through student surveys, unstructured interviews, and the analysis of final project reports. The collected data was analyzed following a qualitative approach that is based on Grounded Theory (GT), using *Nvivo 12* as a supporting tool.

Observational case study was ruled out as a validation methodology for this thesis because in observational case study the researcher does not intervene in the evaluation process. In our case, there was an intervention as the validation was dependent on the researchers. As researchers, we intervened by creating the GLUX guide, which resembles the customized version of the framework to the students' case, as well as asking

them and showing them how to use it. Similarly, single-case experiment was excluded as a possible validation methodology because one of our goals was not to merely apply the framework to see what happens, but to also help the subjects. The subjects here are a model of the stakeholders and can benefit from the solution. In our case, adopting the GLUX framework provides the students with a technique to learn and apply Lean UX principles and tactics in their software development projects. Finally, the choice of TAR over Action Research (AR) was based on the need to test the artefact we are building, as in AR the researchers' main goal is to examine whether or not the problem is solved or improved. The difference between TAR and other forms of action research is that TAR is artefact driven. All other forms of action research are problem driven, because they work with the client to solve a problem without the goal of testing a particular artefact.

Chapter 3

Problem Investigation

3 Problem Investigation

3.1 Introduction

As discussed in the previous chapter (Chapter 2), this doctoral research iterates over *design* and *investigation* following a sequence that starts with investigation. Therefore, the first activity in our DSR framework, as shown in Figure 4, is problem investigation. This activity involves identifying, describing, and explaining the problem to be treated before an artefact is designed. The goal is to explicate and define the problem context including the significance of the problem, and the underlying causes and consequences in order to improve the problem situation.

This chapter primarily presents and discusses the descriptive knowledge from the literature of agile and UX integration and highlights the characteristics and the environment of the key problem this thesis aims to address, i.e. the developers' mindset towards UX in the integration of agile and UX. We also attempt to include explanatory knowledge about the causes and consequences of the problem. In a nutshell, we aim to understand what is the developers' mindset problem? Why it is significant? And why it occurs?

Figure 4. Problem investigation is the first activity in the design cycle of this doctoral research.

According to (Wieringa, 2014), problem investigation can be done in many ways of which exploring the scientific, professional, and technical literature is a common technique that can provide sufficient knowledge to understand the problem. Consequently, we first explore and analyze the current state of the research in agile and UX integration in section 3.2 to provide a brief overview of the field as grounds to our problem investigation. In section 3.3, we aim to summarize and map the major challenges of integrating agile and UX as identified in the literature. Finally, in section 3.4 we explain and analyze the key challenge which this research aims to address that is related to the focus of agile teams on functionality over UX features and issues.

3.2 Recent Developments in Agile and UX Integration

The literature of agile and UX integration (i.e. agile UX) is massive (Curcio *et al.*, 2019). Many researchers in the field of agile and UX integration have carried out studies analyzing and understanding the different ways and conditions under which agile and UX can join forces (Brhel *et al.*, 2015a; Da Silva, Silveira and Maurer, 2015; Magües, Castro and Acuña, 2016; Da Silva *et al.*, 2018). For instance, Da Silva *et al.* (2018) presented a comprehensive and high-level overview of the evolution of agile UX over the past two decades from analyzing peer-reviewed literature of the field. According to Da Silva *et al.* (2018), research in this field has three main themes that represent the type of UX integration into agile processes: process and practice integration, people and social integration, and technology and artefact integration. Process and practice integration refers to merging UX activities and process with agile processes in a unified process incorporating both perspectives. People and social integration refers to making changes to the team structure to bring experts from the two different disciplines together as well as the social interaction among them and knowledge sharing. Technology and artifact integration refers to the use of technological tools to support the integration activities as well as the utilization of artifacts from both processes to facilitate communication and create a shared understanding. The work of Da Silva *et al.* (2018) shows that research in the integration of agile and UX can be segmented across three periods with a five-year interval that mark the maturity of the field: early years (2001-2005), middle years (2006-2010), and recent years (2011-2016) (Da Silva *et al.*, 2018). The early period research provided speculative studies highlighting the importance and value of UX in agile. In the middle years, agile UX started to establish its identity and research originality in relation to other fields. Finally, during the recent years, research has become more mature forming a larger community discussing challenges, limitations, and improvements. In general, the findings mainly suggest that there are evident tensions between the two approaches and they should not be managed and controlled as two separate processes (Da Silva *et al.*, 2018).

While the integration of agile and UX is challenging, it is considered a leap when compared to running each process individually. Given the maturity established in the integration of agile and UX, discussions about the significance of integrating UX into agile processes are no longer necessary. However, there are a handful of unresolved issues and unaddressed gaps, as discussed in the following section. In principal, research in the field should be directed towards managing the integration as one process and understanding users, not only by representing them, but involving them in the development process on a regular basis (Da Silva *et al.*, 2018). Da Silva *et al.* (2018) predict that just as agile is leading software development today, agile UX will be core for software development in the near future.

More recently, Curcio *et al.* (2019) developed a tertiary study and a critical analysis to identify the different strategies and forms of integrating usability and agile software development as well as the challenges of this integration. The article analyzed 14 secondary studies including systematic literature reviews, systematic mapping studies, and non-systematic literature reviews, published between 2001 and 2018. Although the authors made it clear that they distinguish the term usability from UX, their search strategy covered all publications related to the field of usability and UX, such as user-centered design (UCD) and user interaction. The distinction between usability and UX is a whole other argument that this thesis does not cover.

In their tertiary study, Curcio *et al.* (2019) outlined four research questions to guide their study which attempts to identify secondary studies' research questions, explore the main ways to integrate usability in agile, assess the quality of the selected papers, and finally list the challenges published in the literature

related to the integration of usability and agile. The authors confirm that there is an increasing interest in researching the integration of agile and UX over the past few years, which is evident by the numbers of publications proposing new integration approaches. According to the analyzed secondary studies, Curcio *et al.* (2019) categorized the main forms of integrating UX in agile into six categories which are formed from a slightly different perspective compared to the themes presented by Da Silva *et al.* (2018). The six categories are: processes, techniques, practices, recommendations, principles, and different approaches.

Like Da Silva *et al.* (2018), Curcio *et al.* (2019) confirm that the trend is managing and controlling the integration of agile and UX as one process. Nonetheless, this new way of integration presents its own new issues and challenges such as organizational culture and difficulties with the team itself, as discussed in the following section. The identified points of tension do not mean that the integration is unfeasible. Those issues essentially point out to particular aspects in which further research is needed to improve the integration process. Curcio *et al.* (2019) recommend that future research should be conducted to investigate further evidence about the challenges and tensions of the integration process and developing artefacts to address such challenges and facilitate communication and decision making.

3.3 Challenges of Agile and UX Integration

As new approaches and attempts to integrate agile and UX emerge, so do different types of challenges. To the best of our knowledge, the work of Curcio *et al.* (2019) and Da Silva *et al.* (2018) present the most recent and extensive analysis that outlines and maps the challenges of the integration. Although both articles draw related challenges, each study present it from a different perspective as summarized in Table 1.

Table 1. A summarized list of the challenges of agile and UX integration as reported by Curcio *et al.* (2019) & Da Silva *et al.* (2018)

Theme	Challenges as reported by Curcio <i>et al.</i> (2019)	Challenges as reported by Da Silva <i>et al.</i> (2018)
Test	<ul style="list-style-type: none"> • More emphasis on functional tests than on usability tests. • Scheduling usability testing. • Accessing users for usability testing. • Methods of usability testing • Shorter time to iterate design. • Lack of time for usability test. 	<ul style="list-style-type: none"> • The need for new framework proposals to synchronize usability evaluations (UXD perspective) with unit testing or acceptance testing (Agile perspective).
Time	<ul style="list-style-type: none"> • Lack of time for upfront activities. • Short time of each iteration to exploit several design options. 	
Work Balance	<ul style="list-style-type: none"> • Optimizing the work dynamics between developers and UCD practitioners. • Power struggle between UX designers and developers (UX overloaded). • Participation of both roles (developers and designers) in the process of creating design solutions. 	<ul style="list-style-type: none"> • The daily operation of an Agile UXD process. • Tools are needed to support developers in acquiring and sharing UXD and software engineering best practices.
Modularization	<ul style="list-style-type: none"> • Difficulties of design modularization. • Usage of low fidelity prototyping for the key features of entire systems prior to development. 	

	<ul style="list-style-type: none"> • Different views of the application design (up-front vs. avoid an extensive up-front design phase). 	
Feedback	<ul style="list-style-type: none"> • To get feedback from customers and users to evaluate the interface and the interaction flow prior to development. • To evaluate the systems with real users. • Different focus on customer versus end-user. 	
Prioritization	<ul style="list-style-type: none"> • Difficulty of prioritizing UCD activities. • Different focus on software delivery (functional vs usable). • Lack of a UX vision of the whole product. 	<ul style="list-style-type: none"> • Adopting UXD as an organizational culture and mindset.
Documentation	<ul style="list-style-type: none"> • Lack of documentation. 	<ul style="list-style-type: none"> • The need to share the results of UX work with the whole team on an ongoing basis, allowing the team to build a shared understanding. • The need for digital artifact-mediated communication for distributed teams.

The challenges identified by Curcio *et al.* (2019) are relatively more exhaustive and systematic. They present issues reported from different experiences in the literature and were categorized into seven main themes: test, time, work balance, modularization, feedback, prioritization, and documentation, as illustrated in Figure 5. The theme “test” covers issues related to usability testing such as scheduling, accessing users, and methods. In this context, both Curcio *et al.* (2019) and Da Silva *et al.* (2018) highlight on the need to synchronize usability evaluation with unit testing and/or acceptance testing. Curcio *et al.* (2019) argues that usability testing is usually neglected due to the emphasis of unit and acceptance testing having a higher priority to include in the development iteration, particularly in short sprints.

Figure 5. A graphical representation of the seven themes of agile and UX integration challenges based on the work of Curcio et al. (2019).

The theme “time” covers issues related to the lack of time to perform usability activities due to short iterations. Agile teams struggle with the lack of time to perform up-front design activities due to the different perspectives of design (Salah, Paige and Cairns, 2014). 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. This theme also includes issues such as the difficulty to experiment with and evaluate several design options in each iteration prior to development (Bertholdo, Kon and Gerosa, 2016).

The theme “work balance” covers issues related to the work dynamics between developers and UX designers such as the participation of both roles in the design process (Curcio *et al.*, 2019). For instance, the daily operation of the integration process between designers and developers presents several pain points. Measuring success and progress is a major challenge as user satisfaction is cited 7th by developers behind velocity and iteration burndown (Da Silva *et al.*, 2018).

The theme “Modularization” covers issues related to difficulties to break design into chunks or cycle-sized pieces adding design elements incrementally to the overall product design. While breaking work into increments in agile feels natural, breaking design into cycle sized pieces can be critically challenging. Major challenging aspects under this theme include the difficulty to determine the appropriate design chunk size per each iteration, and to plan and maintain the dependency between design chunks (Salah, Paige and Cairns, 2014).

The theme “feedback” covers issues related to maintaining user involvement to evaluate interfaces and interactions. Key issues include the difficulty to evaluate the product with real users and the different focus of customer versus end-user (Sohaib and Khan, 2010). Evaluating the product with real users is an essential

practice in usability engineering and UCD, but with the less time in agile iterations, running usability evaluation becomes infeasible. The other issue is that the customer does not necessarily represent the end-user. Thus, involving only the customer during the evaluation process would not ensure that the system is usable (Bertholdo, Kon and Gerosa, 2016).

The theme “prioritization” covers issues related to the different focus of software delivery in agile vs. UX. While agile teams focus on delivering functional software more frequently, UX teams commit to delivering usable software. Building a shared understanding and goals between developers and designers is still a concern. Challenges in this category are considered extremely complicated because a) it is context dependent, and b) it involves mindset and cultural change (Da Silva *et al.*, 2018). While there are different teams in different contexts experimenting with different techniques to create a shared understanding, this aspect remains hardly explored (Curcio *et al.*, 2019). The challenge is to promote adopting UX as a team culture instead of a role to help both developers and designers to pivot their processes and build a shared understanding among the whole team. Additionally, Da Silva *et al.* (2018) argue that a smooth and successful workflow in integrating agile with UX can be achieved if teams adopted the UX mindset. Hence, the authors predict that User eXperience Driven Development (UXDD) may start to gain more popularity and become more widely used in the next few years. The main principle of UXDD is very similar to Test-Driven Development (TDD), except that in UXDD the team is building features to satisfy the users’ not to pass the functional tests. In a nutshell, before the team get into coding, customers would have to sign off on wireframes and storyboards for each task at the beginning of each iteration. So instead of leaving the UX evaluation until the end of the iteration, if the team has time by then, user testing is moved forefront of the development process. It is worth noting here that the key problem which this thesis aims to address, i.e. the developers’ mindset towards UX, which is explained and discussed in detail in the following section, emerged from this class of challenges

The last theme of agile and UX integration challenges according to Curcio *et al.* (2019) is “documentation”, which covers issues related to the lack of documentation. Minimal documentation in agile methods have been misinterpreted in some cases. Poor documentation of decisions and intermediate findings can lead to a deficient understanding and unsound alignment among the entire agile team (Salah, Paige and Cairns, 2014). In cases when the UX and agile are integrated as a single process, UX work tends to be distributed throughout the development cycle. This implies continuous research, design, and evaluation which must be shared with the entire team on an ongoing basis to build a shared understanding. Da Silva *et al.* (2018) points out that while there are a few of recent suggestions and proposals in the literature to deal with such issue, there are still gaps and limitations. The authors attribute those gaps and limitations to the context-dependent nature of agile and UX integration, where every agile team tailor a specific solution for their particular situation.

The challenges of integrating UX in agile can vary in nature and range from practical challenges such as lack of tools to support UX integration in agile (Da Silva *et al.*, 2018), to more fundamental challenges such as lack of shared understanding of UX vision and principles (Kashfi, Feldt and Nilsson, 2019). The analysis of Curcio *et al.* (2019) shows that although the publications in the field of agile integration is abundant and covers a wide range of aspects, methods, practices, etc., research on human factors (teams) is still scarce. Some research questions remain unresolved regarding the difference of the integration process in different business scenarios, different teams, and different types of software.

3.4 The Challenge of Prioritizing UX Activities in the Agile Development Cycle

The investigation of the challenges of agile and UX integration identified in the literature have led us to identify the key problem which this thesis aims to address: the developers' mindset towards UX in the integration of agile and UX. **The challenge emerges when agile teams continually prioritize delivering fast and “working” features, without managing UX aspects with equal attention. Over time, UX issues start to pile up resulting in “UX debt”. This leaves the team with compounding UX issues that may result in costly clean ups and can adversely impact the users.**

The work of Curcio *et al.* (2019) categorizes the problem of developers' focus on functionality rather than usability as a single challenge under the theme of “prioritization”. However, when extensively and deeply examining the problem, by tracing the root causes and associated consequences of the issue in the original resources where it has been primarily reported, we noticed that it is indeed a more complex and serious issue that may not represent a single challenge but rather embodies a class of issues. Below we extensively discuss those issue.

Among four secondary studies analyzed in Curcio *et al.* (2019) which discuss the challenges of integrating agile and UX, three papers listed few challenges related to software engineers' mindset towards UX work. Overall, those papers mostly agree that agile teams have a different focus regarding software delivery in comparison to UX folks. Agile teams tend to focus on developing functional features, on a frequent basis, rather than focusing on usability, UX tasks, or user research. While agile promotes delivering working software on a frequent basis, it also encourages continuous attention to good design in the same degree as to technical excellence. The root causes and consequences of such phenomena are not thoroughly investigated in the literature, but few researchers have somewhat mentioned and highlighted relevant issues. To begin with, Sohaib and Khan (2010) argue that the term “working software” in the agile manifesto has been misinterpreted by developers. It is true that for software to be usable, it must be working. However, the opposite does not hold true; working software does not equal usable software. Due to this misinterpretation, software developers aim to fit as much as functional features within an iteration and to deliver as quickly as possible. Because “working software” suggests code quality, this in part has led to a decrease in developers' motivation towards usability issues. The misinterpretation of “working software” emphasizes how agile has evolved over the years.

Developers' focus on working software have also caused them to neglect usability testing and favor unit testing and acceptance testing (Sohaib and Khan, 2010). Salah, Paige and Cairns (2014) point out the same challenge of developers focus on accomplishing functionality features. The authors affirm that this issue can lead to other issues such as difficulty in including and prioritizing UX activities into the project iterations. For a good integration between agile and UX, it is critical that UX activities are viewed with equal importance and value to the overall quality of the product and business. Bertholdo, Kon and Gerosa (2016) assert on the significance of developers' involvement in the UX design process. When only the UX team is involved with creating design solutions, having a shared understanding among the entire team to focus on building what is valuable for the user and the business becomes impracticable. The collaboration of the entire team where different views and ideas are discussed stimulates innovation and alleviate tensions between agile and UX. Darin *et al.* (2019) discuss a similar challenge with respect to integrating UCD in the development process. The authors report that their company's development team used to be more focused on code implementation and technical challenges, without genuine concerns of the end users' needs. Therefore, design choices were often not satisfactory with respect to the users. Based on the authors'

experience, they claim that engaging the development team in the UCD process would help make the integration a more natural process. However, they emphasize on the importance to keep the team motivated and committed. Otherwise, integration strategies may fail.

More researchers reiterate over the observation among agile teams in which developers focus on functionality and deliverables, and neglect usability issues (Bak *et al.*, 2008; Lizano *et al.*, 2013; Bornoe and Stage, 2014; Larusdottir, Gulliksen and Cajander, 2017). A key issue was found to be, as firstly labeled by (Bak *et al.*, 2008), the “developers’ mindset”. Developer’s mindset generally reflects the collection of thoughts and beliefs a developer has about a problem that shapes his attitude and habits towards solving the problem. In the context of agile and UX integration, developers’ mindset refers to the way developers think about and act towards UX work. In most cases, their mindset favors functionality and code quality over usability and UX of software. Some argue that such issue essentially emerged because developers do not think of UX work as equally important as development work (i.e. programming or coding) (Bak *et al.*, 2008). For example, a developer might feel that fixing a code bug is more important than fixing a usability issue. Even if they believe that UX work is as important as development work, they think of UX as not their responsibility and is a burden and an overhead (Feng, 2014) and are more interested in making the code more efficient (Bak *et al.*, 2008). Developers’ mindset towards UX can also be influenced by the lack of knowledge and competencies developers have about UX or usability (Bornoe and Stage, 2014). This may lead to shift most of their focus and attention to sprint completion, which in turn result in developers losing track of the holistic view of the software, and thus neglecting the usability perspective. Lack of knowledge about UX techniques may not just result in developers lack of interest, but also developers lack of confidence in performing UX activities (Øvad *et al.*, 2015). Lizano *et al.* (2013) describe a similar phenomenon as “lack of usability culture” which reflects the software engineers’ understanding and opinion about usability and their participation in UX activities. They suggest more emphasis on usability activities and training programs to nurture the usability culture. As Da Silva *et al.* (2018) put it: “UCD must mesh with the Agile organizational culture in such a way that everyone in the team will understand UXD as a team discipline rather than a role in the team.”

Resistance to adopting a user-centered design process is considered one of the main causes of developers neglecting usability (Lizano *et al.*, 2013). Current studies also associate the resistance to integrate UX into agile with the “developers’ mindset” (Kashfi, Feldt and Nilsson, 2019). This obstacle is linked to the general resistance towards change, which is a common issue in product development (Gunther, Janis and Butler, 2001; Bak *et al.*, 2008) and software process improvement (Allison and Narciso, 2015). Getting developers to accept a new process of software development with an increased focus on UX can be challenging (Riemenschneider, Hardgrave and Davis, 2002). Other challenging issues related to developers’ mindset mentioned in the literature include low acceptance of usability evaluation results (e.g. “the user does not know what he wants”), and difficulty to think like the user (Ardito *et al.*, 2011). In some cases, software engineers may be subjective to the “curse of knowledge” (Twomey, 2017). By working on developing a software for a long time, developers start to consider themselves the user who knows more than the real user or assume that the user have the background to understand. This is a harmful situation where cognitive bias makes it difficult for developers to remove themselves from the user shoes and evaluate the product with actual real users.

Based on our analysis of the root causes and significance of prioritizing functionality over UX in agile, and in efforts to extend the work of Curcio *et al.* (2019), we outlined four main challenges that can extend the theme “prioritization”. Our goal in this context is to provide a more precise description and position of the

phenomenon which this thesis aims to address. The four key challenges associated with the developers' mindset towards UX illustrated in Figure 6 are as follows:

1. The tendency to focus on code quality and efficiency over UX. This is partially caused by the misinterpretation of “working software” in the agile manifesto (Sohaib and Khan, 2010). It has led developers to prioritize functional development work over UX practices, particularly in short sprints (Bak *et al.*, 2008; Salah, Paige and Cairns, 2014), and it decreased developers' motivation and interest in UX aspects (Sohaib and Khan, 2010).
2. Lack of motivation, which is a result of lack of knowledge and training in UX (Bornoe and Stage, 2014), low confidence to carry out UX tasks (Øvad *et al.*, 2015), and/or perceiving UX activities as an additional costly burden that jeopardizes the agility of the process (Feng, 2014).
3. Resistance to user-centered design methods, which is linked to the general resistance to change in development processes (Bak *et al.*, 2008), software process improvement (Allison and Narciso, 2015) and organizational culture (Da Silva *et al.*, 2018; Kashfi, Feldt and Nilsson, 2019).
4. “The curse of knowledge” which represents the issue of developers' belief that they know better than the user. It also encompasses the difficulty to think like users. This causes low acceptance of usability evaluation results and developers involving users less than necessary (Ardito *et al.*, 2011).

Figure 6. A fishbone diagram illustrating the phenomenon of the developers' mindset towards UX in the integration of agile and UX. It represents a fish's skeleton with the key problem at its head and the influencing factors of the problem feeding into the spine.

The previously-mentioned issues jointly compose the key problem which this dissertation aims to address: the developers' mindset towards integrating UX practices into agile resulting in the prioritization of functional issues over UX issues. The mindset of agile teams towards UX represents a recurrent challenge when integrating UX with agile. When agile teams favor releasing as many functional features as it can fit in a sprint such that UX aspects and issues are not given equal attention, UX issues can pile up over time resulting in costly cleanup efforts and unhappy users. Upskilling agile teams and creating a multi-discipline team that truly reflects agile team cross-functionality have never become this crucial to achieve a successful integration of agile and UX. While UXD techniques are a valuable set of skills for UX designers to have, the value pales when it is brought to a team of software engineers and project managers (Gothelf, 2019).

In today's software industry, UXD is a must have skill in any development team (Øvad *et al.*, 2015). UX work should be everyone's responsibility. This creates an environment of shared understanding and aligned goals among all team members. The value of reinforcing developers' skillset with basic UXD techniques is significant in both scenarios of handling UX work: the existence or the absence of a dedicated UX designer or UX team. In the case of a dedicated UX team, reinforcing developers with UX skills can alleviate the challenge of communication issues between UX folks and developers with involving developers in the design process which can help create a shared understanding. When an agile team does not have the necessary resources to hire usability specialists or UX consultants (at least at the early stages of a project), maintaining the basic UXD skills can be beneficial in this case. However, reinforcing developers with UX skills requires a change in the mindset and behavior of developers towards UX. For this change in mindset to happen, it requires some unlearning of habits as much as learning new practices (Liikkanen *et al.*, 2014).

In contemporary software development, the user's experience is prioritized as highly as rapid and frequent delivery (Regan, 2019). It promotes shifting software team's focus away from conventional metrics of progress, velocity, and efficiency towards increasing user experience and software quality. To that end, the artefact designed in this thesis contributes to modern software development in general and attempts to address the challenges of agile and UX integration in particular. By approaching the problem from the teams' perspective, the GLUX framework aims to improve agile teams' mindset towards UX work by employing gamification to motivate and encourage them to carry out Lean UX activities. Lean UX by its nature calls for lightweight UX tactics that can be easily incorporated into agile, particularly Scrum.

Chapter 4

Treatment Design

4 Treatment Design

4.1 Introduction

As shown in Figure 7, this chapter represents the second activity of the design cycle (i.e. treatment design) of the DSR approach by Wieringa (2014). This activity involves designing an artefact to address or improve the problem identified in the previous activity (i.e. problem investigation). This thesis design problem is expressed in the following problem statement:

Improve the integration of agile and UX

By designing a framework

That facilitates the integration of Lean UX and Scrum using gamification elements & techniques

In order to improve agile teams' mindset towards a more user-centered development approach by motivating and engaging them to learn and carry out UX activities.

Scrum, Lean UX and gamification set the scene for the design of the main artefact of this research. The Gamified Lean UX framework (GLUX) employs gamification techniques to improve the integration of agile and UX. It is designed mainly to address the developers' mindset problem discussed in Chapter 3. It aims to do that by gamifying the integration of Lean UX tactics into Scrum to motivate and engage software engineers in carrying out lightweight UX activities throughout the development process.

Figure 7. Treatment (artefact) design is the second activity in the design cycle of this doctoral research.

This chapter is composed of three sections. Section 4.2 provides an overview of the techniques used to develop the artefact and the knowledge used from previous work to guide the design process. Section 4.3 demonstrate the design process followed to develop the artefact of this thesis (i.e. the GLUX framework) and explains the design decisions and choices that were made to produce the artefact. Finally, section 4.4 describes the result of the design process in terms of the characteristics of the framework and the guidelines to effectively employ the framework in the problem context.

4.2 Background and Related Work

4.2.1 Scrum and Lean UX

Born out of the manufacturing industry, Scrum is an agile framework used primarily for software development projects in which work is framed into a series of time-boxed cycles called sprints (Schwaber and Sutherland, 2017). It is a structured framework, yet not entirely rigid, to help teams manage and adapt to changing conditions and requirements with the short release cycles and the continuous re-prioritization built into the process (Drumond, no date). Needless to say, it is the most popular agile methodology in software development given that more than 75% of agile teams use Scrum, a Scrum hybrid, or Scrum-based derivative (Scrum Alliance, 2017). Nonetheless, modern software development prioritizes the user's experience as high as the frequent and rapid delivery of software increments. In this context, UX practices and issues are inadequately addressed in Scrum, hence the various proposals and attempts to integrate UX into agile as discussed in the previous chapter.

The work of Brhel *et al.* (2015), Da Silva *et al.* (2018) and Curcio *et al.* (2019) have been of great value to identify and outline existing methods and techniques of combining UX processes and practices with agile. In this thesis, we opted for Lean UX as a UX process to gamify and integrate with Scrum after a thorough analysis of existing approaches and comprehensive search for a well-defined, systematic and lightweight process that extends the traditional UX design to embrace agile methods. Therefore, the choice of Lean UX mainly stem from the lightweight and iterative nature of the process that makes it work in alignment with agile development processes.

Lean UX defines an iterative and a collaborative process to apply UX practices. Its underlying principles are based on design thinking, lean startup, and agile. It aims to reduce waste and increase value by obtaining user feedback as early as possible. Lean UX mimics agile in the sense that work is done in an iterative and rapid cycles but requires a greater deal of collaboration. Unlike traditional UX, Lean UX does not focus on detailed deliverables but rather relies heavily on the notion of assumptions. An assumption in Lean UX is essentially a statement of the team's best guess of what is valuable to the user based on the current data. In order to test the team's assumptions, they are framed into *hypotheses*. A hypothesis is a tactical and testable statement that is written collaboratively by the entire team to generate a shared understanding around the main ideas and solutions of the developing product. The team then collaborate in the *design* of lo-fi prototypes and wireframes of the solutions and concepts brainstormed in hypotheses creation and then build a *minimum viable product (MVP)* which represents the most basic version of the concept as possible. *User research and testing* are based on the same principles of traditional UX except that in the Lean UX approach it tends to be quick and lightweight. The responsibilities of design and user testing are spread across the entire team which resolves the bottleneck created from having a single UX resource and increase the level of shared understanding.

Because integrating agile with UX continues to be a challenge for many agile teams, the authors of Lean UX, Jeff Gothelf and Josh Seiden, proposed a practical approach to integrate Lean UX tactics into Scrum (Gothelf and Seiden, 2016). Starting with brainstorming hypotheses during product backlog refinement, the team creates a list of hypotheses of their assumptions about the needs of potential users and discuss it with the product owner (PO). The team kicks off a sprint with a design sprint or a design studio in which the team picks a hypothesis as a theme to guide the work of the upcoming sprint/s and start sketching and discussing design ideas accordingly. Immediately after the design studio, the sprint planning meeting should

take place. In a typical Scrum framework, the team decides which user stories they are going to develop during the sprint planning meeting. Here in the integrated Lean UX approach, in addition to developing the sprint backlog with user stories, during sprint planning, the team plans for a weekly user experiment in which they put into test a version of their developing product, i.e. an MVP. Those plans are captured in what is called “experiment story”. The main goal of running weekly user experiments is to validate the teams’ hypotheses and collect constant feedback to improve the product. The team can use the artefacts created in the ideation session as base material for user experiments. Typically, in the early stages of a project, the team should aim to test for value using low-fidelity artefacts. Once the value is established, further experiments with higher-fidelity should aim to test the usability of the product.

4.2.2 Gamification

Gamification has received considerable attention over the past few years as a powerful strategy that can potentially influence human behavior. Shortly after it emerged from the business and marketing sector around the year 2010, gamification began to attract various domains such as education (e.g. duolingo.com), fitness (e.g. Nike app), and voluntary contributions (e.g. Stackoverflow.com). While there is no standard definition of gamification yet, the term is used to describe the incorporation of gameful elements and techniques into non-game contexts (Deterding *et al.*, 2011). The inclusion of such gameful techniques are aimed at motivating an individual or a group of people towards a targeted behavior, practice, or habit, such as a healthy lifestyle or exercise. Serious games, on the other hand, are fully-fledged games designed for purposes other than, or in addition to, entertainment. Another related concept is game-based learning (GBL) which is a broader term that describes the use of games for the particular purpose of learning. Essentially, gamification employs game techniques to drive motivation and increase engagement, as well as to create fun and joyful experiences. Examples of typical game techniques and elements adopted from traditional games include reward systems, feedback systems, and social relations and status (Pedreira *et al.*, 2020).

The literature of gamification is massive and increasing at a phenomenal pace, which is only natural in any field that has great potential and enthusiasm. Koivisto and Hamari, (2019) conducted a comprehensive review of the research on gamification analyzing 819 articles and examining empirical evidence on the usefulness and limitations of gamification. While the positive findings about the effectiveness of gamification outnumber the negative, a significant amount of mixed results is still reported. This can be attributed to the lack of coherence and consistency in gamification research, besides the inconsistency across measurement instruments and research methods. Accordingly, the authors provide 15 agendas suggesting future foci and directions of gamification research (Koivisto and Hamari, 2019). It is worth mentioning that the main two themes of empirical research in gamification are education and learning, and healthcare and exercise. Based on the authors findings, the nature of the activities associated with those two domains requires continuity and long-term engagement, and often riddled with procrastination. Therefore, gamification makes the perfect fit. While the most commonly used game elements are points and leaderboards, the authors stress on the need to direct future gamification research to explore the incorporation of collective and cooperative gamification approaches given the nature of humans as social beings. The authors also highlight a very critical observation in which the research of gamification is directing most of its focus on the impact of gamification on human behavior, neglecting the determinants of its effectiveness, or the success factors. In addition, gamification research lacks detailed analysis of the context and the characteristics of the involved individuals as these two factors should be taken with adequate consideration when designing a gamified experience to achieve more favorable outcomes. When the gamification environment and the characteristics of the gamified activities are not thoroughly reported, this

introduces the risk of producing inconclusive results and a limited view of the whole gamification experience which may not, or cannot, be applicable to any other context except the very exact setting in which it was produced.

In terms of evaluating the success of gamification experiences, Koivisto and Hamari, (2019) strongly advise researchers to not only evaluate the experience as a whole, but rather dig deeper into examining and measuring the effectiveness and usefulness of each incorporated element and analyze the contributing factors in the gamified experience. However, without adopting controlled experiments as a research methodology, investigating which, and by how much, each gamification element has contributed to the results can be significantly challenging. With the exception of pilot studies where a prototype or a concept is being validated for the first time with a small group of participants, a crucial point to consider when evaluating gamification experiences is to prolong the time span of the study long enough to reduce the “novelty effect”. The novelty of introducing gamification into a particular process or environment may introduce the risk of distorting the findings resulting in misleading or inaccurate conclusions. We highlight and discuss a similar phenomenon later in this thesis when discussing the risks and limitations of the validation study in Chapter 5, but we refer to it as “regression to the mean” based on the concept introduced by (Kahneman, 2011).

4.2.3 Gamification in Software Engineering

In the context of software engineering, gamification is seen as a promising technique to improve software engineers motivation and engagement and to promote SE best practices (García-Mireles and Morales-Trujillo, 2020). Research and practice in gamifying software engineering is motivated by the challenging nature of software development which can entail different practices and processes that are considered repetitive and time consuming, such as code review and bug hunting (Pedreira *et al.*, 2015). The growing number of publications in this area, evidenced by the recent tertiary study of García-Mireles and Morales-Trujillo (2020) shows the significance and promising potential of gamifying software engineering processes and practices. While the benefits of gamification is established to a certain extent, the empirical evidence on the impact of gamification in motivating software engineers towards a particular practice (e.g. continuous delivery) or process (e.g. agile) is still insufficient (Pedreira *et al.*, 2015; Alhammad and Moreno, 2018a, 2018b, 2020; García-Mireles and Morales-Trujillo, 2020). Nonetheless, the existing evidence shows that gamification can be effective in several areas of software engineering such as in requirements engineering to engage users in requirements elicitation, negotiation and prioritization (Cursino *et al.*, 2018), in software project management to promote collaboration and teamwork (Machuca-Villegas and Gasca-Hurtado, 2018), and in software engineering education to motivate and engage students in learning the key concepts and practices of software engineering (Alhammad and Moreno, 2018a).

To provide methodological and technical support to the research of gamification in software engineering, García *et al.* (2017) proposed a framework called “GOAL” designed to introduce gamification into software engineering environments. The framework is composed of three elements: 1) an ontology of the key concepts; 2) a methodology that provides a step-by-step process to gamify a software engineering environment; and 3) a supporting tool. While the authors highlight that early results may depend significantly on the company, they report in a case study that the ontology and methodology had helped a company to gamify their work environment. Still, the authors also found that it may not be feasible to integrate a new standalone gamification tool in a real organization. While many research papers in the field of gamifying software engineering present customized gamification tools as a proof of concept, introducing

a new process with a new tool different than what the company is accustomed to can face rejection and potentially a negative impact of the gamified experience (García *et al.*, 2017). A similar observation was also reported in a study in the field of gamification in software process improvement (SPI) when participants reported the feeling of pressure and lack of fun when introduced to a new gamified tool “Gamiware” (Herranz, Colomo-Palacios and Seco, 2015; Alhammad and Moreno, 2020). Accordingly, in efforts to improve the GOAL framework, the authors developed a new study where they extended the framework to a more abstract level to present an “architecture” instead (Pedreira *et al.*, 2020). The architecture promises to transform the work environments of software organizations into gamified ones while maintaining its tools and original work processes. To validate the architecture, an instance of the architecture was implemented, referred to as “gamification engine” and was applied to gamify the workplace of a real company. While results show potential, they are still inconclusive as the authors mentioned few limitations in the architecture and gamification engine which are work in progress (Pedreira *et al.*, 2020). Even though the work of García *et al.* (2017) provides a high-level approach to gamify software engineering processes in the form of an abstract framework (“GOAL”), in this work, we aim to provide a more concrete approach to gamify the integration of Lean UX tactics into Scrum such that the effort on agile teams to adopt the approach is reduced.

In the context of this thesis, gamification was employed in the design of GLUX as a strategy to motivate and engage software developers to carry out UX activities. The decision to do so is inspired by the promising potential of gamification in general, and in the field of software engineering in particular. As mentioned earlier, it was shown to be effective in encouraging software engineers to follow software development best practices (Pedreira *et al.*, 2015) and motivating them to adopt agile practices (Alhammad and Moreno, 2018b).

4.2.3.1 Gamification in Software Engineering Education, agile, and SPI

In previous work, we published three systematic mappings of the literature of gamification applied in SE education, agile, and SPI, referred to throughout this chapter as S1, S2, and S3 respectively. The main goal was to explore how gamification was put into practice in the field of software engineering, and to what extent it improves or impacts the motivation and engagement of software engineers. The general conclusion is that gamification in the field of software engineering is in its early stages and very few empirical studies exist, which is mostly consistent with the findings of Pedreira *et al.* (2015) and García-Mireles and Morales-Trujillo (2020). We also found that even though the definition of gamification is simple, its effective application may deal with multiple variables. Accordingly, the three studies point to several critical issues that need to be carefully considered by practitioners who aim to gamify software engineering processes. Our analysis also highlighted few trends in terms of gamification processes, elements, and tools. In Table 2, we share the identified challenges and trends in the form of 17 “lessons learned” which were taken into consideration throughout the design process of GLUX. For example, the choice of the 6D gamification framework was the result of identifying it as the most followed gamification approach, besides its popularity in other fields. Below, we provide a summary of the goals and most relevant findings of each study.

S1. In the first systematic mapping titled “gamification in software engineering education” which was published in the Journal of Systems and Software (Q1) in 29 March 2018, our main goal was to explore how gamification has been applied in software engineering education. That is because the pilot validation of the GLUX framework is to be carried out in an educational setting with graduate students from the software engineering program at La Universidad Politécnica de Madrid. The study identified 21 primary papers which is consistent with the novelty of the field. From these papers, around 77%

provide empirical data about their experience with gamification (either evaluation or validation), resulting more positive than negative or no impacting results. This gave a relatively good indication about the potentially positive impact of gamification reported in other fields. The systematic mapping identified some interesting trends that assisted in the design of the GLUX framework. For instance, we found that gamification does not necessarily have to be costly in terms of technological solutions. Positive results were found in cases where implementation was done simply through the use of Google spreadsheets or email notifications. However, the introduction of gamification may require a major effort from educators to adapt the learning materials and the course workflow.

- S2. The second systematic mapping aimed to get a snapshot of the state of the practice of gamification in agile software development, and to see what evidence there is about the benefits of this integration. The results of the study are relevant in the sense that we are gamifying an agile framework. Therefore, our goal was to explore how and why agile processes were gamified. The article was published under the title: “What is going on in agile gamification?” in the 19th International Conference on Agile Software Development (XP2018). Even though results were preliminary, they provide us with evidence for understanding how gamification can potentially improve agile processes with UX activities. In principle, we found that the impact of gamification on agile team performance is perceived to be positive. Other interesting findings indicate that user stories are the most gamified agile practice, and that the most frequently adopted gamification elements are points and badges. Although only four out of six studies performed an assessment of their gamification experience, they all reported a positive improvement in some aspect.
- S3. The third systematic mapping was performed to investigate and report the state of the practice in the field of gamification in SPI. The field of gamification in SPI is relevant to the development of this thesis because we share a similar goal with researchers of that field in terms of improving the development process, except that we are particularly interested in improving agile development process by integrating Lean UX. The article titled: “Challenges of gamification in software process improvement” was published in the journal of Software Evolution and Process in January 2020. Our analysis identified several important challenges and limitations that point to critical issues that need to be carefully considered by practitioners who aim to gamify SPI initiatives. The challenges mainly focus on the complexity of gamifying SPI given the different organizational roles to consider and gamification strategies from which to follow. We noticed that some of the challenges identified have also been found to apply to previous work such as gamification in SE, gamification in SE education, and gamification in agile. We learned that the introduction of gamification should not impose major changes to the improvement process if the gamification experience is to be successful. Change resistance is common in any environment, particularly when people are familiar and comfortable with the way they work and the tools they use. Gamification can be of greater benefit when introduced gradually, starting with no new activities or tools, but merely deploying gamification elements such as points, badges, and leaderboards (PBLs). Additionally, we found that gamification can produce results that are not helpful or are irrelevant to the goals of the SPI initiative. We learned that it is equally important to clearly establish the objectives of using gamification and align gamification with the goals of the original process. This would not only keep the team focused on working towards achieving their goal, but it can also help in raising commitment and engagement to the process.

Table 2. The lessons learned from our previous studies on gamification in SE education, agile, and SPI.

ID	Lesson Learned	S1	S2	S3
L1	Early results on the effectiveness of gamification in SE are promising. But more empirical research is needed to provide more reliable and grounded results and conclusions for the SE community.	✓	✓	✓
L2	The grounds for selecting gamification processes or elements should be explicated and aligned with the goals of the gamified experience as well as to the players' motivational factors.	✓	✓	✓
L3	Points, badges, Leaderboards, and levels are the most frequently adopted gaming elements.	✓	✓	✓
L4	The "6D" gamification framework is the most adopted existing gamification process. However, details about the concrete actions and steps taken to apply this framework into the examined contexts are missing.	✓	✓	✓
L5	Gamification should be introduced in a gradual fashion to reduce the resistance to change and cognitive workload necessary to gamify a process.	✓		✓
L6	Gamification should not be introduced as a separate entity, but rather "integrated in" or "combined with" existing processes or activities.	✓		✓
L7	The additional effort introduced by gamifying a process must be managed. For instance, gamifying a process should not impose a significant change that entails steep learning curve or completely transforming the way people work.	✓		✓
L8	When reporting a gamification experience, it is important to provide as much information as possible about the context, objectives, participants' background, validation or evaluation methods, decisions of selecting gamification approaches or elements, and so on. That is to make it possible to repeat the experience and produce generalizable results.	✓	✓	
L9	Gamification does not necessarily require the implementation of costly tools and frameworks. Positive experiences can be achieved using existing gamification platforms or no special tools at all.	✓	✓	
L10	A systematic approach for gamifying SE is needed. If SE practitioners opt for following a gamification approach, they face the challenge of deciding whether to follow an existing gamification framework or designing a new one.	✓		✓
L11	High-intensity gamification may have a negative impact on the gamified experience.	✓		
L12	Gamification has a positive impact mostly on participant's engagement, performance and learning outcome, and the adoption of SE best practices.	✓		
L13	Most gamification experience in SE education were found to incorporate multiple SE processes in their gamified experience. We learned that this is probably due to characteristics of the standard SE curriculum that comprises different stages of the software development life cycle.	✓		
L14	All studies that performed an assessment of their gamification experience in agile reported a positive improvement in some respect. Aspects of improvements include team's performance, engagement and knowledge of the agile process.		✓	
L15	Metrics used to measure the impact of gamification on the agile process include number of quality user stories produced to measure productivity and creativity,		✓	

	sprint velocity to measure team's performance, and the team's comprehension and understanding of the agile process.			
L16	Different gamification strategies and techniques may be required for the different organizational. Not all roles may have the same interest in the SPI initiative, react the same way nor have similar motives.			✓
L17	It is critically important to establish a well-defined goal and common understanding of gamifying a process among the practitioners. The more precise the goals are (e.g. in terms of expected outcomes), the more likely for the gamification process to succeed.			✓

4.3 The Design Process of GLUX

This section describes the design process followed to address the problem context of this thesis. For the design of the GLUX framework, we followed the 6D gamification design process proposed by Kevin Werbach and Dan Hunter as part of their world-renowned gamification course at the Wharton School (Werbach and Hunter, 2015a). As shown in Figure 8, the 6D is an iterative design process that comprises six steps:

1. Define business objectives.
2. Delineate target behaviors.
3. Describe your players.
4. Devise activity loops.
5. Don't forget the fun.
6. Deploy appropriate tools.

Figure 8. The 6D gamification process by (Werbach and Hunter, 2015b).

The main objective of designing the 6D gamification process was to help business leaders to think like game designers to address business challenges. Scaled to several different sectors, they provide an informative guide on how game thinking and game design can be employed to engage and motivate people in certain behavior through six steps of careful and thoughtful construction of a gamification system.

Applied to the integration of agile and UX, we aim to take software engineers through a journey of applying Lean UX tactics into Scrum in a fun and seamless way, progressing through challenges and levels of expertise in a rewarding environment to add a sense of achievement and content. Considering the nature of the scrum development framework and the characteristics of software developers, the GLUX framework is designed at a high-level specification to accommodate specific projects and teams. It is also intended to provide a general framework to promote creative ways of implementation. In the case of this thesis, the

implementation of the GLUX framework was in the form of a detailed step-by-step guide for Scrum teams wishing to apply Lean UX tactics into their development process discussed in Chapter 5.

It is worth mentioning that the approach we follow to design GLUX falls within “intentional gamification”, as envisioned by (Hamari, 2019). Hamari (2019) describes the evolution of gamification to be moving through two primary developments: intentional gamification, and emergent gamification. Intentional gamification refers to the process of transforming activities, services, products, or even organizational structures into game-like or gameful experiences, intentionally. This type of gamification is commonly adopted when there is an intention to facilitate or promote a particular behavior or practice. Hamari (2019) argues that intentional gamification has been stretched out over time to become an umbrella concept that encompasses many relevant concepts such as game-based learning and persuasive technology. On the other hand, emergent gamification refers to the gradual and unintentional transformation of societies and cultures into environments that are reflective of games and gameful experiences due to the consistent and pervasive engagement with traditional games (Hamari, 2019). In other words, individuals and societies may look at real-life non-game activities or experiences as games because they have played or been involved in games their whole life. The manifestation of such phenomenon can be mostly observed in social media. For instance, YouTube where users strive to gain more views and subscribers, or Twitter where users’ status and credibility can be linked with the number of followers.

In the following sections, we explain how we applied the 6D gamification process mapped onto the problem context of this thesis. In the first step, we explain our objectives of employing gamification. Then the list of concrete target behaviors and metrics are outlined, which in this case is five Lean UX tactics described in Chapter 7 of the Lean UX book 2nd edition (Gothelf and Seiden, 2016). After identifying the why and what we are gamifying, we move on to analyze and discuss our target stakeholders in terms of what motivates and demotivates them. Later, activity cycle is discussed which describes two things: engagement loops and progression stairs. Engagement loops narrate the core process that Scrum teams go through in the gamified framework. While progression stairs reflect the experience changes while moving through engagement loops. Then, elements of fun are added to enrich the process. Finally, based on the aforementioned five steps, the appropriate gamification mechanics and components are picked and integrated into the framework.

4.3.1 Step 1. Define your objectives

The objectives of introducing gamification are closely linked to the research goal of this thesis (defined in Chapter 1) and inspired by the challenge of the developers’ mindset (discussed in Chapter 3). By gamifying the tactics of Lean UX, we aim to achieve two major goals:

1. Improve developers’ mindset towards UX activities: based on the potential of gamification in promoting and developing particular behaviors and skills in different sectors, we aim to exploit the motivational factor of gamification to enhance developers’ perception of UX work. Our goal is to move their focus from delivering functional features to delivering valuable and usable features. By involving all team members to collaborate in UX activities, many challenges in the field of agile and UX integration can be addressed. For example, involving developers in the design process can help create a shared understanding among all team member about the product being developed, what it looks like, how the users are going to interact with it, how a particular feature is beneficial to the user, etc. However, involving developers in UX activities requires motivating them in the first place to participate in such activities. This is when gamification comes in handy as it was

shown to improve motivation and engagement. Unlike UX designers, who have intrinsic motivation to carry out UX tasks (i.e. they enjoy UX work, the motivation is in doing the activity itself), software engineers need extrinsic motivators to perform UX activities as they do not view UX work as part of their responsibilities as software developers.

2. Facilitate a seamless and enjoyable integration of Lean UX and Scrum: this objective was driven by the perception of developers on UX work as an additional costly burden that jeopardizes the agility of the process and the general resistance to new ways of working. On one hand, the Lean UX tactics described in the GLUX framework are simple and lightweight techniques that should not impose a major obstacle affecting the agility of the development process. In fact, the benefit of incorporating them may outweigh the drawbacks in terms of delivering a high-quality increment with good UX which should reflect positively on customer acquisition and retention. On the other hand, the gamification elements incorporated in the GLUX framework establishes a fun environment using elements such as badges and achievements to take the edge off introducing additional tasks to the development process.

4.3.2 Step 2. Delineate target behaviors

” You don’t want the system to be too complex or confusing, keep it simple, you might want to give the user a range of activities” (Werbach and Hunter, 2015b).

This step involves outlining the target actions that can contribute to promoting the objectives of the gamified process. Once a list of target behaviors is identified, metrics for success should be developed.

4.3.2.1. Step 2.1. List of the target behaviors

The target behaviors aimed for are five lean UX tactics. The five tactics are described in Chapter 7 of the Lean UX book to create a more productive team and a more collaborative process for integrating agile and UX (Gothelf and Seiden, 2016). Some minor changes to the description and naming of Lean UX tactics have been made to avoid confusion and to provide simplicity, while preserving the original principles. For example, one of the tactics described in Chapter 7 of the Lean UX book is called “design sprint”. The design sprint refers to a collection of lightweight upfront design activities that take place at the beginning of each development sprint. The term “design sprint” is popularized by the design team at Google Ventures and is used to describe a time-boxed collaborative activity in which a team brainstorm, design, and develop ideas. Design sprint in Lean UX is an evolved version of Google’s design sprint. Depending on the size of the hypothesis, Lean UX’s design sprint does not represent an entire sprint dedicated for design but rather a time-boxed collection of light-weight design activities performed ahead of every sprint. It can be as short as an afternoon or as long as five days in a two-week sprint. To avoid confusion with the development sprint, we decided to slightly change the naming of the design “sprint” to become the design “studio”. This way, a “sprint” stands for the entire development iteration. Why design “studio”? the term “studio” was used to explain one way of running a design sprint in which a design studio is used to structure the activity to make full use of team’s collaboration.

1. Writing Hypotheses.

A hypothesis is a tactical and testable statement of what the team assumes, or thinks is true about the needs of potential users and what can add value to the business. Writing hypotheses is a collaborative activity where the whole team brainstorm and prioritize hypotheses. The Lean UX

canvas can be used as an effective tool to guide the construction of hypotheses (see Appendix A – The Lean UX Canvas).

2. Running a design studio.

Design studio is a collaborative, time-boxed activity in which a team brainstorm, design, and develop ideas related to a hypothesis. It typically takes place at the beginning of each development sprint. Depending on the size of the hypothesis, the design studio can be as short as an afternoon or as long as five days in a typical two-week sprint. Proto-personas can be of effective use to guide this activity.

3. Writing Experiment Stories.

An experiment story is an artefact used to capture any discovery work or testing to be done to validate a hypothesis. It is a handy technique to remind the team of the weekly user experiments and to prevent plans to validate a hypothesis from getting lost. An example of an experiment story is shown in Figure 9 . Typically, it includes the hypothesis to be validated, the tactic or method for validation, the responsible team member/s, and a level for effort estimate.

Figure 9. Example experiment story.

4. Creating a Minimum Viable Product (MVP)

An MVP is the smallest thing a team can build to learn something from the user and get sufficient user feedback. There may not be code, like sketching, prototyping, copywriting, and visual design. It may be an increment or a functionality. It may be version 1.0 of the product. It all depends on what the team aims to learn.

5. Conducting weekly user experiments.

Weekly user experiments are testing sessions aimed to validate hypotheses and gather user feedback. The collected feedback is then used to improve and build the developing product. It should be conducted on a weekly basis to ensure that the team has continuous feedback from the user.

4.3.2.2. Step 2.2. Metrics to measure behavior and progress

This part of step two involves translating the target behaviors into quantifiable results. The Lean UX tactics described in the GLUX framework are intended to have general variables instead of specific values. That

is for different teams and different projects to have the opportunity to assign values for each Lean UX tactic based on the significance and relevance of UX to the overall project.

Based on Werbach and Hunter (2015), points are the easiest and most straightforward technique to measure progress. It is important to note here that “points” in this context does not refer to the gamification component “points”, but rather it describes a numerical value to be associated with a particular target action. How these numerical values are assigned depend entirely on the designer of the gamification strategy. However, these numbers should correspond to the designer’s best estimate to the relative value of the activities to the business objectives (step #1) or final goal. So, points here represent the weight or value of each action.

Estimating values for Lean UX tactics requires prioritizing them in an order of value and impact to the overall user experience of a product. However, the Lean UX book treats all tactics on an equal level in terms of importance and potential impact. Hence, all five tactics described in the GLUX framework are given an equal value. However, when the blending of Lean UX and agile is discussed in Chapter 7 of the Lean UX book, the collaboration of all team members has been described as fundamental for the integration process to succeed. Therefore, a doubled value is assigned for team’s collaboration. To sum, a value of “X” is assigned for carrying out each Lean UX activity, and a value of “Y” where “ $Y = 2X$ ” is assigned for the participation of all team members. For example, the value of running a design studio is 5 and the value for running it collaboratively where all team members are involved is 10.

In addition to assigning value to each Lean UX tactic, we define the frequency of applying each Lean UX tactic within a sprint as another metric for success. For example, in the case of a Scrum project of a two-week sprint, the Scrum team should run a design studio at least once every sprint. In the same scenario, an MVP should be produced on a weekly basis. Similarly, weekly user experiments should be conducted every week, that is twice a sprint. Further details on the metrics for success are discussed in the next section.

4.3.3 Step 3. Describe your players

The GLUX framework is designed to be used by Scrum teams who aim to integrate UX practices into the development process. Therefore, a good broad understanding of the characteristics and motivators of software engineers, and particularly agile teams, is essential to design the framework and the interaction. Besides, the analysis of potential stakeholders’ goals and needs is part of the DSR activities.

Software engineers’ motivation is a complex factor in software development and a difficult metric to measure and quantify. The seminal work of Couger and Zawacki (1980) brought light to the diversity of the human and social aspects that may affect the performance and mindset of software engineers. Productivity and software quality in software engineering is reported to be largely impacted by and tightly linked to the motivation of software engineers (Hall, Jagielska and Baddoo, 2007; Asghar and Usman, 2013; França, Sharp and da Silva, 2014; Marques *et al.*, 2017; Perez-Gonzalez *et al.*, 2019). While there is increasing awareness amongst the academic community that motivation is a critical factor, the motivation of software engineers continues to be difficult to understand and manage (Beecham *et al.*, 2007; França and da Silva, 2010; França *et al.*, 2011, 2014; França, Carneiro and Da Silva, 2012; Verner *et al.*, 2014; Beecham and Noll, 2015; Franca, Da Silva and Sharp, 2018; Facey-Shaw, Specht and Bartley-Bryan, 2019).

Verner *et al.* (2014) conducted a survey across four countries to investigate the factors that motivate software engineering teams and the impact of culture on software engineers' motivation. The main finding from their analysis showed a perceived association between failed projects and low team motivation. They have also found that culture, in terms of software engineers' global location and environment, can affect the software engineer as an individual and her/his profile of characteristics. However, what is interesting is that the top motivating factors of software engineers were found to be culturally independent such as a project manager with good communication skills and having a pleasant experience working on projects. The findings of Verner *et al.* (Verner *et al.*, 2014) also emphasizes the significant correlation between team motivation and project outcome. Surprisingly, making money is not listed among the top motivators of software engineers. The principal motivators comprise factors associated with personal aspects such as sense of achievement and professional growth, among other factors associated with customer/user involvement such as having a customer with realistic expectations and regular meetings with the customers.

One of the most cited articles in the discussion of motivation in software engineering is the systematic review by Beecham *et al.* (Beecham *et al.*, 2007). To the best of our knowledge, the work of Beecham *et al.* may be the most comprehensive published effort to outline the reported knowledge on what motivates and de-motivates software developers. The findings suggest that the literature of motivation in software engineering presents conflicting arguments which shows that motivation can be context dependent and there exists a variation in personality of one engineer to another. Even though the most cited motivating factor was found to be the job itself, i.e. software engineering as a discipline, the authors were unable to explain why. Still, the authors present a list of 21 motivating factors that influence a software engineer's motivation based on analyzing the literature. The most frequently cited motivator was found to be associated with "the need to identify with the task". This includes having clear project goals, a personal interest in the project or the team, understanding the purpose of a task, etc. Rewards, feedback and recognition, sense of belonging and relationships, and collaboration with others were among the most influencing factors that motivate software engineers. Other motivating factors are linked to technically challenging work, autonomy, trust, and respect. An interesting finding that was reported by Beecham *et al.* suggests that software engineering as a challenging profession can be motivating and de-motivating at the same time. Researchers in motivation theories provide a possible justification of this phenomenon through a theory in which it suggests that most human behavior is driven by two types of motivation: intrinsic and extrinsic. Extrinsic motivation prompts behavior by external triggers such as rewards and money, while intrinsic motivation comes from the activity itself (Herzberg, 2017).

Franca *et al.* (2011) extends the work of Beecham *et al.* (2007) in an updated systematic literature review which mostly backs up the conclusions of Beecham *et al.* The findings indicate that the motivation of software engineers is a complex and contradictory topic which was evident in the way researchers address the issue of motivation through different viewpoints and approaches within various contexts. They highlight that the lack of empirical evidence may have led to build only a partial picture of software engineers' motivation.

Driven by the relatively small number of publications and empirical studies on motivation in software engineering contexts, Franca, Da Silva and Sharp (2018) further complement the analysis of motivation of software engineers by conducting a case study research to investigate the factors that influence work motivation and engagement of software engineers. The authors identified that software engineers are likely to be engaged in the presence of creative work and motivated by challenging tasks involving problem

solving and research. However, they highlight that the meaning of “challenge” may vary among different people. What is challenging and creative for a certain group of people, may be boring to others. Yet, challenging work appears to be motivating in three out of the four cases presented in (Franca, Da Silva and Sharp, 2018). The authors explain such phenomenon by claiming that challenges can have a motivating effect only if perceived beforehand as a relatively difficult task coupled with the team’s confidence that they have appropriate conditions and resources to accomplish it (this situation is described as “technical confidence”, as quoted by some of the participants in the study). Accomplishments in terms of producing good results and recognition in terms of receiving compliments on the job done was identified as a motivator among the participants of the case study. Both accomplishments and recognition were perceived as forms of feedback that influence an individual’s self-confidence.

Motivation of software developers has been also investigated in the context of agile which to some extent is consistent with the expectations of Franca, Da Silva and Sharp (2018). The topic of motivation in agile was approached from a different angle than in software engineering in general. The research in this field is mostly driven by the theoretical outcome of self-management and autonomy associated with the adoption of agile practices. Researcher have been investigating into how agile processes may influence software engineers’ motivation in comparison with traditional software development methods (Whitworth and Biddle, 2007; De O. Melo, Santana and Kon, 2012; Albuquerque *et al.*, 2016; Liechti, Pasquier and Reis, 2017; Marques *et al.*, 2017; Masood, Hoda and Blincoe, 2018). Although few studies were able to indicate that the correlation between the use of agile practices and the motivation of software engineers is present, they have not been able to justify this association in term of what are the causes or reasons for that (Melo, Santana and Kon, 2012; Liechti, Pasquier and Reis, 2017; Masood, Hoda and Blincoe, 2017). Nevertheless, few studies were able to outline a list of attributes associated with agile software development that can have a positive influence on the motivation of software engineers. The sense of accomplishment, feedback, and involvement with the team was among the top motivating factors in agile processes (Melo, Santana and Kon, 2012). Albuquerque *et al.* (2016) identified that learning new skills and challenges are also motivating factors, which conforms with the findings of Beecham *et. al* (Beecham *et al.*, 2007) that software engineering can be motivating itself.

The motivators of software engineers examined in the literature cover a wide range of aspects. However, the literature of motivation in software engineering does not clearly distinguish the different roles in SE teams in general, nor in agile teams in particular. Therefore, in this thesis, we assume the motivating factors identified in the literature of software engineering motivation are universal to the different roles in a typical SE team. It is also important to note here that gamification as a motivational technique may not be the all-inclusive tool to directly trigger all the motivating factors. Some aspects are influenced by organizational conditions such as senior management support and the opportunity for promotion, growth, and advancement. Others are associated with financial aspects such as having sufficient resources, as well as factors related to the work culture such as work/life balance and flexible hours. Gamification should be exploited as one tool in a collection of techniques to solve or improve situations. It can be helpful in addressing the extrinsic motivation in the individual and social environments. Therefore, this thesis focuses on the motivational factors that can be prompted using gamification for the design of GLUX. Such factors include rewards and recognition, collaboration and sense of belonging, challenges, feedback, progression, and autonomy.

4.3.4 Step 4. Devise your activity cycles

The activity cycle is a technique to model the actions and reactions of a player in a gamified system. Activity cycles are represented in two forms: engagement loops and progression stairs. In this context, the activity cycle describes how Scrum team members interact with and move through the GLUX framework. Engagement loops narrate the core process that Scrum teams go through and engage in within the gamified framework. While progression stairs reflect the experience changes (i.e. progress) while moving through engagement loops.

4.3.4.1 Engagement Loop

The engagement loop is an iterative process through which the users cycle, offering them incentives and positive reinforcement. The engagement loop of GLUX is illustrated in Figure 10. When the scrum team performs a Lean UX tactic, they receive a feedback as a recognition for carrying out that Lean UX tactic, and at the same time it acts as a trigger to perform another Lean UX tactic. Feedback and triggers should prompt the players motivation towards continuing to carry out a similar or a different Lean UX tactic. In a gamified process, feedback and triggers typically come in the form of rewards for performing the target activity or alerts for missing out one. The value of feedback lies in providing visibility of the teams' progress and performance. This is relevant to the players sense of growth, development, and progression in the gamified process. While the importance of triggers is to keep the team engaged in the activity cycle and motivates them to carry out a target action.

Figure 10. Engagement loops provide feedback and triggers to keep the players engaged in the gamified process.

4.3.4.2 Progression stairs

Progression stairs is a gamification technique that provide the users with a perspective of their progress in the process. Progression can be represented through an escalating difficulty. The progression stairs technique ensures that the process is not completely linear. In GLUX, progression stairs are created through “levels” of expertise and “challenges”. Levels provide the team with a visibility of their progress, while challenges are designed to make the process more engaging and provides a sense of progression and achievement. The first level of the GLUX framework is designed to be very simple and encouraging to draw the team members into the Lean UX process. Then the next level, challenges are made a little bit harder. The team moves through levels of expertise based on their performance in carrying out Lean UX activities and challenges.

As challenges are listed among one of the motivational factors for software engineers, the GLUX framework provides Scrum teams with the opportunity to define their own challenges. For example, UX tasks that the team thinks are difficult or boring to do can be allocated for the “sprint’s challenge”. The sprint’s challenge provides a useful technique to overcome issues collaboratively. By having a sprint’s challenge, it directs the teams focus to work on pending issues that may turn into UX debt. It starts by the team identifying the challenging issues they have in the development process and order them from easiest to hardest. Starting with the easiest, as illustrated in Figure 11, the team picks a challenge every other sprint, sets it as the sprint’s challenge and work towards addressing it. The rest period between every other sprint is important for the team to celebrate their achievement and finalize any pending tasks. An example of a sprint’s challenge can be performing two types of user testing in one sprint to collect more and different variety of user feedback. Overall, the aim from the sprint’s challenge is to make the whole team goal-driven, focused, and collaborative.

Figure 11. Challenges are handled every other sprint.

Even though gamification can be a motivating tool to promote target behaviors, it is recommended that it should be integrated gradually and temporarily into existing processes (see lessons learned L5, L6 and L7 in Table 2 of section 4.2.3.1). The gradual introduction is to avoid or reduce the resistance to change towards new ways of working. While the temporary application can be effective in avoiding the situation where users end up considering the whole process as a game, and focusing on collecting rewards and competing against each other, neglecting the core goal of the original process.

4.3.5 Step 5. Don’t forget the fun

In this step, we focused on adding a fun factor to the agile and UX integration process to keep the Scrum team even more engaged with the integration process. Fun can range from easy to hard, but it can also take different shapes. Since the GLUX framework aims to target the people factor in the integration of agile and UX, we opted for the social fun which depends on the interaction and collaboration among team members to achieve a shared goal, that is to apply Lean UX tactics. The social fun is also represented in what we designed as the “Lean UX achievements board” in which the achievements of the team is visually represented (explained in step #6).

4.3.6 Step 6. Deploy the appropriate tools for the job

The final step of the gamification process was to select the appropriate game elements based on the results of the previous steps and deploy them. For this step, we followed the recommendation of Werbach and Hunter (2015) that a gamification strategy should not be overloaded with game elements. In addition, we

took into consideration the lessons learned from our previous work on gamification in SE education (Alhammad and Moreno, 2018a), gamification in agile (Alhammad and Moreno, 2018b), and gamification in SPI (Alhammad and Moreno, 2020) that was outlined in Table 2 of section 4.2.3.1. Particularly in this step, L2, L3, L9, and L11 were addressed.

The engagement loops and progression stairs described in step four provides the skeleton of our gamification strategy. Starting with gamification dynamics, the GLUX framework incorporates *progression* and *relationships* based on the analysis of software engineers' motivators and aligned with the goals of this projects. To recall, the literature of motivation in software engineering indicates that "sense of belonging" and "collaboration" was identified as influencing factors in the motivation and engagement of software engineers. Besides, a core activity in Lean UX is the collaboration of all team members. Hence, the choice of this dynamic seems appropriate. In addition, *Relationships* was selected to avoid the scenario of "pointsification". Pointsification refers to the simplistic use of gamification as a technique to merely reward players with extrinsic motivators such as points (Robertson, 2010). That is why in the GLUX framework we attempt to combine a variety of gamification techniques aligned with the process objectives and the characteristics of software engineers to facilitate the whole experience of applying Lean UX collaboratively. The dynamic *progression* was incorporated in correlation with the motivating factors related to challenging work, sense of recognition and accomplishments. This entails fulfilling the team's perception of growth and development.

Table 3 demonstrates the gamification elements that compose the gamification strategy of GLUX. The two dynamics (i.e. progression and relationships) depict the big-picture aspects of GLUX but are not directly represented. While the dynamic *relationships* is represented through the game mechanic *cooperation*, the dynamic *progression* is represented through three main game mechanics, namely *challenges*, *rewards* and *feedback*. Rewards and challenges in the context of GLUX are aimed to trigger the behavior of carrying out Lean UX tactics and help the team to achieve the integration of UX into the development process. The mechanic feedback is aimed to provide visibility and transparency of the team's progress and achievements.

Gamification *mechanics* and *dynamics* are represented in the gamification strategy of GLUX in the form of more-specific techniques called *components*. Typically, a component can tie to one or more mechanic, and a mechanic can tie to one or more dynamic. For example, in GLUX's gamification strategy, the component *achievements* are used for *cooperation* through the team-based achievements board while the same component is also used to provide *feedback*. To promote *cooperation*, *achievements* in the GLUX framework are *team-based*. *Rewards* are depicted in the form of *points* and *badges*. while *feedback* is depicted in *levels* and *achievements board*.

Table 3. The gamification dynamics, mechanics, and components that compose GLUX's gamification strategy.

Dynamics	Progression: The player's growth and development	Relationships: Social interactions among the players.
Mechanics	Challenges: Tasks that require effort to solve or address.	Cooperation: Players must work together to achieve a shared goal
	Rewards: Benefits for some actions or achievements	
	Feedback: Information about the how the player is doing	
Components	Points: Numerical representations of game progression	Teams: Defined groups of players working together for a common goal
	Badges: A visual representation of achievements	
	Levels: Defined steps in players progression	
	Achievements: Defined objectives done successfully	

4.4 GLUX: A Gamified Framework to Integrate Lean UX Tactics into Scrum

4.4.1 Overview of GLUX

GLUX is a gamified framework within which Scrum teams can address the challenges of integrating UX activities into the development process to help them deliver products of the highest value to the user.

Using GLUX, the Scrum team will learn how to apply Lean UX tactics in Scrum in a gamified and fun way. Figure 12 illustrates the roadmap of how Lean UX tactics, along with the gamification techniques, are integrated into the Scrum framework. The Lean UX tactics to be integrated in Scrum are: Hypotheses, Design Studio, Experiment Stories, MVP, and Weekly User Experiments. Each tactic and how it can be integrated into Scrum are further explained in section 4.4.4.1. Upon adopting these tactics, the team earn and collect points and badges, and are encouraged to carry out some challenging tasks related to Lean UX. The gamification techniques: rewards (points and badges), challenges, and levels, require customization based on the team's preferences and the project nature, as explained in greater detail in section 4.4.4.2.

Figure 12. A general overview of the GLUX framework, integrating Scrum with Lean UX using gamification.

Starting with brainstorming hypotheses during product backlog refinement, the team creates a list of hypotheses of their assumptions about the needs of potential users and discuss it with the product owner (PO). The team kicks off a sprint with a design studio in which the team picks a hypothesis as a theme to guide the work of the upcoming sprint/s and start sketching and discussing design ideas accordingly. Immediately after the design studio, the sprint planning meeting should take place. In a typical Scrum framework, the team decides which user stories they are going to develop during the sprint planning meeting. Here, in addition to choosing user stories during sprint planning, the team plan for the weekly user experiment in which they put into test a version of their developing product, i.e. an MVP. Those plans are captured in what is called “experiment story”. The main goal of running weekly user experiments is to validate the teams’ hypotheses and collect constant feedback to improve the product.

Generally, for each tactic, the team should be rewarded for applying it, and receive additional rewards for doing it collaboratively where all team members are involved. For encouragement, the Scrum team is given special rewards for carrying out Lean UX tactics for the first time as well as for carrying out some Lean UX challenges. Challenges are set by the team based on the current difficulties and issues the team faces regarding UX. The collected rewards should assist the team to move up through levels of expertise in Lean UX and are added to the “achievements board”. We explain later in section 4.4.4.2 how the gamification techniques rewards, challenges, and levels are applied to the five Lean UX tactics.

Below we highlight the main characteristics of GLUX:

- **Simple and lightweight:** The GLUX framework aims to take software engineers through a journey of applying Lean UX tactics into Scrum in fun and seamless way, progressing through challenges and levels of expertise in a rewarding environment with a sense of achievement and content. The gamification techniques incorporated are lightweight and straightforward such that it should not impose a major obstacle affecting the agility of the team. The simplicity lies within establishing a fun environment using elements such as badges and achievements to take the edge off introducing additional tasks (i.e. Lean UX tactics) to the development process.
- **Team-focused:** The Scrum team is the main component around which the GLUX framework was designed. The GLUX framework was designed to bring the Scrum team together by making a special emphasis on collaboration. The gamification techniques are developed on a team level and coordinated by the team itself.
- **Incremental:** The GLUX framework offers the possibility to incorporate the suggested gamification techniques incrementally for the purpose of introducing new concepts gradually to avoid distracting the Scrum team from the original development process.

4.4.2 The Scope of GLUX

- **Target stakeholders:** Since the Scrum framework does not include UX guidance for practitioners, the GLUX framework is primarily designed for small Scrum teams who are struggling with integrating UX activities into the development process, particularly in the absence of UX specialists. Even with the availability of a UX specialist, the GLUX framework can help align the mindset of the whole team towards the value of UX work.
- **Target projects:** The GLUX framework can be effective to build user-centered software projects in highly competitive markets where UX is a determinant for a business success.

4.4.3 General Guidelines

- The team should have a shared understanding of the goal of using GLUX. In general, the goal of GLUX is to promote carrying out UX activities within the development process to produce high quality products. In case the team have additional specific goals, those should be shared among the entire team.
- The framework is intended to be used as an initial and temporary solution. When the team starts to consider the initiative (i.e. the gamified process) as just a game, it should be phased out to avoid a perception problem of the original purpose of employing GLUX.
- The framework does not represent a hotfix to all the challenges of agile and UX integration. It should be utilized as one technique of a whole solution.

4.4.4 GLUX in Practice

To apply the GLUX framework, it is important to first note that the framework is structured into two fundamental parts:

- A. **The Lean UX tactics to integrate into Scrum.** In this part, we outline in detail five Lean UX tactics and provide guidelines on how they can be integrated in a Scrum process.
- B. **The gamification strategy.** In this part, we provide a skeletal support to gamify Lean UX tactics depicted in a customizable gamification strategy.

4.4.4.1 The Lean UX tactics to integrate into Scrum

This section explains in detail the five Lean UX tactic to integrate into Scrum. As demonstrated in Figure 13, the Scrum team starts with brainstorming hypotheses (#1) with the PO and stakeholders, then picks one hypothesis to be the focus of the upcoming sprint and use it as a theme to guide the team in the design studio (#2). Then during the sprint planning meeting, the team creates experiments stories (#3) to capture the work necessary to validate the current hypothesis and remind the team of the weekly user experiment (#5), for which an MVP (#4) is needed as a proof of concept.

Figure 13. The five Lean UX tactics to integrate into the Scrum framework.

Tactic #1: Hypotheses

The first Lean UX tactic to integrate into Scrum is creating hypotheses. Early in the beginning of the project, during the first product backlog refinement, team members should collaborate to brainstorm hypotheses. A hypothesis is a statement of what the team assumes, or thinks is true about the needs of potential users and what can add value to the business. Writing hypotheses collaboratively should play a major role in generating a common understanding among the team around the main idea and solutions of the developing product. Each hypothesis should be “tactical” and “testable”. That is, it should be specific enough that it can be validated or measured.

Hypotheses are composed of four parts:

- Business outcome: what business goal the team would like to achieve and believe as a measure of success for them and the potential customers. Business outcomes are the team's definition of done.
- Target users: which type of potential users the team is focusing on first. The technique "proto-personas" can be helpful in guiding this activity.
- Benefit to the user: defining the user benefit entails answering what will the end users/customers benefit if the feature is well designed and implemented. What is the user trying to accomplish? Or what does he want to feel during and after using this feature?
- Feature: which key features or services of the product (or improvement of a feature) will help to achieve the business outcome and deliver value to the user. A hypothesis should focus on one feature only. More than one hypothesis can be similar in every aspect except the feature.

A recommended format for a hypothesis is:

"We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature]"

Just like user stories, hypotheses should be prioritized. The prioritization is based on two factors: risk and value. Assessing risk includes estimating how bad it would be if the hypothesis were invalid. While estimating value involves assessing how much value the feature will generate for the customer and the business. The list of prioritized hypotheses is added to the product backlog, so that the team picks a hypothesis to work on every iteration, a sprint, or a theme, depending on the size and risk of that hypothesis. Also, like user stories, hypotheses writing is not only limited to the beginning of the project. New hypotheses and modifications to existing ones can occur at any time during a sprint, especially after user experiments. When that happens, new or modified hypotheses should be added to the product backlog along with the rest of hypotheses and user stories. The Lean UX canvas can be used as an effective tool to assist in composing hypotheses (see Appendix A – The Lean UX Canvas).

Tactic #2: Design Studio

Before the sprint planning meeting, the team should run a "UX design studio" in which they sketch and discuss design ideas for the upcoming sprint/s (Figure 14). The team starts by picking a hypothesis. Depending on the size and risk of this hypothesis, the team might create a theme to guide the design and development for the next set of sprints. In other words, a theme is an umbrella concept for the number of sprints required to work on a hypothesis. Once the team picked a hypothesis, a "kickoff design studio" takes place, where the team think, ideate, and sketch ideas to use them in the upcoming sprint/s. The collaboration of all team members is necessary to define the scope of the work that will be done in the following sprints and to build a shared understanding of the problem being solved and the design of the proposed solution. After the sprint review, new insights and ideas might come up. This is why the team should run a shorter version of the design studio at the beginning of the consecutive sprints to revise current designs or create new ones.

The sprint planning meeting should start immediately after the UX design studio, where the output of the design studio (e.g. sketches or Lo-Fi prototypes) is discussed as a team.

Figure 14. The design studio is a collaborative design tactic that involves sketching and discussing the team's design ideas.

Tactic #3: Experiment stories

Experiment stories are created during the sprint planning meeting. It is an artefact used to guide and plan weekly user experiments. The aim of creating experiment stories is to capture and visualize the work necessary to discover and validate user needs so that the team stays focused and constantly reminded about weekly user experiments. Additionally, experiment stories force the conversation among team members about planning experiments with the users.

Experiment stories consist of:

- Hypothesis or what the team is trying to learn or discover about the user: “We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature]”.
- Tactic: what kind of light-weight technique/method will be used to test this hypothesis and get customer feedback to learn something. For example, simplified user testing, interviews, A/B testing, prototype, meeting with the product owner, etc.
- Who will do the work: Assign the work to a team member/s.
- Estimation points: how much effort and time will that take?

See Figure 15 for an example experiment story.

Figure 15. Example of an experiment story containing the hypothesis to be validated, the validation tactic, the responsible member/s, and the estimated effort.

Tactic #4: Minimum Viable Product (MVP)

An MVP is the most basic version of a product (or as we call it Version 1.0). In Lean UX, it is the smallest thing the team can build to learn something from the user and get sufficient user feedback. There may not be code, like sketching, prototyping, copywriting, and visual design. Or it can be an increment or a functionality. It all depends on what the team is trying to learn.

Depending on the team's objective, there are two types of an MVP:

- Simple MVP to understand or learn something about the market or potential users. At this stage, the team is not concerned to deliver a real value. For example: a landing page that does not function. It can be used just to get an idea of traffic or design.
- Working MVP that represents a part or small version of the product built as quickly as possible to deliver value to the market. For example: a website with only the landing page but functioning properly enough to deliver the main value. In this case, the team can also learn something about the market and potential customers as well as delivering the desired value to the market.

The team should create an MVP before the upcoming weekly user experiment. That is, every week, an MVP should be produced, whether it is in the form of a low-fidelity prototype such as paper prototypes (simple MVP), or a shippable increment (working MVP).

Tactic #5: Weekly User Experiments

Weekly user experiments are testing sessions aimed to validate hypotheses and gather customer feedback, and then use the collected data to improve and build the developing product. It should be conducted on a weekly basis to ensure that the team has continuous feedback from the user. The methods of collecting user feedback can vary as it depends on the tactics indicated in the experiment stories. Even though a selected team member/s is usually assigned to be responsible for conducting the user experiment process, all team members are encouraged to participate by observing, taking notes and collecting feedback from the users.

It is recommended to use lightweight user testing methods such as: heuristic usability evaluation, A/B testing, surveys, observation and note taking, etc.

4.4.4.2 The Gamification strategy of GLUX

With the integration of Lean UX into Scrum, there is a learning curve. The GLUX gamification strategy aims to make this learning curve less steep with three motivational techniques: rewards, levels, and challenges (Werbach and Hunter, 2015b). As illustrated in Figure 16, The three techniques can be incorporated all at once, or gradually starting with the rewards technique, then later moving to challenges and/or levels.

Figure 16. The three main techniques employed in the gamification strategy of GLUX.

It is important to note that the gamification strategy in this part requires customization based on the team's preferences and the project nature. While the part on the Lean UX tactics is presented here as fixed, it is yet flexible to minor changes. The customization action for each gamification technique is summarized in Table 4.

Table 4. The customization actions required to apply the gamification strategy of GLUX.

Gamification technique	Customization action	Example
Rewards	Assigning numerical values to each Lean UX activity in terms of "points".	10 points.
	Assigning a doubled value for the team's collaboration in carrying out each Lean UX tactic.	20 points.
	Designing badges, recommended as simple as possible to reduce the effort of preparation. A sample of badges is given in Appendix B – Badges Collection, and can be used as a starting point.

Challenges	Outlining the project's or team's challenges related to UX in general, or Lean UX in particular.	<p>"Involving the entire team in design decisions".</p> <p>"Running user experiments more often".</p>
Levels	Deciding if moving through levels is based on the number of collected points or the number of sprints in which the team applied Lean UX.	<p>"We'll move through levels every 5 sprints of applying Lean UX".</p> <p>"We'll move through levels every 100 collected points".</p>

Below we explain the three gamification techniques and how to customize each one of them:

Rewards:

The rewards technique involves assigning values for carrying out the five Lean UX tactics and a value for carrying it out collaboratively. Whatever value is assigned for each activity, we recommend assigning a doubled value for collaboration. That is because collaboration is the core activity in Lean UX. Translating those values into gamification elements, we recommend using points and badges. Points can be represented in estimated numerical values (like story points), while the titles and design of badges are open for creativity (a collection of badges to be used as a starting point can be found in Appendix B – Badges Collection).

Accordingly, we outline below the rules for the rewards system:

Rule 1. The Scrum team earns "X" points for carrying out a Lean UX activity. For example, the team earns five points for running a design studio.

Rule 2. The Scrum team earns "Y" points (where $Y = 2X$) for carrying out a Lean UX activity collaboratively where the whole team is involved. For example, the team earns 10 points for collaborating on revising a hypothesis.

Rule 3. The Scrum team earns a badge for carrying out a Lean UX activity for "Z" sprints in a row or after collecting "W" points. For example, the team earns a badge for conducting user experiments for three sprints in a row, or earns a badge after collecting 150 points.

Rule 4. In the case of incorporating the challenges technique, a badge is also given in the case of achieving a sprint's challenge (explained later in Challenges).

Rule 5. In the case of incorporating the "levels" technique, the points collected should assist the team to move through level of expertise. For example, upon collecting 5X points, the team move up from "novice" level to "intermediate" level. (explained below in Levels).

The aforementioned rules for the rewards technique are flexible. Meaning that a team can tweak some rules, or even add new ones, based on the project objectives and team's preferences. For example, for the tactic "create an MVP", the team can assign a number of points for delivering an MVP before the end of the sprint, and a doubled number of points for delivering a functionally working MVP that is potentially shippable (i.e. an increment).

The collected points should assist the team to move up through the levels of Lean UX expertise shown in the “Lean UX achievements Board – Points” (Figure 19). Meanwhile, collected badges are added to the “Lean UX achievements Board – Badges” (Figure 20). We explain later the different levels and how to move through them using the collected points.

Challenges:

The challenges technique is essentially based on identifying UX related issues and difficulties faced by the team and assigning them as “sprint’s challenge”. The sprint’s challenge provides a useful technique to overcome issues collaboratively. It directs the teams focus to work on pending UX issues that may turn into UX debt. As shown in Figure 17, the team starts by identifying the challenging issues they have in the development process and order them from easiest to hardest. Starting with the easiest, every other sprint the team picks a challenge, sets it as the sprint’s challenge and work towards addressing it. Challenges should be introduced every other sprint to allow the team to have resting periods in-between sprints. In other words, if challenge A was assigned to sprint 1, no new challenge should be introduced in sprint 2. The team should earn a special badge every time a challenge is achieved to acknowledge and mark their achievements. An example of a sprint’s challenge can be performing two types of user testing in one sprint to collect more and different variety user feedback. Overall, the aim from sprint’s challenge is make the whole team goal-driven, focused, and collaborative.

Figure 17. The challenges technique of the GLUX framework.

Levels:

The levels technique aims to demonstrate how the Scrum team advances in the journey of applying Lean UX. It provides the team with a sense of progression that can consequently improve the team’s motivation to carry on further in applying Lean UX. Moving through levels can be based on the number of collected points, or the number of sprints in which the team managed to apply Lean UX.

The GLUX framework suggests five initial levels that are essentially based on the “Dreyfus model of skill acquisition” (Dreyfus and Dreyfus, 1980) to represent the team’s experience in integrating Lean UX in with Scrum. Demonstrated in Figure 18, the five levels are:

1. Novice: team members have very limited experience with Lean UX and still learning the core basics. In this case for example, for a team who encounters Lean UX for the first time, it may require them one to two sprints (or its equivalence in points, say 5X+5Y) to achieve the novice level.
2. Advanced beginner: team members have applied Lean UX practices for few sprints, and they have adequate understanding of Lean UX. For instance, it is reasonable to assume that after three or four sprints, the team is familiar with applying Lean UX tactics, even if it is not done in the best way possible.
3. Competent: team members are familiar of all the principles and tactics of Lean UX, they are comfortable applying it on other projects and are confident to answer other people’s questions about Lean UX integration with Scrum.
4. Proficient: team members have deep understanding and a holistic view of Lean UX principles and tactics, and can carry out Lean UX activities in a routinely manner.
5. Expert: team members are fully capable of applying Lean UX practices intuitively without any help, can achieve excellence, and are capable to guide other teams to apply Lean UX in Scrum.

Figure 18. The levels technique offers five levels of competency that reflect the team's experience in integrating Lean UX with Scrum.

The levels technique, as part of the GLUX framework, is team-based. A team keeps its level of expertise through different projects. For example, if a team reached the advanced beginner level in project A, then the team starts from the advanced beginner level in project B. The title of each level is open for customization too, as long as the idea of levels is maintained.

Achievements Board:

The achievements board is a complementary technique that call for having a space, physical or virtual, where the collected points, badges, and levels of the team are posted. It is important to provide visibility of the team’s achievements and progress in applying Lean UX throughout the development process. Just like all the components of the GLUX framework, the achievements board is team-based; where the whole team shares one board to mark and celebrate their achievements. We call it: “the Lean UX achievements board”. While achievements boards are not restricted to a certain design, we provide examples of achievements boards for displaying points in Figure 19 and for displaying badges in Figure 20.

Figure 19. Achievements board is a useful technique to provide visibility of the team's collected points.

Figure 20. Achievements board can also be used to showcase the team's collected badges.

Chapter 5

Treatment Validation

5 Treatment Validation

5.1 Introduction

As explained in Chapter 2, within the design cycle of DSR, the designed artefact is validated before it is transferred to the real-world problem context (Figure 21). The goal is to develop a design theory that structures the knowledge about the artefact from the interaction of a prototype of the artefact and a model of the problem context. The design theory describes if and how the artefact improves the model problem context and allows to predict if, and how, the artefact would improve the original problem if it were transferred to the real-world problem context. Although not essential in all cases, the design theory might provide some explanation of why the artefact is effective, or not, in improving the problem.

Figure 21. The validation of an artefact in the design cycle.

Following the approach by Wieringa (2014), the method used to guide the pilot study is Technical Action Research (TAR). TAR is a special approach of using Action Research (AR) in Design Science. It is a case-based research method used for validation. While TAR is artefact driven, AR is problem driven. However, the researcher works with a model client not just to solve a problem, but also to examine the artefact in context. As shown in Figure 22, a validation model is used in TAR studies to represent the target artefact and the target context by similarity. Validation models are created to simulate various aspects. In our case, it is used to simulate the interaction of a model of the artefact with a model of the context.

Figure 22. The validation model is used to represent the real-world context.

To this end, we created a model of the GLUX framework depicted in the form of a step-by-step gamified guide to integrate Lean UX tactics into Scrum (see Appendix D – The GLUX Guide). This guide reflects the “client-specific version” of the GLUX framework as recommended by Wieringa (2014). On the other hand, the sample client (problem-context model) is represented by means of graduate software engineering students who use the GLUX framework in a team agile project. The validation study took place during a quasi-real-world agile project of three weeks to achieve project outcomes. Our data collection occurred through student surveys, unstructured interviews in a separate class by the end of the term (course retrospective), and the analysis of final project reports.

5.2 Validation Goals and Research Questions

The goal of this validation study is twofold. We aim to examine the usefulness of the GLUX framework in developing a motivated mindset towards UX and creating an enjoyable agile and UX integration process for software engineers (knowledge goal). We also intend to explore the impact of gamification on motivating and engaging agile teams in applying UX activities in the development process (improvement goal).

The research questions driving this validation study are outlined in Table 5. RQ1 examines the usefulness of GLUX which entails inspecting the utility and usability aspects of the framework (Nielsen, 2012). The aim is to improve the design of GLUX using the feedback received. The goal of evaluating the utility of GLUX is to measure its ability or effectiveness in motivating and engaging agile teams in carrying out UX activities. The goal of evaluating the usability of GLUX is to measure how easy it is to follow, including the effort needed to learn how to use the framework and the overall satisfaction from following the GLUX framework on whether it provided a fun and collaborative integration process.

Table 5. Research questions of the validation study.

ID	Research question
RQ1	How is the GLUX framework perceived by Scrum teams in terms of usefulness? RQ1.1. How is the GLUX framework perceived by Scrum teams in terms of utility? RQ1.2. How is the GLUX framework perceived by Scrum teams in terms of usability?
RQ2	What is the effectiveness of the GLUX framework in developing a motivated mindset among Scrum teams towards applying Lean UX activities? RQ2.1. Do the Scrum teams develop a motivated mindset and attitude towards UX? RQ2.2. What is the role of the GLUX framework in developing this mindset and attitude?

RQ2 examines the impact of the GLUX framework on the mindset and attitude of the Scrum teams towards Lean UX. Recalling one the main objectives of this thesis is to explore if, and to what extent, can a gamified solution improve the motivation of agile teams to carry out UX activities within the development process. The aim of RQ2 in particular, is twofold:

1. To seek the participants opinions about Lean UX (and UX practices in general) to look for any indications on their mindset towards UX principles and activities. This part serves the descriptive inference activity of this TAR study which involves summarizing and interpreting all data collected from data sources regarding the mindset and attitude towards Lean UX or UX.
2. To investigate whether the GLUX framework can influence the mindset and attitude of its users (the participants of this validation study) towards UX principles and activities. This part serves the abductive inference activity which involves explaining the findings and outcomes. In TAR studies, the goal is to find out whether the experimental artefact (i.e. the GLUX framework) produced the desired or expected effect (i.e. contributes to a positive mindset and attitude towards UX) in the client context (i.e. agile teams of graduate students).

5.3 Validation Study Design

The design of TAR studies requires decisions about client selection, treatment design, measurement design (data collection methods), inference design (data analysis methods) and threats to validity, research execution, and finally data analysis and reporting.

5.3.1 Client Selection and Overview of the Model Problem Context

In TAR, the object is a model client provided with a prototype of the artefact. In this TAR-led validation study, the selected client is represented by 17 graduate students enrolled in the course of “Agile Methods: Agile Techniques and Agile Usability” (4 ECTS) as part of one-year master’s program in Software Engineering at Universidad Politécnica de Madrid (UPM). The workload required in the master’s program is 60 ECTS credits distributed across two semesters. The validation study took place at the Fall semester of the academic year 2019/2020.

The GLUX framework is essentially designed for small-scale Scrum teams. Our client in this pilot study represents a small-scale team, but the environment may not be typical. The decision of validating GLUX for the first time to be with graduate students was based on that educational environments can often support a more flexible, interactive, and reduced-risk approach. However, we should point out that students often have little technical knowledge and professional experience of the industry. Consequently, students may have different goals and perspective to researchers and to practitioners. Understandably, the goals of students tend to relate more to assessments, learning the subject, and progression in their program. In this

context, this factor may influence the results to some extent in terms of validity or reliability (discussed in detail in section 5.3.5).

5.3.1.1 Course Description and Structure

The aim of the 4-ECTS Agile Methods course is to provide students with an overview of the main techniques used in agile methods and how particular quality issues interact with them. Special attention was given to user experience (UX) as it is a fundamental component when developing software to create enjoyable and positive experience for the user.

The course is structured in three parts. The first part is delivered through classical lectures where the course lecturer presents and explains main concepts of agile development processes. Topics mainly include the agile manifesto and the Scrum framework. The second part of the course is focused on the recently introduced “Lean UX” software development philosophy and the integration of Lean UX tactics into agile (Scrum). Discussed topics include the principles of Lean UX and tactics to integrate into the Scrum framework. In addition, the lectures included few exercises such as Scrum City using Legos, planning Poker, and a hands-on workshop using the Lean UX Canvas (see Appendix A – The Lean UX Canvas). The third part of the course involves developing a team project to apply the concepts and techniques discussed during the theory parts of the course.

5.3.1.2 The Team Project

As part of the course, students are required to work on a team project, simulating the work rhythms of a real-world agile project, to develop an application for managing and funding business ideas of entrepreneurs, applying Scrum and Lean UX techniques. The 17 students were divided into three teams. Teams A, B, and C, have five, six, and six members, respectively. Each team is required to develop the project within three one-week sprints. Teams are expected to work with Scrum activities and artefacts such as developing the product backlog, user stories (including estimation and prioritization), get familiar with the notion of sprints, learn how to decompose user stories into tasks and acceptance criteria, assign tasks to team members, etc. In addition to Scrum, the project is expected to be developed in a user-centric way by integrating a few tactics from “Lean UX”. All Lean UX principles and tactics are taught in lectures. Students are also given a guide on how Lean UX tactics can be integrated into their Scrum project. Upon adopting these tactics, the team earn and collect points and badges, and are encouraged to carry out some challenging tasks related to Lean UX. The guide exemplifies the GLUX framework customized to be used by students of this course, and specifically for their team project. Details of the guide are described in the next section (5.3.2). Finally, teams are given the freedom to choose the technological “ecosystem” to be used such as development IDE, automated testing environment, configuration management system, etc.

5.3.1.3 The students (i.e. the model client)

The relevance of TAR as a validation method in the context of this thesis is established based on two aspects. Firstly, as researchers, part of our goal is to validate the utility of the GLUX framework (i.e. the artefact). Secondly, the model client (the students representing agile teams) may not be primarily interested in our research goal, but they have a problem to solve for which the GLUX framework can be of use. The issue is closely related to the main problem this work attempts to address which is developers’ mindset and attitude towards usability and UX issues. During previous experiences of the Agile Methods course over the past two years, it was observed that usability issues were not given adequate consideration when developing the projects, unless asked to do so as part of the project grading criteria. Therefore, in addition

to validating the GLUX framework, we aim to help students learn and apply Lean UX techniques integrated into Scrum.

To assist in customizing the GLUX framework to be of help with the case of the students, it was necessary to perform an initial assessment of the students' knowledge and experience in agile (specifically Scrum), and UX (specifically Lean UX). The assessment was done through a short questionnaire (QNR1) distributed to students in the introductory class of the course that directly inquire about their knowledge in experience in both agile and UX (see Appendix C – Questionnaire QNR1 and Responses). The questionnaire also included asking about demographic information, qualification and educational background, and work status (e.g. full-time, part-time, not working, etc.). Collecting such information is essential not just to customize the GLUX framework, but also to design the lectures and exercises accordingly. For example, understanding the students' knowledge and experience in Lean UX would be helpful in defining the extent to which the Lean UX tactics should be explained and detailed in the GLUX guide that was given to the students.

Results of the questionnaire show that around half of the students have a previous experience using Scrum (~54%), but none of them have previous experience with Lean UX (0%). The students rated their knowledge of Scrum as minimal to average (~62%), as very good (~23%), and as certified (~7%). The source of knowledge in Scrum may be of theoretical as in lectures, readings, or workshops, or of practical as in actual work experience. In terms of their knowledge in Lean UX, around 85% of the students have no knowledge of Lean UX, and only 15% rated their knowledge in Lean UX as minimal to average. The source of knowledge is purely theoretical as in readings from blog posts and online material about Lean UX. This suggests that Lean UX should be given a special emphasis through lectures and the design of the customized GLUX framework for the students to properly learn its principles and tactics. The full questionnaire (QNR1) and the complete results can be found in Appendix C – Questionnaire QNR1 and Responses.

5.3.2 Treatment Design

Treatment design mainly involves customizing the artefact to the client case. Again, one of the goals of this TAR-led validation study that was captured in RQ2 is to explore the effectiveness of the GLUX framework in improving the mindset and attitude of developers towards UX work (the general problem), as well as to help graduate students formed in agile teams to learn and apply Lean UX tactics integrated into Scrum (the client-specific problem). The customized artefact is depicted in what we refer to throughout this thesis as the “*GLUX Guide*” (see Appendix D – The GLUX Guide). The GLUX guide is a gamified, step-by-step, and guided process that describes Lean UX tactics and how to integrate them into the Scrum framework. The GLUX guide was designed mainly based on the GLUX framework with the use of the information gathered from the initial questionnaire (QNR1).

The GLUX framework offers a high-level gamification strategy that requires few initializations and customizations such as assigning values for each Lean UX tactic and designing challenges, rewards, and levels. Therefore, the design of the GLUX guide entailed customizing the gamification strategy to accommodate the current client conditions which is represented by three agile teams of graduate students developing a software project over three one-week sprints.

Below we discuss the customizations performed for this pilot study as well as some considerations for the GLUX implementation strategy in the pilot:

5.3.2.1 Rewards

Following the rules described in the GLUX framework, we now explain how points and badges were designed for the GLUX guide. For each Lean UX activity carried out, the team earns five points, and an additional ten points for collaboration. Special case was made for the MVP in assigning points where the team earns the additional ten points for delivering a working MVP (i.e. functional and usable feature). The reason for this modification is driven by the value of delivering a working MVP in contrast to a non-working MVP. In a working MVP, the students can learn something from the market and actually deliver value to the end user. While in a non-working MVP such as a paper prototype, the students can learn something from the market or the user, but the end user does not get any value delivered yet.

As for designing badges, the GLUX framework suggests rewarding the team with badges in two instances: 1) in the case of applying a Lean UX tactic for the first time, and 2) in case of achieving a challenge. However, the GLUX framework does not provide details or specific recommendations on how to design those badges. We think that this activity is context dependent, meaning that different teams in different projects can design relevant badges in their own creative way (e.g. [heyta.co.chat](https://heyta.co/chat)). Therefore, we designed a collection of eight badges simply using Microsoft PowerPoint (Figure 23). The badges were carefully designed such that each Lean UX tactic has its own color-labeled set of badges to make it easier for the teams to know which badge belongs to which Lean UX tactic. The design of badges also involved giving a title (in terms of a fun name) for each badge to make it representative of the corresponding Lean UX tactic. For example, a badge for creating a hypothesis for the first time is labeled with “the thinkers” and is colored with “red” (the color assigned to hypothesis). Another example is the “senior designers” badge that is rewarded upon achieving the challenge of running three design studios in a row and is colored with “yellow” (the color assigned to design studio).

Figure 23. The badges collection designed and included in the GLUX guide (i.e. the client-specific version of the GLUX framework).

5.3.2.2 Challenges

Although the GLUX framework recommended that challenges are identified by the team itself based on the UX issues they have during the development process, and due to the fact that we are applying the framework in a learning environment with low-experienced team, we opted to slightly change this rule and give the teams only one challenge per Lean UX tactic. As mentioned, the conditions we deal with in the validation process is a special case; it is an educational environment. Students are still learning about agile principles, Scrum, and Lean UX. Therefore, students were exempt from the activity of designing the challenges. Instead, the challenges were given to them by the researchers and they were deliberately easy. This decision is driven by one of the lessons learned from our previous work on gamification in SE education related to the impact of gamification on the learning process (Alhammad and Moreno, 2018a). We found that in studies with negative gamification experiences, there was an extra cognitive workload for the students from the high-intensity gamification strategy where students had to carry out a lot of new gamification activities. This have caused a sense of confusion or frustration which prevented them from fully grasping the core learning objectives and enjoying the “game”.

The challenges given to the students do not introduce any new activity beside Scrum and Lean UX techniques. They were merely based on encouraging continuous habits. In other words, the challenges entailed that participants (in terms of students’ teams) have to apply Lean UX tactics throughout every sprint of the project. For example, we challenged them to run a design studio for three sprints in a row (which is the whole duration of their project). Similarly, we asked them to write/revise hypotheses, deliver an MVP, and conduct user experiments for three sprints in a row.

Following the guidelines related to challenges in the rewards technique in the gamification strategy of the GLUX framework, the team earns a special badge upon achieving a challenge. The design of badges is discussed earlier in the rewards technique. To repeat, a collection of eight badges was designed. Four badges are given for encouragement purposes upon carrying out a Lean UX tactic for the first time. The rest are rewarded upon achieving the challenges.

5.3.2.3 Levels

The GLUX framework offers the levels technique to demonstrate how the Scrum team advances in the journey of applying Lean UX. The guidelines indicate that levels can be designed based on the number of collected points, or the number of sprints in which the team managed to apply Lean UX. For the GLUX guide, we opted for using collected points as a measure for advancing through levels. That is to give additional value to collecting points. Also, due to the short time frame in which students have to develop their projects, the five levels suggested in the GLUX framework were scaled down to three levels corresponding to the three sprints of the course project. The levels and the required points for the three teams in our validation study are illustrated in Figure 24 and are as follows:

- Level 1 (practitioners): the team finishes level 1 after achieving 70 points which can be earned by the first sprint if the team managed to apply all five Lean UX tactics within that sprint.
- Level 2 (specialists): the team finishes level 2 after achieving 140 points which can be earned by the second sprint if the team managed to apply all five Lean UX tactics within that sprint.
- Level 3 (experts): the team finishes level 3 after achieving 210 points which can be earned by the third (i.e. final) sprint if the team managed to apply all five Lean UX tactics within that sprint.

Figure 24. The gamification strategy of the GLUX guide uses a point system to move through its three levels.

5.3.2.4 Achievements board

The final step in the GLUX framework offers a complementary technique to provide visibility of the team's achievements and progress in applying Lean UX through an achievements board. The GLUX guide includes this technique by incorporating two achievements boards: The Lean UX achievements board for collected points and levels, and the Lean UX achievements board for collected badges. The points and levels board was designed to demonstrate points and levels visually (Figure 25), while the badges board shows the badges related to each Lean UX tactic in accordance to when they take place in the Scrum framework (Figure 26).

Figure 25. The Lean UX achievements board of collected points and levels.

Figure 26. The Lean UX achievements board of collected badges.

5.3.2.5 The implementation of GLUX’s Gamification Strategy

Throughout the validation process of GLUX, we adopted the philosophy of Lean Startup (Reis, 2011). Lean Startup uses a feedback loop called “build-measure-learn” to minimize risk and waste by delivering a prototype of the proposed solution to inspect the developing product as quickly as possible and learn something on how the product can be improved and evolved. The GLUX framework is designed as a proposed solution for improving the integration of agile and UX – a hypothesis. We assume that the gamification strategy of the GLUX framework will improve the integration of agile and UX by motivating and engaging agile teams in UX activities. Our goal is to validate this assumption as efficiently as possible. Therefore, the GLUX framework was implemented as an MVP. Teams keep track of their progress and share their achievements via an achievements board posted in Google spreadsheets.

The decision to keep the implementation of GLUX as simple as possible was also triggered by the lessons learned from our previous work on gamification in SE education (Alhammad and Moreno, 2018a), gamification in agile (Alhammad and Moreno, 2018b), and gamification in SPI (Alhammad and Moreno, 2020). In the previous chapter, we outlined 17 lessons learned that was taken into consideration throughout the design and development of GLUX (Chapter 4, section 4.2.3.1). In lesson learned L5, we found that “gamification should be introduced in a gradual fashion to reduce the resistance to change and cognitive workload necessary to gamify a process”. Additionally, fellow researchers reached a similar conclusion. García *et al.* (2017) found that presenting a gamified tool of their framework “GOAL” in a company as a proof of concept was not feasible nor successful. The company simply was comfortable with their current tools and found that gamifying its own tools (either through a plug-in or re-designing the same tool) is much easier than introducing and training their employees to the new gamified tool “GOAL”.

This should clarify why no technical tool was developed at this pilot study and to validate the initial design and hypotheses before building the final product following a lean approach. We also found in lesson L9 that gamifying a process does not necessarily require the implementation of costly tools and technical platforms. Positive experiences can be achieved when no special tools are used. Introducing a new tool with

new activities and gamification might introduce a steep learning curve that may not be helpful to achieve the goals of gamifying the process.

5.3.3 Measurement Design

Measurement design refers to defining the data collection methods used to capture the data of the artefact under investigation and its interaction with the model context. It entails defining the data sources, data collection instruments, and data collection schedule (pre-test, post-test, longitudinal, etc.). The guidelines by Johannesson and Perjons (2014) were followed to define the data collection methods since such research methods are not thoroughly discussed in the main source followed for DSR and TAR (Wieringa, 2014).

Although it is common to use only a single data collection method, employing several data collection methods can improve accuracy and broaden the picture (Johannesson and Perjons, 2014). Therefore, this validation study follows a *triangulation* approach in which the data collection occurred using three qualitative techniques: questionnaires, interviews, and documents. Questionnaires were used as the main source of data describing the situation, while the interviews and documents were used to further explain, confirm, or even add on, the findings. The data collection process occurred as post-experience, meaning that they were applied at the end of the validation process.

4.3.3.1. Questionnaire

A questionnaire is a written document that contains a list of questions to be distributed to a number of participants for their response. Following the guidelines of Johannesson and Perjons (2014), and inspired by the System Usability Scale (SUS) (Brooke, 1996), a reflection questionnaire (QNR2) (see Appendix E - Questionnaire QNR2 and Responses) was designed with straightforward and brief questions to evaluate the students experience with Lean UX integration into Scrum in general, and with the GLUX guide in particular. The goal is to primarily measure the utility and usability of the GLUX guide, and to capture the students' perception about the role of GLUX's gamification strategy in helping them to learn and apply Lean UX.

The questionnaire QNR2 contains four sections of both open-ended questions and closed-ended questions (see Table 6). Section one includes five general questions on the students experience with applying Lean UX in Scrum. Section two includes five questions asking the students to rate their experience with using the GLUX guide as a document in terms of understandability, learnability, ease of use, and effectiveness in motivating them to carry out Lean UX tactics. Section three includes 11 questions to inspect the impact of the gamification strategy of GLUX in motivating and engaging the team in Lean UX activities. Finally, section four includes only one question to examine the extent of the additional effort introduced by gamification on the agility of the process. The closed-ended questions are designed using a 5-point Likert scale ranging from strongly disagree to strongly agree. For positive statement questions, the scoring is as follows: 1=strongly disagree, 2=disagree, 3=somewhat agree, 4=agree, 5=strongly agree. While the negative statement questions scoring is the opposite as follows: 1=strongly agree, 2=agree, 3=somewhat disagree, 4=disagree, 5=strongly disagree. For evaluating the results, the average score would be compared against the ideal average score that is five points, such that 3-5 points is considered within the range of positive results, and 1-2 is considered within the range of negative results. For example, a question with average response score of 3.0 points is considered a positive result. While an average response score of 2.9 is considered a negative result.

The questionnaire was constructed using Google Forms and distributed among students during the last course class “course retrospective”. The questionnaire was conducted live, meaning that respondents participated all at once with the PhD student guiding the process. Students were given a brief overview of the purpose of the questionnaire and ethical concerns including confidentiality of any personal data. They were also given instructions, written and verbal, on how to answer the questions. Each question was explained by the researcher. Students were then asked to answer the questions individually, section by section.

Table 6. The structure of QNR2

Section 1 of 4 - You experience in general
1.1 How would you rate your knowledge of Lean UX after having this class?
1.2 I would be willing to apply Lean UX in my future projects.
1.3 Based on your answer for the previous question, please tell us the reason of your choice.
1.4 I view UX work as equally important as development work.
I thought that Lean UX tactics (hypotheses, design studio, experiment stories, MVP, and user experiments)
1.5 were distracting and reduced the team's agility (i.e. less working software is delivered, less flexibility, etc).
Section 2 of 4 - Your experience with the GLUX guide.
2.1 I found the GLUX guide easy to understand.
2.2 In case you "don't" find the GLUX guide easy to understand, which parts you think are difficult?
2.3 I thought the Lean UX tactics listed in the GLUX guide was easy to use in Scrum.
2.4 I think using the GLUX guide increased my motivation towards doing UX work.
2.5 I needed to learn a lot of new concepts before using this guide.
Section 3 of 4 - Your experience with the gamification strategy of GLUX
3.1 I think that earning and collecting "points" was encouraging and motivating to apply Lean UX tactics.
3.2 I think that earning and collecting "points" made applying Lean UX tactics more fun to do.
3.3 I think that earning and collecting "badges" was encouraging and motivating to apply Lean UX tactics.
3.4 I think that earning and collecting "badges" made applying Lean UX tactics more fun to do.
3.5 I think that the "challenges" of Lean UX tactics given in this class were engaging and fun to do.
3.6 I think that challenges alone (without earning points or badges) would be motivating enough to apply Lean UX
I thought that moving through levels of Lean UX expertise (practitioners, specialists, experts) provided us
3.7 feedback on how well we're doing applying Lean UX tactics.
3.8 I felt that the "Lean UX Achievements Board" helped us in gaining visibility of our progress in Lean UX.
3.9 I felt that earning points and badges "as a team" has improved our collaboration.
3.1 Which gamification element you enjoyed the most?
3.11 Which gamification element you enjoyed the least?
Section 4 of 4 - Your experience with gamifying the development process
I thought that the gamification part (i.e. earning points and badges, and managing the achievements board) was
4.1 distracting and reduced the team's agility (i.e. Less working software is delivered).

4.3.3.2. Interviews

An interview is a communication meeting between a researcher and a participant, typically in which the researcher drives the meeting by asking questions. According to Johannesson and Perjons (2014), interviews are more effective than questionnaire when eliciting complex data such as emotions, attitudes, and opinions from the participants. To validate the GLUX framework, the type of information that would be helpful is not restricted to how usable or useful the artefact is, but also related to human factors such as how developers' mindset and attitude towards UX has changed to investigate the impact of gamification. Our goal to conduct interviews was to allow the students to express freely about their ideas and thoughts of the GLUX guide and their overall experience using it to integrate Lean UX into Scrum. Therefore, the interview conducted with the students for the purpose of validation was collective and unstructured. The interviews took place immediately after the students filled the questionnaire QNR2 and lasted for approximately 20 minutes. Interviews provided us with the opportunity to go into depth about the students' responses to the questionnaire. Indeed, the questionnaire may provide only a partial picture of the students' opinion as most of the questions were closed-ended which can be subjective to the views of the researcher. Besides, questionnaires can have a low response rate as respondents can ignore a request to answer the questionnaire, skip some questions or provide random answers.

To conduct the interview, the researcher initiated the discussion by asking the students to talk about their thoughts on integrating Lean UX into Scrum, what challenges they had, which parts they enjoyed, and what they learned from the overall experience. The main aim of discussing the experience as a whole is to look for any clues or evidence related to their mindset towards UX work, without directly asking them to avoid bias. For example, a response such as *"I do not think that design is important at this stage, I feel we should first deliver the functionalities, and then think about the design"* would clearly indicate that the problem may still exist and the GLUX framework was not successful in addressing the problem properly or entirely.

The researcher employed three techniques to facilitate the discussion: prompting, probing, and checking. The first technique involved encouraging the respondent to start speaking when a participant shows interest in responding. The second technique involved enriching the respondents' input by requested more details using a follow-up question. The third technique involved repeating the respondent's answer to make sure that the researcher has correctly understood what the respondent stated. Throughout the interview, students shared some of the thoughts and challenges during the project development process. Most importantly, they proposed some ways of improvements related to the management of the gamification strategy of GLUX. The researcher concluded the interview with a thank you statement. The full interview with the students was audio recorded and transcribed for further analysis and interpretation (see Appendix F - Transcript of The Retrospective Class Interview).

4.3.3.3. Documents

Documents constitute a source of data (often textual, but can contain images, audio or video) that of value to the research. In this validation study, students' final project deliverables were considered as an additional source for data collection. The project deliverables demonstrate how each team applied Lean UX tactics into Scrum. Shown in Figure 27, the required deliverable is structured in two main parts: 1) reporting the project iterations and 2) a discussion section. In the first part, students demonstrate and report the work done in each iteration, including Scrum events and artefacts, as well as the applied Lean UX tactics. In the

second part, the students are asked ten questions to provide their reflection as a team about the whole project. The length of students' reports was an average of 60 pages (actual lengths: 32, 63, 81).

<u>PART 1 - REPORTING</u>	<u>PART 2 - DISCUSSION</u>
<ol style="list-style-type: none"> 1. Team members. 2. Brief description of the project. 3. Results of the inception deck. 4. Product Backlog (name and description of user stories, prioritized, and hypotheses). 5. Iterations - per each iteration: <ul style="list-style-type: none"> • Hypothesis of the Sprint. • Sprint Backlog. • UX designs and sketches. • Design Models or any other intermediate results. • Experiment Story of the sprint. • Screenshots of MVP and the sprint's increment. • Summary of user experiment. • Results of the Demo. • Results of the Retrospective. • Sprint burndown chart. • Final Project Burndown. • Lean UX achievements boards. 	<ol style="list-style-type: none"> 1. Which is the technical framework that your team has used in the project? 2. Did you manage to apply all Lean UX tactics in every iteration? What were the most challenging issues of integrating Lean UX into Scrum? And how did you address them? 3. How do you think Lean UX improved the overall quality of your product? Which Lean UX tactics do you think were beneficial and effective in improving the quality of your product's UX? 4. Which are the most interesting findings (lesson learned) that your team got from this project? 5. Which were the most difficult parts of your project and why? For example, detailing the user stories with tasks and acceptance criteria, estimating story points or hours, team collaboration or communication, working with agile tools 6. Do you think that earning points and badges upon carrying out Lean UX activities has helped you or motivated you to learn and apply Lean UX? Do you think they were distracting instead? Discuss. 7. What were the most fun parts of your project? 8. Which were the most boring or least engaging parts and why? 9. What would you have done differently or better in your agile process? 10. Other comments, suggestions, ...

Figure 27. The structure of the students' final project report.

The students' reports were included as a data source because they provide additional descriptive and explanatory answers to our research questions. While the questionnaire provides responses based on the individual opinion, the projects deliverables reveal the collective mindset of the participants as a team. Knowing this is important to understand how the opinion of an individual is affected by the mindset of the whole team, or the other way around?. It should help us to validate one aspect of the GLUX framework of which the main mechanisms are team-based.

Using students' reports as a source of data may present a challenging aspect in terms of quality assessment. Quality assessment of the data sources is essential to generate a reliable and sound analysis. The students' reports are a special case of documents; they are not peer-reviewed as in academic publications, or credible as in government documents, for instance. Thus, it introduces additional effort on the researcher to first create quality assessment criteria, and then evaluate the students reports based on that criteria. The criteria are based on two questions corresponding to the two main parts of the final project deliverable:

- Question 1: Are the required items of each project iteration described and appropriate?
- Question 2: Did the authors answer all the nine required questions of the discussion section?

The questions are scored as follows:

- Question 1: Y (yes), the required items of all project iterations are explicitly described and appropriate, P (Partly), part of the required items of the iterations are described and appropriate, or all the required items are described but some are inappropriate; N (no), none of the required items of the project iterations are described.
- Question 2: Y (yes), all the required questions are answered; P (Partly), part of the required questions are answered; N (no), none of the required questions are answered.

Given the scoring procedure is $Y = 1$, $P = 0.5$, and $N = 0$, each project report must score a total of at least 1 to be included as a data source. Meaning that the two main parts of the report must be completely or partially provided. All three reports provided by the students by the end of the semester passed the quality assessment criteria. Therefore, all three reports were considered as data sources.

5.3.4 Inference Design

Inference design refers to defining and planning the techniques used to analyze the collected data. It entails defining the data analysis methods (quantitative vs. qualitative; descriptive, inductive, abductive, deductive, or analogic) used to infer information from the collected data. Data alone do not provide value without interpretation and analysis. The qualitative data generated from the data collection methods call for a qualitative data analysis method.

To recall, the goal of the validation process is to develop a design theory from the interaction of a prototype of the artefact in a model of the problem context. The design theory allows to explore if the GLUX framework is effective in improving the developers' mindset problem. It should also allow us to predict if, and how, the artefact would improve the problem if it was transferred to the real-world problem context. In this context, the technique employed for data analysis is the qualitative data analysis process by Runeson *et al.* (2012). The approach by Runeson *et al.* (2012) offers a qualitative data analysis process that is based on Grounded Theory (GT) to generate a hypothesis from the collected data. However, the authors refrain from using the term "grounded theory" due to its dual definition and the conflicts over the coding process. Unlike GT where codes should not be defined a priori and should emerge from the analysis process, the qualitative data analysis approach of Runeson *et al.* (2012) starts with "seed categories". Seed categories are an initial set of codes that come from the goals of the study, the research questions, and predefined areas of interest which guides the analysis process.

The qualitative data analysis technique was conducted in a series of steps as illustrated in Figure 28, with few iterations, using NVivo 12 as a supporting tool. First, all material including questionnaire responses, students reports, and interview transcripts were studied in detail. Then, since the main objectives of the research is captured in the research questions, a set of "seed codes" were formulated that reflect the validation research questions as shown in Table 7. The list of seed codes includes interests of the researcher about the artefact and its interaction with the context.

Figure 28. The qualitative data analysis process adopted in this validation study.

The next step involved reading all the material again and marking where the codes fit to the contents. During this process, some codes were reformulated, and others were split into more sub-codes when necessary. For example, the code “mindset” was originally set as “effectiveness”. However, after diving into the material, we found that the term “effectiveness” is too broad and can have a bias effect on coding and interpreting only the positive results. While the term mindset is more specific and can cover both the positive and negative results, and is still representative of the RQ. Similarly, the code “challenges” was decomposed into sub-codes to reflect the types of challenges reported about the integration of Lean UX into Scrum such as collaboration challenge, time challenge, and so on. The code “patch” also emerged from analyzing the challenges which includes the participants’ ideas and solutions to address some of the challenges they faced throughout the integration process. Additionally, new codes were also formulated during the analysis process that represent interesting findings that may not be directly associated with any RQ but could benefit the researcher in improving the artefact or for future research. For example, the code “learning curve” emerged from reading the students comments about the difficult tasks that take effort to do. Formulating new codes and changing existing codes was one part of the analysis where the process was iterative because after that the researcher had to go back to re-read and re-code the material again.

Table 7. The list of "seed codes" and "emerged codes" used to guide the data analysis process.

Seed codes	Purpose
Utility	To mark findings related to RQ1.1
Usability	To mark findings related to RQ1.2
Mindset	To mark findings related RQ2
Challenges	To mark findings related to the challenges of integrating Lean UX into Scrum
Emergед codes	Purpose
Learning Curve	To mark findings related to the effort required to learn and apply Lean UX
Patch	To mark findings related to the participants' solutions to address some of the challenges.

In NVivo, the coding process was managed by creating a “node” for each code (Figure 29), and then placing all relevant text segments under the appropriate node (Figure 30). This gives a clear picture of all relevant data associated with a particular code which should facilitate the next step in which conclusions and patterns are drawn to generate hypotheses. In some cases, it was useful to summarize what has been found in each node and capture the interesting interpretations in “memos”. A memo is simply a text file that contains the researcher’s comments and interpretations about a particular observation (Figure 31). After coding the entire dataset, texts of similar and different codes were closely observed and compared accordingly, looking for patterns and answers to the RQ’s. For example, if a statement was found in node “mindset” and node “challenge” simultaneously, this indicates that there still might be underlying issues with the teams’ mindset towards UX and that the GLUX framework might not be as effective as designed and improvements are necessary. The final step involved generating candidate hypotheses from the conclusions drawn in the previous step.

Figure 29. Nodes are created for each code in Nvivo12.

Figure 30. Relevant text segments are placed under the appropriate nodes.

Figure 31. Memos were used to capture the author's thoughts and interpretations of particular observations.

5.3.5 Validity Threats

This section discusses the threats to validity of this preliminary validation study and countermeasures based on the classification scheme described by Runeson *et al.* (2012). The scheme distinguishes between four aspects of a research validity: construct validity, internal validity, external validity, and reliability.

5.3.5.1 Construct validity

This aspect concerns the degree to which the data collection and analysis methods measure what it is supposed, or what the researcher intends, to measure. It deals with the question "Are we measuring what is intended to be measured?". Multiple countermeasures were taken to improve construct validity of this validation study.

The goals of the study were explicated in the form of research questions to ensure clarity of both the purpose of the study and the intended aspects to be measured. A clearly specified research question should assist in setting out the constructs to be measured. In our case, RQ1 and RQ2 clearly specify the constructs to be

measured explicitly which include “usefulness” of the GLUX framework and its “impact” on the “mindset” of the participants towards UX practices. Setting out those concepts explicitly has also assisted in defining the list of “seed codes” beforehand which generally represent the constructs to be measured.

Triangulation of data collection methods was achieved through employing multiple independent data sources (questionnaire, interview, documents). Given that those multiple data sources measure the same constructs from different angles, it provides multiple sources of evidence to support the interpretations produced. For example, when measuring the mindset of the participants towards Lean UX, direct questions were asked as part of the questionnaire while indirect and explanatory questions were employed as part of the project report discussion part.

Additional countermeasures include employing the “checking” technique, also known as “member checking”, during the interview as explained earlier in section 5.3.3 which involved repeating the respondent’s answer to make sure that the researcher has accurately understood what the respondent said and to avoid any misinterpretations and misunderstandings.

Mitigating risks regarding the construct validity of the questionnaire involved explaining the purpose of the questionnaire and each question to the participants describing what the researcher is aiming to understand and measure. The researcher was available on-site to answer any questions and resolve any confusion. However, we cannot completely ensure if the researcher have unintentionally communicated any expected outcome to the participants either verbally or non-verbally eliciting the desired outcome during the whole process.

5.3.5.2 Internal validity

This aspect concerns the causality (i.e. causal relations) of the results within, or internal to, the validation study. It deals with the strength of assigning causes to outcomes. Unlike laboratory experiments with tightly controlled variables and conditions, this validation study follows a TAR approach, which is a case-based research. Thus, it is difficult to ensure high internal validity due to the difficulty in controlling environment variables.

One of the main RQ’s of this validation study examines whether the GLUX framework contributed or influenced the mindset of the participant towards UX. Therefore, there is a possible risk that the investigated mindset might have been affected by additional factors of which the researcher may not be aware, or over which she has no control. In other words, it may not be possible to unambiguously assign the GLUX framework impact (the intended cause) to the improvement of the participants’ mindset towards UX (the intended effect). Relevant threats to the internal validity of this study include whether other factors than the GLUX framework could influence the participants’ mindset such as experimental set up, psychological or social mechanisms of participants in the validation model, the novelty of Lean UX to the participants, and the risk-free environment. Given that the validation model was allocated in an educational environment, and even though it simulates a real-world Scrum team, it is yet highly possible that the mindset of participants was influenced to some extent by some of the above-mentioned factors. In this case, several “*peer debriefing*” sessions were carried out with the supervisor of this thesis to critically discuss the researcher’s interpretations and analysis. This was facilitated by reporting and documenting the interpretation process to keep an audit trail of decisions and justifications.

Countermeasures to increase the internal validity to a moderate level included directly surveying the participants, in addition to observation, about the impact of the GLUX framework on developing a

motivated mindset towards UX throughout the software development process. Still, participants' feedback can also be subjective to what is known as "measurement influence" (Wieringa, 2014). Meaning that the type of method used to extract this information can influence the responses of the participants. To conclude, even though it is difficult to claim high internal validity in TAR studies, we employed several countermeasures to address the threats of the highest impact.

5.3.5.3 External validity

This aspect concerns to what extent the findings are generalizable, and of interest to other people from outside the validation model (i.e. the current context). Since drawing a representative sample that can be confidently generalized is nearly impossible, the intention of mitigating this threat is to define a theory which enables an analytical generalization based on extending the findings to similar cases with common characteristics. Adopting DSR as a research approach in this thesis has contributed to improve the external validity of this particular validation study in which a theory (design theory) is drawn from the interaction of the GLUX framework with the validation model. This theory should assist in explaining whether the GLUX framework is effective in developing a motivated mindset. It should also help us predict if the GLUX framework would be effective in improving similar contexts.

While this validation study cannot guarantee similar outcomes if the GLUX framework is to be applied in different contexts, it serves as a pilot study on which other empirical studies can be conducted to provide additional data that further approve or disapprove the findings. A replication of this study at several different contexts must be conducted to provide generalizable findings with enough confidence. A key to improving the replicability, and thus the generalizability, of this validation study involved following the TAR approach in the first place. In TAR studies, the context, setting, and design of the validation are thoroughly explicated in detail before embarking upon the validation process.

Because the GLUX framework was designed to be customized, this introduces two types of risks: compliance and treatment similarity. In compliance, similar outcomes to this validation study may not be achievable if the GLUX framework was not implemented as specified. Therefore, this threat cannot be ruled out completely because it depends on how future targets will apply the framework. Treatment similarity is a special threat to TAR studies. Because different clients require different customization of the artefact, similar findings cannot be guaranteed if those customizations were too reconstructive that it does not represent the customization for a real-world client. In our validation study, the GLUX framework (the artefact) was indeed customized to the validation client (Scrum teams of graduate students), but the same customization is applicable, and can be repeated, on a real-world client (any Scrum team) that matches similar characteristics of our teams and the course project.

5.3.5.4 Reliability

This aspect concerns the dependency of the data, analysis, and findings on the specific researcher in terms of whether similar results are reproducible if the study is to be replicated. It deals with the question: "Would similar results be achievable if another researcher conducted the same study later on?" Improving reliability in qualitative studies includes using precision when describing how the study was conducted (Trochim and Donnelly, 2001). Particularly, ensuring reliability in qualitative research calls for describing the changes that occur in the study and if and how these changes impacted any aspect of the research such as the findings.

In this study, adopting TAR as a validation approach helped us to ensure scientific rigor that contributed to improving this aspect of validity. For instance, client selection, artefact customization, and data collection

and analysis methods were all discussed thoroughly including the coding technique for the data analysis to lower the effort on other researchers when replicating the study. In addition, both questionnaires and interview transcript are all documented and available for access (available in Appendix C – Questionnaire QNR1 and Responses, Appendix E - Questionnaire QNR2 and Responses and Appendix F - Transcript of The Retrospective Class Interview) to assist other researchers wishing to replicate this study. Furthermore, an informal “peer debriefing” was employed once again (in addition to internal validity). The process involved inspecting the researcher’s interpretations against what the data says by a colleague to ensure that every interpretation and finding is supported by and consistent with the data.

It is important to note though that in this study direct interaction and collaboration with the model client (graduate students) was feasible. This may not always be the case for other researchers wishing to replicate the study in the same or different context. To obtain an organization’s approval where the experimental artefact can be applied, mutual trust must be built up. Developing mutual trust may require a few months or even a few years for the researcher to be able to conduct the study.

5.4 Validation Execution

This section reports the results gathered from the three data collection methods: questionnaire QNR2, interviews, and documents that describe the perceived and actual effectiveness of the GLUX framework, and the challenges of integrating Lean UX into Scrum. The data collected was saved and analyzed in NVivo, a qualitative data analysis tool.

5.4.1 Quick Summary of the Setting

The validation process took place during the fall semester (September – December) of 2019 at the Escuela Técnica Superior de Ingenieros de Informática, Universidad Politécnica de Madrid. 17 graduate students (formed into three teams) enrolled in the course of “Agile Methods: Agile Techniques and Agile Usability” of the master’s program in Software and Systems. Students were given a series of lectures on the topics of agile, Scrum, and Lean UX and then started working on developing software projects. The project development lasted for three weeks, structured into three “sprints”. Each week, the students developed an increment of the product using the Scrum framework and integrating Lean UX tactics into the development process using the GLUX guide. By the end of each week, the students conducted a “sprint review” with the two course lecturers playing the role of the product owner and the customer. After the third sprint, a concluding class, or a “retrospective class” as we refer to it was conducted to discuss and reflect the development process and the lessons learned. During the retrospective class, two data collection methods were applied: the questionnaire and the interviews. A preliminary analysis of the data collected from the questionnaire and interview was carried out before the students delivered their final project deliverable (i.e. the third data collection method). That was intentional to give the researcher the opportunity to ask the students to provide further details and explanation in their reports about some unclear or partial findings. So, the analysis process began primarily with analyzing the responses of the questionnaire, then seeking supporting evidence and explanations from the interview transcript and project reports.

5.4.2 Results

5.4.2.1 Summary of the Results

This section reports the results of the RQ’s, as well as some general results that are not directly associated with the RQ’s, but rather brought insight into the analysis process and may have influenced the main study

findings. We first present a summary of the results and report the results in greater detail later in sections 5.4.2.2 and 5.4.2.3.

- **General Findings**

- *Perceived knowledge of Lean UX* – Over half of the participants rated their knowledge of Lean UX as good after their gamified experience, while the rest rated it as average. Further investigation showed that although the participants have developed a relatively better knowledge of Lean UX integration into Scrum in comparison to their knowledge before the course, it would still be difficult for them to apply it independently in future projects without further reading, training, or guide.
- *Impact of Lean UX on agility* – Inconclusive feedback on whether integrating Lean UX into Scrum affects the team's agility. Participants argued that the short duration of the project, as well as learning and applying Lean UX for the first time might slowed down work and the response to change.
- *Impact of gamification on agility* – Most participants reported that gamification did not impact the team's agility in a negative way. On the contrary, teams found it to be a useful technique that motivated them to learn and adopt a new way of development.
- *Challenges of integrating Lean UX into Scrum* – Participants reported several challenges from their experience of integrating Lean UX into Scrum that were classified into three themes:
 - *Tensions of the integration process.* This describes the classical tensions between the two processes: agile and UX. It includes compliance with the organization's development approach as the most common challenge, followed by the divided opinion on measuring progress by value delivery and discovery work, and the difficulty to fit Lean UX tactics into short iterations.
 - *Issues with the Lean UX process.* This describes the challenges that are associated with Lean UX specifically. Our teams found that the concept of hypotheses can be frustrating and difficult to define and prioritize, the high collaboration level that Lean UX tactics require can be impractical, and the abstraction and flexibility of the Lean UX process can be confusing for beginner level agile teams.
 - *Learning curve.* This describes the challenges associated with learning and applying Lean UX for the first time. All teams acknowledge that some difficulties of the integration process result from dealing with Lean UX for the first time.

- **RQ1.1. How is the GLUX framework perceived by Scrum teams in terms of utility?**

- Participants found the GLUX framework to be generally effective in motivating and engaging them in applying Lean UX tactics.
- Badges were rated by the participants as the most effective game element, followed by the achievements board. Both elements provided teams with a visual and interactive representation of their achievements and progress in carrying out Lean UX activities throughout the development process.
- Challenges were rated by participants as the least effective element in motivating the participants towards applying Lean UX in Scrum.
- Teams found GLUX to be helpful in improving collaboration, particularly how rewards were obtained by the team as whole, not individually.

- **RQ1.2. How is the GLUX framework perceived by Scrum teams in terms of usability?**

- Understandability: The majority of participants found the GLUX framework to be easy to understand and follow.
- Learnability: Most participants, however, reported that they had to use additional resources about Lean UX in order to follow the framework.

- Again, badges and achievements board were rated as the most enjoyed and fun part of the gamification strategy.
- **RQ2.1. Have the Scrum teams developed a motivated mindset and attitude towards UX?**
 - A relatively positive mindset was observed among the examined teams of graduate students towards UX.
 - A high percentage of the participants (76%) view UX work as equally important as developing functional features.
 - Collective mindset of teams towards UX seems to impact individual mindsets.
 - Most participants expressed their intentions to apply Lean UX in future projects, although there might be some challenges.
- **RQ2.2. What is the role of the GLUX framework in developing this mindset and attitude?**
 - Most participants reported that GLUX had a contributing role in developing a motivated mindset and engaged attitude towards UX work.
 - The GLUX framework might have helped the participants to form a new habit of user-centered thinking.

5.4.2.2 General findings

This section reports the general findings gathered from our study that are not directly associated with the validation RQ's, but rather provide insights into the experience such as factors or opinions that could influence other findings as discussed in section 5.5.2.

5.4.2.2.1 Perceived knowledge of Lean UX

Figure 32 shows how participants rated their knowledge of Lean UX after applying it in the course projects. The goal of this question was to see if, and how, the perceived knowledge of Lean UX can affect an individual's mindset and attitude towards applying Lean UX as the literature of agile and UX integration points out that UX illiteracy can affect developers' motivation towards UX work.

Figure 32. Participants perceived knowledge of Lean UX.

59% of the participants (10 students) thought that they have a relatively good knowledge of Lean UX compared to the knowledge they had before taking the agile methods class and based on their experience with applying it into their Scrum projects. When asked in the retrospective class interview what this rating implies, students responded that they would be able to apply it in future projects, although not independently,

but with the need to use a guiding source in Lean UX such as the GLUX guide, the Lean UX book, or a Lean UX expert. On the other hand, 41% of the participants (7 students) rated their knowledge of Lean UX as average indicating in the interview that although they have a good comprehension of the main concepts and basic tactics of Lean UX, they still need further reading and training in order to apply it in future projects.

5.4.2.2.2 Impact of integrating Lean UX on team's agility

Figure 33 plots the participants' opinions on whether Lean UX tactics were distracting or reduced the team's agility. The results show that 41% of the participants (seven students) somehow agree that Lean UX may affect the team's agility. However, discussion of the retrospective class revealed that this group of participants were not sure or could not decide on whether Lean UX had any impact on their agility due to the short timeframe of the project (three weeks) and short iterations (one week) in which a significant effect cannot be really measured. Participant who thought that Lean UX affected the team's agility (six students) believed that it is most likely because the team is not used to this new way of working (meaning integrating Lean UX with Scrum), and that it is their first time to apply Lean UX. This result can also be linked to the previous finding on the perceived knowledge of Lean UX. We assume that as the knowledge and expertise improve, team's agility should not be significantly affected. As participants discuss in the retrospective class interview, the impact of Lean UX on the team's agility can be a side effect of learning and applying Lean UX for the first time.

"I think mostly because we are not really, umm, used to work in this way. We had a lot of problems understanding new concepts like defining hypotheses. So, I think it mostly the problem with our own experience because we never heard about or worked with Lean UX, so we don't really know how to apply it, so of course it slowed us down a little bit."

"It was hard to think in this type of way, a global way, linking the business value with the user value, with the feature."

We found supporting responses in the students' project reports as well when asked what they have learned from integrating Lean UX into Scrum. Team A acknowledges the value that Lean UX practices have brought to the quality of the product, but they realize the effort required when following a new development approach.

"We think it...", meaning Lean UX, "...improved the quality of our project because it kept us moving in the right direction and it constantly reported to us what we were doing wrong and what we could improve, but we underestimated the initial effort of working in a new development environment."

Team B also confirms:

"A completely new point of view to understand software development. Due to a lack of experience working with experiments to validate or reject the team's assumptions, we found it a little difficult to deal with during the first sprints"

Moreover, team C also confirms how dealing with Lean UX for the first time and integrated into Scrum as a new way of development have affected the pace of the project:

“At the beginning time was passing as we define a very complex hypothesis. The simple reason of this is that we weren’t used to this kind of methodology.”

As for the participants who thought that integrating Lean UX into Scrum did not affect the team’s agility (four students) through the questionnaire QNR2, no explanation was provided as grounds to their opinion.

Figure 33. Participants opinions on the effect of Lean UX on the team's agility.

5.4.2.2.3 Impact of gamification on team’s agility

Looking further and broader into the agility theme of the whole experience of integrating Lean UX into Scrum, we sought the participants opinions on whether gamifying the integration process had affected the team’s agility. To recall, at the time of this validation study, the execution of the gamification strategy of GLUX was completely dependent on a self-organized team. The management of rewards and challenges is carried out by the team itself to monitor their progress and capture their achievements in Lean UX. Figure 34 describes the participants opinions on this issue. Our aim was to measure if, and to what extent, gamification helped or distracted the team in the overall development of their project. Because gamification was closely associated with Lean UX tactics, participants inquired whether they should evaluate the situation considering gamification as an independent entity or as a combined technique with Lean UX (i.e. altogether: gamified Lean UX). Participants were advised to report their opinion about the gamification strategy objectively as an independent technique. The results show that participants mostly disagree or somehow disagree (14 students) that gamification affected the team’s agility, while three participants thought that it did affect the team’s agility.

Details and further discussion of the responses to this question were requested as part of the project report. The results related to the theme of gamification impact on agility as reported by the participants provides supporting evidence that gamification had no impact on the team’s agility. In fact, teams seemed to have enjoyed the gamification strategy and thought that it helped them in adapting to the new way of development, i.e. integrating Lean UX into Scrum. Team A found it to be useful in motivating the team to applying Lean UX each sprint, and kept the team more focused on what needs to be done.

“It was helpful, and it motivated us to include Lean UX activities into our project. It helped us focus and know which tactics we should apply for each iteration. We thought it was a fun alternative to ask for mandatory tasks during the project.”

Figure 34. Participants opinions on the effect of gamification on the team's agility.

Team B thought the gamification strategy was the fun part of the project and not distracting at all.

“The fun part of the project is this, our team enjoyed it while updating the points and badges on every success part of project. Each member got change to Interact with this board of levels. It motivates us to speed up tasks and to collect as much as points we can to complete the Level. As a team, we think it is fun instead of distracting. It is a very small part of our project and we enjoyed investing little time to play with the levels.”

Team C also found the gamification strategy useful in taking the edge off introducing Lean UX to the development process.

“We think that the points and badges helped a bit to face this new way of managing a software project. We think that are needed to do more attractive this part of the project that is a bit philosophic and, for programming-oriented people, can be tedious.”

Notice that the responses received as a team can slightly vary from the responses received from individuals. This explains our decision to employ three data collection methods to complement and further clarify the results.

5.4.2.2.4 Challenges of integrating Lean UX into Scrum

The objectives of exploring the challenges of integrating Lean UX into Scrum are as follows:

1. First, we aim to examine the general feasibility of integrating Lean UX in Scrum to identify the associated challenges as groundwork for further research.
2. Then, by inspecting the identified challenges, we aim to pinpoint any challenges associated with the teams' mindset and attitude towards UX in general, or Lean UX in particular. Significant findings about challenges of the mindset and motivation of the participants would provide valuable clues on whether the GLUX framework was successful to achieve its goal.

Regarding the first objective, the analysis of all three data sources of this validation study identified several challenges related to the integration of Lean UX and Scrum. From analyzing the relevant data associated with the code “challenges”, the challenges were categorized under three general themes: tensions of the

integration process, challenging issues with Lean UX, and the learning curve. Figure 35 summarizes the findings related to the identified challenges.

Tensions of the integration process	Challenging issues with Lean UX	Learning curve
<ul style="list-style-type: none"> • Compliance with the organization's development approach. • Value delivery vs. discovery work. • Short sprints. 	<ul style="list-style-type: none"> • Hypotheses definition. • High collaboration level. • Process abstraction & flexibility. 	<ul style="list-style-type: none"> • Lack of experience or knowledge.

Figure 35. A summary of the challenges of integrating Lean UX into agile.

Tensions of the integration process: This theme describes the challenges associated with the integration process itself, in terms of tensions between the agile development process and Lean UX. Many participants reported that having to comply with the organization's development approach introduces a significant challenge to integrating Lean UX into the development process. Making changes and organizational shifts are not easy. It takes time and effort to convince management and other teams in the organization to transform processes. Many participants of this study shared that their intention to apply Lean UX in future projects is heavily reliant on the company's or team's willingness to adopt some changes in the development process. Because this implies changes to the culture, teams, and process. The whole organization must have a shared understanding of the expected impact and outcomes of integrating Lean UX on the agile process such as embracing UX debt and redefining "done". A related issue reported by the participants of this study involves how teams and management measure progress. While the development team can count discovery work or learning as progress, management may not be able to realize the indirect value delivered from this learning to the end users. Participants complain that even though they are confident about the benefits Lean UX tactics, such as design studio, can bring to the quality of the product, these benefits are hard to communicate to management. One participant during the class interview explained:

"But it is very difficult to explain to your boss the benefits of integrating agile and UX. The problem is that they think UX work is a waste of time. Because you are not delivering direct value to the user quickly, you are spending time designing and improving your work. They want functional product delivered as soon as possible."

Another challenge that occurred during the participants experience of integrating Lean UX into Scrum was the difficulty to fit Lean UX activities into short sprints such as one-week sprint. Teams found it frustrating and overwhelming to carry out development and UX work altogether within one-week iterations of which by the end a product increment must be delivered. When the authors of Lean UX discuss the integration of Lean UX tactics into Scrum, a typical two-week sprint is assumed. This is to give the team one week to validate ideas or part of the work without having to deliver a working increment. But in the case of a one-week sprint, integrating Lean UX might be extremely challenging particularly in situations similar to the aforementioned environment where management is expecting a working increment by the end of each sprint. The severity of the challenge might double up when it occurs in an environment of uncooperative

organization. Some participants of this study described the experience of having to deliver a working increment in a one-week sprint as overwhelming and stressful:

“We felt a bit overwhelmed as we only had one week to do everything.”

“It was quite difficult in the beginning and sprint one was so stressful.”

Challenging issues with Lean UX: This theme outlines the issues that might be specific to the process and tactics of Lean UX in particular as reported by the participants.

Hypothesis definition: First, all teams reported having difficulties with defining hypotheses, especially in the first two sprints. Hypotheses is one of the core concepts of Lean UX. It represents a tactical and testable statement to frame ideas that requires validation and experimentation. While there is a template to help teams translate their ideas into hypotheses, students found it difficult to grasp the concept, especially the business outcome part.

Students thought that this issue might be specific to their own experience as they lack experience, and it is their first time dealing with Lean UX concepts. This might be true, but defining a good hypothesis is not a straightforward activity. Which is why the authors of Lean UX designed two canvases as tools to help teams compose and prioritize hypotheses, the “Lean UX Canvas” and “Hypothesis Prioritization Canvas” respectively. Both techniques were available for the participants to facilitate their discussion and creation of hypotheses. However, most issues were related to the students’ understanding of the hypothesis as a concept and its main components which are business outcome, user benefit, and feature. Common challenges include:

- Defining a hypothesis with a business outcome of an appropriate relation or level of detail as the associated user benefit and feature. In the initial sprints, most teams defined hypotheses that consist of a very high-level business outcome with a very specific “functionality”, such that the functionality itself does not represent a whole feature and may not help the team to achieve the defined business outcome. As we observed during the sprint reviews, the confusion was mostly about the difference between a “feature” and a “functionality”. Because agile teams are used to think of features in the format of user stories which describes specific functionalities of the product, it was probably challenging to think on a higher level. For example, one team’s project idea was similar to the idea of Kickstarter where entrepreneurs post business ideas and ask venture capitals to fund their businesses. This team defined a hypothesis as: “we believe that we will increase our profit from selling entrepreneurial ideas if investors can invest in ideas using the list ideas feature”. In this example, the “list ideas” is a simple functionality that provides a list of all the ideas posted by the entrepreneurs who need funding. In such case, the business outcome and the user benefit may not be achieved by this particular functionality, instead, a more composite feature such as “invest in ideas” or “fund an idea” may be more suitable. The “list ideas” functionality can still be part of the bigger picture feature along with other functionalities such as allowing investors to pledge or bid a certain amount of money. Moreover, in some cases, the feature is not related with the business outcome as it may not contribute to achieve the business outcome. For example, “we believe that we can increase profit margin by 5% if busy users can save time when registering in our website using the one-click signup feature using their Google accounts”.
- Defining a hypothesis with a measurable business outcome. Most teams defined hypotheses with business goals that either cannot be easily measured, or it was poorly expressed such that the real value to the business is unclear.

For example:

“We believe that the support of the university community (students, teachers, staff, etc.) to the entrepreneurial ideas of its students will be encouraged, if the university provides a web application where they can read the opened ideas and provide their comments or ideas.”

This hypothesis defines the business outcome as “the support of the university community”. The business goal here cannot be easily measured and the direct value to the business is not apparent (provide better customer service? Increase market share? Increase profit margin?). The feature is too complex, here it is the whole application.

Another example:

“We believe that the partners will buy ideas if partners can buy an idea by enabling entrepreneurs to put a price to their ideas to sell them”

Regardless of the business outcome definition, this hypothesis defines a feature that does not deliver the specified user benefit. The user benefit “partners can buy ideas” cannot be achieved by the feature “put a price to sell their ideas”

High collaboration level: The second issue reported about the Lean UX process is the high collaboration level it requires throughout the process. All Lean UX activities encourages the involvement of the whole development team from defining hypotheses to running user experiments. This full collaboration might be challenging in situations where team members have different working schedules/hours or are distributed across different locations. In the case of this validation study, most collaboration issues occurred from not being able to arrange face-to-face meetings on a regular basis as participants are graduate students who have different timetables and other courses commitments. Students report that although collaborating in the design process was evidently beneficial, it was the most challenging activity as it requires face to face meetings and inputs from all team members. One student commented during the class interview that the level of collaboration required when integrating Lean UX is much higher than working with an agile framework such as Scrum:

“Yeah I guess it is fine. But Lean UX requires a lot of collaboration, while in agile that is not the case where you can work in isolation.”

By isolation, the student mean that some work can be handled individually such as working on implementing a user story.

Process abstraction and flexibility: Finally, an interesting issue that was reported about Lean UX is the abstraction and flexibility of the process. Although this should be an advantage, participants reported that the lack of details on some aspects of Lean UX can be confusing. Examples mentioned by the students include lack of one specific way to organize Lean UX work in the product backlog, the details on how to run a design studio, and the methods used to conduct user experiments. Some level of detail was given by the Lean UX process as recommendations, but some room was left for each team to manage those details as per the team preferences and objectives. For example, how a team manages UX work within the product backlog, whether include them as separate backlog, separate product backlog items (PBI's) within a single product backlog with development work, or even one PBI for a feature's development and discovery work, is up to the teams itself. There are recommendations, but it is not a matter of right or wrong. For a team

who is still learning how to integrate UX work into agile, this can create some confusion on what they should do.

Learning Curve: This theme describes the challenges associated with dealing with Lean UX for the first time or integrating Lean UX into agile for the first time with no prior, or little, knowledge and experience. This challenge might be associated with the challenge of defining hypotheses.

Acquiring new skills requires time and effort. In the case of this validation study, all teams acknowledge that some difficulties of the integration process come from learning about and applying Lean UX for the first time. For some participants, it was also their first time learning about agile methods, which made the situation even more challenging at the beginning. All students, nevertheless, reported how they improved as a team in carrying out Lean UX tactics over time. The more students experiment with Lean UX tactics, the more they understood it and got better at managing it within the development iterations. This shows that learning and applying Lean UX has a somehow steep learning curve. Mainly, it has a difficult initial learning process since most participants report that most difficulties occurred in the first two sprints.

Lack of experience in Lean UX was cited as the main cause for the challenges of its integration to agile. As beginners, participants highlighted the importance of practicing Lean UX tactics for a longer time before they can confidently and seamlessly integrate it with agile. As team A describes in their project reports, managing Lean UX tactics was difficult mostly in the first iterations:

“Due to a lack of experience working with Lean UX, we found it a little difficult to deal with during the first sprints.”

While team B noted that they observed improvement in defining their hypotheses over time, which suggests that practice is important to make the integration process more seamless:

“Every sprint we made better hypothesis than the previous one.”

The second objective of exploring the challenges of integrating Lean UX into Scrum was to examine if any of the reported challenges are associated with or can be attributed to the mindset and attitude of the teams towards UX. Apparently the challenges reported by participants as discussed earlier may not indicate any negative mindset towards UX. On the contrary, it seems that the teams are aware of the importance and value of UX work to the overall quality of the product that they attempted to address some of the challenges they have to make the integration process work smoothly. The theme “patch” emerged as a sub-code from analyzing the challenges reported by the students which captures the teams’ efforts to improve the integration process.

Both team A and team B expressed that managing Lean UX tactics within the product backlog was challenging. However, team B demonstrated a simple technique to handle Lean UX tactics in the product backlog to make them distinguishable and easier to track among other product backlog items such as user stories. Using Trello as their project management tool, they labelled product backlog items with different colors such that each color represents a card type that the whole team understands. For example, hypotheses were labeled red, experiment stories labelled black, and so on (Figure 36).

Another challenge was managing Lean UX tactics and development tasks in a short sprint such as a one-week sprint. Team A mentioned that this was an issue for them as they found that defining hypotheses and running the design studio took longer than expected. However, they soon discovered that brainstorming

was a helpful technique to speed up all types of tasks. The brainstorming sessions involve giving each team member the opportunity to present an idea or a suggestion such as improvements to the hypotheses definition or a design solution, and the team discuss and find solutions for any pending issue.

Figure 36. Screenshot of the Trello board of team B showing how the team color labelled product backlog items to distinguish Lean UX tactics.

5.4.2.3 Research Questions

5.4.2.3.1 RQ1 – How is the GLUX framework perceived by Scrum teams in terms of usefulness?

As discussed in section 5.2, RQ1 examines the usefulness of GLUX which entails inspecting the utility and usability aspects of the framework. The goal of evaluating the utility of GLUX is to measure its ability or effectiveness to motivate and engage agile teams in carrying out UX activities. The goal of evaluating the usability of GLUX is to measure how easy it is to follow, including the effort needed to learn how to use the framework and the overall satisfaction from following the GLUX framework on whether it provided a fun and collaborative integration process.

RQ1.1. How is the GLUX framework perceived by Scrum teams in terms of utility?

To assess GLUX's utility, participants were asked to rate the effectiveness of the main features (Rewards (points & badges), challenges, levels, and achievements board) in providing an engaging and collaborative experience to integrate Lean UX with Scrum. The rating scale is from one to five, where one represents very ineffective and five represents very effective. The average of participants ratings for each element of the GLUX gamification strategy are shown in Figure 37.

Figure 37. Participants' rating of the effectiveness of each gamification element of the GLUX framework.

Badges were found to have the most effective influence on teams' engagement and motivation on carrying out Lean UX tactics in Scrum. In the GLUX gamification strategy, badges were earned upon carrying out a Lean UX tactic for the first time or achieving a Lean UX challenge. Badges provided teams with a visual representation of their achievements which might be very motivating and encouraging. The achievements board was also found to have an almost equal influence on the participants experience in integrating Lean UX into Scrum. We assume this is because the achievements board provided the teams with a dynamic and interactive visual space where they can track and share their collected pointed and badges and offered them constant feedback on their progress in applying Lean UX. Ratings regarding points and levels were relatively high in terms of the techniques' effectiveness in helping the teams to have an engaging experience. However, challenges were rated as the least effective part of the GLUX gamification strategy.

As collaboration is a core activity in Lean UX, the GLUX framework was designed with a special emphasis on collaboration by rewarding teams with additional points and badges for carrying out any Lean UX activity collaboratively. In this validation study, besides evaluating the mindset, motivation, and engagement of the participants, we aimed to measure the effect of the GLUX framework on teams' collaboration. Participants were asked if the GLUX framework was effective in improving the collaboration among team members towards achieving their goal. Figure 38 indicates that around 65% of the participants (11 students) found the framework helpful in improving their collaboration, particularly how rewards were awarded to the whole team, not individually. Teams draw special attention during the retrospective class discussion to how the improved collaboration

Figure 38. Teams opinions on whether the GLUX framework improved their collaboration.

was especially useful in the Design Studio tactic in comparison to previous situations where teams struggled to involve the whole team in the design process which led to the lack of shared understanding about the product. Even though most teams said that they enjoyed collaborating, around 35 of the participants (six students) thought that the GLUX framework might have improved their collaboration but only to a certain extent. This group of participants shared some collaboration issues they faced during the development process in which the GLUX framework may not be particularly effective. For example, because the participants are graduate students who have different schedules and other courses, teams underscored that some collaboration issues occurred from not being able to arrange face-to-face meetings on a regular basis which led to poor communication.

“We enjoyed collaborating as a team to develop a great idea (for ideas), learning more about agile and discovering lean UX.”

“It was very difficult for us to have face-to-face meetings, we ended up having just one per sprint which revealed to not be enough”

“It requires a lot of collaboration.”

“We had a hard time trying to find space in our schedules that would allow us to get together and have a face-to-face meeting”

RQ1.2. How is the GLUX framework perceived by Scrum teams in terms of usability?

To evaluate the usability of the GLUX framework, we asked the participant two types of questions. A question on whether they found the GLUX framework in general easy to understand and easy to follow. Other questions inquired if some particular parts of the framework provided a collaborative and fun experience. Figure 39 plots the participants opinion on the overall usability of GLUX in terms of understandability and learnability. The majority of students (around 94%) found the GLUX guide (which is an implemented version of the framework) easy, or somewhat easy, to understand. However, most participants had to learn additional information from further other resources in order to be able to use or

follow the GLUX framework. Resources include the course material on Lean UX (e.g. slides) and online resources such as blog posts.

Figure 39. Participants opinions on the overall usability of the GLUX framework.

Participants were asked to inform us which parts of the framework they found insufficiently explained or hard to understand. All teams faced difficulties in understanding the concepts of Lean UX tactics, not the GLUX gamification strategy. Lean UX concepts include defining and prioritizing hypotheses and including them into the product backlog, experiment stories, and design studio as it may not be thoroughly explained in the GLUX framework. The rationale of this decision was to give the teams the opportunity and freedom to decide on the specifics to fit the team's preferences. However, we did not realize at the time of designing the guide the extent of details required to help the participants in this context who are teams of beginners in agile and Lean UX. Participants provided feedback for improvement in the questionnaire and the final project reports. They wrote:

"We had a lot of difficulties when selecting a proper hypothesis, and sometimes it became frustrating. Knowing how to integrate Lean UX tactics in the backlog was also quite challenging, especially since it does not seem there is one right answer."

"Selecting Hypothesis is not an easy task.."

"I understood but I think is important to make more exercises to have more skills about that."

"Design Studio for me was kind of abstract, I didn't understand at first glance which specific tasks I am supposed to do."

"The experiment stories."

In general, even though the usability of the GLUX framework in terms of understandability was acceptable, the learnability aspect was inadequate. However, the above findings provide suggestions and insights on how the GLUX framework can be improved.

We further asked the participants to rate each component of the gamification strategy based on whether it made the integration process pleasant and fun. The goal here is slightly different than the goal from measuring the utility. There, we were measuring the *effectiveness* of the gamification components in motivating and engaging the teams in the integration process, while here we are measuring if it added a *fun factor* to the integration process. Together, both aspects complement each other to measure the *usefulness* of GLUX.

Participants were asked to tell us which part of the GLUX framework they enjoyed the most and they think it made the integration process more fun to do. The results are somehow compatible with the findings from measuring the utility. Figure 40 shows that badges were rated again as the most enjoyed part of the gamified experience in addition to having the most influencing impact on motivating and improving the teams' mindset. The same pattern continues with finding the achievements board as the second most enjoyed, and as seen most effective, part of the gamified experience of integrating Lean UX in Scrum. Teams pointed out in their final reports how much they enjoyed updating their achievements board whenever they earned a badge:

“Our team enjoyed it while updating the points and badges on every success part of project. Each member got change to Interact with this board of levels. It motivates us to speed up tasks and to collect as much as points we can to complete the Level”.

“Whenever we open it, it helps us to see how much extra we can do to get special badge for each level.”

Figure 40. The most enjoyed gamification elements of the GLUX framework.

Again, challenges received the most noteworthy feedback rated as the most enjoyed part by only around 12% of the participants. This percentage is not significant, but it is consistent with the findings from measuring the utility when challenges were found to have the least effect on improving the team's motivation towards the integration process.

5.4.2.3.2 RQ2 – What is the effectiveness of the GLUX framework on developing a motivated mindset among Scrum teams towards applying Lean UX activities?

As discussed in section 5.2, RQ2 examines the effectiveness of the GLUX framework in motivating and engaging the Scrum teams towards integrating Lean UX into Scrum. Our aim is to first explore the participants' mindset towards adopting Lean UX tactics during the Scrum development process, and then investigate the role of GLUX in facilitating the integration process and influencing the participants' motivation and engagement towards user-centered development.

RQ2.1. Do the Scrum teams develop a motivated mindset and attitude towards UX?

RQ2.1 aim to seek the participants opinions about Lean UX or UX practices in general to look for any signs or indications of improved mindset and/or attitude on UX principles and activities. In general, the results from all three data collection methods suggest a relatively positive mindset and attitude among the examined teams of graduate students towards UX when compared to what has been reported in the literature and our experience in previous courses. Nevertheless, it is critical to investigate whether the GLUX framework has contributed in any sense to this positive mindset and attitude.

First, to elucidate, mindset towards UX in this context refers to the collection of thoughts and opinions a team or an individual have about UX. On the other hand, attitude towards UX is reflected in the team's or individual's behavior and actions regarding UX. Normally, mindset is manifested in attitude. For example, if a software engineer believes that UX practices are worth the effort because they improve the quality of the software product (i.e. mindset), she or he is most likely to apply UX practices throughout the development process (attitude).

The questionnaire responses regarding how the three Scrum teams view UX work in comparison to development work revealed an average score of 3.5 (on a scale of five points) which indicates a relatively positive mindset. We specifically asked the participants whether they think UX quality is as important as functional quality. Figure 41 shows that around 76.5% of the 17 participants (13 students) agree, or agree to some extent, that UX quality is as vital as functional quality. In the retrospective class discussion, this group of participants argue that even if a software is free from bugs and function properly, it may not be successful nor useful if it is not built with moderate to high usability and offer the users with a good experience. The two components, functional quality and UX quality, are both crucial in developing any software product.

Figure 41. Participants agreement rates to the statement "I view UX as equally important as development".

The teams realized the importance of UX practices on the overall quality of the software product and reported their thoughts in the final project reports. For example, team A reports:

"We believe that one of the main benefits we found implementing Lean UX was that each feature we considered to add to our web app was thought on the basis of the business outcome and the user

value we wanted to provide, instead of just building many features we were not sure about their benefits they could bring to the business or the users”

Nevertheless, it is reasonable to assume that the above finding was based on a collective opinion. In the responses from the questionnaire, four participants do not think that UX work should be given equal importance to programming work. None of those views were mentioned in the teams’ reports. In the retrospective class interview, this group claimed that functionalities are prioritized over usability aspects because for a software to be usable, it has to function properly first. Hence, assuring that a product is free from bugs is more critical than assuring that it provides a good UX. Moreover, one student argued that not all software projects require special attention to UX aspects such as API’s (Application Programming Interface). Therefore, UX quality is *not always* as critical as functional quality.

Participants were asked about their willingness and intentions to apply Lean UX in their future projects, and the rationale of their response. The aim was to further assess their mindset and attitude toward applying it voluntarily, not just because it was part of the course project, and to deeply understand the incentives and/or barriers of adopting UX practices. The data gathered present some interesting results. Figure 42 summarizes the participants opinions on whether they are willing to apply Lean UX in the future. The majority (70.6%) indicated that they may possibly apply it in future projects, but there might be some challenging factors that can prevent them from doing so. For example, four participants mentioned that the lack of knowledge and limited experience they have with both agile and Lean UX present an obstacle to successfully apply it in near future projects. As one participant explains:

“I had no prior knowledge or experience about Lean UX, so I’m definitely not an expert yet.”

Having to comply with the company’s standards or development philosophy can also create a barrier as mentioned by two participants.

“Because I need to stick with company’s approach on agile lifecycle, but on personal projects for sure”

Other factors are associated with the project nature (e.g. in some R&D projects in which requirements are handed in to the development team) and project type in terms of user centricity.

Figure 42. Participants' willingness and intentions to apply Lean UX in their future projects.

Participants who showed enthusiasm towards applying Lean UX in future projects present 23.5% of the total participants. This group is motivated by the systematic and structured approach of the Lean UX process and the importance of UX and design practices in software development.

“It is very important to consider design in development process.”

“Because it makes sense to focus only on the most important and “user tested”. Because I have been in projects where the product launched nobody cared about, and nobody wanted the product. So, now this makes total sense for me.”

Only one participant found the Lean UX process as difficult and tiresome and showed no intentions to apply Lean UX in any future projects. In the retrospective class discussion, the interviewer (the PhD student) had the opportunity to further explore the different point of views on this particular issue of how Lean UX can be difficult to integrate into agile. One student felt that Lean UX is more focused on the “design” process, rather than the programming process, which is true. Lean UX, as the name indicates, provides an iterative process to user experience design. However, combined appropriately with agile processes, a team can lift the force to choose one methodology over the other and achieve frequent delivery of valuable and validated features. The main challenge of such integration is apparently associated with the increased level of collaboration, as stated by one student.

“But Lean UX is focusing too much on process, rather than programming. It requires a lot of collaboration, while in agile that is not the case where you can work in isolation. I mean compared to agile.”

RQ2.2. What is the role of the GLUX framework in developing this mindset and attitude?

RQ2.2 intends to investigate if, or to what extent the GLUX framework, particularly the gamification strategy, have positively or negatively contributed to motivating the mindset of its users (i.e. the participants) towards UX. In the questionnaire QNR2, participants were directly asked if they think that the GLUX framework has influenced their motivation to carry out UX activities. Figure 43 shows the rate of participants’ agreement on the positive impact of the GLUX framework on developing a motivated mindset towards applying UX practices.

Figure 43. Participant agreement on the role of the GLUX framework on improving their motivation and mindset towards UX.

Most participants (94.1%) either agree (70.6%) or somehow agree (23.5%) that the GLUX framework helped them to acquire a motivated mindset towards UX work. Although this finding establishes a direct connection between the GLUX framework and the improved mindset of the participants, indirect evidence was found in students' reports that confirms this connection. In each of the three students' reports, we observed a pattern in which students, although managed to apply all Lean UX tactics, did not recognize the value of UX activities until the third sprint. Having the GLUX gamification strategy motivating and engaging them was probably valuable in forming a new habit of user-centered thinking and user-centered design, getting them to continue all the way throughout the project and see the real impact of UX practices on their own projects. When asked about the most interesting lessons learned from the course project, team A wrote:

"Although it was difficult to understand and apply at the beginning, once incorporated with current agile methods, it is time-saving (which means there is minimum waste). We learned that applying Lean UX methods to Scrum is a good approach to improve the quality of software development and the final product."

Team B confirms that Lean UX was challenging in the beginning but nevertheless their mindset and attitude towards the tactics seem motivated:

"All of us agreed that the most interesting finding was the concepts of Lean UX since they were completely new for all of us. We discover not only new tools and steps, but also a completely new way to define, execute and evaluate the software development, centered on the user experience."

Similarly, team C acknowledges the benefits they found from integrating a UX process into the development process:

"It is a very interesting approach that provides several benefits on product quality, user satisfaction, and a more efficient and effective project."

On the other hand, only 5.9% of the participants (one participant) thought that the GLUX framework was not directly helpful in increasing her/his motivation to carry out UX activities. Surprisingly, this participant thought that the particular components of the GLUX gamification strategy, namely points, badges, challenges, levels, and achievements board, improved her/his motivation and engagement level in carrying out UX activities. This is interesting because it might be common to observe an opposite stance in different situations where something is appealing in general except some specific parts. Based on this individual's responses to different questions, we understood that it is very likely that the participant thought that the particular components of the GLUX gamification strategy helped in improving her/his motivation to carry out Lean UX tactics in this particular context/project, but it was not necessarily helpful in developing a motivated mindset towards UX work in general. This interpretation is backed up by another response from the same individual when asked about her/his opinion on the importance of UX work in comparison to programming work. The participant responded that UX work should not be given equal importance as programming work in all cases, specifically when a software project does not deal directly with an end user. This finding indicates another point of interest in which participants may responded to some questions in a general sense, even though the researcher emphasized that feedback should be based on their experience in developing their course project.

5.5 Discussion

This section summarizes and discusses the principal findings of this TAR pilot study. One of our goals was to explore the improvements of the teams' mindset towards UX (i.e. improvement goal) and examine the effectiveness of the GLUX framework in developing such mindset (i.e. knowledge goal). Other aspects were worth to explore such as the impact of Lean UX integration to agile on the team's agility. The challenges of integrating Lean UX into agile were useful to explore as well as they can present clues about the mindset of the examined agile teams which could reveal supporting evidence to the actual mindset and attitude towards UX. Apart from analyzing the challenges for the purpose of validating the GLUX framework's effectiveness in improving the motivation and mindset of agile teams, the identified challenges also provide valuable data to the SE community on the integration of Lean UX into Scrum.

5.5.1 Summary of Results

The results of the validation study revealed some valuable aspects that do not directly contribute to the RQ's validation, but rather bring explanations and insights into the experience such as factors or opinions that may influenced other findings. Figure 44 summarizes the general findings that were gathered from our study. As mentioned earlier in section 5.4.2.1, those general findings covered four main aspects: Lean UX knowledge improvement, impact of Lean UX on team's agility, impact of gamification on team's agility, and challenges of integrating Lean UX into Scrum. First, in terms of participants' perceived knowledge after the integration experience, most students believe they developed a relatively good knowledge of Lean UX compared to before working on the course project. However, further training and education was found necessary to achieve expertise and independence in applying Lean UX. By independence, we mean the ability to integrate Lean UX into agile without assistance or coaching. Second, evidence on whether applying Lean UX can affect team's agility was inconclusive due to the short timeframe of the project. To assess the impact of Lean UX on agile, further research on longer-term projects is required. Nevertheless, few participants reported that Lean UX has affected their agility and they attributed it to the lack of knowledge and experience. Third, the impact of introducing gamification was also examined to verify if gamification can serve as a supportive rather than distracting tool. The findings suggest that gamification

had no impact on the team's agility. On the contrary, teams reported that they have enjoyed the gamification strategy as it helped them in adopting a new perspective of software development. Finally, several challenges of integrating Lean UX into Scrum were identified that fall into three categories: tensions of the integration process of UX and agile, challenging issues specific to Lean UX concepts and process, and the steep learning curve introduced by learning and integrating Lean UX for the first time.

Figure 44. Summary of the general findings.

Figure 45 plots the main findings of each RQ. RQ1 focused on the usefulness of the GLUX framework in helping teams to have a seamless agile and UX integration process and develop a good mindset and attitude towards UX practices. In terms of utility, the GLUX framework was generally found effective in helping Scrum teams to develop a motivated mindset towards UX work. Badges and achievements board were found to have the most effective influence on improving the teams' engagement and motivation to carry out Lean UX tactics in Scrum. In terms of usability, understandability of the GLUX framework was acceptable, but the learnability aspect was slightly inadequate regarding the explanation of the Lean UX tactics to be integrated into Scrum. Again, badges and achievements board were rated as the most enjoyed part of the gamified experience in addition to having the most influencing impact on motivating and improving the teams' mindset.

RQ2 examined the mindset and attitude of the participants towards Lean UX in particular, and UX in general and the influence of the GLUX framework in developing this mindset. The results indicate a relatively positive mindset and attitude among the examined teams of graduate students towards UX and its value to the overall quality of the product. The GLUX framework was found to have a motivating role to build this positive mindset and attitude through its gamification strategy. Teams found the framework helpful in facilitating the integration process which helped them acquire improved habits and mentality throughout the development process.

Figure 45. Summary of RQ findings.

5.5.2 Lessons Learned

Collective mindset may override individual mindset

The literature of agile and UX points out that for a good integration between the two processes, it is critical that UX activities are viewed with equal importance and value to the overall quality of the product and business (Salah, Paige and Cairns, 2014). In this validation study, analyzing the mindset and attitude of the examined teams presented an illustrative view on how collective mindset can sometimes override individual mindset. Even though all teams reported that UX quality should be considered as important as functional quality, individual responses revealed some slightly different points of view. Notice that the collective mindset of the teams was reported as part of the project report in which teams express their thoughts about the importance of UX quality specifically in their course project. The collective mindset of the three teams was reflected through two things: 1) their attempts and efforts to apply all lean UX tactics, and 2) their responses when specifically asked about the role of Lean UX tactics in improving the overall quality of their developing software product confirming the value of UX practices on the overall quality. On the other hand, the questionnaire sought individual responses. But more importantly, questions regarding the mindset towards UX may seemed too general, which may have led students to have different individual responses than the responses they provided as a team in the project report.

This points out to the need to carefully design data collection methods. Even though the researcher in this validation study was aiming to measure the mindset and motivation of participants towards UX work in the

particular context of the course project, it was interesting to see the general opinion and motivation towards UX. When the researcher surveyed the participants opinions about the importance of UX quality, the intended context in this case is the development of software products that interact directly with an end user. But as it appeared, this does not necessarily imply that the participants had a similar perception. Many participants provided their opinion in general and argued that UX can be viewed with equal importance only in certain contexts and within cooperative environments in which UX quality is appreciated.

Developers' mindset towards UX could be influenced by a variety of factors

We further assessed the mindset and attitude of the participants by exploring their intentions to apply Lean UX in future projects. The goal was to measure their motivation towards applying it voluntarily, not just because it was part of the course project. The results show that willingness to apply Lean UX in the future was found to be significantly influenced by factors from other findings. As shown in Figure 46, such findings include Lean UX knowledge and experience, the effect of Lean UX on the team's agility, the mindset and attitude towards UX practices in general, or Lean UX in particular. All those findings are considered as factors that shaped their experience with integrating agile and Lean UX may have also influenced each other in one way or another. For example, in this validation study, some teams reported that the impact of integrating Lean UX into agile may depend on the team's knowledge and experience in Lean UX. As the teams who participated in this study are relatively beginners in both agile and Lean UX, it is logical that many reported that Lean UX significantly affected their agility due to lack of knowledge and experience. This concludes that as the knowledge and expertise of Lean UX improve, team's agility should not be significantly affected. Although the mindset towards UX was generally positive, it was somewhat influenced by the lack of knowledge and experience which also affected the team's agility early in the project. This finding is confirmed in the literature where lack of motivation is partially attributed to the lack of knowledge and training in UX and low confidence to carry out UX tasks (Bornoe and Stage, 2014). The slow progress in delivering features because the team is occupied with learning and applying Lean UX for the first time, triggered a general impression that there is no progress and they need to focus more on building functional features instead of improving UX.

Figure 46. Some Influencing factors found in this study that affect the experience of integrating Lean UX into agile.

The learning curve of Lean UX may have influenced the findings on the usability of GLUX

The results from RQ1 suggest that the GLUX framework offers satisfactory usability level in terms of ease of use and understandability. However, it shows average learnability level in terms of accomplishing Lean UX tactics such that some first-time users of the framework may not able to apply it directly without additional reading about Lean UX. But it is not clear whether the explanations provided about the Lean UX tactics in the GLUX framework lack detail, or if this result is attributed to one of the integration challenges identified regarding the learning curve of Lean UX. Because Lean UX introduces new concepts such as hypotheses to the development process, it is more likely that teams with no prior knowledge or background find the tactics difficult to understand or apply at first. We also learned from this validation study that Lean UX presents a learning curve. Teams reported that the integration process was slightly overwhelming in the first sprint because they are in a learning phase where all concepts and practices of agile and Lean UX are encountered for first time. Hence, adding more details might not guarantee that learnability will improve, it might on the contrary make the process more overwhelming by overloading the teams with more information to grasp and process. One way to better understand the situation and know what to do is to run additional validation studies in various contexts. Contexts like agile teams who have sufficient knowledge and experience in agile but no or little knowledge and experience in Lean UX, and teams who are experts at both agile and Lean UX but they lack motivation to integrate the two processes.

Challenging may, or may not, be engaging

We also observed that even though all teams agreed to some extent that integrating Lean UX into Scrum can be challenging at first, all teams found the integration process as the most engaging and enjoyed part of their project. The remark was simple to spot, but the factual reason for that may not be a clear-cut. It is possible, based on the supporting evidence from RQ1 and RQ2, that the GLUX framework contributed to

making the integration process engaging. But it is also possible that this phenomenon is associated with citing challenges as one of the motivating factors in software engineering (Beecham *et al.*, 2007; Franca, Da Silva and Sharp, 2018). But the latter explanation has a weaker evidence in our study as participants rated challenges as the least effective part in motivating them to carry out Lean UX tactics. This evidence is consistent with what the literature indicates that for challenging tasks to be motivating, software engineers need the technical confidence to enjoy it, otherwise, it may not be motivating. In this study, most teams reported that lack of knowledge and experience in Lean UX may influenced the overall experience. This takes us back to the former explanation, which is that it could be an indirect impact of the GLUX gamification strategy, especially that this claim is reinforced in few evidences from the questionnaire and students project reports. When participants were asked directly whether they think the GLUX framework made the integration process more fun, responses were mostly positive. Nevertheless, further explanations can still be viable. For example, it is possible that the participants enjoyed the challenge of learning and trying a new development approach but did not particularly enjoyed the Lean UX challenges given to them as part of the GLUX guide, which explains why challenges were not highly rated as a motivating factor. Moreover, an interesting point of view that was discussed by Runeson *et al.* (2012) suggests that a learning environment can often be more flexible, interactive, and risk-free than a professional or enterprise setting. Therefore, this may lead to students having perspectives, motivations, and objectives, different to practitioners. Whatever motivated our graduate students, may not necessarily work with practitioners. To recall, the aim of this study was to validate the design of an artefact in progress and explore how it interacts with the model context. Further validation studies are necessary to develop reliable conclusions and generate generalizable hypotheses.

Managing the integration of agile and UX as one process has its own new challenges

In their most recent study of the literature of agile and usability integration, Curcio *et al.* (2019) identified several challenges that highlight the tensions between the two processes. Their analysis showed that most of the identified challenges evolve from separating the two processes, usability and agile development, segregating roles and activities into different tracks. Overtime, new proposals have suggested managing agile and UX activities as one process, by arranging and carrying out activities from both fields throughout the entire and same development cycle, in what is referred to as “Agile-UX”. However, new challenges started to emerge from the new Agile-UX approach, ranging from organizational culture and the need for support to maintain the activities from both fields in the development process, allocating activities and resources, through difficulties with the team itself.

In this respect, it is noteworthy that our study identified few challenges that emerged from managing the integration of agile and UX as one process, in addition to further challenges that are specific to Lean UX and the context of this validation study. By one process, we mean that there are no parallel cycles or tracks in which part of the team works on something, while the other part works on something else. Instead, a single development process is adopted where the team alternates between agile (Scrum) and UX (Lean UX) activities within the same development cycle. The issues identified in our case from managing the integration as one process match some of Curcio’s findings. However, this type of issues is still not thoroughly explored in the literature due to the novelty of this single-process approach.

As the GLUX framework aims to address one particular issue that emerge from managing the integration of agile and UX as one process, to which it leads to developers’ focus on functionality features over UX features, some issues remain yet unresolved. For instance, we found that having to comply with the

organization's approach and culture is a common challenge that was highlighted by Curcio *et al.* (2019) as well. Integrating UX activities into the development process implies changes to the culture, teams, and process. If the whole organization does not have a shared understanding of the expected impact and outcomes of such change, especially in terms of what counts as progress and the definition of "done", the integration process can be extremely challenging. While the development team may appreciate and realize the indirect value of discovery work or learning to delivering a pleasant user experience, it may be challenging to communicate such value and its tradeoffs to management. In other words, the benefits of adopting UX practices within the agile development process can call for few trade-offs, but in some cases the former may outweigh the latter. As one friend says: "software engineering is the engineering of trade-offs".

Our teams also reported difficulties in fitting Lean UX tactics into Scrum in terms of lack of time due to the short length of the development sprint. Similar challenges related to time were also reported by Curcio *et al.* (2019) including specifically lack of time to perform usability tests and upfront designs, or very short iterations to exploit several design options. Again, this type of issues is associated with the team's, as well as management's, definition and understanding of what counts as progress. Fitting UX activities may imply delivering a smaller number of functional user stories by the end of each iteration, but it may contribute to developing more mature, even though fewer, features in terms of UX. Due to time constraints or short sprints, tests related to usability or UX are usually neglected to accommodate other types of tests such as unit tests and acceptance tests. This is when "UX debt" occurs.

Like technical debt, UX debt occurs when fast solutions are launched and UX issues are not treated with equal significance as functional issues, which is common in projects under tight schedules or impractical constraints (Anna Kaley, 2018). Few techniques to avoid or address UX debt in agile has been proposed. One technique is to dedicate a specific number of story points to address UX features and/or fix UX issues in each sprint (Anna Kaley, 2018). This approach may require creating separate Product Backlog Items (PBI's) for UX activities in the product backlog. In this case, UX features can be considered as increments. Another pattern that focuses more on value delivery involves including UX activities as tasks in PBI's. In such way, only working product can be considered as increment and a PBI is not considered "done" until both functional and UX features are developed and tested. While those techniques may work well in the short run, teams may not commit all the way, arguably due to mindset (Liikkanen *et al.*, 2014). Consequently, the GLUX framework aims to further contribute to the UX debt issue through addressing agile teams' mindsets and habits towards UX work.

Gamification can be susceptible to the "novelty effect"

While the early results of this pilot study suggests promising potential of the GLUX framework in improving the mindset and attitude of agile teams towards UX, it may be still subjective to the phenomenon "regression to the mean" (or as Koivisto and Hamari, (2019) refer to it as "the novelty effect") due to the novelty aspect associated with introducing the gamification strategy (Kahneman, 2011). Simply put, the excitement and engagement towards applying Lean UX tactics may fade away after the first several iterations. It has been claimed that gamification may lose its effectiveness or even decrease motivation over time (Deci, Koestner and Ryan, 1999; Werbach, 2015). To truly measure the effectiveness of GLUX, future validation and evaluation studies must be conducted on a prolonged period of time, as well as in different contexts and settings. One way the usefulness of GLUX can be measured more accurately is by running

two tracks of case studies, on the same or different agile teams, one track integrating Lean UX in Scrum without the use of GLUX, and the other track employs the GLUX framework.

5.5.3 Improvements to GLUX

To recall one of the goals of this validation study was to examine the usefulness of GLUX in addressing the “developers’ mindset towards UX” problem. By analyzing the participants’ feedback on the utility and usability of GLUX, our aim was to identify deficiencies in the design of GLUX. While all changes are considered minor, below we discuss two themes of improvements that were applied to the new version of the GLUX framework (i.e. GLUX 2.0 in Appendix G - The GLUX Framework v2.0): improvements to the gamification strategy of GLUX, and improvements to the description of the Lean UX tactics included in the GLUX framework.

- *GLUX’s Gamification Strategy - Delay the introduction of challenges until a few sprints through the development process:* Lean UX challenges were rated as the least motivating gamification element in the GLUX gamification strategy. This finding was surprising as it might contradict with one of the theories in the literature of motivation in software engineering claiming that challenging tasks can be a motivating factor that engages software engineers in their work. However, the arguments around “challenges” being a motivating factor, as discussed in Chapter 4, section 4.3.3 might provide an explanation to such finding and provide insight on how to improve this aspect of the GLUX framework. In brief, the literature claims that challenges can have a motivating effect only if perceived beforehand as a relatively difficult task coupled with *the team’s confidence* that they have the appropriate knowledge and resources (i.e. technical confidence) to accomplish it. In the context of this validation study, and in relation to the reported findings on students’ perceived knowledge of Lean UX, participants stated that they are *not* confident yet to apply Lean UX independently without further training and reading on the Lean UX topic, and they do not consider themselves as experts. Therefore, it is safe to assume that the team’s knowledge in Lean UX had affected in some way the results regarding the challenges presented in the GLUX framework. Therefore, while preserving the idea of setting challenges every other sprint, we include a suggestion in the improved version of GLUX (see Appendix G - The GLUX Framework v2.0) to postpone the team’s first Lean UX challenge until further in the development process when the team has learned and adapted to the new development process integrating Lean UX into Scrum.
- *Lean UX tactics - Include further information and additional resources about Lean UX tactics.* Our teams reported some confusing aspects and lack of details about the Lean UX tactics described in GLUX. It was not very clear for us whether the description of the Lean UX tactics in GLUX was inadequate, or if this issue is associated with the impact of the learning curve of Lean UX on the learnability of GLUX as discussed earlier in this section. Therefore, we opted for providing examples and additional external resources of the Lean UX tactics described in the framework. For instance, we noticed that all teams were confused on how to manage Lean UX tasks in the product backlog. The confusion is understandable. Because there is not one way to do that. During the students projects, the author of this thesis provided the participants with some recommendations on how UX activities are commonly managed in the development cycle. Even though, those recommendations had to be documented in the GLUX framework. In this case, two patterns of managing Lean UX tasks were included as recommendations in the GLUX 2.0 documentation. The recommendations were derived from a workshop on Professional Scrum with User Experience (PSU) (Scrum.org, no date) in which the author of this thesis has participated. The recommendations essentially entail that there should be a single product backlog where UX work

can either be included as tasks within PBI's or as separate PBI's. Another example was related to the abstraction of the design studio description. Some teams reported that even though they know what the expected output of the design studio is, but it was not clear how a design studio can be run efficiently. In such case, we included in the documentation of GLUX 2.0 a structured and specific approach of running a design studio as described by the authors of Lean UX (Gothelf and Seiden, 2016). Although the authors of Lean UX suggest that teams should feel comfortable to tweak the design studio process proposed to fit the team's rhythm and environment, we assume that such level of detail would be particularly helpful to novice agile teams. Additionally, most of the feedback we received regarding the challenges of Lean UX evolved around the definition and prioritization of hypotheses. Teams often complained about the difficulty to choose a hypothesis to work on for the sprint. Luckily, the author of Lean UX has recently put together a simple matrix called the Hypothesis Prioritization Canvas (HPC) to help teams decide which hypothesis should be the focus of the sprint based on assessing risk and perceived value. The assessment is basically done through the team's collective best guess. Therefore, the HPC was included as an external resource to help teams prioritize their hypotheses. Finally, in terms of including further information about Lean UX tactics, examples of a hypothesis statement and an experiment story were included to further demonstrate the components of each tactic.

5.5.4 Design Theory

In DSR, the aim of validating an artefact is to produce design knowledge of the artefact and systemize this knowledge in a design theory. In this pilot study, we were able to develop a micro-range design theory, to explicitly structure the knowledge about GLUX in its intended context (small, self-organized, and cross functional Scrum teams), and to extend the theory and use it in the process of cumulative knowledge development. In this context, our design theory states that:

The GLUX framework may improve the process of integrating agile and UX by developing a motivated and engaged mindset of Scrum teams towards integrating Lean UX activities in the development process.

We should, however, stress on the need for further empirical evaluation studies to develop a more mature and abstract design theory of GLUX in its intended context as we may discover different challenges and get different results.

Chapter 6

Conclusion

6 Conclusion

The work of Da Silva *et al.* (2018) on the evolution of agile and UX underscores the significance of integrating UX with agile and its value in the development of great software products. While there are many challenges and tensions to address, the integration of both areas is feasible (Curcio *et al.*, 2019). The literature points out that discussions over the value of integrating UX practices into agile processes are no longer necessary and that research should be directed towards addressing the existing challenges. Despite the maturity of the field, technological means in terms of supportive tools and artefacts still have a long way to go (Da Silva *et al.*, 2018). Particularly, there is still a need to design and develop artefacts that can support the integration activities and help teams to build a shared understanding.

The key goal of this thesis was to design and provide a preliminary validation of an artefact that can improve the integration process of agile and UX. We broke this goal into three phases representing the three key activities in the design cycle of DSR: problem investigation, treatment design, and treatment validation. We first analyzed the challenges of integrating agile and UX as reported in the literature and identified a design problem related to agile teams' prioritization of functional features over UX issues. Then, using the knowledge and lessons learned from the literature of agile and UX and our previous work in (Alhammad and Moreno, 2018b, 2018a, 2020), we designed and presented the GLUX framework which primarily aims to help agile teams to develop a user-centered mindset throughout the development process. Following the design of GLUX, we conducted a TAR study to validate the usefulness and applicability of our proposed framework in motivating and engaging agile teams to carry out UX practices in the development process.

Having accomplished our main goal, this chapter ends the thesis by highlighting the research contributions, discussing the main findings and limitations of the study, and finally suggesting directions for future research.

6.1 Research Contributions

In concluding this doctoral research, we must begin by assessing the achievements of our original goal. The work of this thesis was primarily motivated by the various challenges of integrating agile and UX reported in the literature, set out and discussed in Chapter 3 to identify and explicate a key challenge which this thesis aims to address: the difficulty to prioritize UX activities in agile projects due to developers focus on functional features. To that end, our central research goal deal with the design of an artefact that can improve the developers' mindset towards UX work.

In the second activity carried out as part of DSR, which represents the treatment design, we identified the key techniques to employ in the design of the artefact taking into account our analysis of the problem context in Chapter 3. This included deciding which UX process to integrate into which agile framework, how gamification can be designed such that it triggers software engineers' motivation and engagement, as well as justifying how the components of the proposed artefact would contribute to improving agile teams' mindset towards a more user-centered one. Then we carried out the design process of GLUX following the six steps of the 6D gamification framework by Werbach and Hunter (2015b) to gamify the integration of five Lean UX tactics into Scrum as proposed by Gothelf and Seiden (2016).

The output of the design activity represents the key contribution of this thesis which is **a framework that employs gamification to motivate agile teams to adopt a user-centered mindset and carry out UX activities throughout the development process**. To the best of our knowledge, GLUX is the first

framework proposed in the in the field of agile and UX integration to address the challenge of agile teams' prioritization of functional quality over UX quality. The GLUX framework features two parts:

- a) Five Lean UX tactics to integrate into Scrum: hypotheses, design studio, experiment stories, MVP, and weekly user experiments
- b) A customizable gamification strategy that consists of three components: rewards system, challenges systems, and levels system.

For agile teams who wish to use the GLUX framework to adopt UX practices, the team should simply follow the Lean UX tactics as described in the framework's specifications, and customize the gamification strategy. The customization, as explained in the framework's description in greater detail, requires assigning numerical values for each Lean UX tactic, brainstorming or outlining the teams' challenges related to UX, designing relevant badges, and deciding how the team measures their expertise by the number of sprints or the number of collected points.

As part of the design process, **we also examined how gamification was employed in software engineering, particularly in software engineering education** (Alhammad and Moreno, 2018a), **agile** (Alhammad and Moreno, 2018b), **and SPI** (Alhammad and Moreno, 2020). Our goal was to identify the key trends and challenges. Indeed, we have identified several critical issues that were carefully considered during the design process of GLUX.

It is worth mentioning that the literature of agile and UX covers a wide range of challenges that include various factors and aspects that may not be addressed using gamification exclusively, if not at all. In this respect, we should note that the inclusion of a gamification strategy in the GLUX framework is intended to address a particular issue, namely the "developers' motivation and mindset towards UX", that is acknowledged in the literature of agile and UX integration and established in the problem investigation chapter of this thesis (Chapter 3). Issues that were not, or may not be, addressed using GLUX are captured in the identified challenges of integrating Lean UX in Scrum as well as in the rest of the challenges of integrating agile and UX in general (Curcio *et al.*, 2019). For example, the GLUX framework is intended to be used by small scale independent teams. Therefore, having to comply with the organization's development approach and culture is still an open issue in the design of GLUX in particular, and in the literature of agile and UX in general.

6.2 Key Findings

The contributions made by this thesis are directly aligned with our key goal. Following the DSR paradigm, we approached our goal by presenting two RQs, to which we present the findings below.

In RQ1, our goal was to understand how gamification can be employed as a motivational technique to design an artefact that can improve the developers' mindset towards UX. Addressing this RQ involved two main activities. First, as part of the problem investigation of DSR, we identified, described, and explained the key problem we aim to address. This included analyzing and tracing the root causes and consequences of the developers mindset towards UX as discussed in the literature of agile and UX integration to provide a more precise description and position of the phenomenon and assess the relevance and significance of such a challenge. **We found that the prioritization of functional features over UX aspects embodies a set of issues and it represents a recurrent obstacle when integrating UX with agile.** Based on our analysis, we found that the tendency to prioritize functional quality over UX quality emerges from four key issues: the misinterpretation of the phrase "working software" in the agile manifesto, the lack of motivation

towards UX work, resistance to change, and the curse of knowledge in which agile teams think they know better than the user. **We concluded that the significance of the mindset challenge occurs when agile teams favor releasing as many functional features as it can be developed in a sprint such that UX aspects and issues are not given equal attention. In such scenario, UX issues can pile up over time resulting in UX debt.**

Regarding RQ2, our goal was to examine the usefulness of the GLUX framework in shifting the mindset of agile teams to become user-centered and to see if the framework offers agile teams with a technique to achieve a seamless and enjoyable integration of UX practices into agile. Consequently, we conducted a TAR study using a model of the GLUX framework depicted in a step-by-step guide to integrate Lean UX tactics into Scrum, with a sample client represented by three agile teams of graduate software engineering students. **We learned through the pilot study that our proposed framework (GLUX) was generally effective in motivating and engaging the participated agile teams in carrying out UX practices throughout the development process, with few aspects to improve in the future.** We primarily measured the teams' opinions, views, and thoughts about the integration of UX into the development process and its value to the overall quality of the product, and the effectiveness of GLUX in facilitating such integration in terms of its utility and usability. The measurement occurred through a triangulation of three data collection methods: questionnaire, interview, and the analysis of the teams' final project reports. In terms of the framework's utility, our participants found the framework to be effective in helping them as a team to develop a motivated mindset towards UX work and made the integration process enjoyable and fun. The badges and achievements boards of GLUX's gamification strategy were found to have the most effective influence on improving the teams' engagement and motivation to carry out Lean UX tactics in Scrum. In terms of the framework's usability, the participated teams rated the understandability of GLUX as acceptable. However, the learnability aspect was reported to be slightly inadequate in terms of lack of detailed explanation of the five Lean UX tactics. Once more, badges and achievements board were rated as the most enjoyed and fun parts of the gamified experience in addition to having the most influencing impact on improving the teams' motivation and engagement.

The validation study revealed some interesting findings that did not directly contribute the original goal of the validation, such as influencing factors and side opinions, which helped us to draw more thorough and better-connected conclusions. **Those findings led us to identify and outline six lessons learned and two themes of potential improvements to the GLUX framework.** For instance, we learned that challenging tasks can be motivating and de-motivating at the same time. Some of the controlling variables was found to be knowledge and experience, meaning that challenges can be motivating only when coupled with the sufficient knowledge or technical excellence that enables software engineers to tackle such challenges; otherwise, it can result in a demotivated mindset. This phenomenon was evident in our study when the participated teams reported their feelings of frustration during the early sprints, particularly with no prior experience with Lean UX.

6.3 Limitations

In this section we outline some of the limitations to this doctoral research. Each research method employed in this work has its own shortcomings that may have influenced the findings and conclusions. We have discussed the major limitations to the validation of GLUX in Chapter 5 as part of inspecting the threats to validity, but here we critically reflect on the combined research to identify the areas that might have influenced our conclusions. While some limitations are common in similar studies, such as small sample

size and lack of generalizability, we should point out that the adoption of DSR as a research methodology and TAR as a validation methodology helped us to achieve scientific rigor.

To begin with, our analysis of the problem in Chapter 3 was limited due to the limited number of prior studies that thoroughly examine the challenge related to the developers' mindset towards UX in agile teams in terms of the causes or influencing factors. The number of research articles that report the issue is not small, but rather is the number of studies that analyzes the causes and significance of such challenge. Such work can constitute the basis of our literature review and provided a theoretical foundation that can support our research questions (RQ1). In such case, the likelihood of misinterpreting the data and research bias is present. Nonetheless, through our analysis, we have managed to develop an entirely new research typology on the phenomenon of the developers' mindset which can serve as an exploratory study to lay the groundwork for a complete research study on this topic.

Moving on to the TAR study conducted in Chapter 5 to validate our proposed framework, we adhered to the rigorous guidelines suggested by Wieringa (2014) in the design of the study. Nevertheless, we acknowledge the possibility of unintentional research bias particularly in the data inference stage. Critically reflecting on the overall process, we were uncertain if the researcher has unintentionally elicited from the participants a desired outcome either verbally or non-verbally. We also acknowledge that providing definite results with high certainty can be challenging due to factors specific to our case such as subjectivity of humans, our inability to properly employ gamification or design an effective gamified experience, or our limited or subjective understanding of the developers' mindset problem.

In this respect, we were also limited by the type of client (graduate students) and the educational environment. Despite our efforts at getting to validate our framework with real-world agile team at a well-established company, we realized that validating GLUX for the first time in an educational setting might offered us with a more flexible, interactive, and reduced-risk environment. Accordingly, we should point out that students often have little technical knowledge and professional experience of the industry. Therefore, validating GLUX with a real-world agile team might have provided us with the possibility to generalize the results.

Overall, we accept the limitations of this doctoral research. Our research combines aspects of various disciplines ranging through gamification, the integration of agile and UX, and Lean UX. Hence, the findings of this study offer several directions for extending the knowledge base of the aforementioned domains.

6.4 Directions for Future Research

Based on the research reported in this thesis, there are various directions for future research in the two areas of gamification in software engineering, and the integration of agile and UX. In this section, we discuss some of the relevant ones.

- **Further research should be conducted to examine the various aspects of gamification in software engineering.** The need for more empirical data to provide reliable conclusions on the effectiveness of gamification has remained a critical concern for software engineering practitioners, seeing that it can be context dependent and it integrates complex aspects such as human behavior and motivational theories. Focusing on the benefits of gamification on software engineers is intuitively understandable seeing that the premise of gamification is to influence motivation and behavior. This work has underscored the complexity of gamifying software engineering processes and the benefits of employing

gamification to motivate and engage software engineers to adopt best practices. However, even though a multitude of positive experiences have been previously reported, we must not ignore the mixed results in some cases. Therefore, researchers and practitioners of this field need to establish a theoretical foundation on the best approaches to gamify software engineering processes, and more importantly we need to explore the success factors of gamifying software engineering. More empirical research is needed to obtain reliable data on the usefulness and attractiveness of particular game elements. Future research should attempt to isolate and examine the most motivating and least motivating game elements in different contexts for different types of users. In addition to examining the effects of certain gamification elements, research in the field of gamification in software engineering should investigate the aspects that precedes the gamification experience such as the characteristics and dynamics of the team or organization that can play a role in producing favorable outcomes.

In the context of this thesis, we emphasize on the need to conduct further empirical evaluation studies and investigate the effectiveness of the GLUX framework to develop reliable and generalizable conclusions. As such, future work should examine the effect of applying GLUX on facilitating and promoting the adoption of a user-centered agile development approach with different agile teams on a prolonged period of time. One way to better examine the effectiveness of GLUX can be through running two validation studies where one team would follow GLUX, and the other team would only apply Lean UX into agile without adopting GLUX. Furthermore, we are particularly interested to explore if, and why, GLUX would improve the development process for agile teams with prior unsuccessful attempts or experiences to integrate UX practices. We are also interested to examine if different customizations of GLUX's gamification strategy might produce different results. For instance, what if badges were tangible instead of virtual? Would this create a more engaging and fun experience?

- **Future research should outline and evaluate the different methods of integrating agile and UX.** The interest in exploring new ways of integrating UX into agile is evident by the number of new proposals to integrate the two areas (Curcio *et al.*, 2019). While the abundance of the different methods and approaches to integrate agile and UX may show the richness and maturity of the field, it can present a challenge to practitioners and researchers when choosing which approach to follow or explore. Part of this thesis's work involved deciding which integration approach to follow in terms of the benefits and drawbacks of each approach, the ideal contexts, and expected outcomes. The process also involved examining the different proposals of integrating agile and UX in terms of aspects that are particular to our case such as its appropriateness to gamify and its alignment with the objectives of this doctoral research. The decision process was challenging and not straightforward partially because there is a lack of comprehensive comparative studies and quality assessment of the proposed methods to integrate agile and UX. On top of the challenges reported in the literature of integrating UX practices in agile, lies the challenge of choosing an approach to achieve a successful integration. Therefore, a comprehensive and critical review that compares the different existing methods to integrate agile and UX is needed to offer guidance to practitioners and researchers to support different business scenarios, different types of software, and different teams.
- **The need to develop supportive tools to facilitate the integration of agile and UX.** While the work on integrating various UX practices and processes into agile has received considerable attention, there is still shortage in technological support in terms of assistive tools. A way to extend GLUX is by developing a supportive tool such as a Trello power-up or a Jira add-on that can offer further assistance. For instance, the gamification strategy includes using a physical or virtual space to represent the achievements board. Having to decide what tool to use to exhibit the achievements board and how it

can be represented may impose additional effort on distributed teams. Building a tool that can be integrated to the most common project management software can further facilitate a seamless integration of agile and UX.

- **Future work to improve GLUX.** The GLUX framework proposed in this thesis represents the prototype version of the envisioned artefact that can improve the integration of agile and UX. While the current version of GLUX remains a useful technique for small Scrum teams to adopt a user-centered mindset, scaling it to large scale agile frameworks such as Scaled Agile Framework (SAFe) or Large Scale Scrum (LeSS) can be useful to address a similar challenge in larger contexts and different environments. Future work should also scale GLUX to accommodate distributed agile teams, as the current implementation of the GLUX framework assumes co-located agile teams where communication and collaboration are ongoing and spontaneous.

6.5 Concluding Remarks

Gamification emerged as a promising technique for encouraging user motivation, engagement, and enjoyment in non-gaming contexts. The application of gamification in software engineering has seen a rapid growth in the academic community and the industry (Koivisto and Hamari, 2019). Even though empirical research is still limited, early results paint a positive-leaning picture of the effectiveness of gamification in software engineering (Pedreira *et al.*, 2015; García-Mireles and Morales-Trujillo, 2020). However, from analyzing the literature and from our experience in this study, we take the view that the process of gamifying software engineering is far from straightforward.

This work has explored the use of gamification to improve the integration process of agile and UX. While several challenges continue to be significant hindrance to achieve a successful integration of the two fields, our proposed framework (GLUX) provided support to address the challenge associated with agile teams focus on the delivery of rapid functional features on a frequent basis without paying equal attention to UX features and issues. When increments are released on a rapid pace such that UX issues are not treated with equal significance as functional issues, it results in UX deficiencies piling up causing a costly “UX debt”.

Our research showed that the problem can be traced down to four key issues related to the developers’ mindset towards UX (discussed in Chapter 3). Seeing that gamification shows potential in encouraging and influencing human behavior, we saw an opportunity to employ the motivational factor of gamification to improve the agile teams’ mindset and attitude towards UX and promote the integration of UX best practices into agile development processes. Following a DSR approach, we designed and validated a framework that features a gamification strategy designed to motivate and engage agile teams to integrate five lightweight Lean UX tactics into Scrum. Through a TAR study, we have demonstrated the effectiveness of the framework in encouraging agile teams to adopt UX practices within the development cycle and facilitating a fun and engaging integration experience.

The development of the GLUX framework was mainly motivated by the reported challenges of integrating UX with agile and designed to accommodate contemporary software development in which the user’s experience is prioritized as highly as rapid and frequent delivery of features. We aim with this work to help agile teams to not entirely shift their focus away from conventional metrics of progress, velocity, and efficiency, but rather achieve a balanced style of developing software in which the quality of the user’s experience is given equal consideration.

Research in the area of integrating agile and UX has a long history. Yet, it has a long way ahead before it becomes the core for software development just as agile is today. Both processes can be fully integrated when agile teams understand that integrating UX into agile requires a change not only in the processes and practices, but also in the culture and mindset of the entire team.

7 Bibliography

- Albuquerque, R. *et al.* (2016) ‘Motivating factors in agile and traditional software development methods: a comparative study’, in *Brazilian workshop on agile methods*. Springer, pp. 136–141.
- Alhammad, M. M. and Moreno, A. M. (2018a) ‘Gamification in software engineering education: A systematic mapping’, *Journal of Systems and Software*. Elsevier Inc., 141, pp. 131–150. doi: 10.1016/j.jss.2018.03.065.
- Alhammad, M. M. and Moreno, A. M. (2018b) ‘What is going on in agile gamification?’, *ACM International Conference Proceeding Series*, Part F1477. doi: 10.1145/3234152.3234161.
- Alhammad, M. M. and Moreno, A. M. (2020) ‘Challenges of gamification in software process improvement’, *Journal of Software: Evolution and Process*, (February 2019), pp. 1–13. doi: 10.1002/smr.2231.
- Allison, I. and Narciso, H. (2015) ‘Managing Change in SPI: A Framework for Addressing Resistance’, *Software Quality Professional*, 17(4), pp. 4–14.
- Anna Kaley (2018) *UX Debt: How to Identify, Prioritize, and Resolve*, Nielsen Norman Group. Available at: <https://www.nngroup.com/articles/ux-debt/> (Accessed: 4 August 2020).
- Ardito, C. *et al.* (2011) ‘Usability evaluation: a survey of software development organizations.’, in *SEKE*, pp. 282–287.
- Asghar, I. and Usman, M. (2013) ‘Motivational and de-motivational factors for software engineers: An empirical investigation’, *Proceedings - 11th International Conference on Frontiers of Information Technology, FIT 2013*. IEEE, pp. 66–71. doi: 10.1109/FIT.2013.20.
- Bak, J. O. *et al.* (2008) ‘Obstacles to usability evaluation in practice: A survey of software development organizations’, *ACM International Conference Proceeding Series*, 358, pp. 23–32. doi: 10.1145/1463160.1463164.
- Baxter, G. and Sommerville, I. (2011) ‘Socio-technical systems: From design methods to systems engineering’, *Interacting with computers*. OUP, 23(1), pp. 4–17.
- Beecham, S. *et al.* (2007) ‘Motivation in Software Engineering: A systematic literature review’, *Information and Software Technology*, 50(9–10), pp. 860–878. doi: 10.1016/j.infsof.2007.09.004.
- Beecham, S. and Noll, J. (2015) ‘What motivates software engineers working in global software development’, *Product-Focused Software Process Improvement*, pp. 193–209. doi: 10.1007/978-3-540-73460-4.
- Bertholdo, A., Kon, F. and Gerosa, M. A. (2016) ‘Agile Usability Patterns for User-Centered Design Final Stages’, In *International Conference on Human-Computer Interaction* (pp. 268–278). Springer, Cham., 9731(July), pp. 268–278. doi: 10.1007/978-3-319-39510-4.
- Bornoe, N. and Stage, J. (2014) ‘Usability engineering in the wild: How do practitioners integrate usability engineering in software development?’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8742, pp. 199–216. doi: 10.1007/978-3-662-44811-3_12.
- Brhel, M. *et al.* (2015a) ‘Exploring principles of user-centered agile software development: A literature review’, *Information and software technology*. Elsevier, 61, pp. 163–181.
- Brhel, M. *et al.* (2015b) ‘Exploring principles of user-centered agile software development: A literature

- review', *Information and Software Technology*. Elsevier B.V., 61, pp. 163–181. doi: 10.1016/j.infsof.2015.01.004.
- Brooke, J. (1996) 'SUS: a "quick and dirty" usability', *Usability evaluation in industry*. CRC press, p. 189.
- Couger, J. D. and Zawacki, R. A. (1980) *Motivating and managing computer personnel*. John Wiley & Sons, Inc.
- Curcio, K. *et al.* (2019) 'Usability in agile software development: A tertiary study', *Computer Standards and Interfaces*. Elsevier, 64(December 2018), pp. 61–77. doi: 10.1016/j.csi.2018.12.003.
- Cursino, R. *et al.* (2018) 'Gamification in requirements engineering: A systematic review', *Proceedings - 2018 International Conference on the Quality of Information and Communications Technology, QUATIC 2018*. IEEE, pp. 119–125. doi: 10.1109/QUATIC.2018.00025.
- Darin, T. *et al.* (2019) 'Integrating HCI perspective into a mobile software development team: Strategies and lessons from the field', *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2019*. doi: 10.1145/3338286.3347545.
- Deci, E. L., Koestner, R. and Ryan, R. M. (1999) 'A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation.', *Psychological bulletin*. American Psychological Association, 125(6), p. 627.
- Deng, Q. and Ji, S. (2018) 'A review of design science research in information systems: concept, process, outcome, and evaluation', *Pacific Asia Journal of the Association for Information Systems*, 10(1).
- Deterding, S. *et al.* (2011) 'From game design elements to gamefulness', in *Proceedings of the 15th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '11*. New York, New York, USA: ACM Press, p. 9. doi: 10.1145/2181037.2181040.
- Dicheva, D. *et al.* (2015) 'Gamification in Education: A Systematic Mapping Study', *Educational Technology & Society*, 18(3), pp. 75–88.
- Dreyfus, S. E. and Dreyfus, H. L. (1980) *A five-stage model of the mental activities involved in directed skill acquisition*. California Univ Berkeley Operations Research Center.
- Drumond, C. (no date) *Scrum - what it is, how it works, and why it's awesome*, Atlassian Agile Coach. Available at: <https://www.atlassian.com/agile/scrum> (Accessed: 5 August 2020).
- Engström, E. *et al.* (2020) 'How software engineering research aligns with design science: a review', *Empirical Software Engineering*. doi: 10.1007/s10664-020-09818-7.
- Facey-Shaw, L., Specht, M. and Bartley-Bryan, J. (2019) 'Digital Badges for Motivating Introductory Programmers: Qualitative Findings from Focus Groups', *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, pp. 1–7. doi: 10.1109/fie.2018.8659227.
- Feng, J. (Karolinska I. (2014) 'Developing Mobile Healthcare Apps to Meet User Needs: Integrating Lean User Experience into Agile Software Development Processes'.
- Franca, A. C. C. *et al.* (2011) 'Motivation in software engineering: a systematic review update', in *15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)*. IET, pp. 154–163. doi: 10.1049/ic.2011.0019.
- França, A. C. C. *et al.* (2011) 'Motivation in software engineering: A systematic review update', in *15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)*. IET, pp. 154–163.
- França, A. C. C. *et al.* (2014) 'Motivation in software engineering industrial practice: A cross-case analysis

of two software organisations’, *Information and Software Technology*. Elsevier B.V., 56(1), pp. 79–101. doi: 10.1016/j.infsof.2013.06.006.

França, A. C. C., Carneiro, D. E. S. and Da Silva, F. Q. B. (2012) ‘Towards an explanatory theory of motivation in software engineering: A qualitative case study of a small software company’, *Proceedings - 2012 Brazilian Symposium on Software Engineering, SBES 2012*, pp. 61–70. doi: 10.1109/SBES.2012.28.

França, A. C. C. and da Silva, F. Q. B. (2010) ‘Designing motivation strategies for software engineering teams’, pp. 84–91. doi: 10.1145/1833310.1833324.

França, C., Sharp, H. and da Silva, F. Q. B. (2014) ‘Motivated software engineers are engaged and focused, while satisfied ones are happy’, pp. 1–8. doi: 10.1145/2652524.2652545.

Franca, C., Da Silva, F. F. Q. B. and Sharp, H. (2018) ‘Motivation and Satisfaction of Software Engineers’, *IEEE Transactions on Software Engineering*, (c). doi: 10.1109/TSE.2018.2842201.

García-Mireles, G. A. and Morales-Trujillo, M. E. (2020) ‘Gamification in Software Engineering: A Tertiary Study’, *Advances in Intelligent Systems and Computing*, 1071, pp. 116–128. doi: 10.1007/978-3-030-33547-2_10.

García, F. *et al.* (2017) ‘A framework for gamification in software engineering’, *Journal of Systems and Software*, 132, pp. 21–40. doi: 10.1016/j.jss.2017.06.021.

Gothelf, J. and Seiden, J. (2016) *Lean UX: designing great products with agile teams*. ‘O’Reilly Media, Inc.’

Gunther, R., Janis, J. and Butler, S. (2001) ‘The UCD Decision Matrix: how, when, and where to sell user-centered design into the development cycle’. Retrieved.

Hall, T., Jagielska, D. and Baddoo, N. (2007) ‘Motivating developer performance to improve project outcomes in a high maturity organization’, *Software Quality Journal*, 15(4), pp. 365–381. doi: 10.1007/s11219-007-9028-1.

Hamari, J. (2019) ‘Gamification’, *The Blackwell Encyclopedia of Sociology*, pp. 1–3. doi: 10.1002/9781405165518.wbeos1321.

Herranz, E., Colomo-Palacios, R. and Seco, A. de A. (2015) ‘Gamiware: A gamification platform for software process improvement’, in *Communications in Computer and Information Science*. doi: 10.1007/978-3-319-24647-5_11.

Herzberg, F. (2017) *Motivation to work*. Routledge.

Hofacker, C. F. *et al.* (2016) ‘Gamification and mobile marketing effectiveness’, *Journal of Interactive Marketing*. Elsevier, 34, pp. 25–36.

Jeff Gothelf (2019) *Jeff Gothelf on Twitter: "I've been teaching #leanux and product discovery for a long time."*, *Twitter*. Available at: <https://twitter.com/jboogie/status/1204377633129148416> (Accessed: 5 August 2020).

Johannesson, P. and Perjons, E. (2014) *An introduction to design science, An Introduction to Design Science*. doi: 10.1007/978-3-319-10632-8.

Kahneman, D. (2011) *Thinking, fast and slow*. Macmillan.

Kashfi, P., Feldt, R. and Nilsson, A. (2019) ‘Integrating UX principles and practices into software development organizations: A case study of influencing events’, *Journal of Systems and Software*. Elsevier Inc., 154, pp. 37–58. doi: 10.1016/j.jss.2019.03.066.

- Khandelwal, S., Sripada, S. K. and Reddy, Y. R. (2017) 'Impact of Gamification on Code review process', in *Proceedings of the 10th Innovations in Software Engineering Conference on - ISEC '17*, pp. 122–126. doi: 10.1145/3021460.3021474.
- Koivisto, J. and Hamari, J. (2019) 'The rise of motivational information systems: A review of gamification research', *International Journal of Information Management*. Elsevier, 45(June 2017), pp. 191–210. doi: 10.1016/j.ijinfomgt.2018.10.013.
- Larusdottir, M., Gulliksen, J. and Cajander, ??sa (2017) 'A license to kill ??? Improving UCSD in Agile development', *Journal of Systems and Software*, 123, pp. 214–222. doi: 10.1016/j.jss.2016.01.024.
- Liechti, O., Pasquier, J. and Reis, R. (2017) 'Supporting agile teams with a test analytics platform: A case study', *Proceedings - 2017 IEEE/ACM 12th International Workshop on Automation of Software Testing, AST 2017*, pp. 9–15. doi: 10.1109/AST.2017.3.
- Liikkanen, L. A. *et al.* (2014) 'Lean UX: the next generation of user-centered agile development?', in *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. ACM, pp. 1095–1100.
- Lizano, F. *et al.* (2013) 'Is usability evaluation important: The perspective of novice software developers', in *HCI 2013 - 27th International British Computer Society Human Computer Interaction Conference: The Internet of Things*. doi: 10.14236/ewic/hci2013.40.
- Machuca-Villegas, L. and Gasca-Hurtado, G. P. (2018) 'Gamification for improving software project: Systematic mapping in project management', *Iberian Conference on Information Systems and Technologies, CISTI*. AISTI, 2018-June, pp. 1–6. doi: 10.23919/CISTI.2018.8399415.
- Magües, D. A., Castro, J. W. and Acuña, S. T. (2016) 'Usability in Agile Development: A Systematic Mapping Study', *Proceedings of the XLII Conferencia Latinoamericana de Informática (CLEI)*.
- Marques, R. *et al.* (2017) 'Gamifying software development scrum projects', *2017 9th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)*, pp. 141–144. doi: 10.1109/VS-GAMES.2017.8056584.
- Masood, Z., Hoda, R. and Blincoe, K. (2017) 'Motivation for self-assignment: Factors agile software developers consider', *Proceedings - 2017 IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2017*. IEEE, pp. 92–93. doi: 10.1109/CHASE.2017.18.
- Masood, Z., Hoda, R. and Blincoe, K. (2018) 'Adapting agile practices in university contexts', *Journal of Systems and Software*. Elsevier, 144(December 2017), pp. 501–510. doi: 10.1016/j.jss.2018.07.011.
- Melo, C. de O., Santana, C. and Kon, F. (2012) 'Developers motivation in agile teams', in *2012 38th Euromicro Conference on Software Engineering and Advanced Applications*. IEEE, pp. 376–383.
- Nielsen, J. (2012) *Usability 101: Introduction to Usability*, Nielsen Norman Group. Available at: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> (Accessed: 9 August 2020).
- De O. Melo, C., Santana, C. and Kon, F. (2012) 'Developers motivation in agile teams', *Proceedings - 38th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2012*, pp. 376–383. doi: 10.1109/SEAA.2012.45.
- Øvad, T. *et al.* (2015) 'Teaching software developers to perform UX tasks', *OzCHI 2015: Being Human - Conference Proceedings*, pp. 397–406. doi: 10.1145/2838739.2838764.
- Pedreira, O. *et al.* (2015) 'Gamification in software engineering – A systematic mapping', *Information and Software Technology*, 57, pp. 157–168. doi: 10.1016/j.infsof.2014.08.007.

- Pedreira, O. *et al.* (2020) ‘An Architecture for Software Engineering Gamification’, *Tsinghua Science and Technology*, 25(6), pp. 776–797. doi: 10.26599/TST.2020.9010004.
- Pereira, P. *et al.* (2014) ‘A review of gamification for health-related contexts’, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, Cham, pp. 742–753. doi: 10.1007/978-3-319-07626-3_70.
- Perez-Gonzalez, H. G. *et al.* (2019) ‘Investigating the Effects of Personality on Software Design in a Higher Education Setting Through an Experiment’, *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)*. IEEE, pp. 72–78. doi: 10.1109/conisof.2018.8645872.
- Regan, S. (2019) *What is modern software development? We surveyed 500 pros to find out*, Atlassian. Available at: <https://www.atlassian.com/blog/software-teams/modern-software-development-trends> (Accessed: 5 August 2020).
- Reis, E. (2011) ‘The lean startup’, *New York: Crown Business*, p. 27.
- Riemenschneider, C. K., Hardgrave, B. C. and Davis, F. D. (2002) ‘Explaining software developer acceptance of methodologies: A comparison of five theoretical models’, *IEEE Transactions on Software Engineering*, 28(12), pp. 1135–1145. doi: 10.1109/TSE.2002.1158287.
- Robertson, M. (2010) *Can’t Play, Won’t Play*. Available at: <https://kotaku.com/5686393/cant-play-wont-play> (Accessed: 11 December 2018).
- Runeson, P. *et al.* (2012) *Case Study Research in Software Engineering - Guidelines and Examples*. John Wiley & Sons.
- Salah, D., Paige, R. F. and Cairns, P. (2014) ‘A systematic literature review for agile development processes and user centred design integration’, *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE ’14*, pp. 1–10. doi: 10.1145/2601248.2601276.
- Schwaber, K. and Sutherland, J. (2017) ‘The Scrum Guide: The Definitive The Rules of the Game’, *Scrum.Org and ScrumInc*, (November), p. 19. doi: 10.1053/j.jrn.2009.08.012.
- Scrum.org (no date) *Professional Scrum with User Experience Certification*, *Scrum.org*. Available at: <https://www.scrum.org/professional-scrum-user-experience-certification> (Accessed: 21 August 2020).
- Scrum Alliance (2017) ‘State of Scrum 2017-2018’, p. 35. Available at: <https://www.scrumalliance.org>.
- Da Silva, T. S. *et al.* (2018) ‘The evolution of agile UXD’, *Information and Software Technology*. Elsevier, 102(March), pp. 1–5. doi: 10.1016/j.infsof.2018.04.008.
- Da Silva, T. S., Silveira, M. S. and Maurer, F. (2015) ‘Usability evaluation practices within agile development’, *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2015-March, pp. 5133–5142. doi: 10.1109/HICSS.2015.607.
- Sohaib, O. and Khan, K. (2010) ‘Integrating Usability Engineering and Agile Software Development: A Literature Review’, *Proceedings of International Conference on Computer Design and Applications (ICCD)*, 2(Iccda), pp. 32–38. doi: 10.1109/ICCD.2010.5540916.
- Sy, D. (2007) ‘Adapting usability investigations for agile user-centered design’, *Journal of usability studies*, 2(3), pp. 112–132. Available at: http://www.upassoc.org/upa_publications/jus/2007may/agile-ucd.pdf.
- Trochim, W. M. K. and Donnelly, J. P. (2001) ‘Research methods knowledge base’. Atomic Dog Pub.
- Twomey, E. (2017) *The Curse of Knowledge. What is the Curse of Knowledge?*, *Prototypr*. Available at: <https://blog.prototypr.io/the-curse-of-knowledge-e00593d61051> (Accessed: 25 October 2020).

- Verner, J. M. *et al.* (2014) ‘Factors that motivate software engineering teams: A four country empirical study’, *Journal of Systems and Software*. Elsevier Inc., 92(1), pp. 115–127. doi: 10.1016/j.jss.2014.01.008.
- Werbach, K. (2015) *Pointsification - Gamification in Perspective* / Coursera, University of Pennsylvania. Available at: <https://www.coursera.org/lecture/gamification/11-1-pointsification-uoKpW> (Accessed: 7 August 2020).
- Werbach, K. and Hunter, D. (2015a) *For the Win - Wharton School Press*, Wharton School Press. Available at: <https://wsp.wharton.upenn.edu/book/for-the-win/> (Accessed: 5 August 2020).
- Werbach, K. and Hunter, D. (2015b) *The gamification toolkit: dynamics, mechanics, and components for the win*. Wharton Digital Press.
- Whitworth, E. and Biddle, R. (2007) ‘Motivation and Cohesion in Agile Teams’, *Agile Processes in Software Engineering and Extreme Programming*, pp. 62–69. doi: 10.1007/978-3-540-73101-6_9.
- Wieringa, R. J. (2014) *Design science methodology: For information systems and software engineering*, *Design Science Methodology: For Information Systems and Software Engineering*. doi: 10.1007/978-3-662-43839-8.

8 Appendices

Appendix A – The Lean UX Canvas

Lean UX Canvas (v2)		Title of Initiative:	Date:
		Iteration:	
Business Problem What problem does the business have that you are trying to solve? <i>(Hint: Consider your current offerings and how they deliver value, changes in the market, delivery channels, competitive threats and customer behavior.)</i>	Solutions What can we make that will solve our business problem and meet the needs of our customers at the same time? List product, feature, or enhancement ideas here.	Business Outcomes How will you know you solved the business problem? What will you measure? <i>(Hint: What will people/users be doing differently if your solutions work? Consider metrics that indicate customer success like average order value, time on site, and retention rate.)</i>	
Users What types (i.e., personas) of users and customers should you focus on first? <i>(Hint: Who buys your product or service? Who uses it? Who configures it? Etc)</i>		User Outcomes & Benefits Why would your users seek out your product or service? What benefit would they gain from using it? What behavior change can we observe that tells us they've achieved their goal? <i>(Hint: Save money, get a promotion, spend more time with family)</i>	
Hypotheses Combine the assumptions from 2, 3, 4 & 5 into the following hypothesis statement: "We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature]." <i>(Hint: Each hypothesis should focus on one feature only.)</i>	What's the most important thing we need to learn first? For each hypothesis from Box 6, identify its riskiest assumptions. Then determine the riskiest one right now. This is the assumption that will cause the entire idea to fail if it's wrong. <i>(Hint: In the early stages of a hypothesis focus on risks to value rather than feasibility.)</i>	What's the least amount of work we need to do to learn the next most important thing? Design experiments to learn as fast as you can whether your riskiest assumption is true or false.	

(cc) BY-NC-SA Download this canvas at: www.jeffgothelf.com/blog/leanuxcanvas-v2

Lean UX Canvas (v2)

Title of initiative:

Date:

Iteration:

Business Problem

What problem does the business have that you are trying to solve?
(Hint: Consider your current offerings and how they deliver value, changes in the market, delivery channels, competitive threats and customer behavior.)

1
NOW

Users

What types (i.e., personas) of users and customers should you focus on first?
(Hint: Who buys your product or service? Who uses it? Who configures it? Etc)

3

Hypotheses

Combine the assumptions from 2, 3, 4 & 5 into the following hypothesis statement:
"We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature]."
(Hint: Each hypothesis should focus on one feature only.)

HOW WILL WE

Solutions

What can we make that will solve our business problem and meet the needs of our customers at the same time? List product, feature, or enhancement ideas here.

HOW DO WE
THINK WE
WILL GET
THERE?

Business Outcomes

How will you know you solved the business problem? What will you measure?
(Hint: What will people/users be doing differently if your solutions work? Consider metrics that indicate customer success like average order value, time on site, and retention rate.)

2

User Outcomes & Benefits

Why would your users seek out your product or service? What benefit would they gain from using it? What behavior change can we observe that tells us they've achieved their goal?
(Hint: Save money, get a promotion, spend more time with family)

4

What's the most important thing we need to learn first?

For each hypothesis from Box 6, identify its riskiest assumptions. Then determine the riskiest one right now. This is the one you need to learn first. Write it down. This is the most important thing we need to learn first.

What's the least amount of work we need to do to learn the next most important thing?

Do the minimum amount of work to learn the next most important thing. This is the least amount of work we need to do to learn the next most important thing.

HOW DO WE FIND OUT IF WE'RE RIGHT?

Download this canvas at: www.jeffgothelf.com/blog/leanuxcanvas-v2

Appendix B – Badges Collection

Hypotheses

Figure B. 1. The "Thinkers" badge.

Figure B. 2. The "Philosophers" badge.

Design Studio

Figure B. 3. The "Designers" badge.

Figure B. 4. The "Senior Designers" badge.

Minimum Viable Product (MVP)

Figure B. 5. The "Makers" badge.

Figure B. 6. The "Builders" badge.

Weekly User Experiments

Figure B. 7. The "Testers" badge.

Figure B. 8. The "Senior Testers" badge.

Appendix C – Questionnaire QNR1 and Responses

C.1. Questionnaire (QNR1)

Hola!

Welcome to the class of Agile Methods 2019! We appreciate your collaboration with us by filling this short form. This would help us to know you better and provide a proper and pleasant learning experience for you.

Thank you.

***Required**

Name (optional)

1. Age*

- ☐ 21-25
- ☐ 26-30
- ☐ 31-35
- ☐ 36-40

2. Where are you from? *

3. Previous degree: *

- ☐ BSc
- ☐ MSc
- ☐ MBA
- ☐ Other, please specify:

4. Educational background: *

- ☐ Computer Science
- ☐ Software Engineering
- ☐ Information Systems
- ☐ Information Technology
- ☐ Information Science
- ☐ Other, please specify:

5. I work...*

- ☐ Full-time
- ☐ Part-time
- ☐ Freelancer
- ☐ I don't work
- ☐ Other, please specify:

6. Years of experience working with Scrum or Agile... *

- ☐ none
- ☐ 1-2 year
- ☐ 3-5 year
- ☐ 5+ years

7. How would you rate your knowledge of Scrum? *

- ☐ No previous knowledge
- ☐ minimal
- ☐ Average
- ☐ Good
- ☐ Very good
- ☐ I'm certified

8. Years of experience working in UX... *

- ☐ none
- ☐ 1-2 year
- ☐ 3-5 year
- ☐ 5+ years

9. How would you rate your knowledge of UX? *

- ☐ No previous knowledge
- ☐ Minimal
- ☐ Average
- ☐ Good
- ☐ Very Good
- ☐ I'm certified

10. Years of experience working with Lean UX... *

- ☐ none
- ☐ 1-2 year
- ☐ 3-5 year
- ☐ 5+ years

11. How would you rate your knowledge of Lean UX? *

- ☐ No previous knowledge
- ☐ Minimal
- ☐ Average
- ☐ Good
- ☐ Very good
- ☐ I'm certified

12. What do you plan to do after you finish your master's degree? (optional)

C.2. Responses (QNR1)

ID	Age	Where are you from?	Previous Degree	Educational Background	I work...	Years of experience working with Scrum or Agile	How would you rate your knowledge of Scrum?	Years of experience working with UX	How would you rate your knowledge of UX?	Years of experience working with Lean UX	How would you rate your knowledge of Lean UX?
R1	21-25	Germany	BSc	Information Systems	Only work experience through Internships	none	No previous knowledge	none	No previous knowledge	None	No previous knowledge
R2	21-25	Vitoria, Spain	Computer engineering	Software Engineering	I don't	none	Average	none	No previous knowledge	None	No previous knowledge
R3	26-30	Colombia	BSc	Software Engineering	Full-time	none	minimal	1-2 years	Average	None	No previous knowledge
R4	21-25	France	ERASMUS	Computer Science	I do not work, student	1-2 year	minimal	1-2 years	Average	None	No previous knowledge
R5	21-25	Porto	BSc	Software Engineering	I'm a full-time student	3-5 year	Very good	3-5 years	Good	None	No previous knowledge
R6	21-25	Spain	BSc	Software Engineering	Not now, but I did two internships	none	Average	none	Average	None	No previous knowledge
R7	26-30	Fuerteventura	BSc	Computer Science	Full-time	none	minimal	none	No previous knowledge	None	No previous knowledge
R8	21-25	Seville, Spain	BSc	Materials Engineering	Full-time	1-2 year	Very good	none	Minimal	None	No previous knowledge
R9	36-40	Argentina	BSc	Software Engineering	No	5+ years	I'm certified	3-5 years	Good	None	No previous knowledge
R10	21-25	Spain	BSc	Software Engineering	Full-time	1-2 year	Average	1-2 years	Minimal	None	No previous knowledge
R11	26-30	México	BSc	Information Technology	I'm not working currently, I am studying MUSS full-time	1-2 year	Very good	none	Average	None	No previous knowledge
R12	31-35	Ecuador	BSc	Software Engineering	Freelancer	3-5 year	Average	none	Average	None	Average
R13	31-35	Perú	BSc	Systems Engineering	Freelancer	none	Average	none	Minimal	None	Minimal

Appendix D – The GLUX Guide

GLUX – A Gamified Framework to integrate Lean UX in Scrum

A guide on how to integrate Lean UX tactics into the Scrum process

Overview

Using the GLUX framework, the Scrum team will learn how to apply Lean UX tactics in Scrum in a gamified and fun way. Upon adopting these tactics, the team earn and collect points and badges, and are encouraged to carry out some challenging tasks related to Lean UX. Figure D. 1 illustrates the roadmap of how Lean UX tactics are integrated into the Scrum framework. The Lean UX tactics to be integrated in Scrum are: Hypotheses, Design Studio, Experiment Stories, MVP, and Weekly User Experiments.

Starting with brainstorming hypotheses during product backlog refinement, the team creates a list of hypotheses of their assumptions about the needs of potential users and discuss it with the product owner (PO). The team kicks off a sprint with a design studio in which the team picks a hypothesis as a theme to guide the work of the upcoming sprint/s and start sketching and discussing design ideas accordingly. Immediately after the design studio, the sprint planning meeting should take place. In a typical Scrum framework, the team decides which user stories they are going to develop during the sprint planning meeting. Here, in addition to choosing user stories during sprint planning, the team plan for the weekly user experiment in which they put into test a version of their developing product, i.e. an MVP. Those plans are captured in what is called “experiment story”. The main goal of running weekly user experiments is to validate the teams’ hypotheses and collect constant feedback to improve the product.

Figure D. 1. The integration of Lean UX tactics into Scrum.

For each tactic, the team should earn points for applying it, and additional points for doing it collaboratively where all team members are involved. For encouragement, the Scrum team is awarded with badges for carrying out Lean UX tactics for the first time as well as badges for carrying out some Lean UX challenges.

The collected points should assist the team to move up through the levels of Lean UX expertise shown in the “Lean UX achievements Board – Points” (Figure D. 2). Meanwhile, collected badges are added to the “Lean UX achievements Board – Badges” (Figure D. 3). We explain later in this guide the different types of badges and how points and badges can be earned.

Figure D. 2. Lean UX achievements board for collected points.

Figure D. 3. Lean UX achievements board for collected badges.

The GLUX guide to apply Lean UX tactics into Scrum

Tactic #1: Hypotheses

Description:

Early in the beginning of the project, during first product backlog refinement, team members should collaborate to brainstorm hypotheses. A hypothesis is a statement of what the team assumes, or thinks is true about the needs of potential users and what can add value to the business. Writing hypotheses collaboratively should play a major role in generating a common understanding among the team around the main idea and solutions of the developing product. Each hypothesis should be “tactical” and “testable”. That is, it should be specific enough that it can be validated or measured. A recommended format for writing a hypothesis is as follows:

“We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature]”

Such that:

Business outcome: what business outcome (goal) the team would like to achieve and believe as a measure of success for them. Business outcomes are the team’s definition of done.

Target users: which type of potential users you are focusing on first. Proto-personas can be helpful here!

Benefit to the user: what will the end users/customers benefit if the feature is well designed and implemented? What is the user trying to accomplish? Or what does he want to feel during and after using this feature?

Feature: which key features or services of the product (or improvement of a feature) will help in this situation to achieve the business outcome.

For example:

We believe that “sign-up rate” would increase if “busy users” are able to “save time” with “one-click sign-up using Facebook”.

When an initial list of hypotheses is created, it is prioritized and added to the product backlog. The prioritization process is based on two factors: risk and value. Assessing risk entails answering the question of how bad it would be if we were wrong about this hypothesis. While estimating value involves answering how much value this idea will generate for the customer and the business.

Just like user stories, hypotheses writing is not only limited to the beginning of the project. New hypotheses and modifications to existing ones can occur at any time during a sprint, especially after user experiments. When that happens, new or modified hypotheses should be added to the product backlog along with the rest of hypotheses and user stories.

Rules & Scoring:

For each session of writing/revising hypotheses, the team earns 5 points. Additionally, the team earns extra 10 points for each session of writing/revising hypotheses “collaboratively” where all team members are involved.

As an encouragement, the team is awarded with the “Thinkers” badge (Figure D. 4) for brainstorming hypotheses during first backlog refinement in addition to the collected points. To level up, the team is challenged to review hypotheses with the PO and stakeholders for 3 sprints in a row. If the challenge is achieved, the team is awarded with the “Philosophers” badge (Figure D. 5). All collected badges are added to the “Lean UX achievements Board – Badges” (for example, see Figure D. 6). Please refer to Figure D. 7 for the summary of this tactic.

Figure D. 4. The “Thinkers” badge.

Figure D. 5. The "Philosophers" badge.

Figure D. 6. The Lean UX Achievements Board after earning 3 badges: thinkers, designers, and makers.

Hypothesis	
What is it?	
<p>A hypothesis is a statement of what the team assumes, or thinks is true about the needs of potential users and what can add value to the business.</p> <p>A recommended format for a hypothesis is: We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature]"</p>	
Rules	
<ol style="list-style-type: none"> 1. Hypotheses should be written during product backlog refinement and reviewed every sprint. 2. Hypotheses writing/review should be done as a teamwork. 	
Scoring	
Writing or reviewing hypotheses for each sprint → 5 points!
The participation of all team members → 10 points!
Brainstorming hypotheses during first backlog refinement → The "Thinkers" badge!
Challenge	
Reviewing hypotheses with PO and stakeholders for 3 sprints in a row → The "Philosophers" badge!

Figure D. 7. A summary of "Hypothesis" rules, scoring, and challenge.

Tactic #2: UX design studio

Description:

Before the sprint planning meeting, the team should run a “UX design studio” in which they sketch and discuss design ideas for the upcoming sprint/s. The team starts by picking a hypothesis. Depending on the size and risk of this hypothesis, the team might create a theme to guide the design and development for the next set of sprints. In other words, a theme is an umbrella concept for the number of sprints required to work on a hypothesis. Once the team picked a hypothesis, a “kickoff design studio” takes place, where the team think, ideate, and sketch ideas to use them in the upcoming sprint/s. The collaboration of all team members is necessary to define the scope of the work that will be done in the following sprints and to build a shared understanding of the problem being solved and the design of the proposed solution. After the sprint review, new insights and ideas might come up. This is why the team should run a shorter version of the design studio at the beginning of every sprint to revise current designs or create new ones.

The sprint planning meeting should start immediately after the UX design studio, where the output of the design studio (e.g. sketches or Lo-Fi prototypes) is discussed as a team.

Rules & Scoring:

Like writing hypotheses, the team earns 5 points for running a design studio, and another extra 10 points for involving all team members in the process of ideation and sketching.

The “Designers” badge (Figure D. 8) is awarded to the team upon running a design studio for the first time in the first sprint.

The team is challenged in this activity to run design studio sessions for 3 sprints in a row. Once the challenge is achieved, the team earn the “Senior Designers” badge (Figure D. 9). Please refer to Figure D. 10 for the summary of this tactic.

Figure D. 8. The "Designers" badge.

Figure D. 9. The "Senior Designers" badge.

UX Design Studio	
What is it?	
UX design studio is a collaborative session where the team design together to build a shared understanding of the problem being solved, and the design of the solution.	
Rules	
<ol style="list-style-type: none"> 1. A UX design studio should be conducted at the beginning of every sprint. 2. All team members should participate in sketching and ideating. 	
Scoring	
Running a design studio each sprint → 5 points!	5
The participation of all team members → 10 points!	10
Running a design studio in the first sprint → The "Designers" badge!
Challenge	
Running a design studio for 3 sprints in a row → The "Senior Designers" badge!

Figure D. 10. A summary of "Design Studio" rules, scoring, and challenge.

Tactic #3: Experiment stories

Description:

Experiment stories are created during the sprint planning meeting. It is an artifact used to guide and plan weekly user experiments. The aim of creating experiment stories is to capture and visualize the work necessary to discover and validate user needs so that the team stays focused and constantly reminded about weekly user experiments.

Experiment stories consist of:

Hypothesis or what you're trying to learn or discover about the user: " We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature].

Tactic: what kind of light-weight technique/method will be used to test this hypothesis and get customer feedback to learn something. Simplified user testing, interviews, A/B testing, prototype, meeting with the product owner, etc.

Who will do the work: Assign the work to a team member/s.

Estimation points: how much effort and time will that take?

Rules & Scoring: Experiment stories should be added to the sprint backlog to remind team members of the weekly user experiments. The team gets 10 points for producing an experiment story for each hypothesis and adding it the sprint backlog. Please refer to Figure D. 11 for the summary of this tactic.

Experiment Story	
What is it?	
Experiment story is an artifact used to guide and plan weekly user experiments. Experiment stories consist of: <ul style="list-style-type: none">• Hypothesis or what you're trying to learn or discover about the user: " We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature].• Tactic: what kind of light-weight technique/method will be used to test this hypothesis and get customer feedback to learn something. Simplified user testing, interviews, A/B testing, prototype, meeting with the product owner, etc.• Who will do the work: Assign the work to a team member/members• Estimation points: how much effort and time will that take?	
Rules	
<ol style="list-style-type: none">1. The team should create an experiment story for each hypothesis.2. Experiment stories should indicate what method/s will be used to test the hypothesis and who will be responsible for carrying out this work.3. Experiment stories should be added to the sprint backlog.	
Scoring	
Producing an experiment story for each hypothesis → 10 points!	10

Figure D. 11. A summary of "Experiment Story" rules, scoring, and challenge.

Tactic #4: Minimum Viable Product (MVP)

Description:

An MVP is the most basic version of a product (or as we call it Version 1.0). In Lean UX, it is the smallest thing the team can build to learn something from the user and get sufficient user feedback. There may not be code, like sketching, prototyping, copywriting, and visual design. Or it can be an increment or a functionality. It all depends on what the team is trying to learn.

Depending on the team's objective, there are two types of an MVP:

- An MVP to understand or learn something about the market or potential users. At this stage, the team is not concerned to deliver a real value. For example: a landing page that doesn't function. It can be used just to get an idea of traffic or design.
- An MVP that represents a part or small version of the product built as quickly as possible to deliver value to the market. For example: a website with only the landing page but functioning properly enough to deliver the main value. In this case, the team can also learn something about the market and potential customers as well as delivering the desired value to the market.

Rules & Scoring:

The team should create an MVP before the upcoming weekly user experiment. That is, every week, an MVP should be produced, whether it is in the form of a low-fidelity prototype such as paper prototypes, or a shippable increment.

If the team manages to produce an MVP before the weekly user experiment, the team earns 5 points. If the team delivers a working MVP, they should receive 10 points in addition.

The "Makers" badge (Figure D. 12) is awarded to teams who manage to deliver a working MVP in the first sprint. The team is also challenged to deliver an MVP for 3 sprints in a row. If the challenge is achieved, the team is awarded with the "Builders" badge (Figure D. 13) as an appreciation for their continuous effort. Please refer to Figure D. 14 for the summary of this tactic.

Figure D. 12. The "Makers" badge.

Figure D. 13. The "Builders" badge

MVP	
What is it?	
An MVP is the smallest thing the team can build to learn something from the user and get sufficient user feedback.	
Rules	
1. The team should create an MVP (a working or non-working MVP) before the upcoming weekly user experiment.	
Scoring	
Delivering an MVP before the weekly user experiment → 5 points!	5
Delivering a "working" MVP before the weekly user experiment → 10 points!	10
Delivering a working MVP before the end of the first sprint → The "Makers" badge!
Challenge	
Creating an MVP for 3 sprints in a row → The "Builders" badge!

Figure D. 14. A summary of "MVP" rules, scoring, and challenge.

Tactic #5: Weekly User Experiments

Description:

Weekly user experiments are testing sessions aimed to validate hypotheses and gather customer feedback, and then use the collected data to improve and build the developing product. It should be conducted on a weekly basis to ensure that the team has continuous feedback from the customer. The methods of collecting customer feedback can vary as it depends on the tactics indicated in the experiment stories. Even though a selected team member/s is usually assigned to be responsible for conducting the user experiment process, all team members are encouraged to participate by observing, taking notes and collecting feedback from the users.

It is recommended to use lightweight user testing methods such as: heuristic evaluation, A/B testing, surveys, observation and note taking, etc.

Rules & Scoring:

The team earns 5 points for each weekly user experiment and extra 10 points for the participation of all team members. Teams are also awarded with the “Testers” badge (Figure D. 15) upon conducting a user experiment before the end of the first sprint. The team is also challenged to carry out user experiments for 3 weeks in a row and awarded with the “Senior Testers” badge (Figure D. 16) once the challenge is completed. Please refer to Figure D. 17 for the summary of this tactic.

Figure D. 15. The "Testers" badge.

Figure D. 16. The "Senior Testers" badge.

Weekly User Experiments	
What is it?	
Weekly user experiments are testing sessions aimed to validate hypotheses and gather customer feedback, and then use the collected data to improve and build the developing product.	
Rules	
<ol style="list-style-type: none"> 1. The team should conduct a user experiment every week. 2. All team members should participate; observing, taking notes and collecting feedback from the users. 	
Scoring	
Each weekly user experiment → 5 points!
The participation of all team members → 10 points!
Conducting an experiment before the end of the first sprint → The "Testers" badge!
Challenge	
Carrying out user experiments for 3 sprints in a row → The "Senior Testers" badge!

Figure D. 17. A summary of "User Experiments" rules, scoring, and challenge.

Appendix E - Questionnaire QNR2 and Responses

E.1. Questionnaire (QNR2)

Section 1 - Your experience with applying Lean UX in Scrum

For each of the following statements, mark one option that best describes your experience of applying Lean UX in Scrum.

***Required**

Name (optional)

1. How would you rate your knowledge of Lean UX after having this class? *

- ☐ Minimal
☐ Average
☐ Good
☐ Very good

2. I would be willing to apply Lean UX in my future projects. *

- ☐ Absolutely!
☐ Probably...
☐ No way!

3. Based on your answer for the previous question, please tell us the reason of your choice.

4. I view UX work as equally important as development work. *

1 2 3 4 5
Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

5. I thought that the Lean UX tactics (hypotheses, design studio, experiment stories, MVP, and user experiments) were distracting and reduced the team's agility (i.e. less working software is delivered, less flexibility, etc.) *

1 2 3 4 5
Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

Section 2 – Your experience with the GLUX guide

For each of the following statements, mark one option that best describes your experience with the GLUX guide.

***Required**

1. I found the GLUX guide easy to understand. *

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

2. In case you did not find the GLUX guide easy to understand, which parts do you think were difficult?

3. I thought the Lean UX tactics listed in the GLUX guide was easy to use in Scrum. *

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

4. I think using the GLUX guide increased my motivation towards doing UX work. *

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

5. I needed to learn a lot of new concepts before using this guide. *

This means that even AFTER reading the guide, some tactics were still not clear that I had to go and read more about the subject.

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

Section 3 - Your experience with gamification

For each of the following statements, mark one option that best describes your experience with Lean UX gamified.

***Required**

1. I think that earning and collecting "points" was encouraging and motivating to apply Lean UX tactics. *

	1	2	3	4	5	
Strongly disagree	☐	☐	☐	☐	☐	Strongly agree

2. I think that earning and collecting "points" made applying Lean UX tactics more fun to do. *

	1	2	3	4	5	
Strongly disagree	☐	☐	☐	☐	☐	Strongly agree

3. I think that earning and collecting "badges" was encouraging and motivating to apply Lean UX tactics. *

	1	2	3	4	5	
Strongly disagree	☐	☐	☐	☐	☐	Strongly agree

4. I think that earning and collecting "badges" made applying Lean UX tactics more fun to do. *

	1	2	3	4	5	
Strongly disagree	☐	☐	☐	☐	☐	Strongly agree

5. I think that the "challenges" of Lean UX tactics given in this class were engaging and fun to do. *

	1	2	3	4	5	
Strongly disagree	☐	☐	☐	☐	☐	Strongly agree

6. I think that challenges alone (without earning points or badges) would be motivating enough to apply Lean UX *

	1	2	3	4	5	
Strongly disagree	☐	☐	☐	☐	☐	Strongly agree

7. I thought that moving through levels of Lean UX expertise (practitioners, specialists, experts) provided us feedback on how well we are doing applying Lean UX tactics. *

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

8. I felt that the "Lean UX Achievements Board" helped us in gaining visibility of our progress in Lean UX *

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

9. I felt that earning points and badges "as a team" has improved our collaboration *

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

10. Which gamification element you enjoyed the most? *

- ☐ Points
- ☐ Badges
- ☐ Challenges
- ☐ Levels
- ☐ Achievements Board
- ☐ None

11. Which gamification element you enjoyed the least? *

- ☐ Points
- ☐ Badges
- ☐ Challenges
- ☐ Levels
- ☐ Achievements Board
- ☐ None

Please provide us with any comment or improvement ideas regarding your experience with the GLUX guide.

Section 4 – Your gamification experience in agile

For each of the following statements, mark one option that best describes your experience with gamification in agile.

1. I thought that the gamification part (i.e. earning points and badges, and managing the achievements board) was distracting and reduced the team's agility (i.e. Less working software is delivered) *

	1	2	3	4	5	
Strongly disagree	☐	☐	☐	☐	☐	Strongly agree

E.2. Responses (QNR2)

Section 1

ID	How would you rate your knowledge of Lean UX after having this class?	I would be willing to apply Lean UX in my future projects	Based on your answer for the previous question, please tell us the reason of your choice.	I view UX work as equally important as development work	I thought that Lean UX tactics were distracting and reduced the team's agility (i.e. less working software is delivered, less flexibility, etc)
R1	Good	Probably...	I am not the person who decide that on my company	2	5
R2	Good	Absolutely!	It is very systematic way to organise our user stories and help us to develop features on time	4	2
R3	Good	Probably...	Lean UX Rocks!!!!	4	3
R4	Good	Absolutely!	Because it makes sense to focus only on the most important and "user tested". Because I have been in projects and in product development where when the product launched np body cared and no body wanted the product. So now this makes total sense for me.	5	1
R5	Average	No way!	Difficult and tedious to apply	2	4
R6	Average	Probably...	I think it's useful when we need to know the users opinion opinion about the functionalities that we are implementing. Also is useful when the developers need to think that the users don't have the same skills as their	4	3
R7	Average	Probably...	Because i need to stick with companie's approach on agile lifecycle, but on personal projects for sure	3	3
R8	Good	Probably...	It helps us to check if what we are assuming is correct before starting to develop the project	4	4
R9	Average	Probably...	I work in a research envirotnment where the requirements and functionality are not celestes, so I think It os very difficult apply LeanUX in this envirotnment.	2	3
R10	Good	Probably...	I think Lean UX can be used only in groups with experience in Agile methodologies since, other way, it incorporate a lot of concepts at the same time for a very beginer team.	3	3
R11	Good	Probably...	Maybe it depends on application type and support from all stakeholders to use UX with Agile	3	4
R12	Good	Absolutely!	Because nowadays a designer should be part of the team that takes a valida decision to make great software	4	2
R13	Good	Probably...	Because is important to include usability design when we create software projects	5	4
R14	Average	Probably...	I think for design Lean UX can be really valuable. However, combine with Agile I feel it is a bit complicated	5	4
R15	Average	Probably...	I had no prior knowledge about Lean UX, so I'm definitely not an expert yet	4	3
R16	Average	Probably...	I find it diffcult to understand so unless you know it well, it might be dangerous to the project.	2	3
R17	Good	Absolutely!	It is very important to consider design in development process	3	2

Section 2

ID	I found the GLUX guide easy to understand.	In case you "don't" find the GLUX guide easy to understand, which parts do you think are difficult?	I thought the Lean UX tactics listed in the GLUX guide was easy to use in Scrum.	I think using the GLUX guide increased my motivation towards doing UX work.	I needed to learn a lot of new concepts before using this guide.
R1	5		4	4	2
R2	4		4	5	2
R3	3		3	3	3
R4	3	I think that adding gamification was easy to understand. But it might be difficult for other people because many are kind of dealing to understand scrum, Lean UX plus gamification might be overwhelming.	5	5	4
R5	4		2	2	3
R6	5		4	4	3
R7	3	Design Studio for me was kind of abstract, i did'n't understand at first glance which specific tasks i am suppose to do	5	4	4
R8	4		3	4	1
R9	4		4	4	2
R10	4	The experiment stories	4	4	1
R11	3	selecting Hypothesis is not easy task..	3	3	3
R12	2	Maybe a website containing the same information of the guide could be more interactive.	2	3	2
R13	4	I understood but I think is important to make more exercises to have more skills about that	4	3	4
R14	4		3	5	3
R15	3	Hypothesis part was a bit difficult to understand.	3	5	4
R16	4		3	4	4
R17	4		2	4	2

Section 3

ID	I think that earning and collecting "points" was encouraging and motivating to apply Lean UX tactics.	I think that earning and collecting "points" made applying Lean UX tactics more fun to do.	I think that earning and collecting "badges" was encouraging and motivating to apply Lean UX tactics.	I think that earning and collecting "badges" made applying Lean UX tactics more fun to do.	I think that the "challenges" of Lean UX tactics given in this class were engaging and fun to do.	I think that challenges alone (without earning points or badges) would be motivating enough to apply Lean UX	I thought that moving through levels of Lean UX expertise (practitioners, specialists, experts) provided us feedback on how well we're doing applying Lean UX tactics.	I felt that the "Lean UX Achievements Board" helped us in gaining visibility of our progress in Lean UX	I felt that earning points and badges "as a team" has improved our collaboration	Which gamification element you enjoyed the most?	Which gamification element you enjoyed the least?
R1	4	4	4	4	3	1	3	5	4	Badges	None
R2	5	5	5	5	4	2	5	3	5	Badges	None
R3	3	3	3	3	3	3	3	3	3	Achievements Board	Challenges
R4	5	5	5	5	5	1	1	5	3	Badges	None
R5	4	4	4	4	4	2	4	5	5	Badges	Levels
R6	5	4	5	5	4	2	3	4	4	Badges	None
R7	3	3	3	3	3	4	4	4	3	Challenges	None
R8	5	5	5	5	5	1	5	5	5	Achievements Board	None
R9	2	2	2	2	2	2	4	2	3	Challenges	None
R10	2	2	2	2	3	2	2	5	3	Achievements Board	Points
R11	2	3	3	3	3	3	3	3	3	Achievements Board	Challenges
R12	4	4	5	5	4	3	2	3	4	Badges	None
R13	4	4	4	4	4	3	4	4	4	Achievements Board	Points
R14	4	2	4	2	3	2	2	3	4	Badges	Levels
R15	5	5	5	5	5	3	4	5	5	Badges	Challenges
R16	4	4	4	4	4	2	4	4	4	Badges	None
R17	4	4	5	5	3	3	3	3	4	Badges	None

Section 4

ID	Please provide us with any comment or improvement ideas about your experience with the GLUX guide.
R1	
R2	I enjoyed playing the badges and point strategy but if we could have more levels for other sections of agile like planning, requirement , testing phase it would be more fun to do just like we follow waterfall model.
R3	
R4	About this question: I thought that moving through levels of Lean UX expertise (practitioners, specialists, experts) provided us feedback on how well we're doing applying Lean UX tactics. I think that the feedback provided during presentations was much more helpful about how we were improving than the levels. Because the levels May said that we did something, but doesn't not validate of we did well.
R5	
R6	There should be one or two challenges hard to complete, so if a team want to complete all of them they need to work more on that hard one. Also having more time to do this project would be more interesting, for example see how we improve: First without any gamification And then using gamification
R7	Include a feedback section, to improve it on future sprints
R8	
R9	
R10	
R11	
R12	
R13	
R14	
R15	
R16	
R17	

Section 4

ID	I thought that the gamification part (i.e. earning points and badges, and managing the achievements board) was distracting and reduced the team's agility (i.e. Less working software is delivered)
R1	1
R2	1
R3	3
R4	1
R5	2
R6	2
R7	2
R8	1
R9	3
R10	3
R11	5
R12	1
R13	4
R14	4
R15	2
R16	3
R17	1

Appendix F - Transcript of The Retrospective Class Interview

Title: Transcript for the retrospective class discussion – unstructured interview with the students

Interviewer: Manal Alhammad, PhD student.

Interviewee: Graduate students from the agile methods class in the master's program of systems and software.

Interview Setting: Interview was conducted as part of the agile methods retrospective class that was held on Friday 13th December 2019 at 15:00 for approximately 30 minutes.

Affiliation with interviewee: The PhD student has taught the class two 2-hours lectures on Lean UX as part of the teaching material of the agile methods class and has played the role of the PO for their course projects. The PhD student was also involved in every sprint review as an instructor to provide feedback and guidance on the development process.

Note: This interview has not been edited nor does it represent a perfect transcript. It does, however, provide insight into the interview topics of discussion. Long pauses, laughs, and irrelevant side notes were omitted.

(Start of Interview)

Interviewer: Ok, are you all done with the questionnaire? Ok. So, in the final report, you'll notice that some of the questions we'll ask you there are similar to the questions we asked you here. But please don't feel that you don't need to answer them, they are required. Because you will have more time and space to think and reflect about your experience. Here, you provided short answers, but we need all your thoughts, all your comments in the final report. And now for the next 15 or 20 minutes, we just want to discuss in general your experience with the course and project. What do you think was the most difficult part? I have some expectations that it was the concept of hypotheses. But I just want to have an open conversation with you. What do you think we can improve? I mean you might not be able to witness those improvements but for you colleagues next year, and for you future selves when you work in a company or startup or start your own business to apply some of the tactics you learned in this course, and by then you'll probably be able to see how things work and improve some aspects by yourselves. So, what do you think? Who wants to start?

Interviewee (student A): Um, I don't really remember but, I like how we did a lot of exercises, it was, like the concept of Lean UX and agile that I understood the most we actually did exercises about them. So when it was explained in class, we were like oh ok I understand. But then when I go back home and think about things I had to go back to the slides and say oh. And for the project part, I think that was a nice way to learn, to apply. However, I feel that merging directly agile and Lean UX together was a lot of things to learn and digest. Sprint one was so stressful. We didn't know what to do.

Interviewer: Yeah I agree and I can see how that was a lot of information to process. And it is not that you just need to process it but to also apply it, like immediately. But you shouldn't have the expectation that you should get everything right from the beginning. This is how you learn. Now you might've messed it up, but this is still good for you because in your next project you are going to be a lot better. And this is the only way to learn, to try. I mean yeah the semester is really short. If it was a two semester course, one semester theoretical lectures and then the second semester would be developing you projects.

Interviewer: we wanted to include the Lean UX part because it is like the added value you get from this course, because you can learn Scrum from anywhere. But the Lean UX part is not very straightforward. It is not easy because there's not a lot of documentation and learning material. And the integration between usability and agile is a trending theme. Sooner or later, you may have to learn about it. Also Scrum has a huge community where people discuss and argue, while Lean UX is still a relatively new thing. So you might not find this kind of rich resources.

Interviewee (Student A): It was really interesting to learn. Maybe if we had a two-week sprint it will work better. Because I was going to say that maybe on the first week, we just focus on the hypothesis and design studio and then the next week we try to do the experiment. And we only had three sprints and the sprint is only one week!

Interviewer: Yeah I agree. In fact in the Lean UX book when they discuss the integration of Lean UX into Scrum, they assume a two-week sprint. So if you have two weeks a sprint, you'll probably have more time to develop and run experiments.

Interviewee (Student A): Or maybe start the project, like, because we have only three weeks, making the first sprint really light like integrating bit by bit of the Lean UX tactics.

Interviewer: I see. Really valuable input thank you so much

Interviewer: Ok what did you think about, um, for example, the design studio? Do you think it helped you to get some kind of a shared understanding about what the application should look like? I mean in any case, without Lean UX, you would still have to do the design, but there are many ways to do it, like upfront design at the beginning within a design sprint. But do you think that breaking down the design into mini-designs in the form of a design studio at the beginning of every sprint had helped you? For example, for one sprint you do two or three design elements, and then work on developing that part, then the next sprint, you do another design element or improve one, and continue on. What are your thoughts on that?

Interviewee (student B): Yeah I thought it was useful. Because, at least from the background I have, sometimes my colleagues and I would start developing without having a clear idea of what are we going to put there or there. Sometimes it becomes a mess, and it takes longer. And when you're going to do tests with the users you're going to be like oh this is missing or this shouldn't be here. So for me I think you should have an idea about the content, the layout, and I don't know how to say this but it allows us to have more time do things that matter well. So I guess it saves us time. And when you're developing it is easier because you know, and the whole team knows, exactly what you need to do.

Interviewer: Yeah so you have the final picture in mind, or somewhere.

Interviewee (student B): Yeah it solves all the programming issues like, oh where should I put this.

Interviewer: And do you think that you as a software engineer, were interested in doing the design bit.

Interviewee (student C): Yeah I guess it is fine. But Lean UX is focusing too much on process, rather than programming. It requires a lot of collaboration, while in agile that is not the case where you can work in isolation. I mean compared to agile.

Interviewer: I don't understand what do you mean by "work in isolation in agile".

Interviewee (student C): I mean it does not require as much collaboration as in Lean UX.

Interviewee (student A): Yeah I think you mean that for example, you choose one user story and then you go and develop this user story.

Interviewer: aha you mean as the owner of this user story you can work on that alone. But with Lean UX all the tactics should be done a collaborative activity. So you didn't like this part of Lean UX? Was collaboration an issue?

Interviewee (student C): It depends. In a typical project you may have time to do collaborative activities. And it depends on the application if it needs that much attention to UX.

Interviewer: Of course. You mean some application might not be directly interacting with an end user. But I'm not sure. I feel that UX is needed nowadays in many project. The users are different. They have high demands and high expectations

Interviewee (student C): For a company if they want to adopt Lean UX, they have to be agile. That is an issue.

Interviewer: Not necessarily. Both Lean UX and agile are iterative processes. Lean UX is a design process that can viewed as a separate process. While agile is a development process. If you look at the Lean UX book, how the authors introduce Lean UX, they first introduce it as an iterative process to validate design ideas. But because the nature of Lean UX is iterative and incremental, it can fit well into agile, particularly Scrum. And that's what they do in Chapter seven; how the process of Lean UX can be integrated into Scrum using some tactics.

Interviewee (student C): But how Lean UX can be used alone? How can we validate what we designed?

Interviewer: Well, if you are only validating the design, you can do your experiments using low-fidelity prototypes. You might not need to build or develop or implement a software increment. Remember when we discussed in lecture the types of MVP. An MVP can be a working software or even a paper prototype, or even just the idea itself. So, you can validate your hypotheses without writing any piece of software, well depending on what is your hypothesis if it involves testing a functionality or the value of something. Many cheap methods can be used for experiment, like paper prototypes, mockups, wizard of OZ, fake feature, and so on.

Interviewer: Any other thoughts? Ok let's go quickly through the responses of the questionnaire to see if there are any interesting point to discuss.

(Interviewer setting up projector to present questionnaire responses)

Interviewer: Is it ok to view the responses live with you? Your names are confidential. Ok. The first questions I asked you to rate your knowledge of Lean UX at this point. Most of you rated their knowledge as good. If you did that, this mean that you should be able to apply Lean UX in your future projects without coaching or without someone telling what to do next. Did you have the same assumption as mine?

Interviewee (student D): We are comparing to our previous knowledge before this course.

Interviewer: Ok I see. For the willingness to apply Lean UX in future project, I can see the reasons you provided for your answers. Some of you mentioned a very common issue. Like: “I’m not the person who decides that at my company”. This problem is acknowledged in many areas of software engineering. Teams can struggle with senior management to apply new or innovative ways of development. The problem with Lean UX might be more complex. Because some organizations might not count the learning as progress. They want something delivered to their customers on a regular basis. But you might be able to convince your company sometimes that what you learned can actually help the business and benefit the customer. So, it takes a lot of effort to convince management.

I can see someone said that it is difficult and tedious to apply. Well I agree that it can be difficult for the first time, especially that you are still learning agile methods. So you might feel it’s too much. Would someone want to comment on that? No? Ok. Let’s move on.

Few people do not view UX work as equally important as development work. Hmm, why do you think is that? Nobody? I just want to understand your point of view.

Interviewee (student E): Because you are not delivering direct value to the user quickly, you are spending time designing and organizing your work, this kind of things. Sometimes, your boss would say: this needs to be done by tomorrow. So you have to develop fast and not waste time on this kind of thing. Just start develop what you need.

Interviewer: I see. So you cannot go to your boss at the end of the day and say: well I didn’t finish implementing the feature BUT I learned that customer might not like this feature.

Interviewee (student E): Yes in my case I would probably get fired.

Interviewee (student F): can I add something?

Interviewer: yeah sure.

Interviewee (student F): The problem with this, you are not being agile. If your boss says something like, it is because you organization is not agile. Because in this case they are not even interested in delivering value, they are just saying the work is done.

Interviewee (student E): Yes it very common. But it is very difficult to explain to your boss the benefits of agile and good UX The problem is that they think UX work is a waste of time. They need functional product. Maybe it depends on the work environment.

Interviewer: Ok I see. But what you mentioned is your manager’s point of view, not your point of view.

Interviewee (student E): Yes, but I need to do what my manager tells me. I need my salary. I need to eat.

Interviewer: Well, at least now you have the knowledge for the day when you become the manager you can understand that there are other ways to do things.

Interviewee (student G): I think it is important, but because of integrating UX, there can be a lot of changes in the development process. Anyways, at the end, based on the UX, if the user didn’t like the product, you have to do some improvements and increase the effort of development to have good UX and maybe redo a lot of work.

Interviewer: Ok I want to move to this question which I was very interested to see your responses on whether Lean UX distracted you or reduced your agility. I can see that half of you were not sure, and only somehow disagreed. I want to understand why you were hesitant or not sure? Can anyone share their thoughts? Was the question was unclear, or you're really not sure if it did affected the agility or not. Nobody? Ok what about the people who said yes, it did affect our agility. Why do you think is that?

Interviewee (student H): I think mostly because we are not really, umm, used to work in this way. We had a lot of problems understanding new concepts like defining hypotheses, and having to decide on a hypothesis, it been like no this is not correct, so we have to re-write it. So a lot of repetitive work that we had to do. So I think it mostly the problem with our own experience because we never heard about or worked with Lean UX, so we don't really now how to apply it, so of course it slowed us a little bit.

Interviewer: So you think just because it is a new way, a new process to follow, or at least different than how you worked before, that it affected your agility in terms of having frequent delivery?

Interviewee (student H): yeah yeah, that's what I mean.

Interviewer: Well yeah I think I agree because with any new way it might be difficult to adjust and adapt very quickly. And you might need some time until you get comfortable with this new way.

Interviewee (student H): Yeah, I think that to really see the benefits, or if it actually makes sense or if it is efficient to implement, I think we might need a bit more time to master it and understand it. I mean for every sprint we had, half of the sprint goes on writing a hypothesis, and then we discover it is not right we cannot validate it, and having to change it again and again.

Interviewer: I see. So you're saying that most of your time was spent on defining a hypothesis, compared to developing something. And do think this is good? Bad? What are your thoughts on this issue?

Interviewee (student H): Yeah I think ummm....

Interviewer: the work was unbalanced?

Interviewee (student H): Yeah maybe. I think the way it works in agile alone is more straightforward. It is hard to think in this type of way, a global way, linking the business value with the user value, with the feature.

Interviewer: Yeah I remember I found a similar difficulty when I was first introduced to Lean UX. I couldn't think from this different perspective. And probably most of you agree, because I noticed during your projects how you struggled to define a proper hypothesis that can be validated. So it was difficult for you to think deeply about the business and user value of each feature you want to build. Is that what you're saying?

Interviewee (student H): Yeah yeah.

Interviewer: Ok, that's good. Now we're going to go through the rest very quickly. So most of you thought the GLUX guide was ok, but there's one person who think that it was not. So we're probably going to see why. Experiment stories, selecting hypotheses were not easy. Some of suggested posting the guide in an interactive website. Like something to refer to. Mostly as I expected, the hypothesis is on the top of the

most difficult things. I can see that five of you felt pressured using the guide. Why? Where is the pressure coming from? Did you think that you have to it right?

Interviewee (student I): maybe...

Interviewer: can anyone explain please? Nobody? Ok let's move on. We're going to move the gamification part. I want to see which parts you enjoyed the most and the least. I can see most of you liked badges. In general you thought it was fun. Lots of middle answers. I want to understand your point of view. Does anyone have an input on that? No one? Ok takeaways, badges is the winner. And also achievements board. Oh that's nice. Can anyone explain in the suggestions section what do you mean by include a feedback section? I don't understand.

Interviewee (student J): I didn't write it but I think he or she mean that write down the feedback you gave us after each presentation in the sprint review.

Interviewer: Ok I see. Ok can anyone tell me if there's a challenging part about the gamification part? No one?

Interviewee (student B): Maybe it related to what I asked before, if we should evaluate the gamification techniques alone or with the Lean UX tactics.

Interviewer: aha yeah I remember you asked me that because you think Lean UX affected your agility, and gamification here is related to Lean UX, whether you should evaluated it as an isolated technique or linked to Lean UX. And I told you to evaluate it in isolation, because I want to see what you think about the gamification strategy itself.

Interviewer: Ok Thank you for your feedback. I would really appreciate it if you provide detailed answers to the discussion questions in your final report. It would be very helpful and valuable. Thank you again.

(END OF INTERVIEW)

Appendix G - The GLUX Framework v2.0

GLUX 2.0: A Gamified Framework to Integrate Lean UX Tactics into Scrum

Overview of GLUX

GLUX is a gamified framework within which Scrum teams can address the challenges of integrating UX activities into the development process to help them deliver products of the highest value to the user.

Using GLUX, the Scrum team will learn how to apply Lean UX tactics in Scrum in a gamified and fun way. Figure G. 1 illustrates the roadmap of how Lean UX tactics, along with the gamification techniques, are integrated into the Scrum framework. The Lean UX tactics to be integrated in Scrum are: Hypotheses, Design Studio, Experiment Stories, MVP, and Weekly User Experiments. Upon adopting these tactics, the team earn and collect points and badges, and are encouraged to carry out some challenging tasks related to Lean UX. The gamification techniques: rewards (points and badges), challenges, feedback (levels), and cooperation (the Lean UX achievements board), require customization based on the team's preferences and the project nature.

Figure G. 1. A general overview of the GLUX framework, integrating Scrum with Lean UX using gamification.

Starting with brainstorming hypotheses during product backlog refinement, the team creates a list of hypotheses of their assumptions about the needs of potential users and discuss it with the product owner (PO). The team kicks off a sprint with a design studio in which the team picks a hypothesis as a theme to guide the work of the upcoming sprint/s and start sketching and discussing design ideas accordingly. Immediately after the design studio, the sprint planning meeting should take place. In a typical Scrum framework, the team decides which user stories they are going to develop during the sprint planning meeting. Here, in addition to choosing user stories during sprint planning, the team plan for the weekly user experiment in which they put into test a version of their developing product, i.e. an MVP. Those plans are captured in what is called “experiment story”. The main goal of running weekly user experiments is to validate the teams’ hypotheses and collect constant feedback to improve the product.

Generally, for each tactic, the team should be rewarded for applying it, and receive additional rewards for doing it collaboratively where all team members are involved. For encouragement, the Scrum team is given special rewards for carrying out Lean UX tactics for the first time as well as for carrying out some Lean UX challenges. Challenges are set by the team based on the current difficulties and issues the team faces regarding UX. The collected rewards should assist the team to move up through levels of expertise in Lean UX and are added to the “achievements board”. We explain later how the gamification techniques rewards, challenges, feedback, and cooperation are applied to the five Lean UX tactics.

As the product backlog is the single source of requirements in Scrum, it is worth to mention how UX tasks can be managed in the product backlog. Few patterns exist, but two of them are recommended. One pattern entail creating separate PBI's for UX tasks with its own story points. Pros of such approach include dealing with UX features as increments such that each UX feature gets inspected at the sprint review. In addition, this pattern feels natural to most teams. The cons to such pattern may include the need to manage dependencies between development PBI's and UX PBI's. The second pattern involves including UX activities as tasks within relevant PBI's. Such pattern focuses on value delivery in terms of only working product is considered as increment, which can be a pro and con at the same time, depending on the team's definition of done and progress. In this case, the team's velocity is only measured by working increment, while in the former patter, any work in between is counted as part of the team's velocity. A con of the second pattern is that it may lead to high volume of “work in progress” (WIP) which spans multiple sprints as the team carry out development and UX work in individual PBI.

Below we highlight the main characteristics of GLUX:

- Simple and lightweight: The GLUX framework aims to take software engineers through a journey of applying Lean UX tactics into Scrum in fun and seamless way, progressing through challenges and levels of expertise in a rewarding environment with a sense of achievement and content. The gamification techniques incorporated are lightweight and straightforward such that it should not impose a major obstacle affecting the agility of the team. The simplicity lies within establishing a fun environment using elements such as badges and achievements to take the edge off introducing additional tasks (i.e. Lean UX tactics) to the development process.
- Team-focused: The Scrum team is the main component around which the GLUX framework was designed. The GLUX framework was designed to bring the Scrum team together by making a special emphasis on collaboration. The gamification techniques are developed on a team level and coordinated by the team itself.
- Incremental: The GLUX framework offers the possibility to incorporate the suggested gamification techniques incrementally for the purpose of introducing new concepts gradually to avoid distracting the Scrum team from the original development process.

The Scope of GLUX

Target stakeholders: since the Scrum framework does not include UX guidance for practitioners, the GLUX framework is primarily designed for small Scrum teams who are struggling with integrating UX activities into the development process, particularly in the absence of UX specialists. Even with the availability of a UX specialist, the GLUX framework can help align the mindset of the whole team towards the value of UX work.

Target projects: the GLUX framework can be effective to build user-centered software projects in highly competitive markets where UX is a determinant for a business success.

General Guidelines

- The team should have a shared understanding of the goal of using GLUX. In general, the goal of GLUX is to promote carrying out UX activities within the development process to produce high quality products. In case the team have additional specific goals, those should be shared among the entire team.
- The framework is intended to be used as an initial and temporary solution. When the team starts to consider the initiative (i.e. the gamified process) as just a game, it should be phased out to avoid a perception problem of the original purpose of employing GLUX.
- The framework does not represent a hotfix to all the challenges of agile and UX integration. It should be utilized as one technique of a whole solution.

GLUX in Practice

To apply the GLUX framework, it is important to first note that the framework is structured into two fundamental parts:

- A. **The Lean UX tactics to integrate into Scrum.** In this part, we outline in detail five Lean UX tactics and provide guidelines on how they can be integrated in a Scrum process.
- B. **The gamification strategy.** In this part, we provide a skeletal support to gamify Lean UX tactics depicted in a customizable gamification strategy.

Part A: The Lean UX tactics to integrate into Scrum

This section explains in detail the five Lean UX tactic to integrate into Scrum. As demonstrated in Figure G. 2, the Scrum team starts with brainstorming hypotheses (1) with the PO and stakeholders, then picks one hypothesis to be the focus of the upcoming sprint and use it as a theme to guide the team in the design studio (2). Then during the sprint planning meeting, the team creates experiments stories (3) to capture the work necessary to validate the current hypothesis and remind the team of the weekly user experiment (5), for which an MVP is needed as a proof of concept (4).

Figure G. 2. The five Lean UX tactics to integrate into the Scrum framework.

Tactic #1: Hypotheses

Early in the beginning of the project, during the first product backlog refinement, team members should collaborate to brainstorm hypotheses. A hypothesis is a statement of what the team assumes, or thinks is true about the needs of potential users and what can add value to the business. Writing hypotheses collaboratively should play a major role in generating a common understanding among the team around the main idea and solutions of the developing product. Each hypothesis should be “tactical” and “testable”. That is, it should be specific enough that it can be validated or measured.

Hypotheses are composed of four parts:

- Business outcome: what business goal the team would like to achieve and believe as a measure of success for them and the potential customers. Business outcomes are the team’s definition of done.
- Target users: which type of potential users the team is focusing on first. The technique “proto-personas” can be helpful in guiding this activity.
- Benefit to the user: defining the user benefit entails answering what will the end users/customers benefit if the feature is well designed and implemented. What is the user trying to accomplish? Or what does he want to feel during and after using this feature?
- Feature: which key features or services of the product (or improvement of a feature) will help to achieve the business outcome and deliver value to the user. A hypothesis should focus on one feature only. More than one hypothesis can be similar in every aspect except the feature.

A recommended format for a hypothesis is:

“We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature]”

For example:

We believe that “sign-up rate” would increase if “busy users” are able to “save time” with “one-click sign-up using Facebook”.

Just like user stories, hypotheses should be prioritized. The prioritization is based on two factors: risk and value. Assessing risk includes estimating how bad it would be if the hypothesis were invalid. While estimating value involves assessing how much value the feature will generate for the customer and the business. The list of prioritized hypotheses is added to the product backlog, so that the team picks a hypothesis to work on every iteration, a sprint, or a theme, depending on the size and risk of that hypothesis. Also, like user stories, hypotheses writing is not only limited to the beginning of the project. New hypotheses and modifications to existing ones can occur at any time during a sprint, especially after user experiments. When that happens, new or modified hypotheses should be added to the product backlog along with the rest of hypotheses and user stories. The Lean UX canvas can be used as an effective tool to assist in composing hypotheses.

Tactic #2: Design Studio

Before the sprint planning meeting, the team should run a “UX design studio” in which they sketch and discuss design ideas for the upcoming sprint/s (Figure G. 3). The team starts by picking a hypothesis. Depending on the size and risk of this hypothesis, the team might create a theme to guide the design and development for the next set of sprints. In other words, a theme is an umbrella concept for the number of sprints required to work on a hypothesis. Once the team picked a hypothesis, a “kickoff design studio” takes place, where the team think, ideate, and sketch ideas to use them in the upcoming sprint/s. The collaboration of all team members is necessary to define the scope of the work that will be done in the following sprints and to build a shared understanding of the problem being solved and the design of the proposed solution. After the sprint review, new insights and ideas might come up. This is why the team should run a shorter version of the design studio at the beginning of the consecutive sprints to revise current designs or create new ones.

The sprint planning meeting should start immediately after the UX design studio, where the output of the design studio (e.g. sketches or Lo-Fi prototypes) is discussed as a team.

Figure G. 3. The design studio is a collaborative design tactic that involves sketching and discussing the team's design ideas.

Tactic #3: Experiment stories

Experiment stories are created during the sprint planning meeting. It is an artifact used to guide and plan weekly user experiments. The aim of creating experiment stories is to capture and visualize the work necessary to discover and validate user needs so that the team stays focused and constantly reminded about weekly user experiments. Additionally, experiment stories force the conversation among team members about planning experiments with the users.

Experiment stories consist of:

- Hypothesis or what you're trying to learn or discover about the user: " We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature].
- Tactic: what kind of light-weight technique/method will be used to test this hypothesis and get customer feedback to learn something. Simplified user testing, interviews, A/B testing, prototype, meeting with the product owner, etc.
- Who will do the work: Assign the work to a team member/s.
- Estimation points: how much effort and time will that take?

See Figure G. 4 for an example experiment story.

Figure G. 4. Example of an experiment story containing the hypothesis to be validated, the validation tactic, the responsible member/s, and the estimated effort.

Tactic #4: Minimum Viable Product (MVP)

An MVP is the most basic version of a product (or as we call it Version 1.0). In Lean UX, it is the smallest thing the team can build to learn something from the user and get sufficient user feedback. There may not be code, like sketching, prototyping, copywriting, and visual design. Or it can be an increment or a functionality. It all depends on what the team is trying to learn.

Depending on the team's objective, there are two types of an MVP:

- Simple MVP to understand or learn something about the market or potential users. At this stage, the team is not concerned to deliver a real value. For example: a landing page that does not function. It can be used just to get an idea of traffic or design.
- Working MVP that represents a part or small version of the product built as quickly as possible to deliver value to the market. For example: a website with only the landing page but functioning

properly enough to deliver the main value. In this case, the team can also learn something about the market and potential customers as well as delivering the desired value to the market.

The team should create an MVP before the upcoming weekly user experiment. That is, every week, an MVP should be produced, whether it is in the form of a low-fidelity prototype such as paper prototypes (simple MVP), or a shippable increment (working MVP).

Tactic #5: Weekly User Experiments

Weekly user experiments are testing sessions aimed to validate hypotheses and gather customer feedback, and then use the collected data to improve and build the developing product. It should be conducted on a weekly basis to ensure that the team has continuous feedback from the customer. The methods of collecting customer feedback can vary as it depends on the tactics indicated in the experiment stories. Even though a selected team member/s is usually assigned to be responsible for conducting the user experiment process, all team members are encouraged to participate by observing, taking notes and collecting feedback from the users.

It is recommended to use lightweight user testing methods such as: heuristic usability evaluation, A/B testing, surveys, observation and note taking, etc.

Part B: The Gamification strategy of GLUX:

With the integration of Lean UX into Scrum, there is a learning curve. The GLUX gamification strategy aims to make this learning curve less steep with four motivational techniques: rewards, challenges, feedback, and cooperation (Werbach and Hunter, 2015b). As illustrated in Figure G. 5, the three techniques can be incorporated all at once, or gradually starting with the rewards technique, then later moving to challenges and/or levels.

Figure G. 5. The three main techniques employed in the gamification strategy of GLUX.

It is important to note that the gamification strategy in this part requires customization based on the team's preferences and the project nature. While the part on the Lean UX tactics is presented here as fixed, it is yet flexible to minor changes. The customization action for each gamification technique is summarized in Table G. 1.

Table G. 1. The customization actions required to apply the gamification strategy of GLUX 2.0

Gamification technique	Customization action	Example
Rewards	Assigning numerical values to each Lean UX activity in terms of “points”.	10 points.
	Assigning a doubled value for the team's collaboration in carrying out each Lean UX tactic.	20 points.
	Designing badges, recommended as simple as possible to reduce the effort of preparation.
Challenges	Outlining the project's or team's challenges related to UX in general, or Lean UX in particular.	“Involving the entire team in design decisions”. “Running user experiments more often”.
Feedback	Deciding if moving through levels is based on the number of collected points or the number of sprints in which the team applied Lean UX.	“We'll move through levels every 5 sprints of applying Lean UX”. “We'll move through levels every 100 collected points”.
Cooperation	Designating a physical or virtual space to represent the “Lean UX Achievements Board”.	A Google document or a white board can be used to post the team's achievements.

Below we explain the three gamification techniques and how to customize each one of them:

Rewards:

The rewards technique involves assigning values for carrying out the five Lean UX tactics and a value for carrying it out collaboratively. Whatever value is assigned for each activity, we recommend assigning a doubled value for collaboration. That is because collaboration is the core activity in Lean UX. Translating those values into gamification elements, we recommend using points and badges. Points can be represented in estimated numerical values (like story points), while the titles and design of badges are open for creativity (a collection of badges to be used as a starting point can be found in Appendix B – Badges Collection).

Accordingly, we outline below the rules for the rewards system:

Rule 1. The Scrum team earns “X” points for carrying out a Lean UX activity. For example, the team earns five points for running a design studio.

Rule 2. The Scrum team earns “Y” points (where $Y = 2X$) for carrying out a Lean UX activity collaboratively where the whole team is involved. For example, the team earns 10 points for collaborating on revising a hypothesis.

Rule 3. The Scrum team earns a badge for carrying out a Lean UX activity for “Z” sprints in a row or after collecting W points. For example, the team earns a badge for conducting user experiments for three sprints in a row.

Rule 4. In the case of incorporating the challenges technique, a badge is also given in the case of achieving a sprint’s challenge (explained below in Challenges).

Rule 5. In the case of incorporating the “levels” technique, the points collected should assist the team to move through level of expertise. For example, upon collecting 5X points, the team move up from “novice” level to “intermediate” level. (explained below in Levels).

The aforementioned rules for the rewards technique are flexible. Meaning that a team can tweak some rules based on the project objectives and team’s preferences. For example, for the tactic “create an MVP”, the team can assign a number of points for delivering an MVP before the end of the sprint, and a doubled number of points for delivering a functionally working MVP that is potentially shippable (i.e. an increment).

The collected points should assist the team to move up through the levels of Lean UX expertise shown in the “Lean UX achievements Board – Points” (Figure G. 8). Meanwhile, collected badges are added to the “Lean UX achievements Board – Badges” (Figure G. 9). We explain later the different levels and how to move through them using the collected points.

Challenges:

The challenges technique is essentially based on identifying UX related issues and difficulties faced by the team and assigning them as “sprint’s challenge”. The sprint’s challenge provides a useful technique to overcome issues collaboratively. It directs the teams focus to work on pending UX issues that may turn into UX debt. As shown in Figure G. 6, the team starts by identifying the challenging issues they have in the development process and order them from easiest to hardest. Starting with the easiest, every other sprint the team picks a challenge, sets it as the sprint’s challenge and work towards addressing it.

We recommend that challenges are not introduced until after few iterations of integrating Lean UX tactics into the development cycle. That is to acquaint the team with the new concepts and tactics before dealing with additional challenges. We also suggest that challenges should be introduced every other sprint to allow the team to have resting periods in-between sprints. In other words, if challenge A was assigned to sprint 1, no new challenge should be introduced in sprint 2. The team should earn a special badge every time a challenge is achieved to acknowledge and mark their achievements. An example of a sprint’s challenge can be performing two types of user testing in one sprint to collect more and different variety user feedback. Overall, the aim from sprint’s challenge is make the whole team goal-driven, focused, and collaborative.

Figure G. 6. The challenges technique of the GLUX framework.

Feedback:

The feedback technique aims to demonstrate how the Scrum team advances in the journey of applying Lean UX using levels of Lean UX expertise. It provides the team with a sense of progression that can consequently improve the team's motivation to carry on further in applying Lean UX. Moving through levels can be based on the number of collected points, or the number of sprints in which the team managed to apply Lean UX.

The GLUX framework suggests five initial levels that are essentially based on the "Dreyfus model of skill acquisition" (Dreyfus and Dreyfus, 1980) to represent the team's experience in integrating Lean UX in with Scrum. Demonstrated in Figure G. 7, the five levels are:

6. Novice: team members have very limited experience with Lean UX and still learning the core basics. In this case for example, for a team who encounters Lean UX for the first time, it may require them one to two sprints (or its equivalence in points, say 5X+5Y) to achieve the novice level.
7. Advanced beginner: team members have applied Lean UX practices for few sprints, and they have adequate understanding of Lean UX. For instance, it is reasonable to assume that after three or four sprints, the team is familiar with applying Lean UX tactics, even if it is not done in the best way possible.
8. Competent: team members are familiar of all the principles and tactics of Lean UX, they are comfortable applying it on other projects and are confident to answer other people's questions about Lean UX integration with Scrum.
9. Proficient: team members have deep understanding and a holistic view of Lean UX principles and tactics, and can carry out Lean UX activities in a routinely manner.

10. Expert: team members are fully capable of applying Lean UX practices intuitively without any help, can achieve excellence, and are capable to guide other teams to apply Lean UX in Scrum.

Figure G. 7. The levels technique offers five levels of competency that reflect the team's experience in integrating Lean UX with Scrum.

The levels technique, as part of the GLUX framework, is team-based. A team keeps its level of expertise through different projects. For example, if a team reached the advanced beginner level in project A, then the team starts from the advanced beginner level in project B. The title of each level is open for customization too, as long as the idea of levels is maintained.

Cooperation (Achievements Board):

The collaboration of the entire team is a core principle in the Lean UX process. This is why GLUX offers the “Lean UX achievements board” as a complementary technique to promote cooperation among the team members. The Lean UX achievements board call for having a space, physical or virtual, where the collected points, badges, and levels of the team are posted. It is important to provide visibility of the team’s achievements and progress in applying Lean UX throughout the development process. Just like all the components of the GLUX framework, the achievements board is team-based; where the whole team shares one board to mark and celebrate their achievements. We call it: “the Lean UX achievements board”. While achievements boards are not restricted to a certain design, we provide examples of achievements boards for displaying points in Figure G. 8 and for displaying badges in Figure G. 9.

Figure G. 8. Achievements board is a useful technique to provide visibility of the team's collected points.

Figure G. 9. Achievements board can also be used to showcase the team's collected badges.