Agile UX

Unifying design and engineering for optimal product development workflows

Johan Pricam

Author: Johan Pricam
Title of the thesis: Agile UX: Unifying design and engineering for optimal product development workflows.
Department: Design
Degree programme: International Design Business Management
Year: 2022
Number of pages: 141
Language: English

Keywords: Product development, UX design, Software engineering, Agile software development, Agile UX, Digital products

Aalto University
School of Arts, Design
and Architecture

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Master of Arts thesis abstract

Acknowledgements

I am expressing my sincer gratitude and appreciation to:

- Elina Hilden and Antti Salovaara for their supervision, guidance, and precious advices along the way
- The authors and content creators from the literature and Web sources, for providing me with solid bases to formulating the direction of this research
- The 24 research respondents for allocating their time and sharing their valuable insights
- My family and friends for their support, patience, and encouragements all along

Warmest thanks to all of you!

Abstract

Digital products, ranging from mobile to desktop applications, from platforms to websites, have become extremely prominent in our everyday life. Their role in tackling real-life challenges is huge, and their place in the modern industry is central. Actually, the rise of digital products is only at its outset, and will be exponential in the next coming years and decades. In the conceptualization and developement of digital products come two major disciplines, among others, into play; UX design and software engineering. In order to deliver great quality products, addressing people's needs, fulfilling businesses' objectives, while providing users with a delightful overall experience, optimal work and collaboration between UX designers and software engineers need to happen in the background. Nonetheless, issues and challenges between the design and engineering ways of working are many, regardless of the context. Agile UX, a modern and rising approach to product development, is seen as a potential solution to alleviate these challenges. In a nutshell, Agile UX aims at unifying Agiles software development (ASD), the dominant software engineering approach and philosophy, and UX design (UX) processes and practices, in order to optimise workflows between designers and developers.
This thesis reveals exisiting gaps with the Agile UX approach and current Agile UX frameworks, preventing them to truly be actionable in the industry, and therefore to be a practical solution for reconciling design and engineering in product development. These gaps are adressed here through an empirical study looking at understanding the wide array of contexts of the industry and their specificities.

This thesis research is a two phases study of the Agile UX modern approach to digital product development. The first phase assembles extensive literature and Web content reviews in order to investigate and map the challenges between design and engineering, and understanding to what extent is Agile UX a solution to alleviating those challenges. The main findings of this first phase highlight the aforementioned gaps of the Agile UX approach. Agile UX is indeed a solution to reconciling UX design and software enginnering, but it lacks of clear guidelines and attention to contextuality, preventing it to be trully actionable. Agile UX as it is defined today also has very little considerations on the evolution of the UX design and software engineering disciplines, and fails at anticipating the future of digital products.
Based on these insights, the second phase of the thesis brings up qualitative insights from industry practitioners, including UX designers, software engineers, managers and leaders, in order to address the limitations of the Agile UX approach. This second phase encapsulates the main objective of this thesis; understanding how to contextually adopt an Agile UX approach to product development.

The findings of this thesis highlight the diversity of the industry and the uniqueness of each product development context. The design-engineering relationship varies a lot across organisations, and many contextual facets impact on the ways processes and collaboration practices are defined. Common trends have been highlighted, as well as numerous divergence points. Clear signs show that Agile UX is spreading in the industry already, but quite inequally. At the same time, Agile UX seems to align with the future trends of product development. In addition to the extensive findings, this research concludes with the draft proposition of a methodology. This could be the first building step towards a more developed solution, aiming at supporting teams wishing to optimising their product development workflows, throught Agile UX.

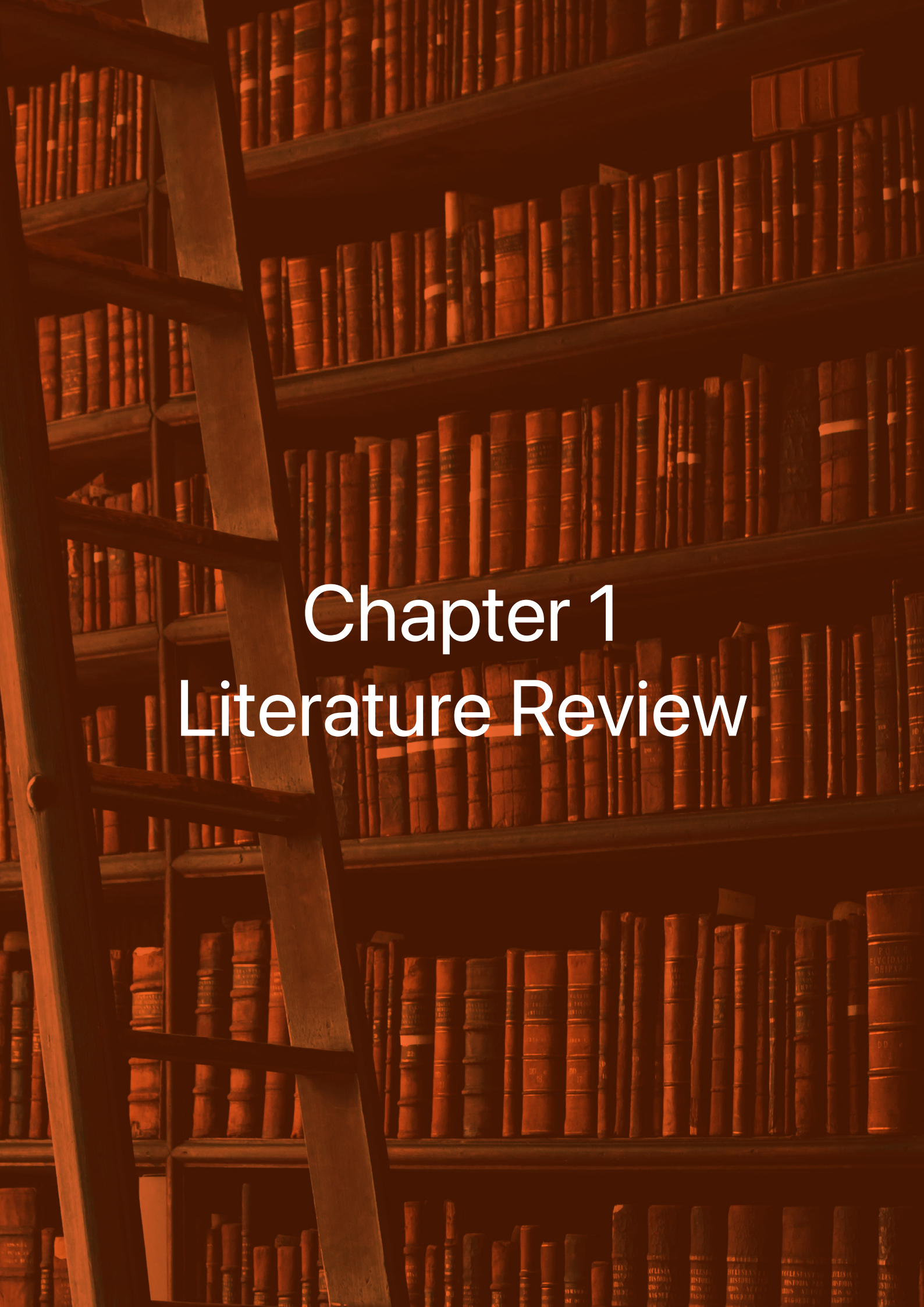
Table of contents

Introduction

Nowadays, the importance of digital products in businesses is huge and their role in tackling real-life challenges is major. A digital product is refered here as any software product, mobile or web application, platform or website. Digital product development has risen during the 90's, following the democratization of the personal computer. At this time, the need to create useful and usable digital products started to be felt. Creating successful products goes through defined processes, including various activities and skills. Two major disciplines come into play here: Software engineering (SE) and user-experience design (UX).
Software engineering is a branch of computer science embedding the development and building of application software and computer systems software. Agile software development (ASD) is an effective approach and philosophy widely used nowadays in SE. It is described as a lightweight model that was proposed to overcome the traditional development methods' limitations (Alsaqqa et al. 2020). ASD has appeared in the early 2000's following a need from SE practitioners to build products faster and more efficiently, which ultimately resulted in the publication of the agile manifesto in 2001 (Beck et al. 2001).
User-experience (UX) has been coined for the first time in 1995 by Don Norman, and refers to the overall experience of functionalities and qualities of a product (Kashfi et al. 2014). The UX discipline has evolved over the last two decades and is today practiced in all types of digital organisation. UX is today commonly used and understood as intrinsic to digital product development, and is highly inspired by the human-centered design philosophy and principles (HCD) (designkit.org), (ISO 9241-210). Applying HCD in product development aims at satisfying users via producing usable and understandable products that meet their needs and interests (Salah et al 2014). Althought appeared before ASD, it took time for UX to be recognized and adopted by the industry. UX is today acknowledged as being an essential part of product development, as much as the SE discipline (Da Silva et al 2018).

Combining design and engineering has proven to lead to better software, useful, usable and satisfying to their end-users, then successful for businesses. Concerns and proposals of methods for reconciling design and engineering already emerged in the early 2000's (Göransson et al. 2003). Those were nevertheless very scarce initiatives. There has been an increased interest on the topic over the last decade, particularly on the UX and Agile integration, as those two disciplines have become the standards. Lots of theoretical and practical studies have been conducted. Agile user-experience design (Agile UX) seems to be a well understood way of emcompassing this trend of integration (Hartson and Pyla 2019), (Da Silva et al. 2018). Agile UX has emerged in response to the conflicts between designers and engineers, and challenges in blending UX with ASD. Those challenges are numerous and vary across contexts.

The first part of this thesis, the background research, includes two buliding blocks, a literature and Web reviews. The Literature review (chapter 1) gathers insights from 28 sources, and aims at providing a first dip in the topic. More precisely, the review looks at the design-engineering sources of challenges, and the nature of the Agile UX approach, being the claimed solution to these. The Web review (chapter 2) gathers insights from 75 sources, and aims at complementing the literature review. The Web review presents a more accurate framing of the challenges, and an extensive understanding of the Agile UX approach. The ultimate takeaways of the background research are uncovered gaps of the Agile UX approach. These gaps frame the problematization of the thesis, addressed in the second part (Chapters 3 and 4) through an empirical study. The main research question, driving the empirical study, is the following: *How might an Agile UX approach to product development be contextually applied, today, and tomorrow?*



Chapter 1
Literature Review

Introduction

This first chapter presents a literature review on user-experience design (UX), Agile software development (ASD), and Agile user-experience design (Agile UX). It constitutes the first part of the background research, and establishes the base and direction for this thesis research.

This literature review aims at depicting the challenges in combining ASD and UX processes in software projects as well as to understanding the emergence of Agile UX in response to those. An overview of this modern trend is provided, and some examples of Agile UX models and frameworks from the selected literature are listed. The end of the review highlights the gaps of the current Agile UX trend and provides insights for further research directions.

The first section presents the methodology of this literature review. The second section gives an overview of both the ASD and UX disciplines and explains their complementarity. The third section presents the challenges faced when combining ASD and UX. The fourth section gives an understanding of the Agile UX approach. The fifth section highlights the current gaps of Agile UX. The summary wraps-up the review at the very end.

1. Methodology

1.1 Selection process

The original idea of direction for this thesis and literature review was to focus on the practical collaboration between developers and designers in software projects. This motivation had emerged from the author's own experience of being a UX designer in a small startup, and the challenges he had encountered working with developers. After some preliminary research and readings around the topic, the author decided to refine this direction idea to the topic of Agile software development and user-experience design integration, providing a more holistic view and understanding on product development processes.

69 relevant sources were first pre-selected using the following keywords: Agile software development, UX design, Software engineering and design, Agile UX, Agile and UX integration, Lean UX, Agile processes, UX processes, UX design and software engineering, UX design and software development, Collaboration developers and designers, Design Thinking and Agile processes, UX design and Scrum. Google Scholar was the search engine used to collect the sources. Google search was also utilized to search some standards websites (e.g. (agilemanifesto.org) or (ISO 9241-210)).The pre-selected sources came from a wide array of digital libraries. Those were AIS electronic library, Springer link, ACM digital library, Sage Journals, Science Direct, Research Gate, IEEE.org, IGI Global, Taylor and Francis online, beei.org, books.google.com, scienceopen.org, scirp.org, Morgan and Claypool publishers as well as different university library platforms.

The final selection for this literature review includes 28 sources. Those were further investigated, red and documented by the author. The final selection includes books, journal articles, conference papers, and standard sources. The research types among the selected sources vary from literature review, empirical and action research, and a mix of them. The contribution types among the selected sources include general standards, high-level models, practical frameworks, guidelines, principles and insighful discussions. This final selection was made relying mostly on the pre-reading of the titles and abstracts of the first selection. The dates of publication of the selected sources range from 2001 to 2020, with exactly 75% of them being 2015 or later, and only 2 of them being early 2000's (2003 and 2001). The detail is: 5*2020, 6*2019, 3*2018, 3*2016, 4*2015, 3*2014, 1*2013, 1*2012, 1*2003 and 1*2001.

The two main criterias of selection were a high focus on ASD and UX integration, as well as being published after 2015. Some exceptions have been made when more focus on either ASD or UX was needed, or when the source was seen as a major contribution to the topic and highly relevant to select. Being peer-reviewed was another important criteria for selection.

1.2 Limitations

The author recognizes some limitations to this literature review. Firstly, biases might have affected the selection process in the extent of self interpretating or understanding the titles and abstracts of the sources. Due to a willingness to provide a condensed summary of the key trends on the topic, and considering the fields of ASD and UX as very rich, a lot of content selection and prioritization was done. Some other important contributions might have been discarded or missed. Google Scholar being the sole search engine utilized to collect the sources might have also reinforced this scheme. Finally, author's biases might have affected the documentation, analysis, and interpretation of the sources themselves.

In addition to the aforementioned limitations, the author's young experience in undertaking academic research and writting needs to be considered.

The author is aware of those limitations and embraces them, as it is an inevitable part of any research advancement and knowledge construction. All risks inherent to those limitations suspected to affect the quality of the research were considered seriously and mitigated as much as possible.

2. UX and ASD

2.1. Agile software development (ASD)

Agile software development (ASD) is the current dominant approach in the software engineering (SE) field and aims at delivering software quickly.

The term "agile" literally refers to the rapid and flexible nature of this approach. Agile methods and frameworks include interconnected practices for improving software teams' productivity, efficiency, and responsiveness (Sedano et al. 2020). Crontradicting some misunderstandings about the field, Agile is neither a framework, process nor a methodology. Agile is a set of practices, values and principles (Alsaqqa et al. 2020). ASD aims at providing different methods than the traditional plan-based development model and focuses mainly on adapting to change and delivering high-quality products through light and simple work-processes (Da Silva et al 2018). Another aim of ASD is to enhance the value of the delivered software to satisfying the customer's requirements (Schwartz 2013). ASD is defined as a lightweight approach that was proposed to reduce the overhead and cost and tackle the limitations of the traditional development methods, such as the waterfall model, which are characterized as heavyweight approaches (Alsaqqa et al. 2020). This is what has triggered the formulation of the agile manifesto in 2001 by the Agile alliance (Beck et al. 2001). Twelve principles and four values compose this manifesto and represent the major reference and standard for ASD.
The four values of the agile manifesto are: Individuals and interactions over processes and tools; Working software over comprehensive documentation; Customer collaboration over contract negotiation; Responding to change over following a plan.

ASD has been characterized as being a compromise between absence of process and excessive process (Salah et al 2014). Agile processes have risen attention mostly due to their openness and flexibility towards changed requirements, design ideas, and value contribution, to give some examples (Bruun et al. 2018). They are based on iterative, incremental and adaptative development and aim at delivering working applications as fast as possible (Navarro et al. 2016). ASD embraces change by reducing costs througout the entire process. Reducing costs is done by prioritizing work and tasks based on the business value, following small incremental iterations and short feedback cycles, cooperating and communicating, delivering software early and continously. Efficiency in ASD is ensured by flat management structure, close exchanges with the customer and the use of self-organizing and cross-functional teams (Kuusinen 2015). Agile processes also claims to marginalizing unnecessary bureaucracy and optimizing the management process so that no time is wasted and efforts are mainly put on the core objectives.

Among the many, Scrum is the most utilized and known Agile framework nowadays. It is especially characterized by its high focus on the product and ability to satisfy the business needs all along the development process (Khaled Al Ghanmi and Shahida Mohd Jamail 2020). Scrum focuses on project management for projects and situations hard to be planned in advance. It introduces feedback loops, self-organized teams, and 1-4 week sprints as its core elements (Dobrigkeit et al. 2019). It gives particular attention on how the team should interact for the work to be accomplished effectively and efficiently, which makes it the typical Agile method (Braz et al. 2019). Some other examples of Agile frameworks are Test-driven development method (TDD), Feature-driven development method (FDD), Extreme programming method (XP), Dynamic system development model method (DSDM) and crystal methods (Alsaqqa et al. 2020). Kanban is also another popular and widely adopted Agile method nowadays.

ASD is often described as fitting better and being more popular among smaller organisations and teams (Liikkanen et al. 2014). Nevertheless, leading organisations in the tech industry, such as IBM or Fusion5, have firmly adopted this approach too (Patterson and Erturk 2015).

2.2. User-experience design (UX)

User-experience design is a discipline and approach of the field of human-computer interaction (HCI).

User-experience is defined as the perceptions and responses of users that result from their experience of using of product (Salah et al 2014), or in other words, the overall experience of functionalities and qualities of a product (Kashfi et al. 2014). UX is an umbrella term that can refer to all types of products, services and systems, since they automatically induce a user-experience (Patterson and Erturk 2015). Nonetheless, UX design is nowadays mainly understood as intrisic to creating digital products, thus being strongly tied to the software engineering (SE) field. Depending on the context, UX design can also be called product design, digital product design, or UX/UI (user interface) design. Those terms are similar in their meaning. UX is also highly connected to human-centered design (HCD) or Design Thinking (DT), which are approaches that support the values and principles of the discipline, and often are guiding the UX process itself.

The overall goal of UX is to ensure the usefulness, usability and quality of the user-experience of the software (or product) by proposing a design and development process driven by user-centeredness (Navarro et al. 2016). UX considers all users' emotions, feelings, preferences, beliefs, perceptions, responses, behaviors and accomplishments that occur before, during and after the use of the software or product (Patterson and Erturk 2015). Software's UX aspects are divided into task-related (pragmatic) and non-task related (hedonic). UX practitioners (UX designers, or UX specialists) are devided between research and design, which are the two main categories of UX tasks. The UX research part aims at understanding problems and aligning the project vision with end-users' needs. The UX design part aims at proposing and designing solutions to the framed problems.

UX has been coined for the first time in 1995 by Don Norman, which is a pioneer in the discipline. Since then, the field has evolved, and is today practiced in all types of team setting, organisation and across a varitey of domains. Although appeared before the emergence of ASD, and whereas the SE field and ASD methods have more rapidly and easily been implemented in the industry, UX has taken more time to be understood and adopted widely. The journey is not even totally achieved yet. It is nevertheless recognized as essential in creating successful products, and the fact that a good UX can be a significant competitive advantage in the market for a software is acknowledged and has been proven (Kuusinen 2016). UX design seems nowadays to have reached the state of standard practice in the software industry (Ananjeva et al. 2020), and to have matured into a confident discipline (Hartson and Pyla 2019).

As said the UX design process is highly based on the HCD approach. HCD is characterized as the dominant and best approach for designing great products (Sedano et al. 2020). HCD is defined as a set of practical techniques, methods or processes, as well as a hight level philosophy that places the user as the core focus of the software design and development process (Salah et al 2014). The goal of applying HCD in software projects is to attempt to satisfy users via producing usable and understandable products that meet their needs and interests (Salah et al 2014). Following an HCD approach induces eminent success of the developed software due to users being at the core of the design (Patterson and Erturk 2015). The website designkit.org created by the leading design firm IDEO is a renowned reference for HCD. It depicts the approach in three main phases: Inspiration, ideation and implementation. Another HCD reference is the ISO standard (ISO 9241-210).

Design Thinking (DT) is another approach often used as guiding reference for UX design. DT differs from HCD by being more detailed and more often considered as a practical framework, as opposed to HCD being more of a high-level philosophy. They are otherwise very aligned with their values and principles. Design Thinking promotes the understanding of users' needs (desirable) considering what is technically (feasible) and economically (viable) possible (Pereira and Russo 2018). The Design Thinking process includes five distinct phases: Empathise, define, ideate, prototype, and test.

2.3. The complementarity of UX and ASD

As seen in the previous sections, UX and ASD are nowadays the two core processes and set of practices in software projects. Indeed, over the last two decades, they have become standards in the software industry (Ananjeva et al. 2020), and have changed the way software is built and designed (Jones and Thoma 2019). Both disciplines fundamentally aim to build quality software (Ferreira 2012) as well as to contribute to the economic success in highly competitive markets (Brhel et al. 2015). UX and ASD are complementary practices in software projects since they each bring unique benefits to the team table, and utilized together, enhance the process and outcome of the software development effort (Ferreira 2012). Da Silva et al 2018 clearly state that UX and ASD processes should be managed and controled together, and Braz et al. 2019 mention Design Thinking as being compatible and sharing common objectives with some of the Agile principles.

As Khaled Al Ghanmi and Shahida Mohd Jamail 2020 put it: "UX design and Agile development are often needed together as the business need to delight the users and to ship the product faster." On one hand, ASD aims at ensuring the rapidity, flexibility, velocity and adaptability of the development process (Ananjeva et al. 2020), (Brhel et al. 2015). On the other hand, UX aims at placing the end-users' goals and needs at the core of the process in order to ensure usability and user satisfaction (Brhel et al. 2015). UX provides software teams with principles and practices determining what to build, while ASD provides principles and practices determining how to build it efficiently (Sedano et al. 2020).

As mentioned earlier, a good UX is nowadays described as a competitive advantage in the market as well as an important component of software success (Kuusinen 2016), (Kashfi et al. 2014). Consequently, it is more and more common to see companies that aim at better adressing their users' needs trying to adopt UX design and make it part of their development processes (Khaled Al Ghanmi and Shahida Mohd Jamail 2020), which often leads to combining UX and ASD in practice. Gothelf and Seiden 2016 add: "if you are in the software business, you are probably in the UX business". To complement ASD with UX also results in a more accurate understanding of the problem from the software team, which allows the design of better aligned solutions (Signoretti et al. 2019).

In a recent review exclusively focused on ASD (Alsaqqa et al. 2020), technical Agile roles explicitely describe UX related tasks, as chief architect, a role that aims at defining software's information architecture, a typicall UX task; or ambassador user, a role that aims to represent the users during the process, also very UX related. This article does not explicitly evoke the need for pure UX role in the team, such as UX designer or UX specialist. This goes into contradiction with (Kuusinen 2016) who states that "designing for user interaction and UX requires special skills that are beyond the skills that software developers are expected to possess. Thus, a role of a UX specialist is often needed to ensure fluent user flow and desired UX from use of the software under development."

ASD alone seems to lack an awareness of usability issues (Ferreira 2012) as well as lack of practices to properly involving end-users' along the development process. On the other side, UX alone would be obsolet for transforming designs into working products.

UX and ASD are complementary practices for creating quality software, but neither of those disciplines provides clear guidance on how they should fit with one another. While looking great together on paper, their combination faces a lot of challenges in practice. Those challenges are depicted in the next section.

3. Challenges in combining UX and ASD

As detailed in the previous section, UX and ASD are complementary disciplines that need to come in pair in order to creating successful digital products. Nevertheless, the selected literature highlights challenges in combining those two approaches.

After reviewing the 28 selected sources, the author proposes a taxonomy of the challenges that have been depicted in the literature. 14 main challenges have been mapped. All those challenges are interconnected to each other and can easily be influenced one by another. Some of the sub-challenges detailed bellow can possibly fall into different boxes. For the sake of clarity, and considering the difficulty of this mapping exercice, the author has avoided duplications.

1) UX resources

A first challenge encountered when combining UX and ASD is the lack of resources put into practicing UX (Kuusinen 2015). This can be a lack of expertise, due to a low number of UX practitioners, if at all. Another scheme can be a lack of consideration or recognition for design in the organisation, or a shortage in time dedicated to its practice.

Consequences of this challenge can lead to UX designers having to spend time advocating their discipline instead of focusing on the core of their practice (Bruun et al. 2018). Not enough time or human resources may hinder the benefits of UX on the resulted software, then arming its success and quality (Salah et al 2014). A lack of human resource in UX can also lead to UX designers having to take care of many tasks at the same time, impacting on the quality of their work.

2) Guidance

This challenge refers to the lack of guidance in how UX should be practiced within the specific context of a project, and how to integrate it in the given Agile environment (Kuusinen 2016). Combining UX and ASD might require changes in the project management structure, to make compromises in the practices and tools employed, as well as learnings and un-learnings from both disciplines. At the end, combining UX and ASD is very much a transformation process in itself (Ananjeva et al. 2020), which can be challenging to go through arbitrarily.

3) Awareness/Understanding

A lack of awareness and understanding of UX design by engineers and other roles has proven to arm the design process, then the ability for designers to impact on the project (Khaled Al Ghanmi and Shahida Mohd Jamail 2020). This can be due to a lack of clear explaination of the UX process and its activities at the forefront. Design understood solely as aesthetical advantage is also an issue that falls into this boxe.

The lack of awareness and understanding can also be related to designers knowing too little about the Agile philosophy, which makes the communication with developers harder, and impacts on the quality of the collaboration (Patterson and Erturk 2015).

4) Users/Stakeholders involvement

Users involvement is reported as being a major challenge when combining UX and ASD as both disciplines have their own set of techniques to do so, which contradict each other; e.g. small unit testing in Agile VS qualitative usability testing in UX. Furthermore, it is one of the core aspect of UX, whereas it is seen as secondary and resource consuming from the ASD standpoint (Brhel et al. 2015). Getting regular access to potential users all along the development cycle is not always easy (Salah et al 2014). Poor end-users involvment has been proven to seriously hinder the quality of a product.

5) Coordination/Timing

Another big challenge refers to the ability of the UX and ASD processes to synchronize with each other (Kuusinen 2016), in other words, how to coordinate and time the respective activities so that they can be applied optimally, while respecting both Agile and UX values (Schwartz 2013). A good example to illustrate this is the debate on how much up-front design time should be allocated before the development (Ananjeva et al. 2020), or when to schedule usability testings so that the results can be efficiently validated and implemented.

6) Mindset

The perspectives in UX and ASD are radically differing with one another. UX designers focus on the users and designing a useful and usable software, while developers focus on the technology and developing the software quickly and efficiently (Sedano et al. 2020). This philosophy gap has been reported has a big challenge in the software creation process.

7) Global vision

Maintaining a common ground between the designers and developers has been reported as challenging (Khaled Al Ghanmi and Shahida Mohd Jamail 2020), (Ananjeva et al. 2020). This refers to the struggle in aligning the vision of the project and understanding the problem the software aims to tackle. Designers and developers may have a different understanding of the solution they are building, which can generate confusion in the process.

8) Context and setting

The particular software project setting can be challenging for combining UX and ASD practices. This refers to the company culture, values and management traditions, which may not provide an optimal environment or team organisational structure for mixing those two disciplines (Ferreira 2012). The hierarchy system of an organisation might influence a lot the ability of developers or designers to perform their tasks properly, consequently, the ability for UX or ASD to align with their respective principles (Jones and Thoma 2019).

9) Physical seperation

Another reported challenge is the physical location of the design team and development team which are often seperated (Kuusinen 2016), (Jones and Thoma 2019). This can generate a need for more documentation to support the communication between the two teams, which is time and resource consuming. A physical seperation also creates the overall feeling of a desunified team which does not share common responsibilities and ownership of the project. UX designers seen as outsiders is another likely consequence of this challenge (Kuusinen 2015).

10) Decision-making

The decision-making process has been described as very challenging to share between designers and developers (Jones and Thoma 2019). Taking shared decisions often requires trades-off in the management process as well as aditional organisational efforts. In the case of decision-making being disjoined or unfunded, trust issues or communication issues between the two disciplines have been observed (Ananjeva et al. 2020).

11) Practices/Activities

UX and ASD induce different set of practices and activities. A reported challenge has been the unique or contradictory nature of practices and activities from one field with the principles of the other (Jones and Thoma 2019). As an example, iterative (thrown-away) UX prototyping goes into contradiction with the evolutive working software development valued by the Agile philosophy (Sedano et al. 2020). Practices can be incompatible in their very nature. UX activities are also reported to be hard to prioritize based on the ASD activities in place (Salah et al 2014).

12) Nature of the processes

The very nature of the UX and ASD processes are fundamentally different. Agile processes are characterized as evolutive, incremental and time-boxed, which contradicts with the continuous and iterative nature of the UX process. ASD approach as been designed by developers for developers, and does not consider the design phases at all. Agile is not an optimal environment for applying design practices. As an example, the up-front design activities, valued by UX, are very hard to perform within a faced-paced and chunked environment (Schwartz 2013), (Ananjeva et al. 2020). Resulting from this challenge are clashes between the processes, which can compromise the time and efficiency of the team (Khaled Al Ghanmi and Shahida Mohd Jamail 2020).

13) Language

This challenge refers to the differing terminologies employed between the UX and SE field. The frameworks and guidelines developed in UX have generated new terminologies while the SE field has kept its own specific terms. This language gap can impact on the ability to communicate and understand each other from one discipline to another (Kashfi et al. 2014).

14) Communication

Communication has been reported to be a challenge in merging UX and ASD. This refers to designers and developers dynamic being broken, misinterpretations of the information or lack of documentation in the process (Salah et al 2014). The use of different tools for designing and developing (Navarro et al. 2016) as well as different team collaboration tools highly influence the nature of this communication aspect. Resulting from communication issues between developers and designers can be risks to formulate different understandings on the project (Kuusinen 2015) or to generate trust issues, among many others.

The above 14 challenges in combining UX and ASD, will be re-visited and reformulated in the next chapter (Chapter 2: Web review). With additional insights from the Web, complementing those from the literature, the author will narrow down the challenges to six main patterns.

4. Agile user-experience design (Agile UX): Integrating UX and ASD

4.1. Agile UX overview

As detailed, UX and ASD put together, generate an important additional value to product development, while challenges in combining those two disciplines are many. This paradigm is the driving motivation for researchers and practitioners to focus on the topic of UX and ASD integration (Salah et al.2014). A term to define this modern tendency is Agile user-experience design (Agile UX).

In short, Agile UX aims at mixing values and practices from both UX and ASD approaches (Navarro et al. 2016).

The author insists here on the difference between combining UX and ASD, which only consists of putting the two approaches together without deep considerations on what efforts are involved in practice, and integrating UX and ASD (Agile UX), which is a rising trend and research topic that aims at understanding the challenges of combining them, and proposing solutions, methods, guidelines and principles to overcome them, leading to new approaches. Integration implies that UX and ASD are recognized as distinct but mutually dependent practices and disciplines (Ananjeva et al. 2020).

Concerns and proposals of methods for integrating UX into SE (when ASD was not too popular) were already existing in the early 2000's (Göransson et al. 2003). Those were nevertheless very scarce. The increased interest has especially sparked during the last decade, where lots of theoretical and practical research, studies of processes, challenges and success factors have been conducted (Ananjeva et al. 2020).

Other terms have been employed to refer to the concept of Agile UX, as user-centered Agile software development (UCASD) (Brhel et al. 2015). Lean UX (Gothelf and Seiden 2016) also has a lot in common with Agile UX, with the difference of being tailored for innovation and business creation types of project, as it integrates the Lean Startup and ASD approaches. In any case, the common idea of reconciling ASD and UX disciplines with the proposal of an integrated process is seen.

Agile UX is depicted in three levels. The practices and process level, which is intrinsic to the activities of designing and developing products and how they are arranged; the people and social level, which refers to the team collaboration and human interactions; the technology level, which refers to the tools, produced artifacts and outcomes, including the released software (Da Silva et al. 2018), (Brhel et al. 2015). Those three levels have not evolved simultaneously. The practices and process level is the most advanced, meaning that the UX and ASD integration through the angle of the activities and their flow is acknowledged and adopted. The people and social level is crossing the line nowadays, which tells that the integration journey from the angle of team collaboration and human relations is processing. The technology level is the least advanced, meaning that integration through the angle of the produced artifacts is still to be done (Da Silva et al. 2018).

The importance of the Agile UX approach is today acknowledged, while its practical definition is in progress. Applying Agile UX seems to generate products better targetted to the end-users' needs, and more usable, while making the development and collaboration process smoothier (Pereira and Russo 2018). In the future, Da Silva et al 2018 think that Agile UX will be at the core of software development culture and a standard followed by software teams. Both UX and ASD processes will be fully integrated, and infrastructures will be available for a seemless adoption of this approach.

4.2. Agile UX values

Following the review of the literature, the author classifies and details six Agile UX values. These values aim at providing a synthesis of the essence of the Agile UX approach, and allow to understand that Agile UX is a relevant solution to alleviating the challenges between UX and ASD detailed in the previous section. Indeed, these values seem to directly respond to the challenges, at least on paper. These values are:

- Continuous communication
- Active collaboration
- Mutual awareness
- A sense of one unified and cross-funtional team
- End-users as actively part of the process
- Transformation guidance and facilitation

Agile UX advocates *continuous communication* between developers and designers (Navarro et al. 2016), (Ananjeva et al. 2020). Live prototyping, inducing working on the same tool and sotware artifact is a proposed example to ensure this aspect (Navarro et al. 2016). Artifact-mediated communication has been depicted as providing huge benefits in software creation collaboration (Brhel et al. 2015).

Active collaboration brings designers and developers together, each following and being involved in both disciplines and engaging each other for their inputs (Ferreira 2012). This aspect induces coordination and synchronization of efforts between UX and ASD (Salah et al 2014). The debate between design being ahead of development or evolving in parallel did not advance a leading answer yet. The ahead-iteration model has its supporters (Bruun et al. 2018) as much as the within-iteration one has its own (Kuusinen 2016), (Brhel et al. 2015). Regardless of which configuration is prefered, active collaboration induces shared ideation, problem-solving, and co-creation (Jones and Thoma 2019).

Another value is *mutual awareness* from both UX and ASD disciplines, ensured by shared knowledge across the team, training developers on UX tasks as well as extending designers' range skill to the SE field (Khaled Al Ghanmi and Shahida Mohd Jamail 2020), (Patterson and Erturk 2015). A solid understanding of each other's discipline can help both developers and designers to adjust their practices and techniques in the sake of better communicating with each other (Da Silva et al 2018). The idea of mutual awareness also induces taking joint decisions all along the process (Jones and Thoma 2019).

A sense of a unified and cross-functional team is claimed as being beneficial for applying Agile UX (Navarro et al. 2016). Unifying the team helps in building common vision and shared understanding of the project. A sens of unified team can be created by co-locating designers and developers (Jones and Thoma 2019) as well as to balance the voice and power resources equally between the disciplines.

Another Agile UX value is to make *the end-users actively part of the development process*, allowing early and constant feedback to inform the project decisions. This allows to keep the focus of the development aligned with what the final UX should be (Patterson and Erturk 2015).

The last value is to *guide and facilitate the transformation* induced by the adoption of Agile UX. This goes through considering the required organisational shifts, the eventual new roles to assign, practices to adjust, new tools to adopt, and other efforts to make to successfully implementing Agile UX (Signoretti et al. 2019). Clear integration guidelines are considered crucial to successfully adopt Agile UX (Da Silva et al 2018). The transformation can be supported by practical toolkits and being facilitated by dedicated roles.

The above six Agile UX values will be re-visited and reformulated in the next chapter (Chapter 2: Web review). With additional insights from the Web, complementing those from the literature, the author will detail the understading of Agile UX values and principle, and will present eight Agile UX cores.

4.3. Some Agile UX models and frameworks

Various models and frameworks aligned with Agile UX have been introduced in recent years. The author lists here ten examples from the literature selection. These can differ from their level of practical details, scope, and the context they fit for. Some are high-level models (e.g. Hartson and Pyla 2019) while others are more focused on one aspect of the Agile UX process (e.g. Navarro et al. 2016). Whether it is HCD with ASD, Scrum with Design Thinking, or other combinations, they all aim at reconciling UX design and ASD.

Agile UX (Hartson and Pyla 2019)
This model aims to be practical rather than formal and theoretical. An entire book is dedicated to it "The UX book: Agile UX design for a quality user experience". The book aims at providing guidance for fitting UX into the ASD standard approach. It depicts processes, methods and techniques in the sake of understanding what a good UX is and how to achieve it.

Lean UX (Gothelf and Seiden 2016)
This model is inpired by the Lean Startup and ASD approaches. The book "Lean UX: Designing great products with Agile teams" depicts it in details. Lean UX aims at fitting the UX design workflow in the fast-paced and time-boxed Agile environment, while encouraging close collaboration of all members of the team. Lean UX is structured by the loop model "build, measure and learn". This approach works especially for startups and innovative projects (Liikkanen et al. 2014), as the concept of minimum viable product (MVP) is at its core. It is worth mentioning that Lean UX and Agile UX are different but have many overlaps. Adopting one of them often involves the other.

SIBAP - A Script-Based Prototyping Framework to Boost Agile-UX Developments (Navarro et al. 2016)
This framework values prototyping, relying on scripting languages to code behaviors, as a common channel for communication between designers and developers. This allows to enhance the Agile UX collaboration process.

BoB (Best of both worlds) - A Framework for Organizing Within-Iteration UX Work in Agile Development (Kuusinen 2016)
This framework aims at integrating UX work into ASD while adopting the within-iteration model (design and development practiced during the same iteration). This method also advocates cross-functional team setting as being an optimal environment for Agile UX.

Dual-Track Development (Sedano et al. 2020)
This framework reconciles HCD with ASD. It depicts two parallel linear tracks, design and development, evolving at the same pace. One takes care of keeping the software's vision aligned with users' needs (product design), the other one focuses on programming and delivering (development).

UX-aware model of requirements, UX-aware process model: Models for Integrating UX into Software Engineering Practice (Kashfi et al. 2014)
Those two frameworks aim at rasing awareness and knowledge about UX, enhancing communication regarding UX and encouraging UX-aware decision-making. They present the relation between UX, functional and quality requirements, and SE concepts necessary to cover UX. UX concepts are integrated in alignement with existing SE terminologies and practices.

InnoDev: A Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup (Dobrigkeit et al. 2019)
This framework integrates UX and ASD and is inspired by the Design Thinking, Scrum and Lean Startup methods. The goal of InnoDev is to make softwares more user-driven. ASD remains the overall management approach, while Design Thinking is spread all along the process, and the cooperation designers and developers is increased. It claims to fit well for both innovative or on-going projects.

Design Thinking and Scrum in Software Requirements Elicitation (Braz et al. 2019)
This framework focuses on software requirements elicitation and unifies Design Thinking and Scrum methods. It aims at better identifying needs, opportunities and project barriers for a better end-users' satisfaction and average completeness with the delivered software.

Integrating Scrum development process with UX design flow (Khaled Al Ghanmi and Shahida Mohd Jamail 2020)
This framework aims to aid organisations in integarting UX into their development processes. It manages to adress problems of time and quality in design and development. The workflow is characterized by two sprints evolving one after another (design sprint and development sprint). Designers and developers are encouraged to be actively involved into both sprints and to extend their range skill and knowledge towards the other discipline.

Boosting Agile by Using User-Centered Design and Lean Startup (Signoretti et al. 2019)
This framework brings together ASD, HCD and Lean Startup. It provides complete and detailed recommendations for organisations to support the transformation induced by this integration, including changes in roles, techniques and tools. Dedicating a transformation team as well as providing a practical toolkit are the two major strategies. The method also claims cross-functional teams, composed of three roles, as the optimal structure.

5. The current gaps of Agile UX

Althought Agile UX seems to tackle and to provide a response to the challenges mentioned in section 3., there are limitations to be highlighted. As said, Agile UX is a current topic of research, and many aspects still need to be deeper investigated and understood.

The author presents six main Agile UX gaps, synthesizing the limitations depicted in the literature:

1) The context and situatedness nature of software projects: each context is unique
Emphasis has been put on the influencial nature of the given project context in Agile UX integration (Liikkanen et al. 2014). The context, or situatedness, refers to the project settings, the organisational culture, values and management styles. The developed frameworks seem to focus too much on the Agile UX processes and tools aspect, whereas the context and human aspects are secondary (Ferreira 2012). It has been claimed that formulating principles based on the manifold contextual factors is hard, as they are different and unique for each and every project (Brhel et al. 2015).

2) The rigidity of the proposed frameworks
Each Agile UX framework, including some of those presented in 4., have been formulated based on specific practical or theorethical researches. They are then contextualized, meaning they would fit well in the context they have been developed in, where they could be obsolet in others. Some methods also lack of wider empirical validations, especially when it comes to scalability (Kashfi et al. 2014), (Dobrigkeit et al. 2019). Considering this, flexibility should also be a core value for adopting Agile UX frameworks, as no perfect recipe exists, nor a single recipe can fit for every context (Alsaqqa et al. 2020). Defining integrated UX and ASD processes does not induce one single way to go (Schwartz 2013), and multiple alternatives can possibly yield the same results. The relevance of generalizing frameworks and methods can be questioned (Ananjeva et al. 2020).

3) A demanding adoption
The process of adopting Agile UX has been described as a transformation process. Transformation processes are not especially arming, but in the case of Agile UX, it has often been reported to be demanding and hard to maintain on the long-term (Ananjeva et al. 2020). Some Agile UX frameworks induce big organisational shifts, trainings in the team, or a need for transformation coaches (Liikkanen et al. 2014). This transformation can lead to big time and resource investments at first, which can be daunting for organisations. Furthermore, the literature focuses on large-scale transformation solutions rather than starting with smaller details and aspects of the integration (Ananjeva et al. 2020).

4) No framework that satisfies all ASD or UX principles
It has been demonstrated that not any single framework perfectly balances both ASD and UX principles (Schwartz 2013). Compromises always have to be made either one or both sides (Ananjeva et al. 2020). The debate on the iteration-ahead VS little up-front design approach is a sign that finding the right balance between UX and ASD principles is not straighforward.

5) A room for improvement in practice

Althought Agile UX methods seem to yield positive results, improvements remain to be done. The user involvement aspect needs to operate a shift from end-users being represented in the project to end-users fully taking part of the process (Da Silva et al. 2018), which some Agile UX methods tend to ignore. Synchronizing the testing methods also remains a challenge (Da Silva et al 2018). Many frameworks lack of details in terms of day-to-day activities, project flow and roles (Ferreira 2012), (Pereira and Russo 2018). Another point is that UX is often perceived through the lence of roles, while it has been argued that it would be better understood as a culture, discipline and set of tasks (Da Silva et al. 2018).

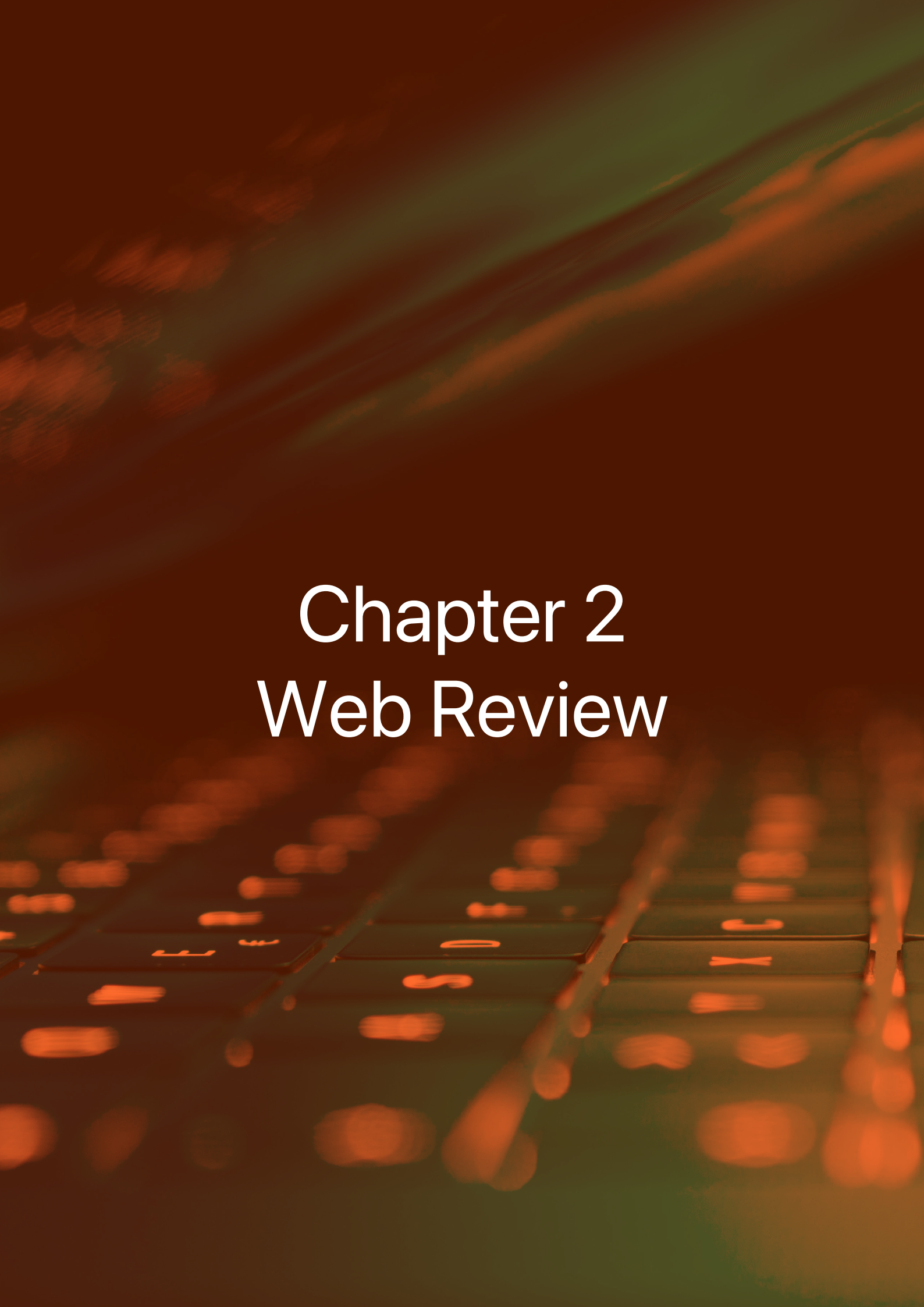
6) A lack of guidance and understanding from the technology dimension

The technology dimension of Agile UX is the least understood and advanced, compare to the practices and process, and the people and social dimensions (Da Silva et al. 2018), (Brhel et al. 2015). A lack of guidance focused especially on artifacts used along the process is felt. An example is the problem of prototypes thrown-away; a model for reusing them, which would save time a resources, is lacking (Navarro et al. 2016).

The above six Agile UX gaps will be re-visited and reformulated in the next chapter (Chapter 2: Web review). With additional insights from the Web, complementing those from the literature, the author will further detail the current limitations of the Agile UX approach in two main gaps. These two main gaps will be the base of the problematization driving the second phase of this thesis, the empirical study (Chapter 3 and 4).

Summary

This literature review has provided an overview of both user-experience design (UX) and Agile software development (ASD), being the two standard approaches in product development nowadays. Their complementarity and respective benefits have been highlighted. Challenges in combining UX and ASD are many. The author has reported 14 main challenges from the selected literature sources. Agile user-experience design (Agile UX) is a current research topic and trend that aims at integrating UX and ASD in order to alleviate those challenges, and enhance product development workflows. The review has given an overview of this modern approach, depicted six Agile UX values, and given ten examples of Agile UX frameworks and models. Finally, six Agile UX gaps have been formulated, in order to picture the current limitations of this approach.



Chapter 2
Web Review

Introduction

Following the literature review (Chapter 1), having laid down the base of the research, this second chapter presents a Web review. It constitutes the second part of the background research. Insights from the Web aim here at complementing and extending the academic and scientific foundations laid by the literature review. The open-source, rich, and valuable knowledge of the Internet allows to deepen and sharpen the perspective on UX design, Agile software development, and Agile UX, as well as ensuring to formulating an accurate problem statement to be taken further for the upcoming empirical study (Chapter 3 and 4).

In the above literature review, investigations have been made on three main aspects: First, the sources of challenges in combining design and engineering, and UX and Agile models, then, the Agile UX approach in response to those, and finally, identifying the current limitations in this evolving approach. This Web review aims at taking these conclusions further, by looking at the wide array of open-source contents in the Internet. Three types of Web sources have been considered: Blog posts, podcasts and Youtube videos. By selecting and analysing 75 Web sources in total, the author has been able to reformulate the insights from the literature review and deepen their level of detail. Undertaking this Web review has been motivated by the huge amount of relevant information on both UX design and software engineering's fields available online, as well as the presence of wide communities and experienced practitioners. The knowledge of the Web on design and engineering is today very recognized and trusted. Digital practitioners, the author included, rely on it for inspiring their daily practice, as well as staying up-to-date with the trends of their discipline.

The first section of the review presents the methodology, including the selection process and the limitations. The second section depicts the reformulated sources of challenges between UX and ASD. The third section goes through the details of the Agile UX model, including an accurate definition, the presentation of the eight Agile UX cores, a depict of some practices and trends supporting Agile UX, as well as a short clarification about the differences and overlaps between Agile UX and Lean UX. The fourth section reformulates the Agile UX gaps and details them. The summary wraps-up the review at the very end, and outlines the research direction for the next part of this thesis research.

1. Methodology

1.1. Selection process

The research direction for this Web review has been driven by the insights of the literature review. Then, the content selection process mostly targeted sources treating about the collaboration between designers and developers and the Agile UX topics. Very few selected sources focus only on UX or ASD, since these aspects have been investigated enough in the literature review.

75 Web sources have been selected and analysed by the author. Out of those 75 sources, 22 are blog posts or web articles, 23 are podcasts, and 30 are Youtube videos. Three distinct selection processes have been undertaken for each of these categories.

Regarding the blog posts and web articles, Google has been the search engine utilized. The search keywords used to find the sources were: Agile UX, Lean UX, Agile user-centered design, UX and Agile approaches. The sources come from various websites and online platforms: nngroup.com (5), interaction-design.org (4), uxmatters.com (3), medium.com (2), coderchronicles.org (1), thedesignsquiggle.com (1), mckinsey.com (1), toptal.com (1), justinmind (1), uxdesign.cc (1), withintent.com (1), xd.adobe.com (1). The dates of publication of the sources range from 2002 to 2021 (1*2002, 1*2012, 1*2015, 2*2016, 3*2017, 4*2018, 3*2019, 3*2020, 2*2021 and 2 not being dated).

Concerning the podcasts, they have all been found on Spotify. The search keywords used to target the sources were: Agile UX, Lean UX, Agile user-centered design, UX and Agile approaches, collaboration designers and developers. Some sources have been found without keywords, directly by browsing through podcast channels. The sources come from various podcast channels on the topic of UX design and software engineering: UI Breakfast (3), User Defenders (2), Tech Done Right (2), UI Narrative (2), Design Systems Podcasts (2), Bloc Thinking (1), Conversations with Hans Van Dam (1), Delta CX (1), Design Better Podcast (1), Design Details (1), Design Huddle (1), Design Life (1), Developer Tea (1), Honest UX Talks (1), UX Podcast (1), Village Global's Venture Stories (1), What is Wrong with UX (1). The dates of publication of the sources range from 2015 to 2021 (1*2015, 1*2016, 2*2017, 3*2018, 4*2019, 9*2020, 3*2021).

Regarding the Youtube videos, the search keywords used to find the sources were the same as for the podcasts: Agile UX, Lean UX, Agile user-centered design, UX and Agile approaches, collaboration designers and developers. The sources come from various Youtube channels on the topic of UX design and software engineering: NNGroup (11), Jesse Showalter (2), Micke Locke (2), Career Foundry (1), Google Developers (1), Google Chrome Developers (1), Aj&Smart (1), Bent Taylor (1), Design Gal (1), Codeacademy (1), Flux (1), React Conferences by GitNation (1), blondiebytes (1), ChariotSolutions (1), DeltaCX (1), Designlab (1), Malewicz (1), vaexperience (1). The dates of publication of the sources range from 2016 to 2020 (1*2016, 7*2017, 7*2018, 7*2019, 6*2020).

The main criteria of selection for any type of source was a high focus on the topics of Agile UX and the collaboration designers-developers. The author prioritized sources being published no longer than five years ago (2016), even though some exceptions were made when particular relevance was perceived in regard of the topic. Unlike for academic and scientific content, there is no peer-review like model to evaluate the quality and trustability of Web content. To deal with it, multiple factors for sizing the reliability of the sources were looked out by the author: The number and proportion of likes (for Youtube videos), the number of followers and/or popularity of the channel/platform in its field (for all type of sources), the profile and experience of the author(s) or people involved in the source content creation (for all type of sources).

1.2. Limitations

The author recognizes some limitations to this Web review. Firstly, biases might have affected the selection process in the extent of self interpretating or understanding the titles and descriptions of the Web sources, as well as partially relying on already known Youtube channels, podcast channels, and online platforms. Furthermore, prioritization needed to be done due to time and resource constraints, meaning that some other relevant sources might have been discarded. Google search being the only tool utilized to find plog posts, Spotify being the only platform utilized to find podcasts and Youtube being the only platform utilized to find videos might have also limited the scope of the selection. Finally, author's biases might have affected the documentation, analysis, and interpretation of the sources themselves.

In addition to the aforementioned limitations, the author's young experience in undertaking academic research and writting needs to be considered.

The author is aware of those limitations and embraces them, as it is an inevitable part of any research advancement and knowledge construction. All risks inherent to those limitations suspected to affect the quality of the research were considered seriously and mitigated as much as possible.

2. Reformulation UX and ASD combination challenges

The first aim of this Web review is to complementing the understanding of the challenges in combining UX and ASD depicted in the literature review, as well as to re-visiting and reformulating these challenges.

2.1. Conclusion of the literature review

In the literature review (Chapter 1) the author depicted 14 main challenges:

1) UX resources
The lack of resources put into practicing UX design in a given organisation.
2) Guidance
The lack of guidance in how UX should be practiced and how to integrate it in the given Agile environment.
3) Awareness/Understanding
The lack of mutual understanding between designers and developers.
4) Users/Stakeholders involvement
The difficulty of involving end-users throughout the entire process.
5) Coordination/Timing
The difficulty for the UX and ASD processes and activities to synchronize with each other.
6) Mindset
Dealing with the radically differing perspectives of UX and ASD.
7) Global vision
The difficulty of sharing and maintaining a big picture between designers and developers.
8) Context and setting
Organisations' culture, management traditions, team structure, and project settings not being optimal to integrate UX and ASD.
9) Physical seperation
Dealing with a physical separation between the designers and developers, often within two different teams.
10) Decision-making
The difficulty of taking shared and transparent decisions throughout the process.
11) Practices/Activities
Dealing with the unique or contradictory nature of practices and activities from one field with the principles of the other.
12) Nature of the processes
Dealing with the difference of nature between UX and ASD processes.
13) Language
Dealing with the differing terminologies employed between the UX and SE field.
14) Communication
The difficulty of maintaining a continuous and clear communication between designers and developers.

2.2. Remapping the challenges

Reviewing the 75 Web sources allowed the author to better understanding challenges in combining UX and ASD. The proposed remapping narrows down the challenges more logically and details them more accurately. Most of the challenges from the literature review have been reconsidered, but some broken down or merged with each other. The challenge 2) Guidance is the only one which was removed from the list. After reflection, this aspect appears to be the consequence of other challenges, such as lack of UX support and resources, more than a challenge in itself.

The author clarifies here the term *challenge* as being the potential source of emergence of conflicts between designers and developers and/or the trigger of patterns suspected to harm the efficiency of the process, and the quality of the resulting software product.

Six challenges in combining UX and ASD are listed and detailed here. These challenges are highly interconnected and likely to influence one another in any given project:

- 1) Lack of UX support and resources
- 2) Mutual misunderstandings
- 3) Unshared vision and differing mindsets
- 4) Inadequate settings and context
- 5) Clashing processes and practices
- 6) Broken communication and decision-making

1) Lack of UX support and resources

As already concluded in the previous chapter, a lack of UX support and resources, or a lack of design buy-in from organisations' executives, is a major source of challenges in combining UX and ASD.
This is directly tied to the aspect of design maturity within the organisation [23]. A low level of design maturity, as for example considering design at the look and feel level only, reveals to be a real pain for designers to deal with. It is no doubt that this is a very common pattern. As a study about DesignOps (the design operation practices within an organisation) shows, companies only do 22% of the recommended DesignOps efforts, do not have DesignOps-dedicated roles, and have a low DesignOps maturity overall [3].

Under such environments, designers are likely to end-up with a lack of time, or the leftover time, to practice their discipline [38], and in worst cases, crucial UX activities being overlooked [2], as for example up-front or continuous user research being cut off in spite of a sole validation phase at the end of the project [9], [18], [43]. Designers in such situations need to deal with pressure, short deadlines, and a very fusy idea of what the project big picture is. Furthermore, they don't have any priority on the workflow [34], [43], [44]. A lack of resource can also lead them having to work on too many projects, and to carry too many responsibilities [13], [16], [32], [38]. This can lead designers to be disconnected to one given project, and not being able to perform their job optimally [4], [70]. Overall, this pattern reduces the scope of their work to a minimum, with emphasis on making the workflow efficient [9]. With this spirit, designers under Agile environments can be seen as slowing down the process [41].

This issue of UX support and resources in software product development is the direct consequence of the Agile model only adressing engineering. Indeed, Agile has generated misconceptions on what UX design is and its importance [37], which has influenced how organisations perceive it. The Agile manifesto has not included the UX discipline at all [2], [15], [55], [37], [38], and it is organisations and product teams' task to include it blindly [54]. In Agile teams, no one usually takes into consideration the UX time and resources needed to deliver high-quality, user-oriented products [9]. What is even more problematic is that Agile often takes over the whole product development process [38]. Indeed, the SAFe model, which is a scaled version of ASD, only puts design into one box, where engineering takes 90% of the process [38]. More recent books on Scrum suggest to train non-UX roles to do UX designers' job, which shows how little awareness and consideration about the discipline Agile has [17]. More than ASD itself, the fact that some companies have rigidly and blindly implemented it, with more concerns about following the trend than creating great products, makes it even harder to be flexible and create space for UX [23].

As a direct consequence of this challenge, it is sometimes part of UX designers' job to educate people within their team and organisation why design matters; in other terms, advocating for UX [63], [25], [50]. This takes time on their schedule, and funny enough, it might not be the best way to make managers buy-in, since educating is a separate discipline in itself, and surely not in every designer's range of skills. Furthermore, it's not a lightweight of a task to convince people about the value of design when they don't think they need it [37].

2) Mutual misunderstandings

Mutual misunderstandings from one another is a second major source of challenges between the UX and SE fields [16].
This aspect partially joins the lack of UX support and resources, with a focus on the specific designer-developer duo. From one side, developers can struggle to properly understand what UX is [23], [64]. In some cases, they might even ignore, underestimate, and take control of it [38]. Similarly to the issue of design maturity withing organisations, it happens that developers only understand design as limited to graphic, visual and UI design [50]. This, again, creates a need for designers to advocate for their discipline, and explain to developers the methods and principles of UX [63], [25], [50].
From the other side, designers can be unaware of how developers work, and what working within Agile environments induces for them. They usually understand developer's roles, which is not especially reciprocal, but often miss aspects of feasibility in software development. Designers might ignore that software engineering processes are a lot of thinking, and not only concist in executing their design and code [33]. Without input from developers and knowledge on engineering possibilities in a given project, it often happens that designers create designs that cannot be technically implemented under the time constraint [71]. Such scenarios can lead them to perceive their peers as idea breakers, always pushing to more simple alternatives [40]. Moreover, designers within Agile environments often feel that some aspects are vague, as for example the project's big picture and problem solved [44].
Misconceptions in both designers and developers's mind from one another's field can lead to serious problems such as bad assumptions, lack of trust, lack empathy and willingness to collaborate [32], [40].

This mutual misunderstanding issue is also strongly linked to the language aspect, which can reinforce this scheme. Indeed, designers and developers talk differently [23], [40], [47]. UX and SE have developed different terminologies, which can lead to jargon barriers if one lacks of knowledge on the other field [23]. For example, the recent term design system is still confusing for most of the people; it can even be interpreted differently within the design field itself [23].

When it comes to developers not understanding design properly, there is no doubt that ASD has played a role in it. As explained in the first challenge, the Agile Manifesto describes a development process that does not include them [38], [55], [63], [68], nor the resources they need in order to create excellent design [2]. Even today, Agile rarely trains on UX or working with UXers [17], [38]. The SAFe (scaled Agile) latest framework still considers designers as consultants for the development team, and UX as a job that can be done in couple of hours [38].
The fact that design is still volatile and inconsistent across companies does not help engineering teams to better understand the discipline either [24], [37], [50]. Furthermore, when Agile started to become popular, most of the tech organisations didn't have designers. Back in the time, engineers had to carry UX tasks themselves, surely not optimally [39].

3) Unshared vision and differing mindsets

Another source of challenges between designers and developers working together is an unshared vision of the project and differing mindsets.
Indeed, aligning the entire product development team to a common ground is difficult [16]. Even in the cases where both designers and developers understand pretty consistently what are the project goals, they remain with different priorities and concerns in mind [67]. Developers have a technology-centered philosophy, whereas designers have a user-centered mindset [16]. In addition, the fast pace rhythm of Agile often generates tunnel visions of the project, with the team being too focused on one aspect of the product, for example one feature, while forgetting about the broader perspective [16]. As explained in the fourth challenge, this sense of unity towards the project vision is harder to create, and more likely to be lost on the way if the design team and development team are put apart, within two different departments [23].

UX design and software engineering are described as carrying two cultures; designers and developers are perceived as having two different personalities [12], [67]. The first group focuses on the users, the second on making the idea real [40]. Within Agile environments, emphasis is put on delivering working software, and this is defined as being the primary measure of progress and success [18]. The problem is that this theory forgets that software shouldn't just work, but they should work well [38]. A well working software is a software that allows its users to easily achieve their goals, which is the focus of UXers [18]. Another issue of the Agile mindset is the obsession of making the workflow as efficient as possible [9]. This high concern on building an actual product also decreases the lack of attention to producing documentation, which is a way to maintain common vision [10].

A project big picture is not only shared by designers and developers. Managers, business representatives and the client need to be at the same table too. In the case of the designers-developers duo, the most crucial unshared aspect is the picture of user's needs, or users' voice [12]. As said, users are the main focus of designers. Nonetheless, it is crucial that developers understand it too. Generating and spreading the understanding of the users is a UX task, wheareas understanding the users is the entire teams' duty. Developers not feeling concerned about this priority is a problem, since it is crucial and useful for them to perform their work too [55]. Again, Agile doesn't really help with this issue when it prescribes to cut off thorough research in spite of a small validation phase at the end. Scrum tries to minimize up-front planning in favor of working code quickly, and favorises in-situ workshops to gather requirements [18]. Nevertheless, thorough research and up-front planning are the best way to picture and spread the users' voice properly within the team. The paradox here is designers requesting more time to do research and testings are seen as slowing down the project from an Agile perspective [41], [43].

Balancing these user-centered and engineering-centered mindsets, as well as strengthening common visions in software projects is a challenge that needs to be estimated and dealt with in order to generate seamless collaboration experiences, and create great products [43].

4) Inadequate settings and context

Another source of challenges which has been reported in the literature review, and highly validated by this Web review, is inadequate project settings and context.
This encompases different aspects. Firstly, organisations' culture, internal processes and managerial traditions have a huge impact on the collaboration between designers and developers [23]. It can be harmful in some cases where decisions have to go through many rounds of review, or being approved by various people, and consequently, the process is highly slowed down [23]. This scheme is particulary common in bigger companies where project teams often feel like having many elements to overcomes [31]. The way of reporting their work also influences the ability of designers and developers to deliver impact. A study, focused on design teams, has shown that most of the teams report to product management, which affects their ability to greatly impact on business metrics [8].
An aspect that has been depicted before already, which is the way organisations have implemented Agile, but also how they adopted UX on top of it, strongly impact collaboration experiences and processes. Organisations practicing Agile dogmatically and rigidly, a scheme also noticed in large organisations mostly, generates huge bottlenecks for designers, and for developers [2], [23], [24]. When it comes to poorly practicing UX, joining the aspect of design maturity, huge biases on the practice of design are likely to arise, as for example relying solely on quantitative data, and underestimating the need for deep qualitative user research [2]. As said in the second challenge, UX is understood and practiced differently across organisations [24], [37], [50], inducing this aspect of the context being inconsistent.
Organisations' traditions also are the very cause of designers or developers having to work on too many projects, which either way leads to disconnections on one given project, and prevents them to work optimally, being overwhelmed [32], [38], [70]. Interestingly enough, Design Thinking, often used as the UX process outline, often has to break traditional approaches of organisations to be successfully implemented [45].

The physical location of team members, particularly the location of designers in regard to developers, is another aspect of this challenge. Indeed, teams are often put apart, which has a consequence on the overall project workflow and decision making processes [4]. Tied to this location aspect, the scheme of defining different departments within organisations, especially distinguishing design and engeneering as two very separate entities, prevents product teams from creating a sens of unity, trust and understanding. It often leads them to silos, mostly UX being out of the process, and the overall mindset of "us vs. them" [2], [23], [38]. Considering not only design and development teams put appart, but each and every team members, it has been described that Agile, consequently Agile with UX, is difficult for remote collaboration [44].

The team structure and composition also falls into this settings and context issue. Distributing power and room for influence equally is often hard to establish and maintain [24]. Power dynamics are even more challenging to deal with in cases of multiple product teams working on the same project [72]. There are extreme cases where teams are even split across different companies, if designers or developers are outsourced, and where maintaining track of the project and managing power is even more challenging [51]. A study has concluded that product team structures are equally split between centralized, decentralized and hybrid models [8]. This shows that power dynamics are far from being consistent across various organisations. When it come to the team being transdisciplinary, which is the case with designers and developers, Agile has been described as not fitting very well, which creates another paradox [24].

A last parameter of the project context is the client, which also influences a lot the collaboration process. Certain clients might need extra time and resources to understand design and/or engineering practices [31]. Problems particularly arise when clients think they already know their problem, and are emotionally attached to their assumptions [31].

5) Clashing processes and practices

Another major source of challenges when UX is combined to ASD are the clashes between processes and practices from each field.
Firstly, the high level nature of UX and Agile processes are contradictory. Designers and developers work differently [38]. The UX design process, including the Design Thinking model, is more similar to the Waterfall than Agile model [9], [58]. A strong emphasis is put into the research phase, and each phase is carried out individually, even thought integrating some level of iteration [9]. This is very much a structured model. There is a conflict here in how to bring the UX process and make it Agile; developers often report the challenge of integrating certain design aspects in their process [75]. On the opposite, ASD is an incremental and chuncked approach to building software, where no real overall structure is defined at first. Both ASD and UX processes are iterative, but this level of iteration is very different; being past-paced, short and defined by default in Agile, while being longer and undertaken when a need of it is felt in UX [18]. As said in previous challenges, no place is built for design in Agile [15]. Creating this space appears to be even more difficult, considering Agile being often used as the overall project management model [44], which is not the case for design models. This generates arming feelings of unpredictability for designers [75].

Important aspects of this challenge are timing and coordination. Time management is often reported as being the most challenging part when blending UX into Agile [14]. UX phases and practices need time, and the fast phased product development cycle does not embrace it at all [14]. The number of designers reporting having to carry out too many responsibilies and tasks within two-week sprints, in order to stay on track with their engineering peers, is endless [16], [38]. In addition, deadlines are also felt to be very short and ambitious [44]. This scheme not only generates bottlenecks for designers in their collaboration with developers, but also within their design team, as for example role assignement or UX delays [38]. A solution to partially solve these timing issues is to accord more time for UX to operate, by putting designers further ahead of developers. But again, Agile comes here with the paradox that this practice can arm the feeling of flexibility in the process [37].

When it comes to more specific methods and activities in the product development process, contradictions remain between design and engineering. Traditional UX techniques, not working optimally within Agile, do not generate the same level of result [13]. Deep qualitative user research at the forefront of the project are not welcomed by Agile, and continuous user testings not supported either [16], [43]. Developers only rely on quick validation at the end of the project [9]. Consequently, designers have to re-adjust and trade off multiple of their activities. Many engineering methodologies also exclude designers during the development processes, as for examples requirement gatherings, sprint dailies, plannings and reviews; this generates a need for extra efforts to stay involved [17]. Finally, some activities in UX and Agile overlap with each other, which can create confusion in task assignement between UX and SE [55]. An example is product requirements in Agile with problem definition and concept ideation in UX .

6) Broken communication and decision-making

The last major source of challenges reframed by the author between UX and SE is broken communication and decision-making.
Communication issues are pretty common in the case of designers and developers working together [40], [23]. Different aspects come into play here. First, communication is highly tight to giving feedback, which is crucial for any type of collaboration [35]. If not well provided, or received, feedback lead to misinterpretations and conflicts [40]. Misinterpretations lead to assumptions to each other's ideas and opinions, which are most of the time wrong [40]. Not only providing funded and contructive feedback, but also letting room for discussions, questions and trade-off, can help to prevent these patterns [40], [42]. Communication issues can be due to many factors in software projects. Collaboration tools, or sometimes, the hight number of tools to handle, can bring complexity in the interactions [35]. Complexity can also be felt in regards of translating particular activities' insights into actionable recommendations; for example, some design workshop outcomes have been described has challenging to transcribe in actionable and clear directions for developers [58].

More than interpersonal conflicts between designers and developers, a broken communication leads to result issues. Without an understanding from the tech perspective, designers work in silos when developing design and prototypes, which leads to overpromesses in front of clients [53]. In that case, the lack of communication and exchanges with engineers lead designers to arbitrary take micro-decisions, and to generate concepts technically not implementable under the given project time and resources [71]. Micro-decisions are small decisions independently taken by designers or developers due to a lack of understanding or specifications by their peers. This scheme of micro-decisions also arises with engineers while implementing designs that have been poorly documented [30]. Again, regular and open communication prevents these patterns to arise. Micro-decisions aren't the only component of decision-making issues between UX and SE. Due to multiple factors, as for example the designers and developers being dispersed across different locations [4], [2], or the fact that Agile takes over the project management process, it is common to see important unshared-decisions. These are major decisions taken by the design or engineering team without informing the other team. They are highly armful for the collaboration as they break the feeling of individual ownership of the project, as well as the sens of trust and unity within the global team. As micro-decisions, they lead to issues in the results; too ambitious concepts that lead to project delays and potential additional costs.

An important aspect of project communication and decision-making is the traditional hand-off from designers to developers; designers passing their design to developers for implementation. As explained above, it is common for technical problems to arise [42]. During hand-offs, frustrations can be felt from UXers when implemented products aren't aligned with their design, and felt from engineers when the proposed designs are too ambitious [42], [66]. Hand-offs often are lands of issues because developers lack of specifications, which are crucials [36]. In some companies, it is common to see designers providing screenshots and visuals only to engineers [35]. Nevertheless, screenshots tell nothing, developers need more details [67]. They want to understand the process behind the design, and have all the UI components at their disposals [36]. Without specifications and documentation, an important part of project communication, engineers have to take time deciphering designs, and without having the same eyes and perspective, this is certainly not great of a practice [35].
The hand-off process is often a source of conflicts because it is seen as the only phase where designers and developers truly collaborate, sometimes not even that. The concept of hand-off has generated misconceptions around the overall collaboration model between design and engineering. Hand-offs are just components of a larger converstion [46], [36], which should be spread from the beginning to the end of the entire product development cycle. The definition and conception of hand-off is more and more questioned nowadays, as it seems to be a major cause of this armful "us vs. them" mindset carried by both designers and engineers [46].

The above six re-visited challenges between UX design and software engineering are important to acknowledge and bear in mind in order to understand the emergence and relevance of the Agile UX approach. These challenges can be considered as the main motivation justifying the need for an Agile UX approach to product development. Agile UX, being a potential solution to alleviate these challenges, is detailed further in the next section.

3. Better defining Agile UX

The second aim of this Web review is to complementing the understanding of Agile UX detailed in the literature review, and more accurately sizing this modern approach.

3.1. Conclusion of the literature review

The literature review gave an overview of the Agile UX approach, presented some examples of Agile UX models and frameworks, and depicted these following six Agile UX values:

1) Continuous communication
2) Active collaboration
3) Mutual awareness
4) A sens of one unified and cross-funtional team
5) End-users as actively part of the process
6) Transformation guidance and facilitation

3.2. Agile UX definition, goals, and application

Due to the difficulty of blending UX into ASD [63], as the six major challenges in the last section show it, a new approach based on both design and engineering's perspectives is to be considered; this is where Agile UX steps in.

Agile UX is a modern approach to digital product creation, design and development. It's a model inspired by both ASD and UX models, to truly generating optimal collaboration environments and working processes for designers and developers. Agile UX is not to take UX practices and Agile methodologies, and blending them together, it's about reconsidering and updating these approaches to allow UX practitioners and software engineers to perform their job in the best possible ways, to create the best possible solutions [6], [7], [20]. Although inducing inevitable shifts and trade-offs in both UX and Agile, this modern approach tries to keep their traditional values and principles as much as possible [20]. Agile UX is also about recognizing UX design and SE as equal fields of contribution in software creation, and putting designers and developers at the same level.
Agile UX doesn't limit itself to UX design and software engineering, but these disciplines are the base of the model. UXers and software engineers are the main roles in Agile UX. Other stakeholders likely to be involved are project managers, product owners, organisations' executives, project clients, products' end-users, and sometimes, stakeholders from other departments within organisations.
As described in the literature review, Agile UX is depicted in three distinct levels: The practices and process level, which is intrinsic to the activities of designing and developing softwares and how they are arranged; the people and social level, which refers to the team collaboration and human interactions; the technology level, which refers to the tools, produced artifacts and outcomes, including the released software.

Ultimately, the goal of Agile UX is to integrate UX and ASD; but it goes beyond. As said, Agile UX aims at fostering optimal collaborative environments for designers and developers, and unifying them. This enhances their interactions and ability to perform their respective job [7]. Who says better collaboration and work processes says better results; by doing so, Agile UX also aims at creating better software products; softwares that are aligned with their business goals, and delivering a great user experience [18], [29]. As it is based on the Agile model, Agile UX is also made to create these great products quickly. Added on top of it the UX philosophy, Agile UX delivers great and functional products fast, that are well designed and validated, in other terms, useful and usable [15]. Internally, Agile UX is a great way to easily respond to customer demands, set a budget for a project, and staying flexible for requirement changes and concept refinement [15].
On a higher level, Agile UX radically differs from Agile, in that it truly captures and embraces the UX design discipline and user-centered principles. Therefore, Agile UX plays a major role in design advocacy within tech companies and environments; as a McKinsey & Company study shows, embracing design is a major factor allowing businesses to stand out from the crowd [21]. Agile UX also induces change in the overall selling strategy and business mindset. Unlike Agile, which recognizes working and tangible softwares as being the primary source of value, Agile UX focuses on the impact, the changes in user behaviors, measurable by the key performance metrics (KPIs) [29]. After all, software products are means to achieve particular goals, to answer specific users' needs, and would be devoided of any value if not the case.

Interestingly enough, Agile UX would be irrelevant in a world where every software solutions would be known in advance, any contingency foreseen, and all requirements would be fixed and stable. A Waterfall UX model would be the most efficient model in that case [16]. Nonetheless, this is never the case; user needs, business requirements and technical environment are volatile, and ever changing [1]. This is why Agile UX matters.

3.3. The 8 Agile UX cores

Following the review of the Web content, the author was able to remap and re-detail the Agile UX values, principles, and key aspects, which were named "the 8 Agile UX cores":

- Agile UX core 1: A high design maturity level
- Agile UX core 2: UX designers showing leadership
- Agile UX core 3: A mutual knowledge and understanding between design and engineering
- Agile UX core 4: A unified and cross-functional team
- Agile UX core 5: Shared project vision and goals
- Agile UX core 6: A continuous collaboration
- Agile UX core 7: An active and transparent communication
- Agile UX core 8: A unified and flexible process; the Agile UX process

The 8 Agile UX cores are fundamentals for adopting an Agile UX approach to product development, regardless of the team, project, and organisation. There are obviously cases where totally aligning with those cores will be difficult, depending on the many factors and aspects that define projects and collaborations. Nevertheless, the cores should always serve as a driving direction in the attempt of fostering great design-engineering workflows, for the sake of creating nothing but great products.

It is worth noticing that these 8 Agile UX cores are highly related to the six challenges in combining UX and ASD. Indeed, Agile UX is one direct solution to overcome these. Nevertheless, it goes beyond into creating true design-engineering friendly collaborative environments and work processes. All the cores are highly tightened with one another; consequently, the level of one core has a direct influence on the others.

Agile UX core 1: A high design maturity level

A high design maturity level is the ability for an organisation to operate and understand design at a strategic level, as opposed to design at the aesthetical level, or absence of design. Design maturity is a concept well illustrated by the design ladder (Danish Design Center 2001), which depicts four levels of design integration within organisations. Design at a strategic level induces that the discipline is fully integrated into organisation's processes and culture, and a key element influencing the business strategy itself. For companies, a high design maturity level is nothing more than a key to stand out from the market crowd nowadays [21]. For Agile UX, this aspect is a big deal because it ensures that an entire organisation understands and supports the role of UX designers [12], [43], which allows them to fully unlock their potential, and operate their job in the best possible way.

Organisations with a high design maturity level understand and embrace the value of UX design; they know it is a careful and user-centered approach taking place during the entire software work process, essential for teams to create great results [2], [21], [37], [43], [53]. They value UX designers as equal contributors, along with engineers, in the creation of software [6], [9], and allow them to have a big impact [63]. Therefore, a high design maturity level allows leaders and managers to fully consider UX work and processes [9]; in that way, all UX needs are taken into account [9]. Undertaking research and connecting with end-users is one of these needs [2]. Such environments also place UX tasks and issues in the same system as other technical concerns [17]. More than a role, or set of tasks, UX design within high design maturity level companies is a shared spirit, and as to the overall business vision, is everyone of a responsibility [55]. Such organisations empower the UX design principles and make them visible for everyone [57]. Understanding UX design doesn't limit to a broad and fuzzy overview of the discipline, it's about knowing UX practices and tools and trusting the benefits they each bring. As an example, managers should understand personas as an easy way to put project teams into end-users' shoes, and an efficient way to alleviate biases in decision-making [17].

More than understanding it, executives and managers within high design maturity level environments support UX. They allocate the right resources and time to designers to properly practice their discipline. This includes allocating sufficient time to operate up-front research, as well as to continuously connect to end-users along the project, including undertaking usability testings [6], [12], [17], [20], [59]. This requires designers to always work ahead of developers [9], [17], so that every piece of software can be designed and validated before the implementation. Organisations should constantly stay tuned and flexible for designers to meet the needs of end-users [15]. Furthermore, they should avoid to split UXers across too many projects (also valid for software engineers). Ideally, one designer shouldn't be involved in more than three projects [17].
Supporting UX also goes through allocating designers authority and authonomy. They need the freedom to estimate their own time and set of tasks from the project plannings, since they are the best people to do so [17], [38]. They also have to be key contributors in drafting the overall project roadmap [59]. Therefore, designers within high design maturity environments spread their voice from projects' kick-off, and are trusted to properly plan and undertake their work [68].

Agile UX core 2: UX designers showing leadership

Within well operated Agile UX environments, UX people show a certain level of leadership, [2], [63]. They take time to reach out to their colleagues to make sure the entire team works with clients' and users' requirements in mind [12]. They explain their UX practices and processes, and justify why following UX principles allows to produce better products [2], [57]. When needed, they push their peer back to the project vision and software's use context [2]. Showing leadership is considered as an important UX soft skill nowadays [43].

Falling into this leadership aspect, educating and advocating for UX, is essential [64]. Considering the many remaining misconceptions about the field nowadays, this is the major way to leveling up design maturity within organisations. Advocating for UX differs a lot from one company to another [25]; it can be about selling particular UX practices and tools, convicing to allocate more time for usability testings and research [2], [6], or justifying about the very relevance of the field. In any case, designers need to step up, get out from behind their desk, to demonstrate the value of what they do [43]; demonstrating why the user-centered approach matters [54].
As said in the Agile UX core 1, UX should be everyone of a responsibility [25]. Nevertheless, designers still remain the main user advocates. Therefore, being a design leader also requires being a user defender. When it's needed, designer should show why users are struggling and demonstrate how good UX will resolve it [63]. Relying on key performance metrics (KPIs) and numbers is often a good convincing way.
More than explaining about their process, designers also have to set their colleagues' expectations, developers, managers, and even clients, on how they will contribute to it, including when they will offer feedback [64].

In well practiced Agile UX, UX designers are proactive and engaged from project plannings [54]. More than convincing about their practices, they bring their coworkers with them, involve them in the research to build understanding towards the problem, include them in design ideation to generate concepts around a common ground, and invite them in usability testings to maintain empathy [2], [44]. When needed, designers plan and facilitate collaborative activities, such as braintorming sessions, where everyone in the team is represented [16]. Facilitation also comes up in engineering activities, as for sprint plannings, where UXers have to be involved and make sure to adress the right issues and concerns [12].
Designers also take care of documenting at the right level, which is crucial to constantly exploring and developing solutions [6]. They also make sure to share progress when needed, as early user's insights to build context's awareness [44], or usability testing findings to backup their recommendations [64]. They constantly maintain active communication and continuous collaboration with their colleagues [44].

UXers know the disctinction between design leadership and executives' leadership. They respect managers and clients' authority, while making it clear that UX is the resource to take the best decisions for the users and business needs [64].

Agile UX core 3: A mutual knowledge and understanding between design and engineering

Agile UX requires designers and developers to have a certain level of knowledge, understanding, and consideration on their respective discipline [9]. More than a requirement, some designers and developers truly desire to learn about their co-discipline, since they know it will bring them huge collaboration benefits [39], [71].

For UXers, it is crucial to understand processes and frameworks used in software engineering, and have an extended understanding about Agile methodologies, including Agile teams' rituals and organisation [44], [72]. Furthermore, UX designers know how to think like developers, and understand their point of view when creating designs. They think about how their choice will be implemented [46], and know it will save time and efforts during the upcoming phases of the project. At a personal level, there is a lot that designers can learn from Agile thinking, as working more optimally, more quickly, automating parts of their process, and creating more realistic solutions earlier [20].

On the other side of the spectrum, software engineers know the basics of UX and the human-centered design approach [67]. They are curious to know what are the driving factors for design decisions [27], [39]. Understanding complex user behaviors and interactions is an already implemented aspect in Agile, but when it comes to Agile UX, this aspect becomes crucial [15]. Developers have to properly understand the role of their UX peers, as well as how to interact with them [43]. They're aware that better UX work means less waste of their efforts, since fewer fixes and faulty features are to be operated [17]. Moreover, they recognize the value of UX as such, no less important than their work, and consequently, place UX issues and tasks on their own system [17]. Instead of a secondary concern, engineers understand UX as part of their responsibilities [55], and have to expect being involved in design practices and activities, as for example the up-front user research [30].

Both designers and developers know the importance of discussing and making trade-offs. They're aware that each of their fields have many aspects to be considered in a project [33]. They also know that for UX and SE to be performed, the best persons to do the job are to be considered. It's not about getting the work done, but getting it done well. Therefore, engineering are involved in UX activities, but don't take care of UX tasks themselves, as Agile unfortunately encourages to do. Similarly, designers are involved in engineering activities, but don't take care of their core [38]. To provide great UX and UI is the mission of UXers, to build and implement softwares is the mission of developers [48]. Nevertheless, they recognize their common goals, concerns, and interests, as solving problems, and creating the best possible solutions in the best possible way for the team [33].

A mutual understanding between design and engineering also includes language aspects. Indeed, a shared language is key for collaboration [47], [71]. Speaking the same language, or being "billingual", forsters the ability to communicate clearly and continuously, and taking decisions based on a common ground.
UXers have to know some level of code, as it brings them huge advantages [27], [39]. Knowing some level of code allows them to be more integrated into the engineering process, to take more feasible decisions, and to minimize conflicts within the team [30], [42], [51], [67]. They also understand the Agile and software development terminologies, as for example the sprints, product backlog, user stories, or daily stand-ups [6], [44].

From their side, developers understand the design language, and concepts such as Design Thinking, personas, user testings, user journeys or background research. As a great addition, they are eager to master the basics of design software tools, such as Figma or Sketch [51]. Similarly to designers, this allows them to be more embedded into the UX design process, and to deeper understand their peers[32].
More than a shared understanding, a common language helps UXers and engineers to build a shared culture around softwares, projects, and organisations [34]. Interesting enough, the concept of design sprint, widely used by UXers, has been named as such for software engineers to better relate to it [60].

Regarding this mutual knowledge and understanding between design and engineering, profiles with many soft skills are more and more valued, sometimes refered as “T-shaped“, or “unicorn“ profiles. This is the case for both designers and developers, and the more a team is composed of these profiles, the most likely Agile UX is to be applied optimally [32], [43].

Finally, UXers and software engineers treat each other with mutual respect [12]. They value the work of the colleagues. Naturally, they go further than purely practicing their discipline [36]; more than designers and developers, they are partners and team mates. Anytime, they are open to listening to insights and ideas from their counterpart [2], and willing to allocate time and efforts for the sake of an “healthy relationship“ [41]. Furthermore, since they are learning from each other, designers and developers maintain a common trust and credibility, an essential aspect for any collaboration [30], [42], [43], [63] . Overall, respect creates confortable atmospheres, where both UXers and engineers can feel confortable in expressing themselves, and aspire motivation to work together [42], [40].

Agile UX core 4: A unified and cross-functional team

With Agile UX, the concept of unified and cross-funtional team is major [15], [18]. The time where design and Agile teams operate separately is over. Uxers and software engineers are embedded in one single team [2], [12]. Working as a unique team brings huge advantages to all members, as fostering the sens of collaboration, optimizing project’s settings, and allowing designers and developers to achieve better results together [10], [54]. Moreover, cross-functionality allows every aspect of the product development process to be optimized, as every perspectives are constantly taken into account along the way, which generates better outcomes [10].

As detailed in another Agile UX core, the aspect of collaboration is central for creating the best products, therefore, cross-functional teams are highly collaborative [6], [18], [43]. They make sure that all members are involved when running major activities and taking important decisions for the project [18]. Moreover, they are ensuring an involvement of all the team members from the very beginning of the project. The collaboration starts with everyone being on the table, no matter if the project is kicked-off by a brainstorming session, workshop, project brief review or any other activity [41], [70]. Gathering all the necessary expertise early-on allows to maximize collaborative efforts, and helps to form and maintain a commitment to the original project vision [18]. Within a unified team, designers and developers work tightly with each other, they share common activities and objectives [30], [38]; this sens of unity is to be kept until the end of the project, and being maintained all along the collaboration. From beginning to start, UXers are embedded with engineers, and engineers with UXers [63]. This need to work closely with each other is nonetheless subject to variations across the different phases of the project, depending on the different concerns at a given stage.

Cross-functional teams value no more the UX than the SE discipline, and, as detailed in another core, both designers and developers are seen as equally important for a product to be successfully created [6], [9].

Althought designers and developers are the main roles in software projects, Agile UX teams are composed of other actors. A characteristic of such a team is to involve the customer, understood here as the project client, in the case of consultant projects [15], [17]. This helps optimizing the decision-making process and clarifying each other's understanding of the goals, which is important [74]. Managers and executives are also to be considered as important contributors, and being included as well, as they are the coordinators and supervisors of the project. End-users are another major role to bear in mind in the composition of the cross-funtionnal team. Depending on the project, the nature and rythm of their involvment might differ, but regularly reaching them out is essential to create user-centered solutions. Some other internal or external actors are sometimes to be considered within the team.

Ideally, an Agile UX cross-functional team is physically co-located, and every team member share the same office or working space [18], [30]. Being co-located plays a big role in forming the sens of unity among the team. If not the case, a balance between the disciplines' individual dynamics and shared dynamics is more difficult to find, even though still possible.
A sens of unity among cross-functional teams is also maintained by the tools and system used to organise the work and communicate. As mentioned in the previous Agile UX core, UX and engineering issues and tasks should be on the same system. Centralising project's information prevents teams from generating gaps and delays from one discipline to another [17]. This goes for communication; although it stays relevant to consider sub-channels within the team, regarding the manyfold aspects of a project, keeping a main project channel allows to stay on track with everyone more easily [71].
A good practice is also to create collaboration guidelines. This can be done by writing a team manifesto at the oustet of the project. A manifesto tells how people should interact with each other and sets the standards. This is an easy way to prevent conflicts and align people from day one [69].

Another important aspect of unified and cross-funtional teams is trust [12], [24], [43], [63]. All team members trust each other by showing respect and empathy towards one another [64]. They are as much concerned about serving their team than making the project success [36]; they know that these two go in pair. Having a good level of trust is key to allow individuals to unlock their influence on the project [43].
This goes close to the idea of phsychological safety and confortable atmosphere, which many companies point out as a success foundation [23], [40]. Great software teams are teams where all individuals can bear the feeling of creating impact, solving problems, and being empowered by their peers to do so [40], [41]. This generates huge motivation and sens of ownership towards the work, allowing members to generate a synergy between them, learning from each other, and consequently producing powerful outcomes [42].
More than a great work atmosphere, some level of informality can help fostering the team spirit [71]. Individuals feel more included when they have some space for ralaxing together, and interacting about something else than work [70]. As said in the previous core, being a team member more than a colleague is important and beneficial for the collaboration [30], [36], [71].

Agile UX core 5: Shared project vision and goals

Any designed and developed software product solves its users' problem(s), and answers to a particular business need(s). These two aspects are what constitutes the project vision, itself composed of different goals and objectives. For collaborations to be optimal, and Agile UX to be applied properly, sharing a common vision of the project is crucial [2], [59]. Forming this common ground from the beginning, as well as maintaining its commitment along the way, allows individuals to better understand each other and to perfom their work more accurately [18], [69]. Sharing a project vision is also about making sure that all team members understand the project goals the same way. If interpretations or points of view were to differ, the vision would be broken. This would likely to generate inconsistencies in the work and arm the final result. Both designers and developers share the common mission to create great products, therefore, buying to a common vision is their priority [29]. A project vision is good to be visualised and documented to be accessible for everyone, which will help teams taking relevant and shared decisions with the least chance of tensions to arise [41].

As said, the project overall vision includes the business' and end-users' perspectives. Formulated generically, teams should always care about delivering the best experience for the users, while creating the best value for the client (or their own organisation); the success of end-users and the business value are intrinsic aspects [29], [32]. Interesting enough, great process, optimal collaboration, timeline, methods and tools aren't goals; instead, they are means to achieve the project goals [17]. Therefore, they aren't included in the vision. Indeed, it doesn't matter to the client or end-users that the software has been developed in one way rather than another. Teams are then aligned around principles (what), in spite of recipes (how) [29].

As sharing the project vision goes by being aligned with users's needs, designers play a major role in creating this common ground. Indeed, they are in charge of involving the end-users, spreading their voice, and maintaining it along the way [39], [41]. When highlighting the users' requirements to all their peers, by for example generating user personas and user journeys, UXers ensure the right deliverables to be produced, and usability of the product to be optimal [12], [15], [17]. From their side, developers, and other team members, align with this voice. They have the responsibility to put their assumptions aside, and exclusively rely on it. As described in a previous core, UX is everyone's concern [25], [55], which induces that defending the users and understanding their context is a duty for everyone [30].

On a more practical level, sharing a common vision is done by aligning teams on common metrics. Metrics, or KPIs (key performance metrics) are essentials to measure impact, both the impact on business and impact on users' behaviors [29], [31], [38]. With an Agile UX philosophy, the generated impact is the measure of success [63]. For some organisations, those who bear the traditional Agile mindset around selling softwares, and valuing the cababilities, number of deliverables, and amount of technology, a shift is required. Indeed, with KPIs as the primary measure of success, the impact of a software becomes the primary value; sellings strategies have to embrace it [25], [29]. For designers-developers collaboration, metrics are tangible and trackable ways for team members to share a common ground, to align their goals, to understand each other and to measure their work [40]. In short, these are numbers that speak to everyone. It is then crucial, for Agile UX teams, to pick these metrics up and set expectations from the start of their project, and stick to them along the way [31]. Various frameworks can be used for that; the AARRR (Aquisition, Activation, Retension, Revenue, Refferal), and the HEART (Happiness, Engagement, Adoption, Retention, Task Success) are two widely used examples [29], [59].

Agile UX core 6: A continuous collaboration

The aspect of continuous collaboration is key in the Agile UX philosophy [15], [18], [43], [59]. For designers, developers, and all the other roles involved, it constitutes a pay-off in the work process, and a factor for success [6], [20], [27]. Being a collaborative Agile UX team is the ability of involving all members from day one, all the way, until the end of the project [2], [10], [42], [46], [69], [70]. More than the involvement, a continuous collaboration includes aspects of adaptability, flexibility, and mindfulness towards others' practices [15], [6].
Different levels of collaborative relationships are distinguished within Agile UX teams; the designer-developer relationship, which is the main one, the team-client relationship, the team-manager or team-business relationship, and the team-users relationship. More than designers and developers, a continuous collaboration brings everyone to collect the requirements, generate the design and implement the product [1], [29], [45]. Obviously, keeping a constant relationship with the end-users is particularly crucial to create user-centric solutions, the foundation of good UX [16], [17].

Continuous collaboration provides designers and developers with mindsets and methods to enhance their contribution to the project [6], and ways to bridge the gap between their respective disciplines [53]. Naturally, their contribution varies across the different phases, and their level of involvement isn't constant. Continuous collaboration especially allows individuals to stay on track of the timeline. This reduces efforts in communication, risks of unshared decisions, and frustrations. Cases where presenting a work that nobody has seen, inducing potential effects of surprise, is a no go with the Agile UX model [66]. When deciding on the actionable outcomes of major activities, such as research analyses or usability testings, everyone has to be involved [18].
As depicted in the previous core, gathering the entire team early-on allows to build and strengthen a shared vision, as well as trust [31]. Moreover, it allows to quickly set-up an overall awareness towards the project process and collaboration model, in other term, how Agile UX is put into practice and what expectations individuals should have. An early involvement of developers is important to give the direction to designers on technical feasibility [17], [53]. An early involvement of UXers allows for efficient up-front research and an accurate context understanding.
Continuous collaboration is also the key to lighten designers-developers hand-offs; it reduces the amount of heavy documentation, minimizes mis-understandings, and increases the sens of ownership from both sides. The traditional conceptions around hand-offs, which assume design to be crafted separately and thrown to engineers with no follow-up, are completely redefined [30]. Implementation is a conversation [46], and product development is a continuous learning [41]. Again, but never stressed enough, designers are part of the SE process, and developers are part of the UX process [33].

In Agile UX, designers involve themselves with developers in various ways. They attend the development meetings, including plannings, daily stand-ups, and retrospectives [17], [44], [72]. During the technical implementation, they have a major role in following-up their peers to make sure the software to keep aligning with user's needs [18]. Following-up is a monitoring practice, where UXers provide feedback and help to their engineering colleagues, and stay available to hand over any design resource [27], [72]. Furthermore, they bring up their voice when decisions related to what's in the product backlog are made, including features, bugs, and any other task. Designers can also facilitate the development team, which makes the follow-up practice even more efficient [43]; as explained in a previous core, designers are leaders.

From their side, developers also involve themselves with designers in many ways. They are involved in the initial user research, which allows them to get an accurate understanding of the context from early on. Developers also take part in the concept ideation, by providing their peers with a technical perspective, which helps choosing realistic directions [5], [30]. They keep an eye on the overall design prototyping to properly assimilate the software flow and identity, which saves them time for the up-coming coding phases [17], [23]. Moreover, regardless of their design knowledge, they are a good addition to come up with ideas and solutions [5]. Software engineers also attend usability testings to get a quick and clear understanding on improvements to make, with the least amount of documentation to be made [2], [18], [41], [44]. Similarly to their colleagues, they bring up their view on any design decisions, so that proper and commonly validated trade-offs can be made.

On a practical level, continuous collaboration is to be defined, by picking a framework up, setting the rules, choosing the tools, and selecting relevant activities and methods. The way developers, designers, and other team members interact with each other along the journey needs to be thought off. Too little collaboration leads to poor results and frustrations; too much collaboration leads to waste of time and confusion in each other's roles and tasks. A right level of collaboration is crucial to be found, based on the project context.

Additionaly to the existing UX and Agile activities, at some level reconsidered, in order to be more collaborative, various methods are especially thought to bring UXers and software engineers together. These are direct ways to break silos and streamline the overall project flow, with minimized needs for hand-offs [5]. A lot of them are design methods related to the ideation, as the design studio, the whiteboard session, the telephone, or other form of brainstorming and sketching approaches [5], [16], [20], [37]. The cross-functional pairing is an implementation method to split the work between multiple pairs of designers and developers. Design prototyping and design sprints are also activities whose sucess highly relies on close collaboration with software engineers [17], [29], [70]. Again, other non-design and non-engineering roles are to be considered as part of these interactive practices.

Agile UX core 7: An active and transparent communication

Great communication is a golden rule for Agile UX to be properly applied [27], [30], [42], [51], [71]. As for the necessity to collaborate across the project, it has to be active, continuous, and transparent, between all the team members. UXers and engineers have to put consequent efforts to ensuring a good communication, because at the end of the day, it will create trust, motivation, shared understanding, and smoother the process with minimized risks of conflicts between them [30], [42]. Nowadays, it is recognized as a must to have soft skill for both UX and engineering practitioners [28], [39].
Communication in software projects can be depicted in four different levels, each with different objectives.

The first level is the type of communication that sets the general tone for collaboration. This relates to the general exchanges and interactions between the team members, including the way feedback is provided and received from one another. Giving feedback is important in any collaboration, and directly reflects the capacity for people to empower, listen, and learn from each other [35], [42]. More than providing feedback, communication goes into practice for discussions around product decisions and trade-offs. When having conversations, people should avoid patterns such as shutting down other's ideas, let the debates unresolved, or entering into battles of opinions [64]. Within professional settings, arguments aren't to be taken personally, and all other points of view should be respected [64]. Agile UX team members, designers and developers, keep the discussions running, they seek for the best alternatives and build on each other's ideas [40]. Not necessarilly essential, but highly encouraged within Agile UX settings, the ability for each other to adopt a friendly attitude when communicating helps establishing a confortable and more human atmosphere [23], [36]. As explained in a previous core, designers and developers should go further than purely and rigidly practicing their discipline [36].

Another level of communication refers to the necessity for Agile UX teams to organise their work and stay on track of the timeline [71]. For designers and developers, it relates to the ability of sharing the progress, explaining their work, and making sure a continuous understanding and common vision is kept [44]. Maintaining a constant learning and dialogue from early on, and througout the entire process, is capital [12], [39], [41], [71]. In practice, this can be done by asking questions regularly, and welcoming others for solicitation [71], [41]. Weekly reviews, or any other form of regular meetings, are great for ensuring the team to be on track [42], [44]. Furthermore, making use of messaging tools and platforms that centralize the communication is a great way to optimize the organisation and minimize the confusion [71].

The third level of communication in Agile UX relates to decision-making process. All along the project, designers, developers, and other team members, have to make various choices around the software they are building. These can be small, as when deciding to choose an activity over another during the process, or major, when prioritizing a feature in the product backlog. In any case, decisions have to be shared by everyone [18], [36], [37]. The decision-making process is to be started with mutual conversations, and carried upon negociations and trade-offs [33], [41], [42], [70]. In this process, constant transparency is the golden rule, meaning that all members contribute to the discussion and are aligned with each other [33], [37], [42]. Every aspects to be considered in a given decision should always be disclosed and plainly explained to the team, whether they are UX, technical, or other sort of considerations [33].

The last level of communication in Agile UX relates to the hand-off between designers and developers, which is an essential aspect of the software creation process; this is the action of passing over designs for implementation, which mostly is done gradually, piece after piece of software. Hand-offs are lands where communication has to be managed carefully, beforehand and afterwards [46].
Within the hand-off practice, three sub-types of communication can be differentiated.
The first is related to the spoken and active communication between the individuals. In Agile UX, the hand-off process is a conversation [46], [58]. The hand-off spoken communication extends much beyond the time where designers and developers exchange materials. Designers should make early efforts to ensure their peers to have a good awareness of the design they will receive. This saves a lot of time in the implementation phase, since the understanding is shared, and risks of having to re-do designs are minimized [46]. Conversations in hand-off happen beforehand, for designers to truly emphasize with the technical perspective while creating designs, and for developers to assimilate the context, understand design choices, and be aware of the details from early on [42], [71]. Conversations happen afterwards, for developers to be supported by their peers while coding, and for designers to ensure the end-users to stay the core focus [27]. Designers should also be taking extra time to re-iterate design explainations, and ensuring their peers to correctly understand the process [49], [58].
The second type of communication in hand-offs is unspoken, and relates to the product documentation. Being a necessary addition to the prototypes and design materials, the documentation gives deeper details and explainations about the software, helping developers to better understand the design and the decisions behind it [26], [46], [67]. Documentation can take different forms depending on the given project and needs [66]. Specifications are one of these, and purely consist of providing written comments and visual details about the design [46]. Recordings and videos are also great ways to provide extra details to developers. Furthermore, designers being disposed to share the design files and relevant project reports in an organised manner is another component of good documentation [27], [44], [49], [64].
Regardless of the form it takes in a given project, documentation should always stay simple, concise, and actionable for developers to appropriate it as efficiently as possible [18], [23]. The best documentation is the one that tells the more details, in the most high-level way [6].
The third type of communication in hand-offs encompases the design prototypes. These are the artefacts giving an high fidelity preview of what the final software will be. The more designers are able to master the design and prototyping tools, to generate detailed prototypes, including layouts, flows, and interactions, the more streamlined the hand-off is [27], [66]. Ultimately, clickable prototypes, which are becoming more and more advanced with the evolution of the modern design tools, allow designers and developers to hold richer conversations, and minimize the amount of extra documentation [53]. Prototypes are the ways to gather all team members around a common perception of the product to be built, without misunderstandings [53]. Again, this form of communication is unspoken, but capital.
Complementing high-fidelity prototypes, design systems are also an important aspect of the hand-off communication [47], [67]. Design systems lay down the product styles, components, and patterns, in an organised way; they are seen as the internal version of the final product, as a common tool box for designers and developers to interact with each other [34], [71], [66].

Agile UX core 8: A unified and flexible process; the Agile UX process

With the Agile UX model, collaborating efficiently requires designers and developers to follow a unified process, which is going to be refered here as the Agile UX process. Considering they have to manage the outcomes of the product together, following a common process allows design and tech practioners to break the silos, adapting to each other's workflow, as well as staying close to one another [29], [30], [31], [66]. The Agile UX process adopts the characteristics of both Agile methodologies and UX practices, incorporates the needs of UXers and engineers at the same time, while bringing a new overall structure that includes unique methods and activities [6], [9], [14]. Resulting from the UX discipline, the process includes the essential phases of research, design ideation, and validation [10]. Issuing from the Agile workflow, the process embodies the fast-paced, incremental, and iterative attributes.
The Agile UX process unifies UX and SE processes, nonetheless, designers and developers work on two parallel tracks, which are highly connected to each other [38], [72]. Being structured in such a way allows the fields to be operated connectively, while letting the flexibility to practitioners to work the way they are the most confortable with. As said, UXers and engineers have two very different working styles; the Agile UX process recognizes and embraces it [68].

The Agile UX process emcompases high level characteristics, which stay similar across all types of projects. Firstly, the process is iterative. This attributes mostly results from the Agile philosophy, but is also a scheme of the UX process. Being iterative allows the team to support learnings about the product, users, and context, throughout the entire development cycle [14], [16]. This aspect is also a way to ensuring the project direction to stay aligned with business and user's goals. Furthermore, being iterative allows the team to scope, descope, prioritize, and re-prioritize the requirements along the way, since the concept often is undefined at the outset of the project [16], [18]. Fundamentally, the iterativeness is based on sprints, whose length varies depending on the needs [1], [44]. Sprints in Agile UX generally go from one to four weeks long, and include UX and SE activities together.
Highly tight to the iterative aspect is the responsiveness. The Agile UX process is responsive to customer demand, to the market context, to user discoveries, to technical unknowns, all aspects that are influencing the requirements [1], [15], [16].
Another attribute of the Agile UX process is its incremental nature. Similarly to Agile methodologies, Agile UX is divided into chuncks. This incremental approach allows the team to take care of small parts of the solution at a time, and having more focused hypotheses to user struggles [14], [16]. Incorporating both UXers and engineers practices, this chuncked structure stays flexible, and teams should be concerned about producing quality work, over following strict structures and rituals [6].
One of the fundamental of the Agile UX process is also to be able to quickly deliver great digital solutions [16]. The process is fast-paced, and the timeline as short as possible [44]. The length of a given project is never predefined, and varies a lot based on the needs, but the idea of the Agile UX process is to be faster than traditional UX processes. Software typical projects often remain a matter of weeks over months [15].

Additionally to its overall nature attributes, the Agile UX process has a high level structure in place, which should be written down at the outset of the project [71]. A high level structure depicts the different phases and steps of the process. Eventually, it can also incorporate the timeline and different activities falling into each phase. A high level structure is a scheme that is mostly resulting from the UX discipline, as it is particularly important for designers to have a formalized guide [17], [31], [37]. Being visualised and made accessible for all the team members, a structure helps to share a clear idea of the process and direction; referring to it helps designers and developers to take decisions with minimized risks of tensions [41].
The structure encompases five building blocks that stay unchanged across any kind of digital project. These are the discovery, the definition, the ideation, the validation, and the implementation. These five phases are separated into two parts; the discovery and definition phases constitue the research part, the ideation, validation and implemention phases define the solution creation part.

There are multiple frameworks that can be utilized to draft the high level structure of the Agile UX process, such as Design Thinking, the double-diamond diagram, or the phases defined by the human-centered design approach [31], [55], [68].

Within these two parts and five phases, the Agile UX process encompases three key steps.
At the oustet of the project, the kick-off aims at gathering the entire team to draft the vision and goals, and especially, it aims at defining the process and collaboration model based on a common ground [71].
Another step is the hand-off, which describes the action of passing over the elements of design and related documentations from UXers to software engineers. In the case of Agile UX, the hand-off is practiced incrementally and continuously throughout the solution creation part [30]. As depicted in the previous core, the hand-off isn't just a mechanical step, it's a collaborative practice and conversation.
The third major step of the process is the sign-off, which follows the implemention phase. The sign-off defines the official software(s), piece(s) of software, or feature(s) released to the end users. This crucial step includes a follow-up, to measuring the new metrics, and evaluating the impact and success of the project.

The timing of the Agile UX process is really dependent on the project needs. Nevertheless, various trends can be highlighted.
The first one is the coordination of the UX practices ahead of development [10], [16], [38], [59], [72]. Indeed, as said in previous cores, designers need time to properly performing the research, connecting to the users, defining the problem, as well as to coming up with accurate actionable insights and validated designs [2], [59]. Designers staying ahead of development ensures that all software pieces that are being coded and implemented have been priorly designed and tested, so that engineering re-works and fixes are minimized afterwards [72].

The research part, or up-front research, being a major and essential component of the Agile UX process, is seen as a particular entity within the overall process, because it doesn't fall as much into the incremental, iterative and fast-paced attributes, as to the solution creation part [30], [66]. This is explained by the fact that the discovery and definition phases aren't best practiced under such an environment, as they usually need some time and planning.
The research part being undertaken, a good practice is to kick-off the solution creation part by a sprint 0, focused on design production [6], [17], [18]. Following is the first sprint of development, based on the outcomes of the sprint 0. During the first development sprint, UXers work on the next design sprint(s), so until the software is fully validated and built [9]. During the current sprint, designers follow-up their peers, and work on eventual new discoveries or issues that can arise during the implementation. Future sprints take the next priorities in the product backlog; the further it goes, the less focused and defined the requirements are [75]. Generally, sprints can be distinguished between different levels, the current sprints, next sprints, upcoming sprints and future sprints [75].
The sprint model is a core element of Agile UX. This model requires UX activities to be adapted, which isn't the case for the engineering practices. The concept of design sprint is such an adaptation [10], [29]. Design sprints allow UXers to fit their practices into the incremental, iterative and fast-paced environment. They offer a quick and efficient method to designers and Agile UX teams to ideate solutions in response to defined problems, and generate validated designs. Design sprints have to be distinguished from development sprints, as beside their lenght, they differ in all aspects [10].
Other than design sprints, various UX practices can potentially be tweaked to fit into the sprint model. An example are usability testings, which often need to be run quickly, and therefore, might need to be conducted with fewer participants, or prototypes with a lower fidelity; this is called the discount usability method [18].

To be noticed regarding the coordination of the Agile UX process is the constant overlapping of designers and developers from their own process track to the other. Indeed, as a previous core highlights it, UXers and software engineers continuously collaborate throughout the project. The fact that the Agile UX process is divided into two tracks is to provide teams with mental models that help understanding how the UX design and software engineering disciplines correlate with one another, not to generate silos or hinder collaborative efforts.

Process tools and activities are very dependent on the context.
The Agile UX mindset encourages to use collaborative activities as much as possible. Designers usually tend to be in charge of planning and facilitating such types of activities [16]. A common example of collaborative activity is the brainstorming session or workshop, where the entire team is gathered to come up with insights and ideas. Any sort of meeting, where all members of the team come together, to plan, prioritize, give feedback, or do retrospectives, also fall into this type. More specific examples of collaborative activities are the whiteboard session, collaborative sketching, the 6-Up 1-Up, the telephone, or the design studio, which are all ways to generate solutions involving entire teams [5], [16], [37].
In terms of tool, the mindest is similar, involving everyone. Writing user stories has been recognized as a common touchpoint between design and engineering [14]. They are an excellent way of keeping the end-users in mind, and for designers to communicate the UX vision to developers. Other examples of tools aligning with the Agile UX spirit are a shared product backlog, user journeys, and personas. Again, tools can be very dependent on the project given needs and context.

Finally, an important aspect to be highlighted with the Agile UX process is flexibility. There are not any single recipe that can satisfy every single project context. The Agile UX mindset recognizes the process being a guide and a structure to support teams, more than a strict duty [2]. Indeed, the process needs to adapt to the many factors and unexpectancies suspected to arise throughout the project [31]. These unexpectancies can be shift of human or time resources, changes in the timeline, or tweaks in requirements and vision, whatever they come from the business, design, or technical side. Therefore, the process is balanced between parts that are defined at the outset of the project, and parts that are defined on the way [31]. In short, the process is drafted at the beginning, and constantly modified along the way [16]. The process should always fit to the needs of a given project, and constantly ensure to support teams optimally to do their job.
This mindset on the process aligns very much with aligning teams around principles instead of recipe [29]; the overall concern is to create successful and impactful products, not to follow rules [63].

The above eight Agile UX cores, encapsulating the values, principles, and key aspects of the Agile UX approach, are the main material to understand Agile UX in this thesis. They will be used later on in the actionables chapter (Chapter 4), where they will be put into correlation with the empirical research findings. The empirical research will mostly be adressing the limitation of the Agile UX approach, presented in the last section of this Web review (4. the Agile UX gaps), but connecting the findings with the Agile UX core is particularly useful to seize the status-quo of Agile UX in the industry.

3.4. Practices and trends supporting Agile UX

The review of the wide selection of web content has highlighted the fact that various modern practices and trends in the industry are already going towards the direction of Agile UX. As said, the willingness to unify design and engineering processes in digital product development is on its path, but it's not like everything needs to be built from ground.

The author proposes here a small taxonomy of five different practices and trends picked-up from the web sources, briefly describes them, and explains why they contribute to supporting the Agile UX mindset, and how they are aligned with the cores detailed above.

These five elements are not the only on-going practices and trends supporting the Agile UX values, but they already give an idea on how the approach comes into play more concretely.

1) Advocating for UX

Due to the overall lack of design maturity in the industry, with various levels depending on the company, advocating for UX, or advocating for design, is becoming more and more common of a practice, usually carried out by designers. A lack of design maturity has been defined in section 2. as a major source of challenge for collaboration between designers and developers. All sort of initiative coming into play to rise the design maturity level within a company, including getting an understanding by executives and software engineers, having enough and well estimated UX resources, as well as establishing an overall mindset towards delivering user impact over tangible software, directly support the Agile UX core 1: A high design maturity level.

Advocating for UX is becoming more and more of a focus towards well balanced processes in digital product developement. Nowadays, it is even recognized as a skill designers should have [64]; in most of the cases, designers cannot expect project managers, products owners, the client, and the rest of the team, to know everything about UX [64]. Advocating for UX can be a very different practice depending on the company; sometimes, it's only about human resources, or in the most extreme cases, about selling the very relevance of UX [25]. Establishing design principles, and making them visible across entire organisations, also falls into this practice [32], [57]. Established design principles, including laws, guidelines, and design considerations, help designers to meet their goals and make design decisions [57]. As just said, this practice is usually done by designers themselves, but there are other ways to aim for good design maturity levels. A good example is described in one of the sources, a design course for developers, which doesn't teach them to practice design, but give them enough understanding of the field for an optimal collaboration with their UX peers [32]. Another way of rising the design maturity level can be to hire an external expert, or coach, to facilitate the transition [37].

2) DesignOps and DevOps

DesignOps (design operations) and DevOps (development operations), are respectively defining the practices, processes, guidelines, and measures that support design and engineering within a company, so that consistent and quality work can be done [4]. DevOps is still in development in IT companies, but DesignOps is especially new, therefore, big hopes are put into this trend when it comes to rising design maturity, as well as to enhancing UX design processes and practices in the industry.
DevOps and DesignOps touch all aspects of engineering and design, from communication, tools, processes, vision, to human and technical resources. DevOps and DesignOps are very specific to a given company context, it should be unique to every single organisation. If well established, and well aligned with one another, DevOps and DesignOps create an optimal relationship between designers and developers, help growing and evolving teams, allow to find and hire people with the right skills, allow to generate efficient workflows, as well as improving the quality and impact of digital products [4], [41].

3) The evolution of the hand-off practice

As detailed earlier, the hand-off is the practice that defines the act of passing away designs from designers to developers, for implementation. Traditionally, hand-offs are almost the only interaction design and engineering have during a project, and each discipline work in a very siloed way. Obviously, this mindset is the opposite of what has been described in the Agile UX cores, and this siloed way of working is the cause of a lot of challenges for digital product development teams.

Nowadays, the hand-off between design and engineering is evolving towards the direction of Agile UX; it is done more and more continuous across the entire project, piece by piece, and supported by clear and transparent communication. Different aspects of the hand-off are becoming recognized, such as simple, actionable, and high level documentation for developers [27]. This documentation can include visuals, guidelines, or specs (specifications), of any form, as well as any other elements that aim for a clear understanding of designs by developers [18], [46]. The big deal is to find the right balance between keeping documentation as concise as possible, and giving as many details as possible [66]. Hand-offs should also be supported by conversations and collaborative activities between UXers and engineers all along the project, before, and after design delivery [20], [27], [30], [46], [58], [71].

4) The evolution of tools

The evolution of tools is another trend aligning with the Agile UX direction. These include collaboration tools such as platforms for communicating, like Slack, as well as tools where tangible work is produced, such as Figma or InVision [71]. In any case, these tools are allowing more and more members of digital product development teams to be involved. This improves the aspect of shared understanding and vision of the project, as well as transparent decision-making [36]. These tools provide designers and developers with better environments to hold conversations along the project, between themselves, but also externally, with clients and end-users [53]. Nowadays, some specific tools allow designers to create design prototypes with a huge level of detail very quickly, which wasn't possible few years ago. Examples of such tools are Figma, but also ProtoPie, Framer, or Anima. High fidelity design prototypes reduce needed exchanges between designers and developers to communicate, as well as risks of misunderstanding each other [53], [66]. The evolution of these tools helps team individuals to focus on their work, and to faster the process [35]. Tools also centralize more and more processes flow, which helps to keep the work more organized, structured, and less messy [36]. In short, tools are contributing in bridging the gap between designers and developers [36].
It is hard to predict the future, but these collaboration and design tools will keep evolving, to a degree where workflows between UXers and engineers might be very different in few years already from now. They might even redefine the roles and tasks in design and engineering; for example, some say that designers will also take care of front-end development [39].
Today, the evolution of tools support the Agile UX principles and mindset, but tomorrow, they might also influence on the evolution of the Agile UX model itself.

5) The rise of design systems

A design system can be defined as a set of standards and instructions that can be re-used by UXers and engineers [47]. This is the system that clusters all the information, assets, and components, needed to build a digital product. The topic of design system has gained more and more interest in recent years, and is now becoming an important aspect of any digital product development project.
Again, this trend goes in total alignement with the Agile UX philosophy. Design systems ease work and save time for both designers and developers, as they can re-use components or pieces of code, instead of re-creating them [23]. More than just laying down and structuring elements and standards of the design, design systems bring clarity, efficiency, and consistency into the product development process [34]. Design systems are becoming a major touchpoint, and shared material, bringing design and engineering together, and allowing to foster cross-functional collaborations [34], [47], [71]. For developers, a design system is a key to decifer the product design [27]; this is becoming the main piece of reference to support the implementation [67]. Therefore, design systems contribute in creating streamlined hand-offs, by bringing order, and reducing the amount of documentation [27].
Nowadays, design systems are becoming as important to consider as softwares themselves, and start to be seen as internal products for organisations [34].

3.5. A small clarification about Agile UX VS Lean UX

To close this section about defining Agile UX, the author felt the need to clarify in what extent the model differs and overlaps with Lean UX. Indeed, these two approaches, both intrasically tied to digital product development projects, are sometimes used interchangeably, what can bring some confusion. Fundamentally, they both share the same vision, which is to integrate UX design in a more light-weight, iterative, and flexible way into processes, and providing UX designers with environments to enhance their contribution [6], [7], [55]. Both Agile UX and Lean UX aim for great user-experience products. The real difference is the How.

Firstly, Agile UX and lean UX are two different ways of working [1], [7], [13], [20]. Lean UX defines the integration of UX methods with the Lean Startup model, an approach derived from Lean manufacturing, that aims at creating and validating concepts, targetting especially the outset and innovation side of digital product development. As defined earlier, Agile UX describes update of the Agile software development approach with UX design methods [7]. Lean UX focuses on the outcomes, rather than outputs, and aims at generating assumptions and hyptheses to be tested quickly. Lean UX aims at developing minimum viable products (MVP) and/or proof of concepts, through constant measurement and iteration along the learning cycle loop "build-measure-learn" [6], [7], [13]. Lean UX has a bigger emphasis on the research and quality measurement, while Agile UX focuses on designing, communicating, and delivering [20]. Additionally, Lean UX aims at generating multiple products, or concepts, of increasing polish, while Agile UX aims at producing a more polished product [15]. In other words, Lean UX builds the right product, whilst Agile UX builds it right [6].

Despite these distinctions, Agile UX and Lean UX share many common principles and aspects [1], [9]. They both value high levels of collaboration and shared responsibilities regarding research and design, within cross-functional teams [6], [9], [13], [15]. They both encourage to release early and often, so that continuous testing, feedback, and validation can happen [15]. Each of theam is built on iteration cycles, which fosters continuous discovery and improvement, as well as change responsiveness [6], [9], [13], [20]. The two of them focus on solving users' problems and needs, and see customer success and business value as the same thing, therefore, they measure success based on the impact and change in user's behaviors [15], [29]. Both approaches try to maximize the efficiency and speed by eliminating waste in the process and collaboration [6].
Agile UX and Lean UX can be used in any sized business, since they are free to be adapted to different contexts, and neither of them has a rigid process [1]. Both models are based on sets of principles and guidelines rather than a strict framework to follow [6].

Interestingly enough, Lean UX and Agile UX are complementary, and can be applied simultanuously [7]. Some organisations and teams already combine these two models [16].
As described earlier, the research part of the Agile UX model, or up-front research, defining the discovery and definition phases, is seen as a particular block within the overall Agile UX process, because it doesn't fall as much into the incremental, iterative and fast-paced attributes as to the solution creation part [30], [66]. For this part of the Agile UX process, Lean UX is one possible workflow to be merged with.

4. Reformulating the current Agile UX gaps

The third aim of this Web review is to complementing the insights on the current Agile UX limitations laid by the literature review, remapping, and reformulating them.

4.1. Conclusion of the literature review

The literature review has defined six current Agile UX gaps:

1) The context and situatedness nature of software projects: each context is unique
The lack of consideration on project settings, organisational culture, values and management styles.
2) The rigidity of the proposed frameworks
The proposed frameworks being to contextualized and not allowing enough flexibility.
3) A demanding adoption
The tranformation or shifts induced by the adoption of Agile UX being too demanding and hard to maintain on the long-term.
4) No framework that satisfies all ASD or UXD principles
No framework that perfectly balances both ASD and UX principles, leading to many compromises for one or both disciplines.
5) A room for improvement in practice
The current Agile UX frameworks not being clear enough, on aspects such as users' involvment, details on day-to-day activities, roles and project flows, as well as process synchronization.
6) A lack of guidance and understanding from the technology dimension
The technology dimension of Agile UX being the least understood and advanced, compare to the practices and process, and the people and social dimensions.

4.2. Reformulating the current Agile UX gaps

Agile UX is a relatively new and evolving approach to digital product development. Its relevance is being more and more recognized in the industry. This background research, including both literature and Web reviews, demonstrated the ability to theoretically seize and define what this model consists of, and how it brings a solution for optimising design-engineering workflows, in response to the challenges depicted earlier. Nonetheless, limitations remain, especially when it comes to putting it into practice.

Based on the 75 Web sources, complementing the above summary of the literature review, the author was able to narrowing down two major gaps of the Agile UX model:

- Gap 1: Lack of application instruction, and attention on contextuality
- Gap 2: Lack of anticipation to the evolution of design, engineering, and digital products

These two gaps, detailed bellow, will give the direction for the upcoming empirical research.

Gap 1: Lack of application instruction, and attention on contextuality

The background research highlighted Agile UX to be a solution for great processes and collaboration between design and engineering, at least theoretically. The above eight Agile UX cores deeply go through various aspects of digital product development. Nonetheless, there are none or poor instructions when it comes to putting them into practice. Althought detailed, theory is not sufficient to apply the model, and many other aspects are to be taken into account; this is where lies a first gap. The contextualized and uniqueness nature of each project, highly influencing the ways of working, is to be studied to understanding the ability for organisations, teams, and individuals to adopt the Agile UX approach themselves. Above all, context is what defines how processes and collaboration come into practice [16]. Some even argue processes being as important to be designed as products themselves [43]. Therefore, models such as Agile UX should be taken as supportive principles, guidelines, and eventually dynamic toolkits, which can then generate customized frameworks, but certainly not as strict and polished recipes to follow [29].
In digital product development, the aspect of context can be divided in three layers, each having a consequent influence on workflows.

First, the context is defined at the organisation level. This level defines the aspects of company's type, field, current situation, culture, traditions, and settings.
Some argue that companies should experiment and find their own way of fitting anything into their culture [14], [19]. Processes and practices should be customized in alignement with the vision and strategy in place [14]. For implementing a model, such as Agile UX, what matters is the mindset and the goal, not if the model is applied accordingly to its definition. There is no such thing as fixed model working in every context [31], [38]. Indeed, startups' environments are very different than big corporations, in-house contexts very different than consultancies, business-to-business (B2B) very different than business-to-customer (B2C), to list very few comparisons. From one of the Web sources, an employee tells about his company continuously refining combinations of user-centered design practices, Agile, and Lean, for over ten years. They also describe about adjustements being made for each and every project [18]. This is a good example illustrating the necessity for companies to be active and dedicated in implementing models, practices, and mindsets, and being willing to operate shifts if needed. Indeed, applying Agile UX can require an organisation to make changes in its structure, in regards of the designers-developers relationship [25], but also changes in decision-making processes [37], how roles are defined, and how much responsibility and influence they have [12]. As described in the above Agile UX cores, a company's culture and settings are considered as crucial factors. Nonetheless, the degree of malleability of these aspects vary a lot, and this needs to be acknowledged when formulating methodologies to adopting and applying Agile UX.

Secondly, the context is defined at the team and people level. This level embedes the aspects of team composition, roles, structure, size, hierarchy, relationships, as well as people's experience, skills, and connections with each other.
Similarly to the organisation level, this level influences a lot processes and collaboration [41]. The Agile UX model provides teams with some directions, but again, the malleability can be limited. As an example, it might be hard to easily modify the structure of a team depending on its size and setting constraints. Same goes for the composition, and people experiences. Theses aspects might be set in the stone, and any model coming over should deal with them. A context being different from one company to another, it is also different from one team to another, because a team is a local entity, composed of unique individuals [24], [41]. A recent study conducted by NN group allows to seize the diversity in terms of team structure [8]. Three common models have been reported: centralized, decentralized, and hybrid, with all different workflow challenges and concerns. Again, adapting to the local team is more important than blindly following an approach [24], [37]. To truly capturing team member's needs, Agile UX should not only be specific to each organisation, but each and every team as well [16]. Models can help to set standards, but teams should also learn to let go, experimenting, self-organising, and to defining their own appropriate Agile UX workflow [29], [37], [54].

Thirdly and lastly, the context is defined at the project level. This level embedes the aspects of product type, timeline, client, resources, tools, and processes in place.
The two first levels have described the organisation and team as being unique pieces of context; the other piece bringing another degree of uniqueness is the project itself. Each project is unique to an extent. Each project aims at creating a specific solution, a specific product and type of product, focusing on particular users' needs and problem [26]. One given project may require the collaboration by the team with a client, in the case of consultant environments, which will multiply the specificity again. Indeed, clients draw an important influence on team workflows. One given project also goes with a unique timeline, and scope. The length of a project influences a lot the nature of the process; as an example, one blog article stresses the major disctinction on how design activities are fit into sprint cycles, for short-term projects, compare to long-terms projects [10]. In one podcast, A UX designer even tells about their team using different sets of tools based on the timeline [30]. Althought the high level building blocks of the Agile UX process can be used in many contexts, teams often use unique sets of activities, based on their specific project needs [1], [31]. Similarly to the two first levels of context, the aspects defining this third level of context can be tweaked, with some limitations. This induces another degree of customization to consider, for contextual applications of the Agile UX model.

These three layers of context (organisation, team, and project), each including various aspects, seem essential to be looked at in the attempt of adopting Agile UX. To find the most appropritate, unique, and tailored fit, the key is to understanding how to find a good balance between tweaking elements of the context, and elements of the Agile UX approach. Having defined Agile UX, it is now about defining and understanding contexts, evaluating their malleability, so that better methodologies to applying the model can be drafted. This defines the first, and major angle for the rest of this thesis research.

Gap 2: Lack of anticipation to the evolution of design, engineering, and digital products

The current formulation and understanding of Agile UX relies on today's state of the different disciplines and technologies involved in digital product development. Nonetheless, the model would probably be formulated differently ten years ago; it is then relevant to lend a concern on what will be Agile UX ten years from now. Indeed, lots of aspects have evolved in the past years. The fields of design and engineering are evolving very fast. This evolution lies in the knowledge, practices, activities, and tools of each discipline, but also the evolution of design and engineering roles in the industry. It would be too long to lay down all details on the evolution of these fields; the following paragraph gives one great example, the prototyping tools.

Prototyping tools, such as Figma, are changing, and will keep changing how designers work, as well as how they collaborate with their peer developers. Although there is still room for improvement [49], these modern tools allow for faster processes, more seamless interactions within teams, and better understanding of each other. Prototyping tools are also getting so powerful, by generating ready-made code, that they are starting to automize the front-end development process [39]. Website builders such as Webflow, or mobile application builders such as Bravo, already allow to skip this entire front-end development process. This trend of automation, drawn by the enhancement of digital tools and technologies, will drastically influence the future of digital product development workflows.

More than this example, many other emerging trends are slowly starting to paint a new landscape for digital product processes. Jakob Nielsen, pioneer in the field of UX, has presented in 2018, ten UX directions for the future, designers, and all actors of the global digital ecosystem, should bear in mind [52]. Among these, accessibility, security, artificial intelligence, non-screen products, and system UX are some trends whose impact on digital products and processes will be prominent.

Less urgent than the first gap, but still important, this gap defines a second angle to take the thesis research further; understanding the evolution of design, engineering, and digital products, so that the future of Agile UX, or "insert new name of the model defining product development workflows", can be better anticipated.

The above two Agile UX gaps, the first one of a primary importance, the second one of a secondary importance, represent the base of the problematization which will drive the coming empirical research. They will allow the author to formulate research questions to be addressed (Chapter 3). These research questions will then be answered in the actionables chapter (Chapter 4).

Summary

This Web review has complemented and extended the foundations laid by the previous literature review (chapter 1) on the topic of digital product development workflows and design-engineering collaboration. First, six major challenges in combining UX design (UX) and Agile software development (ASD) have been detailed. As a solution in response to those challenges, the author has provided a detailed definition of the modern approach of Agile user-experience design (Agile UX). This definition first went through an overview of the approach, its goals, and application. Then, eight Agile UX cores have been formulated, describing the values, principles, and key aspects of the approach. Five ongoing trends, supporting the Agile UX vision, have been laid down, to complementing this definition. Finally, two major Agile UX gaps have been depicted, giving the direction for the emprical part of this thesis research.

Chapter 3
Insights from industry practitioners

Introduction

The background research, encompassing literature and Web reviews, has provided on one hand a deep understanding of the challenges between design and engineering in terms of collaboration and processes in product development, and on the other hand a complete comprehension of Agile user-experience design (Agile UX), the modern approach aiming at reconciling these two disciplines in order to alleviate those challenges. The aim of the background research was not only to shed the light on Agile UX but also to depicting its current weaknesses and gaps; in other words, areas of improvement.

This third chapter aims at addressing these gaps, by collecting insights from industry practitioners. Overall, this empirical research addresses the main research question of the thesis: "How to contextually apply an Agile UX approach to product development, today, and tomorrow?". To answer this question, the author is interested in understanding the different product development contexts within the industry. More specifically, the author investigates the current level of Agile UX approach to product development and the state of the design-engineering relationship in different organisations, and how processes and collaborations models are defined and managed. Additionally, he tries to get a glimpse on how will product development and digital products evolve in the future. The author relied on one single method for data collection, being semi-structured interviews. 24 research respondents, designers, developers, and managers, coming from various organisations were interviewed.

The first section of the chapter presents the method for data collection. It includes the definition of the research questions, the rational for choosing semi-structured interviews as unique research method, an overview of the research respondents, details on how the interviews were run, and how the data was documented. The second section presents the method for data analysis, being content analysis and affinity mapping. The third section presents the findings of the empirical research depicted in four clusters, being product development contexts, processes and collaboration for designers and developers, people's perspectives on processes and collaboration, and the future of product development, design, and engineering.

The interpretation and actionables of the empirical study are presented in the next and last chapter; Research actionables.

1. Data collection

1.1. Defining the research questions

The empirical research builds on the two Agile UX gaps detailed in the background research, at the end of the Web review (Chapter 2). The first Agile UX gap is about the lack of practical instructions to apply the Agile UX model and a poor attention to industry's contextuality. The second Agile UX gap describes the lack of anticipation of the model to the evolution of the fields, design and engineering, as well as digital products overall. Based on these weaknesses, this empirical research is driven by one main research question, itself depicted in three sub-research questions.

The main research question is the following:

How to contextually apply an Agile UX approach to product development, today, and tomorrow?

The three sub-research questions breaking down the main research question are the following:

1) What is today the level of Agile UX approach to product development, and the state of the relationship between design and engineering in the industry?
2) How are processes and collaboration models defined in contextualized product development?
3) How will product development processes and collaboration models be affected by the evolution of the design and engineering disciplines, as well as the technologies, in the future?

The main research question aims at unifying the research direction and making it digest to bear in mind. The research questions 1 and 2 adress the first Agile UX gap about the lack of practical instructions and attention to contextuality, being the most important gap. The research question 3 adresses the second Agile UX gap about the lack of anticipation to the future.
The aim of this empirical study is two come up with answers to these research questions, and actionables to adress the two Agile UX gaps, ultimately contributing to improving product developement processes and design-engineering collaboration.

1.2. Semi-structured interviews

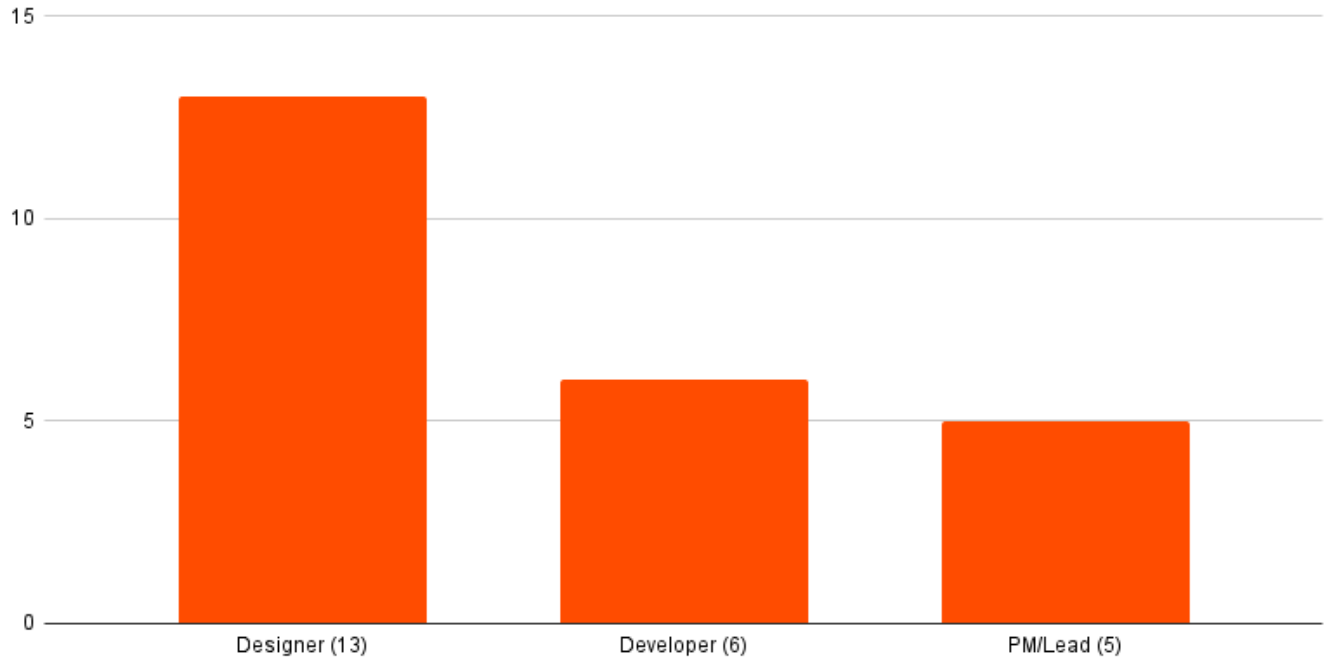
For this empirical research, the author wanted to look at the industry and talk to practitioners, people who can tell about how designers and developers really work together. The aim was to listen to experiences from different roles, coming from different companies, from different fields. Overall, various contexts needed to be studied. To capture the state of the design-engineering relationship in real-life, how things work in different environments, and investigating on how things will evolve in the future, the research method needed to be qualitative and to allow for deep insights to be uncovered. Moreover, the author wanted to listen to as many industry practitioners as possible. Considering the scope and time resource limitation of this thesis work, and for simplicity purpose, the choice to relying on one single research method made the most sense for the author. With all these considerations, semi-structured interviews appeared as the natural choice as to the research method to rely on.

In preparation to the interviews, an interview guide was drafted by the author. This guide was devided in three main sections based on the three research questions detailed above. The guide contained a total of 17 questions, out of which two were introduction questions. The complete interview guide is available in the Appendix section, at the end of this thesis.

1.3. The research respondents

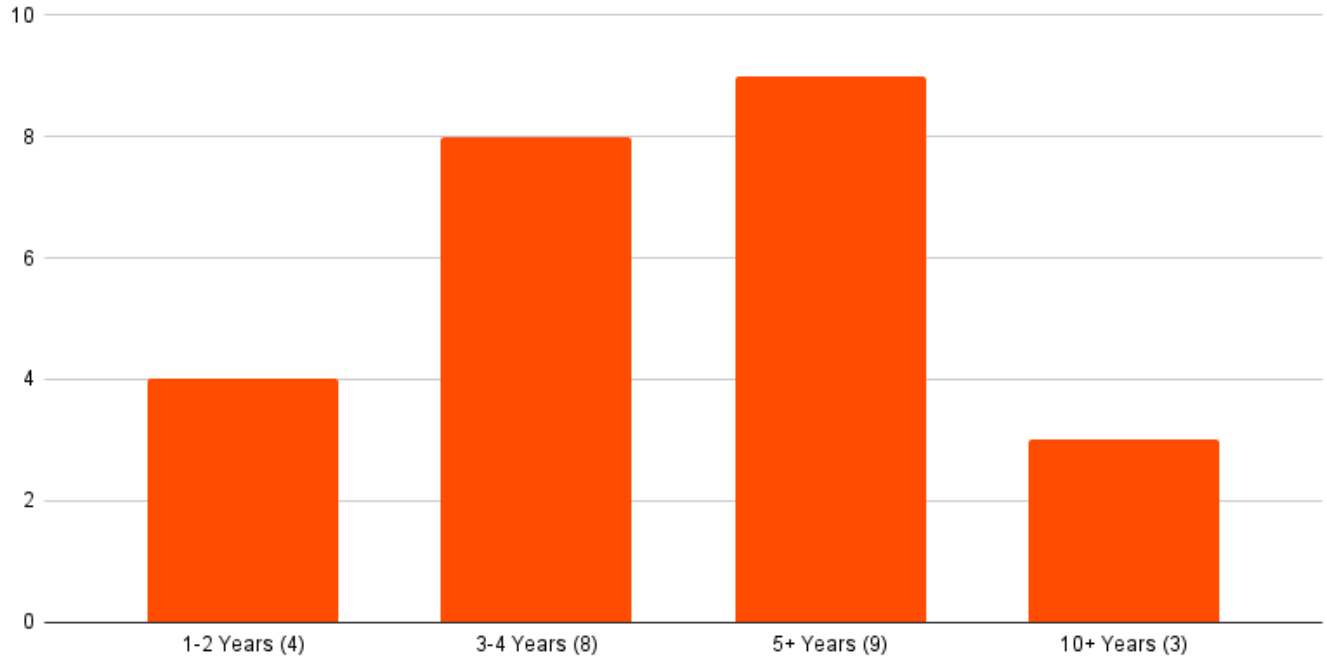
Respondents' profiles

The author reached out to a total of 24 industry practitioners; 13 UX designers, 6 software engineers, and 5 project managers or leaders. The aim of interviewing designers, developers, and product managers/ leaders was to collect insights from all the major perspectives of digital product development. The choice of interviewing this amount of people was not only made to compensate the mono-method nature of the research, but also by the willingness of the author to capture as accurately the state of the industry as possible. Additionally, the respondents all had varying level of experience and seniority within their organisation, which made insights even more rich. The three following graphs summarise the respondents profiles of this thesis empirical research.



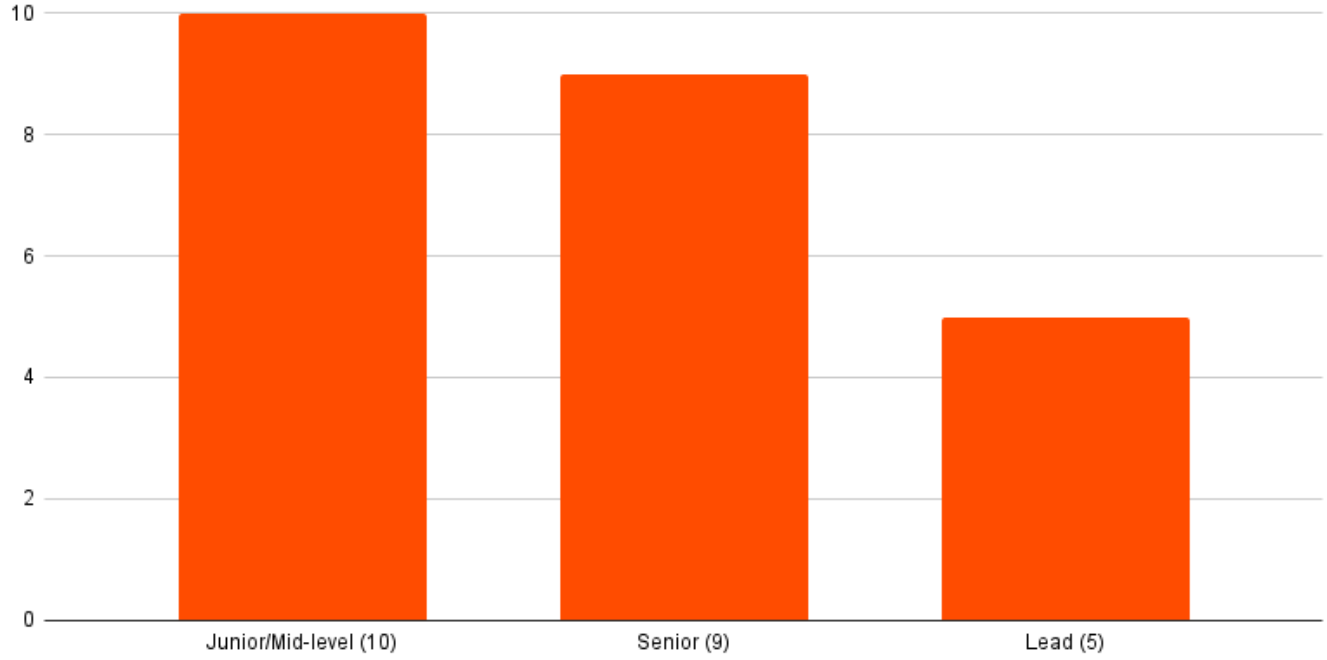
Distribution of respondents' role

The design's perspective was much more represented in this empirical study, considering that project managers (PM) and leaders are mostly design focused as well.



Distribution of respondents' experience in their role

The majority of the research respondents are experienced in their domain of expertise and digital product development. Only 4 out of the 24 respondents could be considered as beginners, with 1 to 2 years of experience. Half of the respondents are very experimented, with over 5 years of experience, of which 3 have over 10 years of experience.



Distribution of respondents' seniority within their organisation

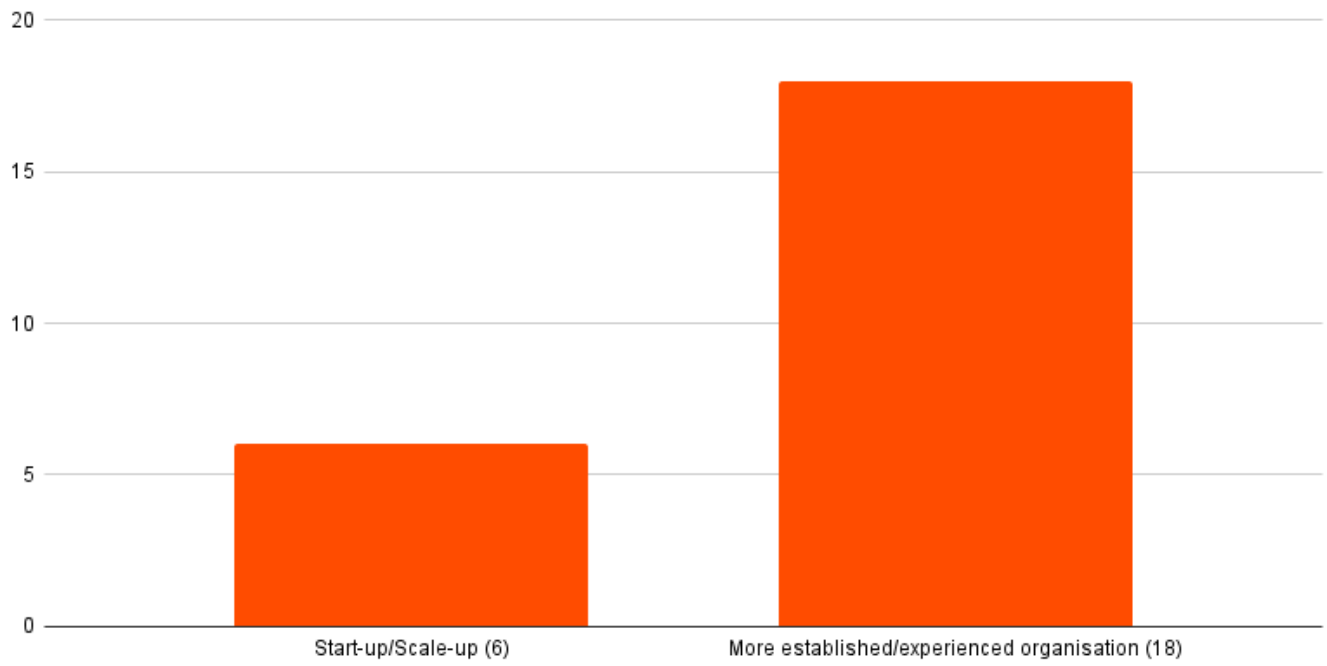
To be distinguished with the years of experience, the seniority of the respondents within their related organisation allows to understand their status in the hierarchy, and aptitude to influence and take decisions. As the graph shows, over half of the respondents are senior or lead roles within their company, and 10 out of the 24 respondents are junior or mid-level roles.

Organisations

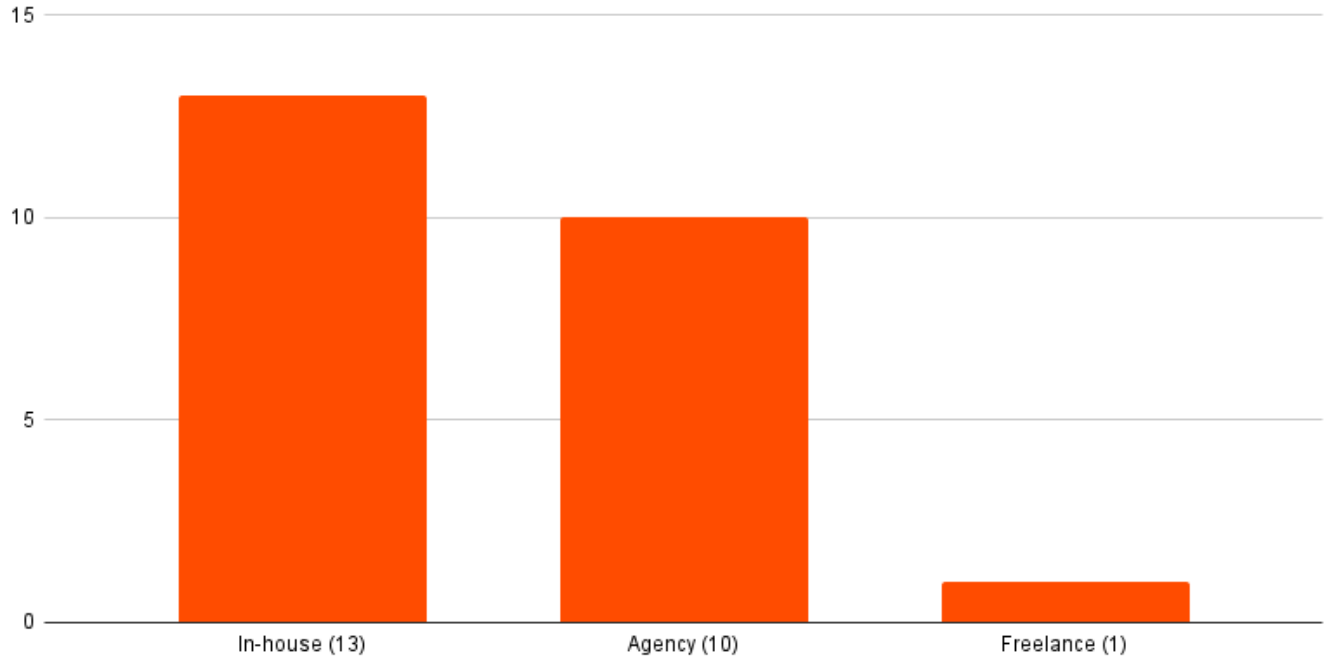
The research respondents came from various organisations. 24 organisations were represented for 24 respondents. Not only hearing from many people, but under different types of organisations, being agencies or product companies, start-ups or established companies, allowed to picture different contexts and how they influence on the relationship, processes, and collaboration between designers and developers.

The many organisations represented by the respondents depicted a wide array of fields, from the Fin-tech to the telecommunication industry, or the transportation to the cyber-security industry. Similarly, the stage of the companies in their development process differed a lot. For the sake of simplicity, a split was made between the start-ups/scale-ups companies on one hand, and the established/experienced companies on the other hand. A distinction between the organisations was also to be made between in-house businesses and agencies. In-houses companies were a bit more represented in this research. One respondent made an exception on that matter, being a freelance manager working for an in-house company. A majority of the companies' business type was business to business (B2B), whereas 8 of them were business to customer (B2C). Finally, the respondents were based in 4 different countries, Finland being largely represented, with 20 respondents, followed by India with 2 respondents, and Japan and Germany, with 1 respondent each.

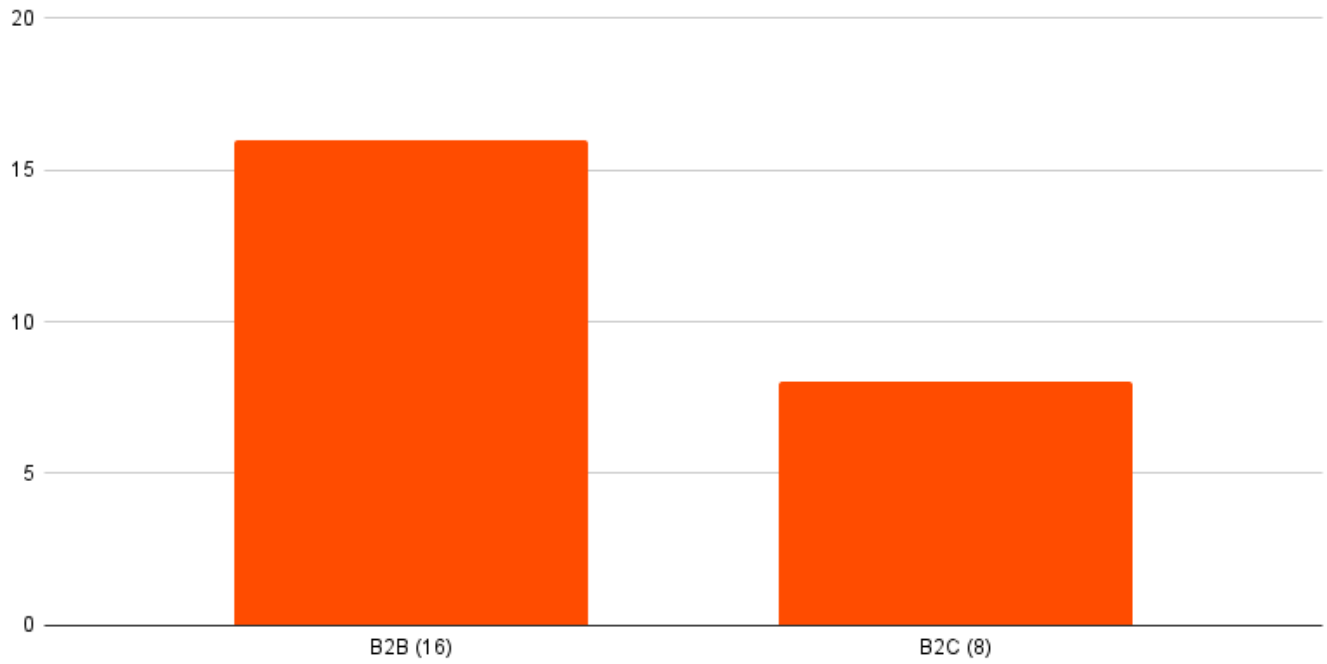
The graphs below summarise the attributes of the organisations represented in this empirical research.



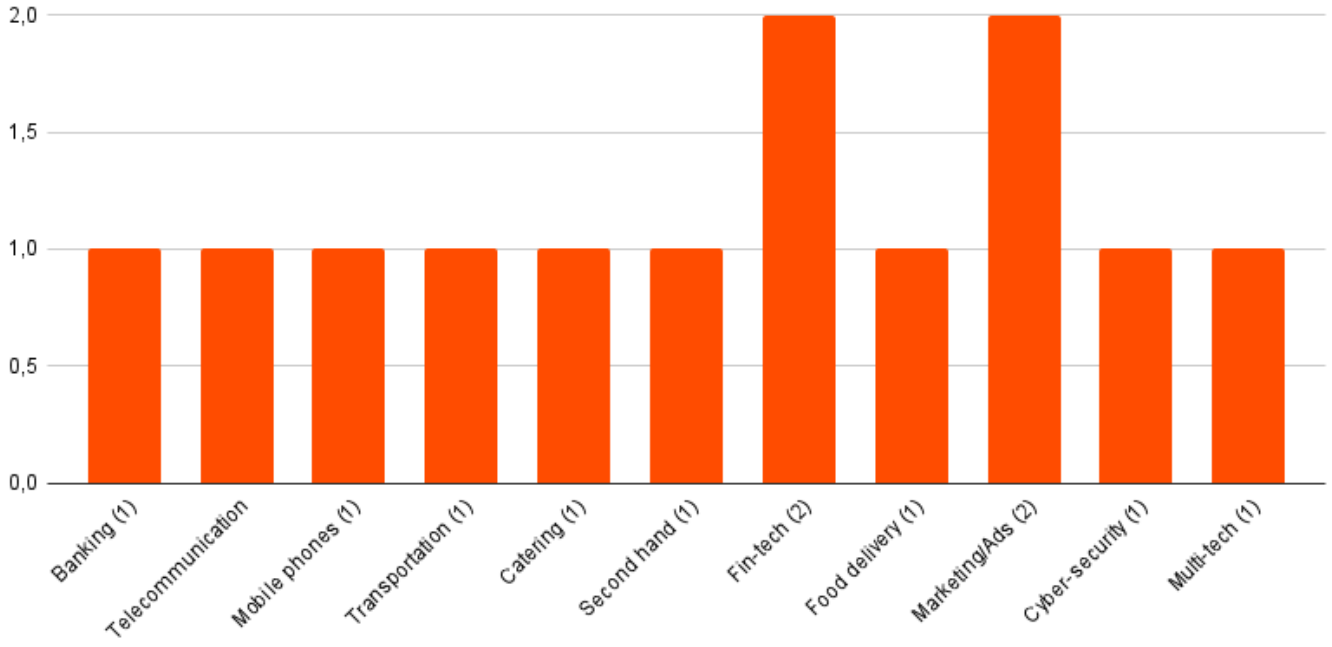
Distribution of organisations' development stage



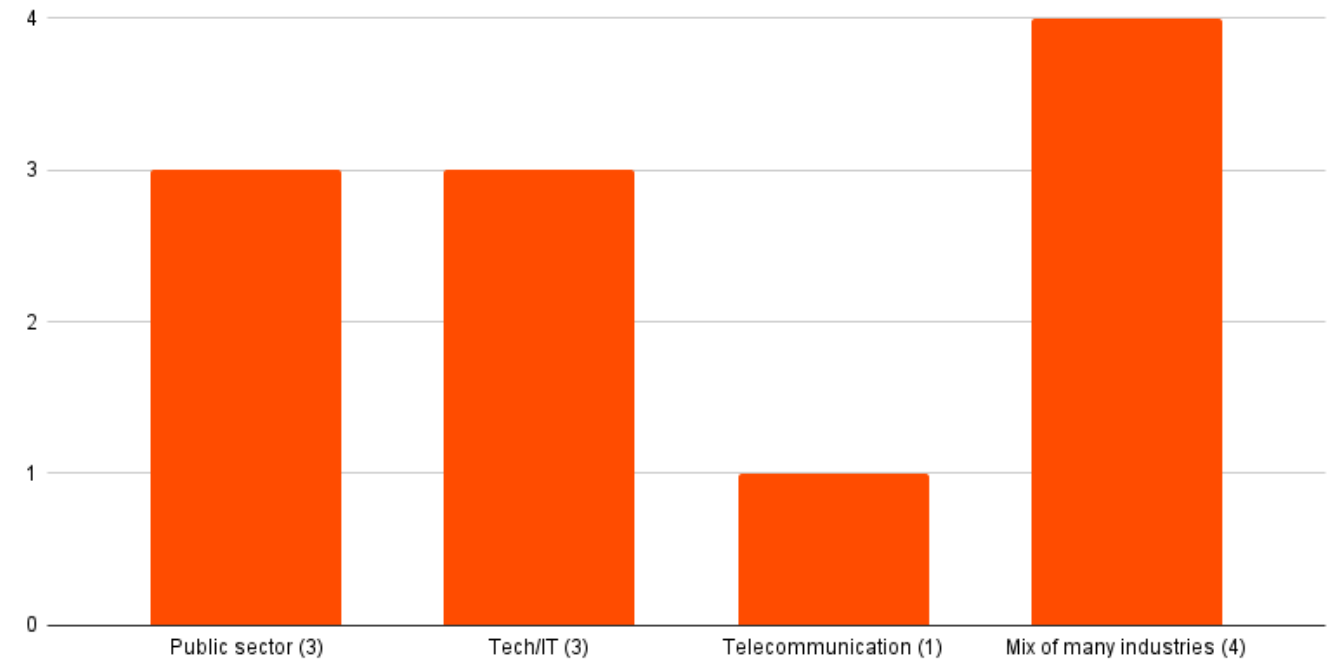
Distribution of organisations' type



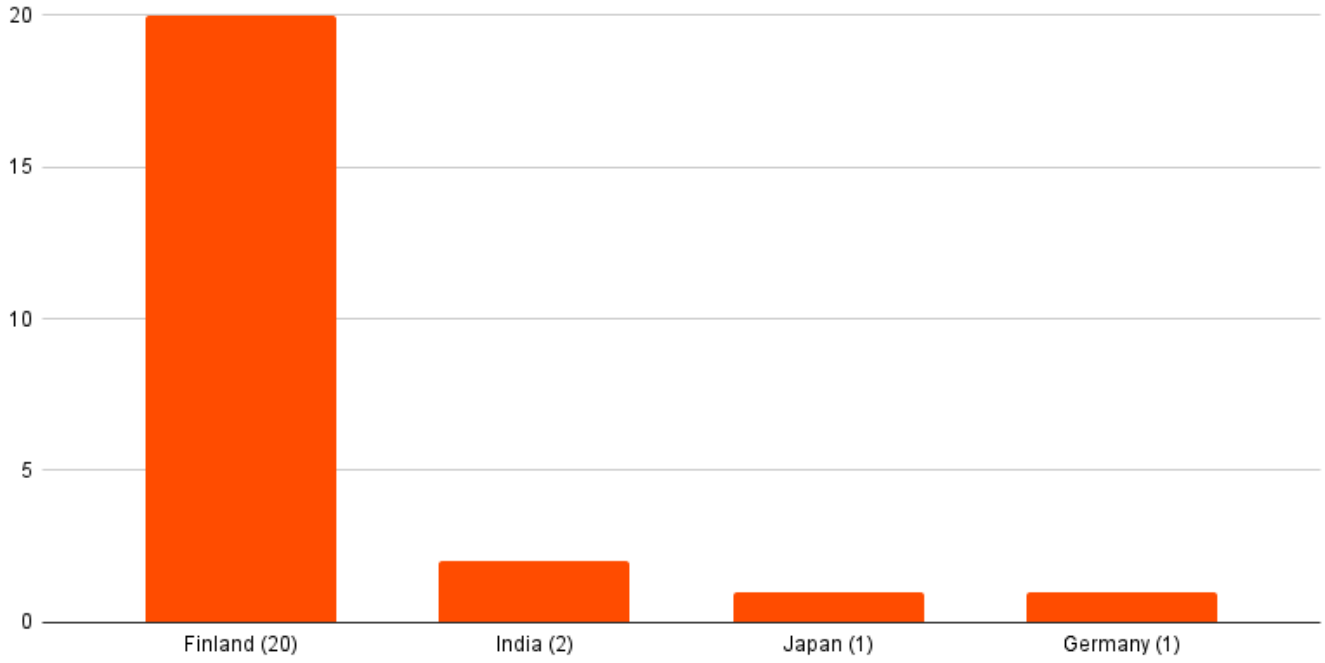
Distribution of organisations' business type



Distribution of organisations' operating field (for In-house organisations)



Distribution of main clients' operating field (for Agency organisations)



Distribution of countries where respondents are based

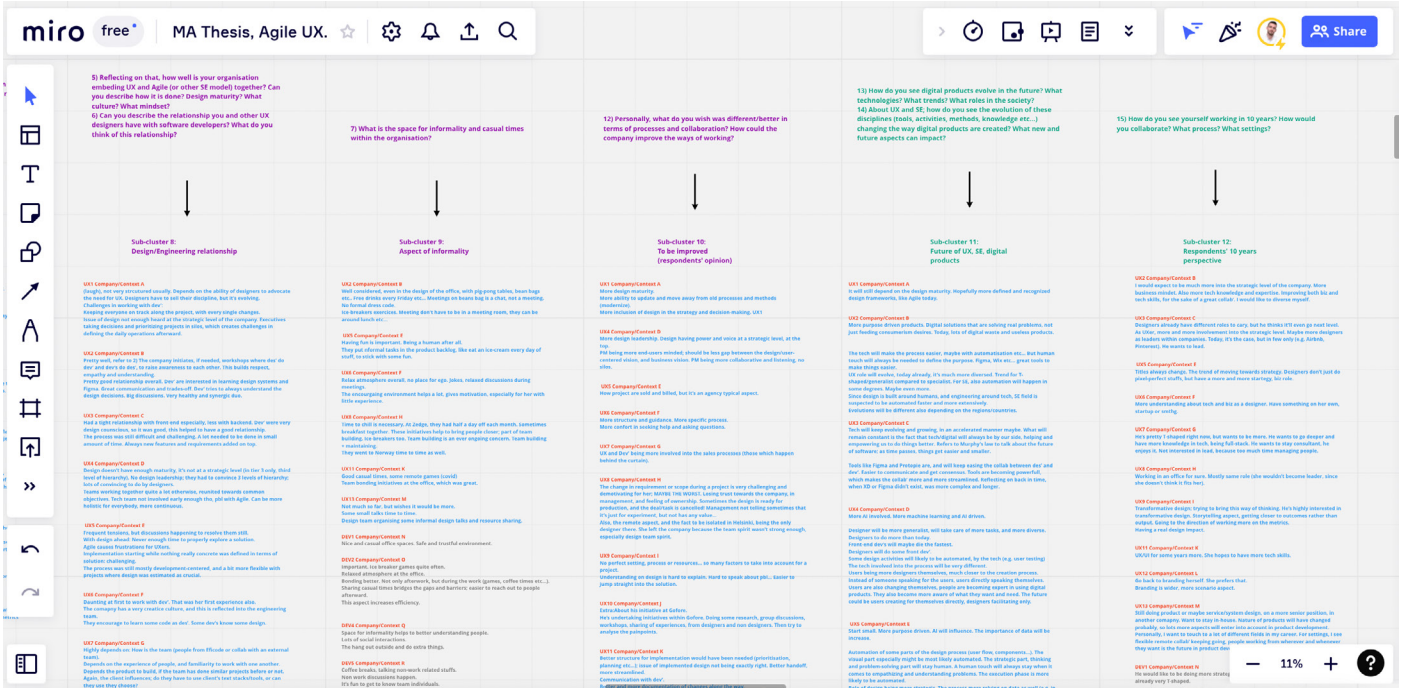
1.4. Running the interviews

The semi-structured interviews were conducted in April 2021, in a fully remote way. This is mostly due to the Covid situation at the time. Additionally, this choice made the 24 interviews convenient and costless to run. The interviews were moderated by the author himself, using the interview guide (Appendix). The author followed the guide as closely as possible, but kept some level of freedom and flexibility in the moderation, for example when specific follow-up questions needed to be asked, or questions to be cut out because of time constraint. The tool used to conduct and record the interview sessions was Zoom. On average, an interview session lasted 45 minutes.

1.5. Documenting the data

The qualitative insights from the semi-structured interviews were collected using a detailed note-taking method. Under verbal agreement of the interviewees, each interview session was recorded. The author took preliminary notes of respondents' insights while running the interviews. Afterwards, each session was reviewed in detail with the recordings in order to lay down the most complete and accurate notes. The author made the deliberate choice not to rely on word by word transcription, which considering the amount of interviews (24) and length of each session (45min on average), would have been time costly. Moreover, video recordings were estimated as more reliable in the sense of cristalising answers with respondents' reactions, tone, and body language, which word by word transcripts cannot do.

Once detailed notes were written down, Miro was used as a tool to organise the insights. The author copy-pasted notes of each respondent's answer and laid them down together under each interview question. Notes related to extra insights were put in a seperate column. The screenshot below gives a snapshot of how notes were visually aranged on Miro.



Arrangement of interview notes on Miro

For the sake of privacy, anonymisation of the respondent and organisations was made. The table below details the anonymised names given to the research respondents and their respective organisation.

Respondent	Organisation/Context	Organisation Type	Business Type	Development Stage	Field/Main clients' field	Country
UX designer 1 (UX1)	Company/Context A	In-House product	B2C	Established/Experienced	Banking	Finland
UX designer 2 (UX2)	Company/Context B	In-House product	B2C	Established/Experienced	Fin-tech	India
UX designer 3 (UX3)	Company/Context C	Agency	B2B	Established/Experienced	Mostly Tech/IT	Japan
UX designer 4 (UX4)	Company/Context D	In-House product	B2C	Established/Experienced	Second-hand	India
UX designer 5 (UX5)	Company/Context E	Agency	B2B	Established/Experienced	Mostly public sector	Germany
UX designer 6 (UX6)	Company/Context F	Agency	B2B	Established/Experienced	Many industries	Finland
UX designer 7 (UX7)	Company/Context G	Agency	B2B	Established/Experienced	Many industries	Finland
UX designer 8 (UX8)	Company/Context H	In-House product	B2C	Start-up/Scale-up	Mobile Phone	Finland
UX designer 9 (UX9)	Company/Context I	In-House product	B2C	Start-up/Scale-up	Transportation	Finland
UX designer 10 (UX10)	Company/Context J	Agency	B2B	Established/Experienced	Mostly public sector	Finland
UX designer 11 (UX11)	Company/Context K	In-House product	B2C	Start-up/Scale-up	Catering	Finland
UX designer 12 (UX12)	Company/Context L	In-House product	B2B	Established/Experienced	Cyber-security	Finland
UX designer 13 (UX13)	Company/Context M	In-House product	B2B	Start-up/Scale-up	Fin-tech	Finland
Developer 1 (DEV1)	Company/Context N	Agency	B2B	Established/Experienced	Mostly Tech/IT	Finland
Developer 2 (DEV2)	Company/Context O	Agency	B2B	Start-up/Scale-up	Many industries	Finland
Developer 3 (DEV3)	Company/Context P	Agency	B2B	Established/Experienced	Many industries	Finland
Developer 4 (DEV4)	Company/Context Q	In-House product	B2B	Start-up/Scale-up	Advertising/Marketing	Finland
Developer 5 (DEV5)	Company/Context R	Agency	B2B	Established/Experienced	Mostly Tech/IT	Finland
Developer 6 (DEV6)	Company/Context S	In-House product	B2B	Established/Experienced	Advertising/Marketing	Finland
Project Manager/Lead 1 (PM/LEAD1)	Company/Context T	Freelance	B2B	Established/Experienced	Mostly telecommunication	Finland
Project Manager/Lead 2 (PM/LEAD2)	Company/Context U	In-House product	B2C	Established/Experienced	Food delivery	Finland
Project Manager/Lead 3 (PM/LEAD3)	Company/Context V	In-House product	B2B	Established/Experienced	Multi-tech	Finland
Project Manager/Lead 4 (PM/LEAD4)	Company/Context W	Agency	B2B	Established/Experienced	Mostly public sector	Finland
Project Manager/Lead 5 (PM/LEAD5)	Company/Context X	In-House product	B2C	Established/Experienced	Telecommunication	Finland

Anonymised names of research respondents and their respective organisation

2. Data analysis

2.1. Content analysis

The method used for analysing and making sense of the qualitative data collected throught the semi-structured interviews was content analysis.

Rational behind the method

As detailed by Marsh (2018), content analysis is a method for categorizing and codifying qualitative data in order to making sense of it. This method is also defined as a way of quantifying qualitative data. Beyond its definition, the author made the choice of relying on content analysis due to its particular and renowned efficiency for dealing with a large amount of qualitative data, which was the case in this empirical research. Additionaly, Marsh (2018) describes content analysis as suitable for treating data coming from semi-structured interviews, and when the analysis is undertaken by a sole individual.

Content analysis presents additional advantages, which have influenced the author's decision. Firstly, this method is great for uncovering notes and text, identifying terminologies used, and determining the frequency of occurence of categories and themes. Content analysis is also particularly cheap to undertake, and doesn't require the use of complex tools. Moreover, Marsh (2018) describes a high level of reliability of this method. Finally, althought this research mostly relied on detailed note taking, content analysis is adapated to cover imagery, video, and audio materials.

6 steps followed for data analysis in this thesis

STEP 1: Getting familiar with the data, by listening to the recordings and reading the detailed notes.

STEP 2: Organising the different datasets on Miro (clusters and sub-clusters) based on the research questions. Each research question was analysed independently. At this point, the author highlighted notes in different colors for dinstinguishing insights coming from designers, developers, and managers.

STEP 3: Identifying the relevant insights and creating categories and themes (independently done within sub-clusters).

STEP 4: Organising categories and themes based on their importance. Reviewing categories and themes within the sub-clusters.

STEP 5: Interpreting and reflecting the results. Establishing connections between, sub-clusters, categories, and themes. Connecting the findings of the empirical research with the backgroung research (Detailed in Chapter 4: Reserarch findings and actionables.)

STEP 6: Creating actionables (Detailed in Chapter 4: Reserarch findings and actionables.)

2.2. Clusters and sub-clusters

The data analysis process allowed the author to transform rough qualitative notes into themes, themselves organised within clusters and sub-clusters. Four clusters and twelve sub-clusters compose the result of this empirical research, which are detailed in the next part (3. Findings).

The clusters and sub-clusters presented here are based on the sub-research questions and interview questions (Appendix). They were properly defined and named while organising the data after collection. Within each of these sub-cluster, the analyis was data driven, and themes emerged based on the data.

Cluster 1: Product development context

Sub-cluster 1: Organisation attributes

Sub-cluster 2: Organisation settings

Sub-cluster 3: Projects and tasks

Cluster 2: Processes and collaboration for designers and developers

Sub-cluster 4: Processes and collaboration

Sub-cluster 5: Freedom and flexibility

Sub-cluster 6: Design-Engineering relationship

Sub-cluster 7: Aspect of informality

Cluster 3: People's perspective on processes and collaboration

Sub-cluster 8: To be improved in the organisation

Sub-cluster 9: Conception of a successful project

Sub-cluster 10: Aspects influencing processes and collaboration

Cluster 4: The future of product development, design, and engineering

Sub-cluster 11: Future of UX, SE, and digital products

Sub-cluster 12: Respondent's 10 years perspective

3. Findings

This third section of Chapter 3 presents the detailed findings of the empirical research, depicted in four clusters and twelve sub-clusters. The interpretation and actionables of these findings are presented in the next and ultimate chapter; Research actionables.

3.1. Cluster 1: Product development context

This first data cluster provides insights on understanding the uniqueness of each product development context and the diversity of the industry. It's composed of three sub-clusters: Organisation attributes, Organisation settings, and Projects and tasks.

Sub-cluster 1: Organisation attributes

This data sub-cluster refers to the overall attributes about the respondent's organisations, which constitute a first level of granularity for understanding product development contexts.
This information is already detailed in 1.3., and shows the diversity in the organisations involved in this research. The organisations are described under the following factors: Development stage, organisation type, business type, country, and field of expertise.

Sub-cluster 2: Organisation settings

This data sub-cluster refers to the respondent's insights on their organisation's settings. The settings constitute a second layer for understanding product development contexts.

Generic insights:
Few generic insights about the settings have been mentioned by the respondents.
One designer pointed out a lack of design maturity, whereas two project managers/leads in contrary found the design maturity relatively decent. One designer indicated their particular position of being the first designer in the company. Multiple designers and developers mentionned about the design ratio being much more inferior than the engineering ratio in their organisation. Developer 3 for instance specified that their company had 300 developers for 20 designers.

Recent changes in organisation:
The respondents pointed out organisation's recent changes being an influencial factor on the settings.
Such examples of organisational change are company acquisition, mentioned by Developer 1, a growing adoption of the Agile model described by Developer 3 and Developer 5, or an overall company shift from product-driven to service-driven mindset facilitated by design roles, evoked by Project Manager/Lead 3.
The biggest source of change mentioned by the respondents has been the company growth itself. One designer, one developer, and one project manager/lead described their company growth as a trigger to an evolving environment, tending to be more structured, with more interdependent teams, and rising responsibilities among members.

Team types and nature:

The biggest aspect of the settings depicted by the respondents has been the team.

Most of the respondents, 15 out of 24, described their team, and/or teams they were involved in, as cross-functional, and being composed of various roles. These cross-functional teams have the similarity of embedding multiple perspectives, usually all the three design, tech, and business together. Some teams were described as mainly unifying design and engineering, with less emphasis on business. Despite this common point, the cross-functional teams also have varying attributes in different organisations. For example, respondents in small companies tented to refer to the entire organisation as being one unified cross-functional teams. When it comes to bigger companies, the scope of the team seemed to be defined per project, or product area. The composition of these cross-functional teams also varies a lot from one company to another, and one project to another. One designer from an agency depicted a typicall project team to be composed of 1 business lead, 1 or 2 project managers, few developers, few data analysts, and few designers. Another respondent, design lead in-house, works in a team of 6 engineers, and 2 data scientists. Roughly, the size of the cross-functional teams varies from 5 people to 20 people. The role ratios within these teams are uneven. Some teams almost have an equal number of designers and developers, whereas others, as a designer in an agency pointed out, include 1 designer alone with a team of engineers. Based on the respondents, it seems most common to have more engineering than design resources in a cross-functional team. A major attribute of cross-funtional teams specific to agency organisations is the cross-organisational aspect of the team. This refers to the team being composed of people from different organisations, the agency, the client company, and potentially other agencies and/or stakeholders companies. One respondent, a developer, has 4 different organisations involved in their team.

Some respondents, 9 out of 24, pointed out their organisation making a specific distinction of design specific teams. In these organisations, one or multiple teams are purely focused on UX work. The size of these design teams ranges from 3 people in a start-up, to over 30 people in a big established company. Although these organisations make this distinction of design teams, it doesn't prevent cross-functional teams to be formed, and designers to be connected to different teams across the organisation. As Project manager/Lead 5 highlighted, team layouts can be complex. To be noticed with design specific teams is that 7 out of the 9 respondents who provided insights on that are from in-house companies.

On the other hand, 4 respondents described engineer specific teams. In most cases, tech specific teams split their focus between platform types (mobile, desktop) and/or front-end and back-end. Here again, the size of the teams varies depending on the company. Simliarly as the design specific teams, engineer specific teams have mostly been described by respondents from in-house companies.

Few respondents provided insights on their team forming process, which applies to cross-functional teams in agency companies or big in-house companies. A common trend can be highligted; new teams are created based on a need for a given project, and people are choosen by their experience, level of familiarity with the topic, and availability. One designer said that project managers usually join a project first, and have the duty to select designers and developers to work with. In another case, a developer highlighted that executives are in charge of forming the entire team. A project manager/lead affirmed new teams to be formed after having an initial research, initial concept, and financial support allocated for the project.

One extra insight on teams has been provided by one designer and one developer, who stated being in the same team for a long time. They pointed out the fact that it gets easier to work with team members over time, once getting familiar to them.

Structure:
Another aspect of settings depicted, highly connected to the team, has been the company structure.
On that matter, 7 respondents mentioned a division of work in their company per product area. A good example to illustrate that comes from UX Designer 2, being from an established in-house company, who described their structure as a system of cross-functional sub-teams, called «pods», each one focusing on different aspects of the company product. This pattern seems to be similar for most respondents who highlighted this division of work. In most cases, these sub-teams are all different in composition, and are formed by people from different departments (design, engineering, analysts, business...). UX Designer 5 also mentioned dedicated project managers being responsible for each product vertical.

Another trend highlighted on the structure by 7 respondents was the distinction of specific design and engineering departments. Obviously, these two departments don't stand alone within these organisations, and co-exist with other units, for instance product, business, marketing, sales, and analytics. UX Designer 7 mentioned the split of design and engineering departments to be justified by the simplicity of management.

An extra note on the matter has been provided by UX Designer 5, who highligted that settings in their company can be different from one department to another and from one team to another, which emphasizes the influence of the company structure on the settings.

Hierarchy:
Few respondents provided insights on the aspect of hierarchy in their company.
3 respondents, all from agencies, mentioned having a flat hierarchy in their organisation. They described this flat hierarchy as working with people devoid of ego, project teams being self-organised, autonomous, and where roles have big space for decision making. One designer mentioned their organisation not being completely flat, but tending to be, with more and more equal decision making and less marked verticals.

3 respondents highlighted distinctions between junior and senior roles within their company. UX Designer 13 mentionned their design lead to be in charge of giving junior roles the tasks to work on. UX Designer 6 and Developer 6 shared the same words, and explained junior roles to work on less and shorter projects, whereas senior roles work on longer and more holistic tasks. Junior roles also seem to be embeded in one team, while senior roles are more likely to be involved in multiple sub-teams.

The most complete description about company hierarchy came from UX Designer 4, who depicted it in three decision making bodies. The primary body involves business, engineering, and product leads. The second body involves product, project, engineering, and design managers. The third body involves execution roles, designers and developers.

Location:
The last aspect of settings highlighted by the respondents has been the location.
Most respondents on that matter described a mix of office and remote work settings. Often, remote work is necessary due to the company being split across multiple countries, even different time zones. In the case of UX Designer 6, not only the company is based in different locations, but project teams themselves. UX Designer 8 described being «isolated» in the Finnish office of the company, with few people, while the headquarters are in Norway. Few respondents from agency companies mentioned that projects can require to be at the client's company, but not always.

Out of 24, only two respondents admited that designers and developers are physically colocated on a regular basis. Both of these organisations are established in-house companies.

Sub-cluster 3: Projects and tasks

This data sub-cluster refers to the respondent's insights on their projects and tasks which constitutes a third layer of product development context. Two major types of product development projects have been distinguished; continuous projects, and finished projects.

Continuous projects:
The author defines here continuous project as long-term or infinite development of a digital product, service, or system. In continuous projects, designers, developers, and other roles in the team, work on an ever evolving product to be maintained and enhanced over time. Out of the 24 interviewees, 16 work on a continuous project within their organisation. Most of these respondents are from in-house companies, and 3 of them make an exception in agencies, being allocated to their client project for a long-term period. Within these continuous projects, the work of respondents is divided in smaller tasks, which can vary in size and length, depending on their scope and goal(s). UX Designer 13, for instance, affirmed that one or two months are needed to get proper design outcomes, and big tasks focused on major improvements or features can take up to three months. UX Designer 2 mentioned smaller tasks no longer than two weeks, while Project Manager/Lead 2 described tasks potentially going up to six months. The respondents affirmed usually working on mutiple tasks at the same time. Two major types of tasks can be distinguished within these continuous projects: new features from one hand, and improvement or fixes of existing features/aspects of the product on the other hand. These continuous projects and tasks vary a lot based on the company, but especially based on the product, its size, its complexity, its field, and its users. Some respondents are involved in a mix of both desktop and mobile breakpoints, as UX Designer 10 for example, and some only focus on mobile or desktop, as UX Designer 1 or UX Designer 13. There are also variations of the tasks in terms of scope. People from start-up companies tend to have a more holistic view on the product, which usually is smaller and less complex, whereas roles in bigger and more established companies seem to focus on one area or aspect of the product. For example, UX Designer 3 focuses on the B2B side of their company product, and Project Manager/Lead 5 focuses on one specific service area of their company offering.

Finite projects:
The author defines here finite project as short-term or medium-term development of a digital product, service, or system. Out of 24 respondents, 8 are involved in finite projects. All of them, 4 designers, 3 developers, and 1 project manager/lead, come from agency companies. Since working in an agency inherently induces consulting for clients, that justifies the finite nature of projects designers and developers are involved in. Nevertheless, as said above, 3 respondents from agencies were included in the continuous type of project, which demonstrates that although finite projects likely apply to agencies, agencies aren't limited to this scope. The nature of these finite projects vary from one respondent to another based on their client and their goals. For instance, Project Manager/Lead 1 mostly does design system consulting, Developer 2 focuses on websites, UX Designer 7 does a mix of product and design system consulting, and Developer 3 focuses on Web and mobile applications. The length of these finite projects are mostly a matter of weeks or months, and can go up to one year for the longest. Developer 2 also made a distinction between projects being renewals of an existing product, and the ones that require to build a product from scratch, which are usually longer and involve design and engineering more equally.

3.2. Cluster 2: Processes and collaboration for designers and developers

This second data cluster provides insights on understanding the state of processes and collaboration for designers and developers, as well as their relationship in the industry. It's composed of four sub-clusters: Processes and collaboration, Freedom and flexibility, Design-Engineering relationship, and Aspect of informality.

Sub-cluster 4: Processes and collaboration

This data sub-cluster, being the core sub-cluster of cluster 2, as well as the bigger sub-cluster of this qualitative research, refers to the respondent's insights on the processes and collaboration models in place in their organisation, which constitutes a first and main layer for understanding ways of working and interacting between designers and developers.

The nature of the process:
Firstly, the insights brought-up different aspects describing the nature of the process.
A trend which has been highlighted by 11 respondents is the fact that process is usually project or task specific. For in-house companies, the process employed for a given sub-project or task highly depends on the work to be done, its scale and importance. A task could be to develop a new feature or improve an aspect of the product; already these two types of tasks induce big distinctions in the process in terms of the level of research needed, or the amount of development, to give two examples. Developer 5 and UX Designer 2 both highlighted that the more important change is to be done on the product, and the newest a feature is, the more people will be involved in feedback, which inherently impacts the process. In agency companies, each project also seems to bring different requirements for the process to follow. UX Designer 10 affirmed not having determined processes to be replicated for each project, and design to be done differently each time. Based on the goals and requirements, the length of a project or task seem to impact on the nature of the process as well. For instance, Developer 3 makes a difference between the process employed for small and big projects, with bigger projects taking the same steps as the smaller ones but iterated multiple times. In agencies, the client seems to impact on the process too. For instance, Developer 1 mentioned that the understanding and recognition of design by the client will influence on the product development process and the balance between engineering and design work.

Another element adressed by 1/3 of the respondents was the timeline. The timeline is considered as an important aspect of the process, mostly consisting of key project milestones, deliverables, and timetable. In agency companies, the timeline is also a core element of the contract to be negociated with the client. Based on the respondents, this timeline is drafted at the outset of the project with some level of flexibility in order to anticipate unexpectations throughout the project or task. The drawing ritual of this timeline varies from one company to another. UX Designer 6 for example stated that timeline is defined by all the team with the client, whereas few other respondents mentioned that timeline is usually drafted by project managers or company executives. In short, the timeline is seen as a high-level and flexible element allowing to guide the product development team and drive the process.

Out of 24 respondents, 16 mentioned Agile as the main driver for the product development process within their organisation. These 16 companies are a perfect mix between agency and in-house companies. By describing their Agile ways of working, respondents often highlighted similar trends. The Scrum framework was evoked by half of them, and the degree of flexibility to which this method is applied varies from one organisation to another. On one hand, some companies seem commited to following Agile processes as closely as possible, and on the other hand, some don't hesitate to customise the model, as in the case of Developer 5 who mentioned not working in sprints, or Developer 4 who described their processes as home-made mix of Scrum and Kanban methods. Designers who expressed themselves on the topic detailed about the way their organisation fit design processes into Agile. In these organisations, designers sometimes follow the same sprint model as their tech peers. UX Designer 5 and UX Designer 8 for example both described the fact that UX related work should be produced within each sprint. In some cases, designers borrow the same model of ticket for task assignment, using common or parallel boards with developers. Three respondents mentioned designers and developers to be using the same tool for task management, Jira. Other Agile components, such as product backlog, sprint planning and retrospectives have been highlighted, as well as the fact that design work is done ahead of development.

A trend that has been expressed through respondents' insights, is the free, flexible, and experimental nature of processes. This aspect was mentioned by 14 respondents in total. As UX Designer 6 highlighted, trying to apply a rigid process for every project and task is not optimal, and customizing and experimenting workflows seems capital. On that note, UX Designer 7 described their need for drafting a specific design process template at the begining of a new project. Few respondents mentioned about their organisation experimenting with different methods, as for example UX Designer 4 who depicted their processes as a mix of Waterfall and Agile models, or Developer 5 with a mix of Scrum and Kanban frameworks. Developer 6 described the development process as an open version of the Kanban framework, and the company employing it based on their own needs. Overall, these insights communicated the idea of constant trials and refinements around the product development process. Few respondents also highlighted the level of autonomy and freedom given to them and their team to lead processes. Project Manager/Lead 3 for instance acknowledged that executives in their company don't give much constraint regarding processes and methods. UX Designer 12 also described being completely autonomous to lead their design process, and not being constrained by the engineering sprint model. Respondents seemed overall to acknowledge the fact that high-level structure and preliminary draft of the process are of great help, while making space for parts to be defined on the way is important for managing unexpectancies. Developer 5 illustrated it well, by pointing out that every 2 weeks sprint, although recurrent practices are in place, is always different.

Regarding process, a third of the respondents explicitely acknowledged that a minimum structure and high level phases are of great support for product development. This has been mentioned by all types of research respondents, designers, developer, and project managers/lead. UX Designer 9 pointed out the ease and comfort of setting up a high-level process structure for each project and task to be done. Few designers highlighted the problem and solution spaces, also expressed through the Double Diamond model, as being recurring major aspects of the process for each project and task. Two project managers/leads recognized the necessity for a separate and defined structure to support the design process, design work being harder to put into tickets than engineering work.

4 designers pointed out the design process not being really defined within their organisation. UX Designer 1 for example admitted being let among designers on their own when it comes to processes and practices. UX Designer 8 acknowledged that not clear phases are defined for product development process.

Finally, regarding the nature of the process, some challenges were expressed by the respondents. Developer 1 pointed out the billing model as a source of troubles for a process, as it pre-determines time and resources allocated for design and engineering beforehand, freezing some level of flexibility. PM/Lead 4 mentioned their struggle of having to deal with various tools. Few respondents highlighted the timeline as being a source of challenge. According to UX Designer 6, tight schedule forces the team to make wise choices and sometimes to compromise on the process, as time waste can be costly. The distribution of roles across different projects, therefore different timelines, brings complexity in managing a given process, as UX Designer 12 stated. In agency context, Developer 5 mentioned the split of the team across mutliple companies as a source of challenge as well. Finally, in in-house context, two respondents mentioned documentation and requirements tracking to be difficult to manage within Agile.

Process phases:
Following the nature of the process, insights from the research respondents allowed to depict different phases of the product development process. 7 phases have been described and synthesized by the author: The kick-off phase, the discovery phase, the definition phase, the design phase, the validation phase, the delivery phase, and the follow-up phase.

The kick-off phase, also referred to as initiation, or introduction phase, has been depicted by 19 out of 24 respondents. According to Developer 5 and Project Manager/Lead 1, the kick-off includes the formulation of the project goals, sometimes detailed by epics, as well as the different requirements and their prioritisation. Some respondents mentioned early design drafts and concept framing to be part of the kick-off as well. Overall, respondents affirmed initial requirements to be brought up by the project manager(s), executives, product lead(s), or product owner(s). Following that, the goals and requirements are reviewed and validated by the team, developers and designers included. The involvement of design and engineering in the kick-off phase is overall equal. Indeed, most organisations seem to encourage all fields to take part of the outset of a project or task. Nonetheless, few companies seem to unbalance this ratio. UX Designer 3 described the involvement of developers in the kick-off rituals to be less important than designers, whereas Developer 1 highlighted the opposite. Kick-off rituals vary from one company to another. In the example of UX Designer 13, project teams create concensus arround a document to be reviewed by all departments. In other cases, get together meetings or workshops are more valued. The kick-off phase is also the phase where the scope, timeline, and milestones are defined. Respondents highlighted the importance for all team members to create a solid common ground during this phase. As UX Designer 7 stated, kick-off meetings allow the team to answer all questions before the start of the work. There are few differences to be noted between in-house and agency organisations. In agencies, the client plays an important role in formulating the goals and requirements, in addition of the whole aspect of sale, negociation, budgeting, and resource allocation. For in-house organisations, strategic work is mainly carried out by executives and managers, as UX Designer 12 acknowledged. Another difference can be made for continuous projects as opposed to finished projects, where kick-offs relate to features or product improvements, rather than all new products or solutions. Few respondents, working on continuous project environment affirmed that the kick-off is not a major phase. Project Manager/Lead 3 described kick-offs as unregular and truly considered for important features only. Finally, Developer 3 highlighted a difference between small and longer projects, inherently influencing the length and importance of the kick-off phase itself.

Following the kick-off phase, the discovery phase has been depicted by 11 out of 24 respondents. This phase is also referred to as the research and analysis phase, which designers are usually in charge of planning and running, as UX Designer 2 described. The respondents expressed similar patterns when it comes to the goals of the discovery phase, which are to understand the users, the business, the competition, and ultimately formulating the right problem to solve and need to fulfill. In some companies, as in the case of UX Designer 2 and UX Designer 4, designers run the research in pair with other roles, as project managers or other business representatives. UX Designer 4 also highlighted the involvement of the whole project team for the analysis of the research, developers included. As UX Designer 13 and UX Designer 10 mentioned, this phase is not always done deeply. Indeed, the research has been described by few respondents as quick and short, and only deeply executed when important aspects of the product are concerned, or for brand new features to built. In these other cases, the discovery demands particular time and efforts, which pushes some companies and teams to consider it outside of the Agile cycle and dynamic ways of working. UX Designer 10 said: "The discovery should be the least close as possible to the development deadline ".

Following the discovery phase, the definition phase has been detailed by 5 respondents. The definition phase comes as a conclusion of the discovery phase, as its objective is to make sense of the research insights. Respondents who described the definition phase mentioned problem framing and KPIs definition as the core of the phase. UX Designer 10 highlighted the importance of defining the outcomes of the project from the user's perspective. When it comes to practices in this phase, workshops and get together activities are mainly evoked. UX Designer 4 for instance described the way their company involves all key people in user story creation, and Project Manager/Lead 5 mentioned designers, developers working together with business people before moving to the solution space of the project.

Following the definition phase, the design phase has been depicted by 14 respondents out of 24. The design phase is also referred to as the ideation phase. According to the respondents, this phase is the solution creation space. The core is to generate ideas and conceptualize. Different activitites falling into the design phase have been described, such as workshops, brainstorming sessions, or design sprints. A strong pattern emphasized by most of the respondents was the design work timing ahead of the development. Another dominant aspect expressed was the iterative nature of the design work, sometimes described as sprint loop model. Both Project Manager/Lead 3 and Developer 2 mentioned the evolution of the design from concept, wireframes (low-fidelity prototypes), to high-fidelity prototypes. In the design phase, designers seem to keep the main role, but all other roles in the team stay involved. For example, Project Manager/Lead 2 affirmed in their case that the initial conceptualization is done between designers and project managers. They also admitted engineers to be solicited for commenting on feasibility. By listening to respondents, it appeared that the level to which the development work is undertook in the design phase varies from one company to another. Developers seem to either start implementing designs in parallel of the design phase, or to wait for the validation and delivery phases.

Following the design phase, the validation phase has been depicted by 12 respondents out of 24. According to their insights, the validation embedes different sets of practices. Multiple interviewees mentioned testing activities, team meetings, and in the case of agencies, validation sessions with the client. As opposed to other phases, more linear, the validation has been described as chuncked and divided. Multiple insigths supported that. UX Designer 3 described early concept validation performed within the design phase, and some aspect of the solution being early validated, while others still being designed. UX Designer 13 mentioned multiple rounds of validation, and a division between technical, business, and design validation.

Finally, in agency context, Developer 2 described a gradual validation model punctuated by series of client approvals, overlapping with the design phase. This level of division of the validation phase varies from one company to another. Developer 3 and Developer 4, in contrary to the aformentioned insights, evoked a linear validation phase coming between the design and delivery phases.

Following the validation phase, the delivery phase has been depicted by 12 respondents out of 24. Based on the insights, the delivery phase marks the start of the software development work. As Designer 4 put it, the process becomes truly Agile at this point since most of the effort switches to the engineering side. The delivery phase embedes a lot of different practices. Aside from Agile practices specific to engineering work, respondents mentioned common rituals between designers and developers such as pre-implementation discussions, planning meetings, and above all, the design handoff. Indeed, the design handoff, following the validation, consitutes the outset of the delivery phase. Based on the insights, handoff practices vary from one company to another, and are not always straightforward as UX Designer 11 described. UX Designer 7 mentioned having a checklist to support them in this process. As Project Manager/Lead 2 evoked, once the solution is implemented, the end of delivery phase includes design reviews and quality assurance tests. Again, these practices brings designers and developers together for an ultimate alignement before the final release of the product.

Finally, following the delivery phase, few respondents depicted a last phase, named the follow-up phase by the author. The product release doesn't sign the end of the project, and as UX Designer 2 described, the goal of this ultimate phase is to observe results, and to measure the sucess of the project. Ideally, this goes through the establishement of key performance indicators (KPIs) during the kick-off phase, which are monitored at the end in order to measure the impact of the work. This practice of KPIs monitoring doesn't seem to be mature in all companies, as only two respondents mentioned this aspect. Tracking the impact of the project informs teams and companies for future features or improvements on the product. Therefore, while the follow-up is the last phase of a project, it's also a bridge to upcoming projects and initiatives to continuously improve the product. Another aspect of the follow-up phase, well described by Project Manager/Lead 1, is the promotion of the outcomes internally and externally. Indeed, multiple other efforts follow the product development process, as marketing, communication, and maintenance, among others.

Collaboration:

Following insights on the nature of the process and product development phases, the semi-structured interviews highlighted patterns on collaboration between design and engineering.

12 respondents highlighted the fact that designers are in close collaboration with engineers. This described close collaboration is maintained throught different aspects. UX Designer 3 and UX Designer 9 both mentioned the early involvement of developers in the project. Common activities were depicted by few other respondents; UX Designer 9 evoked design reviews, UX Designer 12 highlighted alignement workshops, Developer 1 and UX Designer 4 mentioned more specific activities such as feasibility checks and timeline drafting. UX Designer 4 also talked about "healthy clashes" bewteen design and engineering. Project Manager/Lead 2 affirmed that the collaboration design-engineering is devoided of silos, and similarly, Project Manager/Lead 5 talked about parallel work of design and engineering. UX Designer 12 and Developer 4 specifically highlighted a constant communication during the design/ideation phase, and UX Designer 2 outlined the particular importance of a close design/engineering collaboration for projects that demand creating new product components and features.

On the same note as the previous paragraph about close collaboration, 13 respondents highlighted regular meetings, communication initiatives, or rituals by designers and developers. These rituals can have different purposes. Some are carefully planned and dedicated to particular tasks. Sprint plannings for instance have been described by respondents to be focused on task planning, prioritization, task review, demo, and team feedback. Similarly, retrospectives are seen as collaboration checks and task report rituals. These retrospectives seem to be more high-level and less regular compare to sprint plannings. As in the case of Project Manager/Lead 4, retrospectives can be discipline specific, whereas sprint plannings seem to stay cross-disciplinary. Aside from plannings and restrospectives, rituals also include less defined initiatives. Many respondents described regular check-ups, casual discussions, and gatherings. These on-the-go initiatives both include common gatherings with everyone involved, and smaller check-ups between fewer individuals. Developer 3 highlighted the fact that these initiatives allow people to stay on the same page throughout the project, and aligned with the project goals and expectations. Project Manager/Lead 3 also affirmed that an open-minded and collaborative environment makes it rare for people to silo.

Despite close collaboration and collaboration rituals, few respondents acknowledged isolated work to be part of their routine as well. For example, Developer 1 affirmed that post planning activities, they are going more on their "own bubble". They affirmed this to be the case for designers as well. Developer 1 again, from an agency, mentioned some projects to be devoided of design resources due to budget or client mindset, and therefore, highlighted varying levels of design-engineering collaboration across projects. Beyond the design-engineering relationship, Developer 6, being a backend developer, described being distant of front-end developers on a daily basis, and collaboration mostly happening for explanatory purposes. UX Designer 2 and Project Manager/Lead 2, both elaborating on the nature of communication within their organisation, affirmed that most of the interactions between people happen within sub-teams, and less commonly between teams.

Practices and operations:

The semi-structured interviews also provided few insights related to UX and engineering practices and operations within the different organisations.

UX Designer 1 first expressed some gaps in the design practices and operations. They developed by mentioning a lack of time resources, a lack of buy-in and understanding by the executives, as well as UI (user interface) design more valued than UX.

On an all other point, few respondents mentioned initiatives and attempts for standardising processes and practices. UX Designer 7 described their company to be using templates and project handbooks; UX Designer 11 mentioned the use of brand book. Developer 2 evoked attempts from their company to defining project funnels and guidelines to be reused across projects. UX designer 2 described their organisation to have a mature design system. Project Manager/Lead 5 mentioned the definition of clear and defined UX metrics and goals. Project Manager/Lead 5, again, expressed themselves on their company operations, and acknowledged that more and more DesignOps (design operations) experiments are undertaken, encompassing varying aspects such as tools, processes, and design identity. Finally, UX Designer 5 evoked the team dedication to discussions exclusively centered around enhancing practices and operations.

As a side note, UX Designer 10 gave an interesting reflection around design documentation, which they try to keep as actionable and light as possible for developers and other team members. The designer made a comparison with "Holidays pictures" that nobody cares about if too many.

Tools:

Finally, aside from insights on the nature of the process, process phases, collaboration, and practices, respondents mentioned the different work and collaboration tools used in product development. These tools have been structured in different categories.

Regarding design specific tools, few respondents mentioned Figma. Figma has been acknowledged to be a great tool, not only for design specific work, but also for collaboration with developers, thanks to its sharing and comment features. Two other design tools have been mentioned, Sketch and Adobe XD. Abstract, being a design handoff specific tool has also been mentioned.

For task distribution, few respondents mentioned Jira and its system of tickets.

One engineering specific tool has been mentioned, being Github.

Regarding communication tools, Slack seems to be to go-to choice. Respondents particularly appreciate its convenient system of channels and centralised communication.

Zoom and Google Meet have been mentioned as tools for remote meetings.

Miro has been stated as a great tool for running activities, such as workshops, particularly convenient in remote settings.

Regarding company level documentation, Confluence, Notion, and Google Drive have been mentioned.

Finally, few respondents expressed interesting insights and thoughts around tools. Designer 5, being in an agency context, highlighted the fact that tools cannot always be freely choosen by the team members. Developer 2 mentioned in their case the fact that the communication specific tool is usually determined by the client. UX Designer 10 tries to have as sharable and as minimum tools as possible. They think that tools should be as holistic and as less design specific as possible, as inviting other team members to access and edit if needed is important. UX Designer 9 expressed an interesting thoughts by considering tools as minor concern in the product development process. How to use tools appears to be what really matters to them.

Sub-cluster 5: Freedom and flexibility

This data sub-cluster, much shorter than the previous one, refers to the respondent's insights on the aspect of freedom and flexibility for processes and ways of working, constituting a second layer for understanding processes and collaboration between designers and developers.

Flexibility embraced:
Firstly, 9 respondents out of 24 acknowledged their organisation to be embracing freedom and flexibility. UX Designer 6 mentioned to be having some level of process in place, with space to bend it. Project Manager/ Lead 2 described general and high-level values and principles for designers, but a flexible way of putting them into practice. They continued by detailing that what matters in a project is to reach business and UX objectives more than following strict ways of working by the book. Project Manager/Lead 4 expressed themselves about the importance of avoiding to stick people to rules and standards. They also admitted not writting down any process. UX Designer 7 described having a process with high-level phases and parts of it defined on the way. This level of process defined on the way is justified by the unpredictabilities in project objectives and vision. Similarly, Developer 5 affirmed that the best teams are the ones who are able to define the process on the go and adapt to constant changes in requirements. UX Designer 9 described their effort of constantly re-drafting canvaces and guidelines across projects, and highlighted that processes and practices should be tailored instead of rigidly applied. On the same note, UX Designer 13 affirmed that each process needs customisation, and UX Designer 10 noticed that standard blueprints reused across different projects and tasks are too restrictive for them.

Flexibility under conditions:
3 research respondents depicted freedom and flexibility to be appreciated within their organisation, but under certain conditions. Project Manager/Lead 3 described that freedom and flexibility is easier to afford when tasks are not urgent, and that it is a different matter when teams need to work fast and deliver results quickly; rigidity and structure in ways of working are more needed in those cases. On a different note, UX Designer 2 affirmed flexibility to be easy to get if trust is established between people and teams. UX Designer 5 told that flexibility and freedom in their company is not granted without advancing reason for it. Project Manager/Lead 3 depicted the balance that people have to make between following Agile processes and compromising them in order to adapt to unpredictabilities.

Flexibility limited:
Finally, 7 respondents described a limited level of flexibility within their organisation. UX Designer 1, UX Designer 5, and Developer 1 highlighted Agile practices and the Scrum model to be hightly trusted and carefully followed within their organisation. UX designer 4 acknowledged to be working on the same larger process structure over and over across projects. Both UX Designer 9 and Project Manager/Lead 3 agreed on the fact that following a strict process is key in cases of urgent, small, or particularly time-boxed projects. Project Manager/Lead 3 highlighted that structure and rigidity becomes more important for bigger teams. Developer 6 expressed their particular enjoyment in working with strict discipline. This latest proves that flexibility and rigidity are not specifically good or bad aspects, and both can be appreciated differently by different practitioners.

Sub-cluster 6: Design-Engineering relationship

This data sub-cluster refers to the respondent's insights on the design-engineering relationship, which constitutes a third layer for understanding ways of working between designers and developers.

A synergic duo within a cross-disciplinary environment:
The first pattern highlighted by respondent's insights on the relationship design-engineering is depicting a synergic duo operating in a cross-disciplinary context. UX Designer 8 put it this way: "Healthy and synergic duo". This complicity has to do, among other aspects, about the mindset. On that note, multiple respondents described willingness and efforts from both sides to understand each other's discipline on a deeper level. UX Designer 3 highlighted initiatives from developers to learn about design, design systems, and Figma. Both UX Designer 9 and UX Designer 10 described the same idea, designers and developers learning from each other. Similarly, UX designer 6 mentioned about developers willing to understand design decisions. Developer 2 described designers to be aware of the technical constraints and possibilities, as well as developers being design counscious. On the mindset still, UX Designer 11 depicted authentic appreciations from both design and engineering roles to collaborate with each other. UX Designer 13 affirmed responsibilities to be shared on the product from both sides. Developer 5 highlighted a good level of psychological safety between both disciplines, and similarly, Project Manager/Lead 3 described open minded and respectful people, which they claimed is necessary for creating great products.

The complicity and bonded relationship highlighted by the respondents not only has to do with the mindset and attitude of the people, but also the context and practices in place. Many insights depicted initiatives in favor of a bonded relationship between designers and developers. Project Manager/Lead 5 highlighted a pretty strong design culture in place within their organisation. Multiple respondents mentioned various cross-disciplinary initiatives contributing to, as UX Designer 2 put it, fostering awareness, respect, empathy, and understanding between disciplines. Developer 3 and UX Designer 13 affirmed both roles to involve and call for each other spontaneously, and UX Designer 8 described developers to be involved in design reviews. Multiple other respondents elaborated further; UX Designer 4 mentioned common brainstorming sessions, UX Designer 13 technical validation sessions to happen before the design handoff, and UX Designer 3 described aligned objectives across design and engineering. Respondents also touched on the communication aspect, and acknowledged it to be pretty efficient between designers and engineers. Great conversations and trade-offs have been highlighted by UX Designer 1 and UX Designer 9, UX Designer 11 mentioned regular check-ups along the process in order to maintain alignement, and multiple other respondents described design and engineering roles to be staying in the loop continuously. Project Manager/Lead 3 and Project Manager/Lead 4 both evoked centralized communication and work management and acknowledged its benefit on the design-engineering bond. Few respondents described more in detail the balance of each discipline within their organisation. Project Manager/Lead 1 affirmed design and engineering to have an equal voice in decision making, and Project Manager/Lead 2 acknowledged an almost inexistant distinction between both entities. Developer 2 continued by evoking a significant influence of engineering on design decisions, and Project Manager/Lead 5 affirmed both roles to be working next to each other.

On a side note, the Agile model has been described by few respondents to be influencing the design-engineering link in a positive way. For example, UX Designer 11 evoked the system of ticket distribution to smoothly reuniting roles under a common way of working, and UX Designer 13 expressed a similar thought in regard of the sprint model.

Recognising rooms for improvement in the relationship, few respondents evoked ongoing initiatives and efforts from their organisation willing to enhance the bond. Developer 4 and UX Designer 13 described a growing design system contributing to more and more bridging the gap between disciplines. Project Manager/Lead 5 highlighted efforts from the designers to provide better deliverables and specifications to engineers. Finally, other respondents elaborated on the motivations and experiments aiming at tighting and maintaining this link.

A less optimal relationship under a challenging environment:
A second pattern highlighted by the respondents is revealing issues in the organisation's environment, consequently leading to a less optimal relationship design-engineering. UX Designer 1 for instance mentioned a lack of structure in the interactions between the two roles. UX Designer 4 evoked breaks in the process, discontinuities between design and engineering workflows, and developers being involved too late in the process. Developer 1 described an isolated design team, impacting on the communication. Few other respondents mentioned conflicts and tensions to be regular between design and engineering. Developer 6 pointed out an uneven ratio between the two disciplines, design being much inferrior. Beyond design and engineering roles, Project Manager/Lead 5 highlighted gaps with the business side of the organisation, potentially having an influence on the relationship design-engineering.

Regarding internal issues impacting on the relationship design-engineering, the lack of design maturity has particularly been stressed by 7 respondents. Both UX Designer 1 and UX Designer 4 pointed out the fact that UX needs to be advocated and sold at the startegic level, creating lots of additional efforts for designers. UX Designer 5 and UX Designer 12 also described a design discipline not influent enough, leading executives roles to take decisions in silo. UX Designer 11 evoked an early stage design and a process focused on development, while UX Designer 12 pointed out the small size of their design team within their organisation. Developer 1 also highlighted the unbalance of a tech-centered process, as well as executives not understanding the value of design. Developer 3 continued by mentioning that the level of understanding of design by the client, in agency settings, can as well impact the bond design-engineering in a negative way.

Last but not least, few respondents describing a less optimal relationship between design and engineering pointed out the challenging Agile time-boxed rythm as one of the main cause. UX Designer 3 recognised many requirements to be thrown to them in a small amount of time, while UX Designer 4 affirmed that design doesn't have enough time to explore solutions. Similarly, UX Designer 5 highlighted the implementation work to be starting too early, with no finalised solution to start from. Finally, UX Designer 4 affirmed Agile to be causing frustrations for UXers and design to be hard to fit in it. UX Designer 12 continued on a similar thought by pointing out Jira and the ticket model to be illogical for designers, and making it hard to see the bigger picture of the project.

Sub-cluster 7: Aspect of informality

This data sub-cluster refers to the respondent's insights on the aspect of informality, which constitutes a fourth layer for understanding ways of working between designers and developers. This layer comes on a secondary level compare to the previous ones in this cluster, but is worth detailing since it brings an influence on the collaboration.

Informal initiatives in place:
When respondents were asked about the aspect of informality within their organisation, they all described some level of initiatives or practices in place. Very simple initiatives were described, such as coffee breaks, day-to-day social interactions at the office, or casual chats. Some respondents pointed out more in detail the informal tone of the meetings and the relaxed discussions happening in those. More advanced and thoughtful practices have also been highlighted. Project Manager/Lead 2 mentioned a casual Slack channel dedicated to informal interactions. Developer 1 highlighted the casual office space as contributing into creating relaxed interactions. More practically, activities such as ice breakers, team bonding exercices, remote games, or even informal tasks put in the product backlog, were depicted. Events and get together initiatives, for the most dedicated, have been listed by the respondents as well. UX Designer 8 for instance described special company lunches organised regularly. UX Designer 11 evoked half a day off scheduled each month for people to bond together oustide the office space. Company breakfasts, travels, or afterworks, were few other events highlighted.
Only one respondent, UX Designer 13, explicitely stressed the fact that informality is not enough considered within their organisation.

Strong and well considered informality:
Out of the 12 respondents who expressed themselves on the aspect of informality, only 2 described a particular high level consideration and devotion by their organisation on the matter. On one hand, UX Designer 2 highlighted the design of their office to be extremely optimal for informal interactions, including ping pong tables and bean bags. They also mentioned initiatives such as free drinks every Friday, among others. On the other hand, Developer 2 mentioned regular initiatives such as afterworks, games, coffee times. They elaborated on the office atmosphere as well, and depicted informal work interactions to be part of their company culture.

Informality as a trigger for motivation and great collaboration:
7 respondents explicitely acknowledged informality to be a contributing aspect in creating a relaxed atmosphere within their organisation, and ultimately enhancing their quality of work. Developer 1 affirmed informality to be increasing the efficiency of their team. UX Designer 8 acknowledged the benefit of informal initiatives in bringing people together, ultimately building a solid team spirit. Developer 6 continued by detailing an improved team dynamic. Developer 2 described informal initiatives to be bridging people and easing their interactions. On a similar note, Developer 4 acknowledged that informality helps to better understand their team members, and contributes in building close relationships with them. UX Designer 13 expressed themselves on the safe and trustful environment that informal actions can establish, while UX Designer 6 completed by affirming their work environment to be encouraging and without ego.

3.3. Cluster 3: People's perspective on processes and collaboration

This third data cluster provides insights on the perspective of the respondents regarding processes and collaboration within their organisation. It's composed of three sub-clusters: To be improved in the organisation, Conception of a successful project, and Aspects influencing processes and collaboration.

Sub-cluster 8: To be improved in the organisation

This data sub-cluster refers to the respondent's opinion on the aspects to be improved within their organisation, which constitutes a first layer for understanding their perspective on processes and collaboration.

Design maturity:
The first aspect mentioned by respondents when it comes to desired improvements was design maturity. Design maturity was highlighted by 6 respondents. UX Designer 1 wishes for a deeper inclusion of design at strategic level of their organisation. On a similar thought, UX Designer 12 wants design to have more voice and power, and UX Designer 4 believes that design should have more leadership. Project Manager/ Lead 5 pointed out management not understanding design properly, for example when drawing the vision of the company. They expanded their thought by affirming that both business and technical roles should get a wider awareness of design.

While reflecting on the lack of design maturity, few respondents touched upon the design resources. UX Designer 12 expressed themselves on their desire to have a bigger design team. Both Developer 3 and Developer 4 stressed on the lack of design resources. Developer 3 thinks their company's vision to be uncompatible with the current design resources, and Developer 4 believes that designers are dealing with too many tasks. Developer 3 continued on pointing out the unbalanced design-engineering ratio.

Collaboration and people:
In addition to design maturity, respondents highlighted mutliple points regarding collaboration and people.

4 respondents expressed themselves on their wish to see design and business disciplines more connected. UX Designer 4 believes that managers and business roles should be more user-centered. Similarly, UX Designer 12 thinks that management could be more collaborative, listening, and more aware of design and engineering practices. Aligning management and design seems to be hard according to them. Developer 3 also described conflicts between design and business roles, and affirmed language to be one of their possible cause. Project Manager/Lead 5 was quite radical when pointing out issues in the business mindset. They think that the business mindset should be switched, as business roles are taking too much space in design decisions.

Respondents also highlighted the communication and mutual involvement of design and engineering as a source for improvement. UX Designer 11 affirmed developers to be involved too late in the product development process.

Developers 1 went further by highlighting that not only developers, but all roles involved in a project should be involved earlier. They pointed out that some people need to catch-up a process on the way sometimes. Developer 4 thinks there could be more interactions between designers and developers, and Project Manager/Lead 2 believes that developers can reach out to design more actively.

Two respondents stressed on the aspect of team spirit, as they think it could be strengthened. UX Designer 8 expressed a particular isolation of some members of their organisation being located in another country, and constrained to work remotely. Project Manager/Lead 2 thinks that the internal collaboration between the different product teams of their organisation can be enhanced.

Developer 5 detailed their thought on the collaboration with external team members or stakeholders, being part of other organisations. They pointed that out as an aspect to be improved since dealing with different timelines and external communication is challenging.

UX Designer 6 highlighted the aspect of team spirit and psychological safety. They wish to be more confortable in seeking help and asking questions to all team members.

UX Designer 10 expressed their wish for more peer learning and experience sharing between their team members. They believe this to be useful for analysing pain points and improve ways of working together.

Practices and process:
Following design maturity and collaboration, respondents finally pointed out some gaps in the practices and process.

Documentation has been mentioned by 6 respondents has a practice to be improved. Respondents detailed their thoughts and highlighted different types of documentation they wish to be enhanced. UX Designer 11 mentioned the evolution in general and changes happening along the way, UX Designer 13 talked about case studies, Developer 2 emphasized techniques and tools, and Developer 4 elaborated on written documentation about company's processes and structure. Project Manager/Lead 2 affirmed documentation to be useful for unifying people and teams, as well as integrating new comers. Project Manager/Lead 3 highlighted the challenge to practice documentation in an Agile and fast-paced environment.

3 respondents expressed themselves on the aspect of project structure and guidance. UX Designer 2 wishes to work with more defined and clear processes and frameworks. Developer 2 enjoys working within a sprint cycle, and would like to see a more clearly defined product backlog. Project Manager/Lead 4 also highlighted more structure to be needed, which they believe allows to better handle unexpectations in addition to be saving time and efforts.

Two respondents highlighted design system and Design Ops (design operations). UX Designer 13 thinks design systems should go beyond documenting UI only, and expand in order to standardise processes and ways of working. Developer 4 expressed their desire for developers to be more involved in the design system of their organisation.

Two other respondents detailed their thoughts on the selling and billing practices of their organisation. UX Designer 5 believes these to be too strict, and UX Designer 7 thinks they happen in isolation, without involving enough design and development roles.

UX Designer 8 talked about unexpectations and change happening within the process, and thinks that these could be mitigated. They believe this to be arming trust towards management and the feeling of ownership. They also wish for more communication from management on the state of the project over time.

Handoff and implementation practices have been highlighted by UX Designer 11, as they think this process is not streamlined enough and issues of wrongly implemented design still happen.

UX Designer 12 mentioned the ticket system as not being optimal and too narrow for designers. They think design tasks to be hard to split into tickets, especially when being holistic.

In a quite high-level way, UX Designer 9 wishes to get more time to understand the problem instead of jumping into solutions.

Finally, UX Designer 1 thinks that their organisation could improve in their ability to modernise and move away from old methods and processes.

Sub-cluster 9: Conception of a successful project

This data sub-cluster refers to the respondent's insights on their conception of a successful project, which constitutes a second layer for understanding their perspective on processes and collaboration. What respondents consider to be a successful project has been clustered in two trends: the project sucess in terms of results, and the internal success.

Project success in terms of result:
Firstly, respondents mentioned their conception of a project sucess in terms of end-result. The end-result can be divided between the outcome, the impact of the solution on the end-users and business, and the output, the tangible product solution.

8 respondents highlighted the end-users satisfaction to be a core measure of success for a project. According to UX Designer 4, the user satisfaction goes through the fulfillement of real needs, as opposed to fake needs created for the sake of satisfying the business. As UX Designer 13 put it, the solution has to be useful and delightful for the end-users. Other respondents mentioned the aspect of time saved as being another component of the user satisfaction. According to Project Manager/Lead 5, a project is succesful when business challenges have been translated into users' value.

7 respondents mentioned the alignement of the end-result with business objectives to be another core measure of success for a project. They mentioned the importance of the impact on the business as well as the accuracy of its estimation from the outset. Few respondents expanded their thought by mentioning the alignement of the results with both business and UX requirements to be essential.

Another measure of success highlighted by 4 developers is the quality of the working product and features. The less bugs and reworks needed after the release, the better. Developer 1 and Developer 2 expanded on the quality of the code, production, and technical architecture.

After the end-users, 3 respondents went further by mentioning the satisfaction of the client, in agency contexts. UX Designer 7 highlighted the importance for the client to get answers from their question(s).

Developer 6 detailed their thoughts on the security insurance aspect, being another measure of success.

Finally, UX Designer 5 mentioned the environment (nature), and affirmed that successful projects and solutions are the ones taking this aspect into account.

Internal project success:
After elaborating on the end-result, respondents focused their reflection on what they think a successful project is at the internal level.

10 respondents consider a planned, structured, and smooth process to be a core aspect of the internal success of a project. Project Manager/Lead 5 defined a successful process as one allowing designers and developers to focus on their core job. It appeared important for the respondents to define stages and follow them as closely as possible. A smooth process has been described as a process minimizing breaks and discrepancies as much as possible. UX Designer 13 for example considers a project to be successful when extra work is limited. UX Designer 3 also pointed out the importance of experienced project managers when it comes to the efficiency of a process. Finally, Project Manager/Lead 3 defined a structured process as one allowing teams to make use and distribute their resources in an optimal way.

According to 9 respondents, the internal success also goes throught great interactions, communication, and collaboration within and across teams. In a summarised way, respondents think that great collaboration is established when everyone's time and effort is optimised, and when all the different perspectives are taken into account and understood.

5 respondents linked the internal success with internal alignement and understanding of the vision and objectives of the project. They highlighted the importance for all team members to be updated and informed continuously.

Another 5 respondents consider a project to be successful at the internal level when it starts with a proper brief, a well defined scope, and clear formulated objectives. UX Designer 3 mentioned an actionable target to be essential to start with, and Project Manager/Lead 2 highlighted the importance of defining concrete metrics. Project Manager/Lead 5 also reminds here on the necessity for a business rational to be translated into value for end-users.

A good team spirit has been highlighted by 5 respondents again. Respondents think that good team spirit is reached when no major conflict occurs, and when team members enjoy working together while being their true-self. UX Designer 9 went further by affirming that an healthy team spirit is established when no ego takes over in the team.

3 respondents highlighted the aspect of timing. Simply put, they think a project to be successful when it is done on time.

UX Designer 4 and Developer 4 detailed their conception of internal project success with stakeholders' involvement. They both think that involving all stakeholders from the start and all along the project plays on its success. UX Designer 4 also emphasized the involvement of designers and end-users at the outset in order for proper research to be done.

UX Designer 4 and Developer 4 again think a project to be successful when sufficient experimentations and iterations are done. They emphasized the importance of "fail-fast" and "learn-fast" mindset.

UX Designer 4 and UX Designer 5 both highlighted shared decision-making to be another measure of internal success.

On a more personal level, Developer 2 and Project Manager/Lead 3 think a project to be a success when they get the feeling of having learnt something and improved their skills. They both conceive successful teams to be those where people are able to grow with each other.

Finally, UX Designer 6 shared another measure of internal project success, being the ability to manage expectations between the team and the client, in the case of agencies.

Sub-cluster 10: Aspects influencing processes and collaboration

This data sub-cluster refers to the respondent's insights on the influencial factors of processes and collaboration according to them. This constitutes a third layer for understanding their perspective on processes and collaboration.

The overall environment:
First, respondents highlighted general characteristic of the work environment.

8 respondents stressed on a major difference to make in ways of working between in-house and agency environments. UX Designer 13 affirmed that creating a product for a client as opposed to developing an internal product brings a different mindset to the project. UX Designer 1 and UX Designer 2 made a comparison between both contexts, and affirmed processes to be more complex and unpredictable in agencies. UX Designer 2 expanded by highlighting more structure to be in place in in-house settings. A thought resonating in conflict with the previous one came from Developer 3, stating that a lower level for unexpectations is usually tolerated in agencies. UX Designer 3 mentioned the aspect of client management in agencies, bringing another piece in the collaboration and potentially additional tasks. Continuing on that, Developer 3 mentioned the concern, sometimes pressure, in keeping the client to the table and maintaining trust with them. UX Designer 5 elaborated by affirming less need in advocating processes and design to be felt in in-house companies. UX Designer 5 again, mentioned a practical aspect and acknowledged people to be aligned with the product and system more easily in-house. On another practical aspect, Project Manager/Lead 3 highlighted the way designers and developers can potentially switch across different projects in agencies, a trend that doesn't seem to occur in-house. Finally, UX Designer 11 thinks in-house companies have a more flat hierarchy overall, as opposed to agencies.

Another characteristic of the environment influencing on processes and collaboration was highlighted by UX Designer 3 and UX Designer 13. They both stressed on the opposition B2B (business-to-business) vs. B2C (business-to-customers) business model. More specifically, UX Designer 3 reflected on the sale dimension present in B2B contexts, bringing a different mindset to the project.

Project Manager/Lead 1 was the only research respondent operating as a freelance. Therefore they highlighted their particular position in the collaboration compare to their other employed team members. Another distinction is to be made here between freelance and employed roles, inherently influencing the ways of working.

UX Designer 13 highlighted a last environment level influencial factor, being the operating field. Based on their case, UX Designer 13 affirmed the level of expertise of the organisation and team members on the operating field to influence the way research is done. More generally, they also desribed each operation field to bring different constraints and requirements to the project, for example technical fields relying more on analytics as opposed to other fields emphasizing qualitative data.

The project and product:
Respondents went more in detail by describing aspects related to the project and product.

10 respondents highlighted the size and scale of a project, inherently its timeline, to be a big influencial factor on ways of working. UX Designer 10 affirmed bigger projects more likely to be Agile, while UX Designer 12 ackowledged that smaller product development projects can rely more on the waterfal model. 3 respondents mentioned the timeline, in other words the level of urge of the project, to influence on the flexibility in the process. More generally, Developer 6 believes each project to be a unique context, therefore defining unique ways of working.

Developer 3 made a significant distinction between projects with varying level of design work needed. They affirmed ways of working being different for projects centered on execution, compare to projects including conceptual creation. In a nutshell, the balance between design and engineering work needed for a given project is another influencial factor for processes and collaboration.

Few respondents highlighted the nature of the product itself to be affecting ways of working. UX Designer 7 and UX Designer 13 pointed out the difference between desktop and mobile applications in terms of usability and technical architecture. Project Manager/Lead 1, currently working on a design system project, stressed on the divergences between internal projects, such as design system, and working on a product for end-users.

UX Designer 1 and UX Designer 13 mentioned another influencial factor, being whether a project aims at improving existing products, features, or flows, as opposed to those creating solutions from scratch.

UX Designer 13 affirmed the type of end-users to be having an influence on processes as well. As an example, getting feedback and conducting user research with B2B users is significantly different than for B2C users.

From a technical perspective, Developer 2 highlighted the nature of the design and design decisions to be impacting how engineering work is undertaken. They affirmed each design, functionality, or set of requirements to be always a unique implementation process.

The organisation:
Traits related to the organisation have also been highlighted by the respondents.

7 respondents highlighted the development stage and maturity of the organisation to be influencing processes and collaboration in product development. UX Designer 7 detailed their thoughts on growing environments and affirmed growth to generate more interdependencies between teams. Few respondents pointed out differences of processes between startups and more established organisations, and even between startups and scale-ups. UX Designer 13 for example affirmed startups to be releasing features and product pieces on a faster rythm.

4 respondents mentioned the project management style, including the competences, experience, and methods employed by the entitled people to be a consequent influencial factor on ways of working for developers and designers.

The resources allocated, including budget and human resources, has been pointed out by 3 respondents as another factor.

The technical capabilities, resources, and technology available are according to UX Designer 4 and Project Manager/Lead 3 aspects to be considered as well. UX Designer 4 believes technical capabilities to be very project specific. Project Manager/Lead 3 highlighted the technical and back-end infrastructure to be affecting decision making along the project.

UX Designer 6 and UX Designer 9 pointed out the tools, including collaboration and more discipline specific tools, to be another influencial factor on ways of working. They pointed out the familiarity of people to use these tools to be weighting on the balance as well.

UX Designer 8 and UX Designer 9 mentioned the practical settings, and amount of remote work.

UX Designer 6 thinks the degree of mistake allowed, and flexibility of the organisation on wrong estimations to be another relevant factor.

UX Designer 8 was the only respondent to highlight the design maturity, and more specifically the degree of recognition, trust, and respect about design as a distinct influencial factor.

Finally, Project Manager/Lead 5 pointed out the business mindset of the company to be highly affecting ways of working as well.

The people:
After general characterics of the environment, more detailed aspects related to the project and product, and traits regarding the organisation, respondent focused on the people.

9 respondents highlighted the team and people to be playing a major contribution in defining processes and running the collaboration. Various aspects have been mentioned with this idea. UX Designer 2 highlighted the overall team mindset, as well as the level of motivation and empowerment of each team member on the project vision. UX Designer 4 described team dynamics to be influenced by the personalities composing the team. They also mentioned the level of understanding and awareness on each other's role and discipline to be an important factor affecting collaboration. UX Designer 7 reflected on the age, and believes teams composed of younger individuals to be more dynamic. Both UX Designer 8 and UX Designer 9 acknowledged the level of expertise and experience of each individual in the team to be forming a unique mix of skills and knowledge. UX Designer 10 and Developer 2 highlighted the familiarity of the people to work together, and their relationship on a personal level, to be affecting the synergy of the team. On a related thought, Developer 6 described a close link between the degree of respect, trust, and psychological safety from people towards each others, and a great collaboration experience. Developer 2 went more practical, and mentioned the team organisational structure to play a big role on the nature of the processes, and highlighted at the same time the amount of external people taking part of the project to influence on the collaboration. Finally, Project Manager/Lead 2 expressed their general belief being the fact for each team to be unique, therefore each project collaboration as well.

UX Designer 6 and Developer 4 highlighted the communication dynamic to be an important influencial factor as well. UX Designer 6 thinks great communication to be dependent of its form, frequency, and accuracy. They highlighted potential divergences between internal and external communication flows, and stressed on the importance of language and terminologies on enabling good communication. Developer 4 pointed out the major influence of early and continuous involvement of team members on the nature of the processes and collaboration.

The client (for agency contexts):
Finally, few respondents coming from agency contexts have highlighted the client as being another aspect influencing processes and collaboration.

In consultant projects, 7 respondents mentioned the client being a major influence on processes and collaboration. UX Designer 5 affirmed working methods to be varying from one client to another. UX Designer 7 mentioned the consequent impact of the familiarity of the client with product development, design, and Agile practices on the processes. Similarly, Developer 2 affirmed that the better the client understands design, the more likely designers are to be involved continuously in the project. Based on experiences, Developer 1 affirmed the client to be a potential cause of limitation. Developer 2 also shared whitnesses on project delays caused by the client activity. Developer 5 highlighted technical tools and infrastructure to be sometimes imposed by the client. Finally, Project Manager/Lead 4 pointed out the maturity of the client design system and practices in place to be influencing on the amount of work needed.

Another aspect related to the client came from UX Designer 7 while evoking billing methods. They affirmed the level of flexibility of billing methods to be playing an influence on the ability of teams to adapt to change. Billing methods as well as the amount the client is willing to pay have a huge impact on processes.

3.4. Cluster 4: The future of product development, design, and engineering

This fourth and ultimate data cluster provides insights on the perspective of the respondents on the future of product development, design, and engineering. It's composed of two sub-clusters: Future of UX, SE, and digital products, and Respondent's 10 years perspective.

Sub-cluster 11: Future of UX, SE, and digital products

This data sub-cluster refers to the respondent's insights on their view of the future of UX (UX design), SE (software engineering), and digital products. This constitutes a first layer for understanding their vision on the future of product development.

The future of the product development process:
When being asked about their vision of the future of product development processes, 12 respondents mentioned the trend of UX and engineering practices to blend with each other. UX Designer 5 for instance thinks to see more shared logics between design and software development, as design system unifying designers and developers. UX Designer 6 hopes to see less delimited borders between both disciplines, while UX Designer 7 believes in a future of stronger and closer collaboration. While both Developer 1 and Developer 2 expect more knowledge and better understanding from designers and developers on each others, UX Designer 9 elaborated on evolving roles overlaping more and more over both disciplines. On a similar idea, Project Manager/Lead 1 foresees the border between front-end development and design to blur. Developer 4 believes in more co-creation initiatives to occur in the future, and Project Manager/Lead 3 hopes to witnesse the evolution of the designer role to be more technical oriented. More practically, Developer 5 stressed on evolving methods such as design sprints and thinks they will continue to embede design more and more into engineering. Developer 3 visions designers and developers to openly work together in chunks. On a higher level, Project Manager/Lead 5 hopes for UX and software engineering disciplines to be equally seen in more and more companies in the future.

Another future trend highlighted by 8 respondents on the product development process was automation. When it comes to automation in product development, few respondents believe the engineering field to be impacted faster and more extensively than design. UX Designer 5 supports their idea by pointing out design to revolve around human as opposed to engineering revolving around technology. Nonetheless, other respondents specifically highlighted the automation of design processes. UX Designer 11 foresees some level of automation of the visual aspect of design, UI, and also research activities such as user testing. Both UX Designer 5 and Developer 4 affirmed design templates, as well as tools built around templates, to be evolving forms of design automation already. Developer 6 mentioned the idea of design assistance. They affirmed that even design aspects such as the ideation can technically be enhanced and partially automated by AI. Developer 6 continued by describing their vision of a human-to-machine interaction cycle as a component of the future product development process. Developer 4 also believes the evolution of tools to replace some current design and engineering tasks, if not roles. Developer 6, again, mentioned AI generated codes to be more and more common. On a higher level, Developer 1 believes automation to influence the nature of design and engineering roles, and bringing them closer to strategy while pulling them away from execution.

The evolution of tools was another future trend highlighted by 6 respondents. Respondents believe tools to bring designers and developers closer and closer to each other in the future. UX Designer 3 and UX Designer 7 mentioned the spread of more and more sharable tools such as Figma, Protopie etc... already contributing in a better collaboration today. Project Manager/Lead 3 thinks it will be faster and faster to get into high-fidelity prototypes in the future. Both UX Designer 5 and UX Designer 7 believe tools to keep easing communication, streamlining collaboration, and allowing for better concensus to be done between designers and developers. Similarly, Developer 3 and Project Manager/Lead 3 think the evolution of tools to be reducing backs and forths as well as micro-interactions between product roles. UX Designer 13 mentioned remote collaboration tools such as Slack or Zoom having already transformed ways of working in product development.

5 respondents expressed their vision of systematisation and standardisation for product development. Designer 7 elaborated on design systems being at their early stage today. They believe that not only UI components, but the organisation of work in general, could be standardised. Developer 3 expressed their wish for systematisation to help people focusing more on their work, building great products, and less on defining processes and dealing with conflicts in collaboration. Project Manager/Lead 1 believes standardisation and systematisation to help designers and developers in multiple areas, such as handoff practices, project frameworks, and language.

More of a reflection than a concrete trend, 4 respondents believe the evolution of the product development process to be context specific. For example, UX Designer 2 highlighted discrepencies between different areas of the world, and technological gaps already visible today. More specifically, UX Designer 13 believes each operating field to induce a unique evolution scenario for product development.

3 respondents highlighted the evolution of working settings, inhenrently affecting the future of product development processes. While respondents mentioned the prominence of remote work, UX Designer 13 still believes physical and co-located work to happen in the future, as it helps to foster deeper team spirit and collaboration.

Another trend highlighted by 3 respondents was the rising importance of data in the product development process. UX Designer 5 and Project Manager/Lead 2 believe user research to evolve and the emphasis put on qualitative data, or “Thick“ data, to increase.

The last thought on the future of the product development process came from Project Manager/Lead 2, who reflected on the evolution of the project manager and lead role. They believe this role to be a major piece of the puzzle in order to let designers and developers focus on their jobs, therefore mentioned that holistic roles such as project managers will always be needed. As an overlap with the previous paragraph, Project Manager/Lead 2 believes managers and lead roles to be more data oriented in the future.

The future of UX Design:

Following the product development process, respondents reflected more precisely on the future of UX, or digital product design.

Firstly, 5 respondents evoked a significant rise in design maturity within the industry. Respondents believe a high level of design maturity to become a standard in most organisations in the future. UX Designer 1 thinks recognised design frameworks to become as common as Agile is nowadays. Both Project Manager/Lead 3 and Project Manager/Lead 4 affirmed that design will take a consequent weight in executives' opinion, and will be largely valued at the strategic level of most digital companies. UX Designer 10 thinks that design being seen as one bloc within a larger product development process will be an egde case in the future, and Project Manager/Lead 5 believes the image of the designer reduced to a "pixel pusher" to die progressively.

When reflecting about the future of UX design, 5 respondents affirmed the human touch to be always needed. In other words, respondents don't believe to see the role of the designer to die. Both UX Designer 2 and UX Designer 5 believe that uncovering needs, defining a problem, defining a strategy, and creating a solution are tasks that cannot be automated. UX Designer 11 and Developer 6 had a similar thought in regards of the research, and the aspect of empathy, context, and problem understanding. Project Manager/Lead 5 too affirmed that parts of subjectivity within the design process are impossible to automate.

Another set of 5 respondents highlighted the trend in the design role to become more and more a facilitator and a leader. According to UX Designer 9, the designer is the one unifying all perspectives within a project, and a key connection between people. Both UX Designer 10 and Project Manager/Lead 3 believe facilitation and coaching to become more and more required skills for future design positions. As described in the next paragraph, the aspect of co-creation will rise; Project Manager/Lead 5 on that note highlights the crucial role of design in making people work together in order to solve problems. Project Manager/Lead 4 also foresees more and more CEO's to be coming from the design field.

4 respondents highlighted the trend of co-creation between project teams and end-users. UX Designer 4 highlighted users to be coming closer to designers and to be speaking for themselves. Multiple other respondents believe to see the voice of the users taking more and more space within design and product development. Both Project Manager/Lead 2 and Project Manager/Lead 5 are convinced that understanding users becomes more and more important for creating successful products, therefore involving them in the process appears as essential.

Another 4 respondents reflected on the diversification of the role of the UX designer in the future. The diversification already happens today, according to UX Designer 2, and will keep increase over time. UX Designer 4 highlighted the term "T-shaped", referring to skills generalism, to be more and more employed in the design field. UX Designer 4 again believes designers to be taking care of more, and more diverse tasks. On top of that, Project Manager/Lead 3 thinks the designer role to become more and more holistic within product development. Developer 2 gave more concreteness to the reflection, by highlighting the likeliness of designers to be doing front-end development in the future.

3 respondents highlighted the field of design to become more and more strategic in the future. Further away from the narrow vision of the "pixel pusher" sometimes associated to designers, respondents believe design to be widely understood as the science of solving problems.

UX Designer 11 and Developer 2 elaborated their thought beyond the role of design itself, but its demand. Both respondents emphasised the current rise of the UX market, and believe this trend to continue in the future. They expect to see and more design job open positions in more, and more diverse types of companie.

Finally, on the future of UX design, Project Manager/Lead 2 and Project Manager/Lead 5 evoked the fields of sociology, psychology, and anthropology to be taking more weight into the discipline. Project Manager/Lead 5 even believes to see new types of roles and competences to emerge into the design process.

The future of Software Engineering:
Only few thoughts were expressed precisely on the future of SE (Software Engineering).

Firtsly, two respondents reflected on front-end development, and believe this area of SE to be dying over time. Both UX Designer 4 and Developer 4 think that automation will take over the human need.

On the area of back-end development this time, Developer 6 expressed their doubts on automation. They believe back-end development to always need the human touch in order to build and maintain complex systems in a secure way. More generally, Developer 6 thinks that automation will only move the focus of engineering roles, or create new ones, but certainly not cut the jobs.

2 developers highlighted SE to have faster and easier processes in the future. Developer 1 mentioned more in details the auto-generation of pieces of code and the diminution of tools to use. Developer 3 believes it will be easier to set-up technical environment and do change add-hoc.

Finally, on the future of SE, UX Developer 2 affirmed software development roles to become more generalist, similarly as UX design roles.

The future of digital products:
Last but not least, after detailing their view on the future of the product development process, the future of UX Design and SE disciplines, respondents focused their reflection on digital products in general.

11 respondents highlighted the exponential increase in use and prominence of digital products in our society. They believe more and more digital products to be serving and empowering us in more and more fields. Developer 4 affirmed almost every company to be digital, and UX Designer 11 outlined digital to be part of our lives already. They expect this trend to keep accelerating. Similarly, Project Manager/Lead 1 believes today to be only the beginning of digitalisation. Project Manager/Lead 2 stressed on the growth of digital to be unequal in different fields, as they believe some fields to be faster to digitalise than other.

When reflecting about the future of digital products, 8 respondents hope for better products, and products with more purpose. UX Designer 5 wishes for products to be more focused on solving real needs, and less axed on feeding consumerism. They continued by affirming many products today to be waste. Many other respondents highlighted similar thoughts. Both UX Designer 13 and Project Manager/Lead 5 wish digital to better empower humans to do what they really have to do. UX Designer 2 expects more meaningful problems to be tackled, and Developer 6 believes that digital should be more adaptative to users' needs. Developer 5 hopes for digital products to become better for people to not only fulfill their needs, but doing so with delight. UX Designer 6 mentioned some products to be adictive and manipulating. On that note, UX Designer 12 thinks understanding human psychology will become crucial to prevent these dark patterns.

7 respondents highlighted an increased prominence of artificial intelligence (AI) and machine learning technologies in digital products in the future. UX Designer 4 and UX Designer 13 believe these capabilities to considerably influence the nature of digital. Few respondents directly linked this trend with the aspect of automation in product development process. Developer 6 hopes for AI and machine learning technologies to be wisely used and to provide better product experiences, as opposed to solely benefiting processes and business. Regarding AI, Project Manager/Lead 2 had a reflection on the social aspect and ethics; they believe these concerns to be more emphasised when it comes to using such technologies in the future.

Again, 7 respondents highlighted the aspects of accessibility and inclusiveness to become more prominent concerns for digital products. Developer 6 affirmed that making use of digital products in the future should be more simple, and more simple for more people. 3 respondents referred to the EU regulations being a sign of the already growing recognition of the need for accessibility. Developer 4 is curious to see how these rising concerns will affect the nature of the products in the future. Both UX Designer 6 and UX Designer 13 questioned the future space for creativity in digital products, and how this can be balanced with the need for accessibility.

The aspect of ethic in digital products was risen up by 6 respondents. Respondents seem to be concerned about unethical use of people's data through digital products. Project Manager/Lead 5 affirmed that data being used against people is a real issue nowadays. Face to that, Developer 4 hopes to see rising transparency in the use of data in digital products, and UX Designer 12 expects for more standards to emerge in order to prevent dark patterns. As for the aspect of accessibility in the previous paragraph, few respondents highlighted EU regulations on data privacy to be an encouraging step towards more ethical and transparent products. Project Manager/Lead 2 is convinced this rising concern to create positive impacts on digital applications over time.

Without elaborating too much, 4 respondents evoked the aspect of security to be another increasing concern for the future of digital products.

Straightforwardly as well, UX Designer 5 and UX Designer 7 both pointed out the increasing role of data and the data analytic field in digital products.

Both UX Designer 4 and UX Designer 13 reflected on the evolution of the technologies and their potential impact on digital applications. They highlighted multi-sensorial systems, screenless product experiences, voice interaction, and augmented reality being only few technological advances which will shape the digital experiences of tomorrow. UX Designer 13 reminded the nature of digital products to be closely inherent to the technologies, therefore their evolution too.

UX Designer 9 and Developer 2 reflected on the lifecycle of digital products. As opposed to early days of digital, where softwares were finished and sold in a box, both respondents reminded how digital products nowadays are constantly updated, renewed, and therefore how unifished they are. They believe this trend to keep going in the future, on maybe even shorter lifecycles. According to them, the needs for product improvements and renewals won't ever blur.

Finally, UX Designer 4 had a final thought on end-users. They believe digital users to have become, and to keep becoming more experts in using products. UX Designer 4 talked about the "democratisation of digital". Consequently, UX Designer 4 also described users to become more aware of their needs.

Sub-cluster 12: Respondent's 10 years perspective

This data sub-cluster refers to the respondent's insights on the future of their personal career. This constitutes a second layer for understanding their vision on the future of product development, design, and engineering.

Similar but evolving career:
The first and largely dominant trend expressed by respondents when thinking about the future of their career was to stay in a similar but evolving role.

In 10 years, 7 respondents expect themselves to keep their role of designer or developer, with more involvement into the strategy and business. UX Designer 3 for example would like to become a design leader. They pointed out designer CEOs of companies such as Airbnb and Pinterest to be role models. Both UX Designer 5 and Developer 3 also intent to step into the area of strategy and business, rather than purely staying in the execution sphere, as today. UX Designer 2, Developer 1, and UX Designer 6 expect to have a deep understanding of the business field 10 years ahead.

Staying on a similar career path, 7 respondents pointed out the diversification of their role as strongly intented along the next 10 years. UX Designer 3 see themselves designing different forms of solutions, and most likely stepping beyond digital products to the detriment of services and systems. On a different note, UX Designer 13 has in mind to try different operating fields. UX Designer 2 outlined the tendency of the designer role to wear different hats, and believe to be impacted by that. Project Manager/Lead 4 intents to diversify their manager role and putting a stronger focus on rising design maturity within their current or future organisation. As an overlap with the previous paragraph, Developer 4 expects to become more of a leader software developer. Developer 4 and Manager/Lead 2 both intend to step into the area of mentoring and coaching. Finally, Project Manager/Lead 2 believes manager roles to be moving closer to the field of data analysis in the future.

4 designers expressed their motivation for increasing their level of technical knowledge and expertise in the next 10 years. Few of them also went further by elaborating on their intention to become "full-stack designers", in other words, taking care of more tasks, including engineering tasks.

Reflecting not on what they intend to do in the future, but on what they don't, UX Designer 7 and UX Designer 8 mentioned the task of managing and leading. They both deliberately expressed their wish to keep doing what they truly enjoy, designing. Whereas other respondents intent to go that way, UX Designer 7 and UX Designer 8 believe that managing people is too demanding and that leading doesn't reflect their personality.

On a more practical level, 2 respondents expressed their intention of staying in a similar type of organisation in the next 10 years. UX Designer 7 enjoys working in agency contexts, wheareas UX Designer 13 thinks of staying in-house.

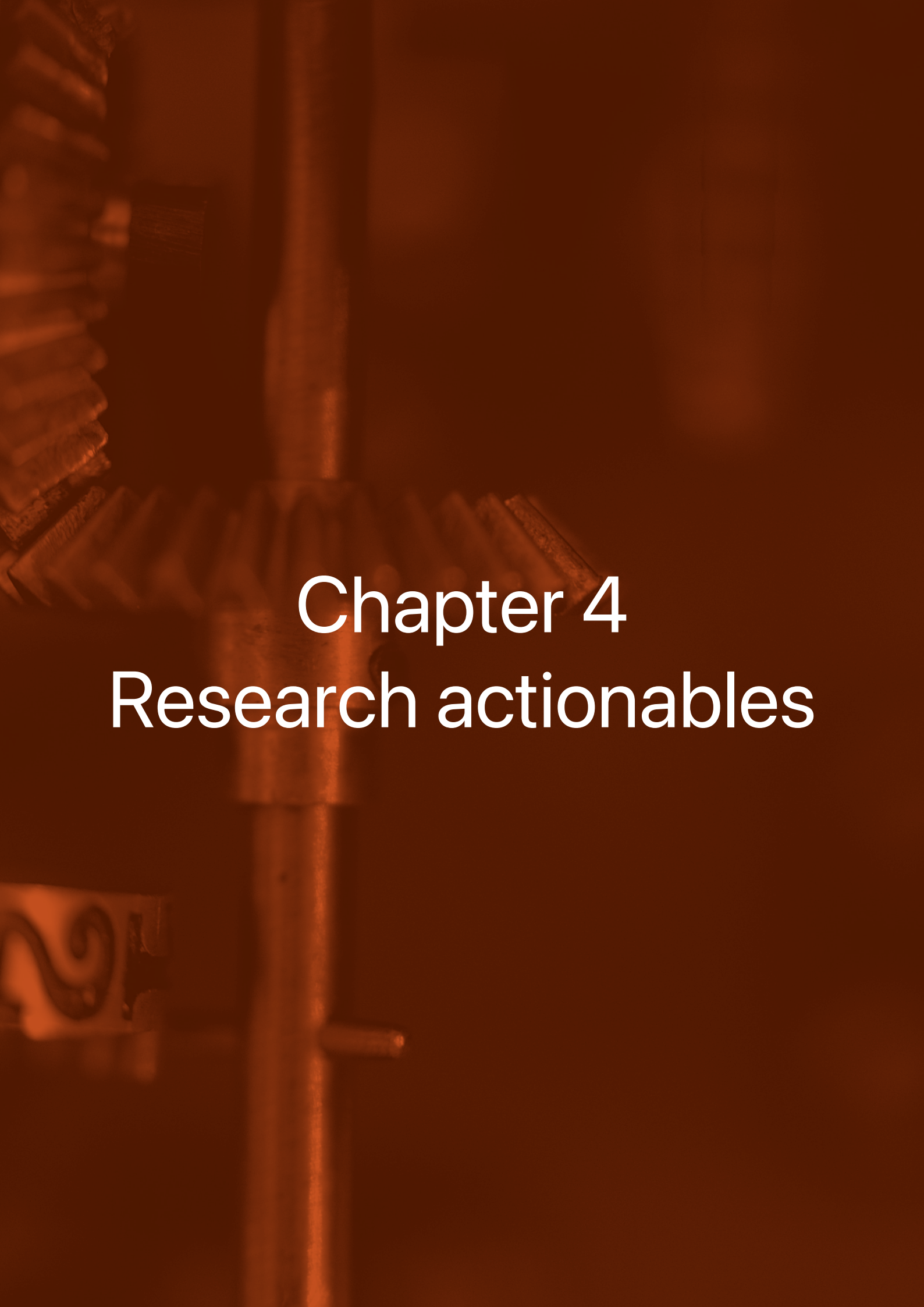
2 respondents reflected on their future work settings. As it becomes more and more prominant nowadays, UX Designer 13 believes flexible ways of working, mixing remote and office hours, to become the norm. UX Designer 8 shared the same reflection, while emphasizing more on their wish to keep working in physical settings, as they believe physical work to be more enjoyable and motivating.

Changing career path:
Only 3 respondents when thinking about the future of their career formed the second trend, being the intention to change their career path. UX Designer 12 doesn't think of continuing as a UX designer, but instead, moving back to the field of branding, where they originally come from. Developer 2 has in mind an interesting switch from software developer to service designer. Finally, Developer 6, being in a temporary role position, has plans to move to research and focusing on data science.

Creating own business:
A third and "micro-trend" was expressed by the intention of only one respondent, UX Designer 6, to create their own project or business, without concrete ideas on this yet.

To wrap-up

Structured in four clusters and twelve sub-clusters, the above findings have laid down the detailed insights from the research respondents having taken part in this empirical study. These findings, presenting answers to each interview question (Appendix), as well as some extra emerging themes and patterns, allowed the author to answering the three sub-research questions, and main research question of this thesis. Structured following these sub-research questions and main research question, the next and ultimate chapter provides the interpretation and actionables of these findings. The answer to the main research question constitutes the draft of a methodology aiming at formulating and adopting Agile UX in a given context.



Chapter 4
Research actionables

Introduction

Following the collection of valuable and extensive data from industry practitioners in the previous chapter (Chapter 3), as a way to addressing the problematization formulated in the background research regarding current limitations of the Agile UX approach, this fourth and last chapter presents the actionables of the thesis. The actionables aim at making sense of the findings. Additionally to answering the three sub-research questions having guided the semi-structured interviews, the core of the research actionables is the proposition by the author of a three phases methodology draft to formulating and adopting Agile UX, as an attempt to answering the main research question of the thesis: *How to contextually apply an Agile UX approach to product development, today, and tomorrow?*

This fourth and ultimate chapter is composed of two sections. The first section "Discussion" opens with a first part dedicated to laying-down some limitations of the empirical research. Then, the actionables are presented, being structured following the three sub-research questions and main research question of this emprical study, reminded below:

SUB-RESEARCH QUESTION 1:
What is today the level of Agile UX approach to product development, and the state of the relationship between design and engineering in the industry?

SUB-RESEARCH QUESTION 2:
How are processes and collaboration models defined in contextualized product development?

SUB-RESEARCH QUESTION 3:
How will product development processes and collaboration models be affected by the evolution of the design and engineering disciplines, as well as the technologies, in the future?

MAIN RESEARCH QUESTION:
How to contextually apply an Agile UX approach to product development, today, and tomorrow?

The second section lays-down some suggestions to researchers and practitioners for further research to be taken forwards, following this thesis.

The conclusion of the overall thesis research is presented subsequently to this fourth and last chapter.

1. Discussion

1.2. Limitations

In addition to the limitations detailed in the literature and Web reviews respectively (Chapter 1 and 2), constituing the background research limitations, the author recognizes some limitations to the empirical research. Firstly, the mono-method nature of the research, althought numerous interviewees in various companies were interviewed, is a limitation in the type of insights gathered. It is also to be noted that research respondents were choosen based on the author's professional network, potentially narrowing down the scope of the research insights. As the distribution of respondent's roles show, and considering the author being a designer himself, there is an unbalance between the design perspective and the engineering and management perspective in this research. Most likely, this does not influence on the quality of the insights and actionables, but implies to bear in mind which lence the look at this research through, and outcomes to potentially be more targeted to designers than other roles in digital product development. Another limitation to be mentioned are unbalances in the respondents and their organisations. Finland is almost the only country represented, business to business (B2B) organisations are twice as more represented as business to customer (B2C) organisations, and established and experienced organisations are three times more represented than start-ups/scale-ups. Finally, the author recognises that biases might have affected the documentation, analysis, and interpretation of the qualitative data.

In addition to the aforementioned limitations, the author's young experience in undertaking academic research and writting needs to be considered.

The author is aware of those limitations and embraces them, as it is an inevitable part of any research advancement and knowledge construction. All risks inherent to those limitations suspected to affect the quality of the research were considered seriously and mitigated as much as possible.

1.2 Actionables

SUB-RESEARCH QUESTION 1:
What is today the level of Agile UX approach to product development, and the state of the relationship between design and engineering in the industry?

Firstly, the relationship between design and engineering varies a lot from one company to another. Two major groups can be distinguished. The first group encompasses companies where design and engineering are considered as a synergic duo and where the environment embraces cross-disciplinarity. Within these organisations, a good mindset, willingness and efforts to understand each others' discipline, and various initiatives and practices to involving each other, can be highlighted. The second group describes organisations where the relationship between designers and software developers has more issues, and the environment is a source of challenges. Within these organisations, sources of challenges similar than those presented in the background research (Chapter 2, 2.2.2) can be found, such as discontinuous collaboration, broken communication, lack of design maturity, or processes constrained by Agile methodologies.
A given company most likely doesn't fit exactly in one or the other group, but this disctinction gives an idea of the inequalities of the industry. Overall, this research highlights that more companies are part of the first group, compare to those where the environment is less optimal. This gives a sign that things are getting better between designers and developers, despite the challenges depicted in the background research. Nonetheless, there is still a lot of work to do for companies clearly part of the second group. The research also highlighted that although a company is put into the first box, synergic design-engineering relationship, still some issues in product development and collaboration are potentially noticed. For the second group, implementing an Agile UX approach to product development would be a bigger effort and also very needed. For the first group, Agile UX would come as a good complement on top of already promising product development workflows.

In regard of collaboration, the research also highlighted the aspect of informality to be embraced in some organisations more than others. Informal initiatives of any sort and form, for example casual hangouts, out-of-office activities, or ice-breakers, according to the respondents allow to tighter the bonds between people and foster a more seamless collaboration. Informality is rarely a major concern in product development, and organisations in general, but people working within environments that totally disregard that aspect did recognise a desire for it. On the other hand, practitioners within organisations valuing informal initiatives strongly acknowledged that it helps creating a safe environment, nurturing motivation, and ultimately enhances the quality of work. More specifically with adopting Agile UX in a given organisation, informal initiatives can have a direct and positive impact on most of the approach's fundamentals, such as design-engineering mutual understanding, unified and cross-functional team, or active and transparent communication.

To go a bit broader than the aspect of design-engineering relationship, the sections below depict the state of Agile UX in the industry by reflecting on each of the 8 Agile UX cores formulated in the background research (Chapter 2, 3.3.3), and highlighting fluctuations across organisations.

Agile UX core 1: A high design maturity level:

The aspect of design maturity fluctuates a lot in the industry, from one organisation to another. The design-engineering ratio for example is not always balanced. One respondent even affirmed to be the first designer in their company. The design maturity, including time and resources allocated for design, is also one of the main aspect that respondents whish to see improved within their organisation. Overall, design processes and practices get much less consideration than for engineering, which can probably justify the DesignOps gaps mentioned by the respondents.

Agile UX core 2: UX designers showing leadership:

Very connected to the first core, the ability for designers to show leadership within their organisation is unequal in the industry. Designers struggle to spread their voice and act as leaders due of a lack of design buy-in, design time and human resources, but also because of the hierarchical model in place. The hierarchical model in place is a key aspect influencing on design leadership. Organisations with flat hierarchy, or those where designers are represented at the top of the pyramid, offer a more optimal environment for design leadership. On the contrary, organisations only having designers in junior roles, with an important emphasis on decision-making levels, provide a very limited room for design to be trully empowered.

Agile UX core 3: A mutual knowledge and understanding between design and engineering:

An aspect that some practitioners wish to see improved withing their team is peer-learning and knowledge sharing. Although this does not give an accurate view on the mutual understanding between designers and engineers, that pin-points room for improvement. Additionaly, as well detailed in the first paragraph of the previous page, over half of the industry seems to consider designers and developers as a synergic duo. Within these organisations, the healthy mindset, efforts to understand each others' discipline, and the various collaborative practices involving each other, clearly favor of good level of knowledge and understanding between designers and developers. This level is probably lower in the other group of organisations, depicting more sources of challenges between design and engineering.

Agile UX core 4: A unified and cross-functional team:

Designers and developers either are part of one cross-functional team, or considered as part of their respective design or engineering team. Indeed, the research highlights a distinction between companies differenciating design and engineering departments, and those embracing cross-functionality within teams, or areas. This Agile UX core 4 varies a lot in the industry, and can even vary within the same organisation. Indeed, the findings are able to tell that settings can differ from one department or team to another. Also, teams cycle is another aspect that fluctuates between different contexts. Individuals being part of an agency organisation are shuffled more regularly and teams re-composed more often, since projects usually are shorter. For longer, or continuous projects, mainly in-house contexts, a given team can stay the same on a long term basis. This influences on the ability and ease for people to work together, and be unified. Additionally, the aspect of team spirit and psychlogical safety has been highlighted to be another point to improve, which definitely demonstrates the need to work on that.

Agile UX core 5: Shared project vision and goals:

No major insight allowed to clearly formulate a reflection specifically on the aspect of shared vision and goals. Nevertheless, the 8 Agile UX cores being interdependent and connected, understanding the other cores allow to indireclty deduce this one. It is for example very likely that a team being unified and cross-functional (core 4), collaborating continuously (core 6), and communicating actively (core 7), has a well shared project vision and goals. On the contrary, an organisation encountering difficulties in linking-up people together will most likely have difficulties in maintaining alignement on the project vision and goals.

Agile UX core 6: A continuous collaboration:

Out of all the organisations having taken part in this research, half seem to embrace and encourage close collaboration between design and engineering. At the same time, this aspect of collaboration, including among others early involvement of design and/or engineering roles within projects, has been one of the points that practitioners stressed should be improved. This depicts again some level a variation on how well designers and engineers work together. People's physical location, impacting on the ability for teams to establishing and maintaining a continuous ground between design and engineering, is also a very distributed aspect. Very few companies nowadays seem to have designers and developers physically collocated (2 out of 24). There is a strong predominance of remote and hybrid work, impacting directly on the nature of the interactions. Regarding the tools employed by practitioners, discipline specific and collaboration tools, many have become sharable and optimal for cross-disciplinary and remote collaboration. Nonetheless, tools can be great, but how people use them really what impacts on the collaboration experience. Another aspect related to collaboration, mostly in agency contexts, is the multi-organisational nature of the team, when people, including developers and designers, come from a mix of two or more companies. This is in some cases an extra source of challenges to deal with in order to ensuring continuous collaboration.

Agile UX core 7: An active and transparent communication:

As to the aspect of continuous collaboration, the communication design-engineering seems to be well managed in half of the organisations. Respondents described meetings, numerous communication initiatives, regular rituals between designers and developers. Again, as the collaboration, the communication is very impacted by individuals' working location. As said above, the industry nowadays seems to favor hybrid or remote working settings, as opposed to fully collocated designers and developers. The tools also impact on the nature of the communication. As detailed above, tools used in the industry have become more cross-disciplinary and optimised for remote settings. Therefore, they can really empower designers and developers to communicate optimally, if making good use of them.

Agile UX core 8: A unified and flexible process; the Agile UX process:

About the overall product development process, the industry again fluctuates, and according to respondents, it even fluctuates across projects within the same organisation. Indeed, based on nature of the tasks, the goals, the timeline, the people involved, and many other aspects, the process has some part of uniqueness each and every time. Some common patterns also have been highlighted by the research. Firstly, the high level phases of the product development process, althought considered and executed differently in the various contexts and projects, seem to correlate. The author was able to depict seven major product development phases based on the insights: The Kick-Off, the Discovery, the Definition, the Design, the Validation, the Delivery, and the Follow-up phases, themselves embeded in three major parts: the Problem space, the Solution space, and the Execution. Practioners involved in this research admited the fact that bearing in mind the major phases of the project helps to aligning the team with the process.

In the industry, Agile methodologies are the main driver for product development in many organisations. Nonetheless, the level of rigidity to which Agile is applied seems to vary a lot. Organisations being flexibe and open with the Agile practices would most likely experience a more seamless switch towards an Agile UX approach, as opposed to those applying them very strickly. Design processes and practices overall, in other word design operations, are not properly considered, and designers often have to define these on the go. The research also depicted some level of initiatives in some organisations to standardising practices and processes, as for example creating and re-using process templates, handbooks, guidelines, and so on. This is a clear sign of an emerging Agile UX mindset. Many challenges were also highlighted with the overall product development process, some big ones being the timeline, the billing model, the tools, the multi-company nature of certain projects, requirements tracking, and documentation. Finally, numerous aspects related to product development processes clearly trigger a wish by practitioners to be improved, particularly the process structure, design and development operations, documentation, and flexibility.

What is detailed above emphasizes the diversity of the industry when it comes to the design-engineering relationship and the level of Agile UX approach to product development. The research also allowed to understand and map numerous aspects intrinsic to a given team or organisational context, suspected to impact on practitioners' experience of product development, and on the ability to establish an Agile UX model. Among these aspects, major ones which have not already been detailed above, are the managerial methods, the company development stage, and the human and financial resources. For more details on all the aspects, refer back to Chap.3, 3.3.3. "Sub-cluster 10: Aspects influencing processes and collaboration".

SUB-RESEARCH QUESTION 2:
How are processes and collaboration models defined in contextualized product development?

Similarly as for the first sub-research question, the research sheds the light here on numerous contextual sides proper to a given context, and influencing how processes and collaboration models are defined. All these characteristics, varying from one organisation, from one team, and from one project to another, demonstrates the diversity of the industry and the uniqueness of each and every product development context, having their own set of characteristics. Therefore, when it comes to defining processes and collaborating in a given context, it is done in a unique way to some extent. Some high-level contextual facets are the organisation type (in-house or agency), the business model (B2B or B2C), or the operating field. Some facets related more specifically to the project and product are the size and scale, the project timeline, the nature of the product, the type of end-users, and the nature of the project (improving or creating from scratch). Some company level aspects are the development stage, the budget, the human resources, the managerial practices, the company structure, the design maturity, and people's working location. Some characteristics related to the people are their experience and their skills. In agency contexts, the client is another major contextual aspect. Added on top of all the possible characteritics defining a given context, specific organisational factors are also suspected to consequently influencing on processes and collaboration at a given point in time, some examples highlighted by the respondents being the level and pace of business growth, business model switches, and company acquisitions.
For more details on all the contextual aspects, refer back to Chap.3, 3.3.3. "Sub-cluster 10: Aspects influencing processes and collaboration".

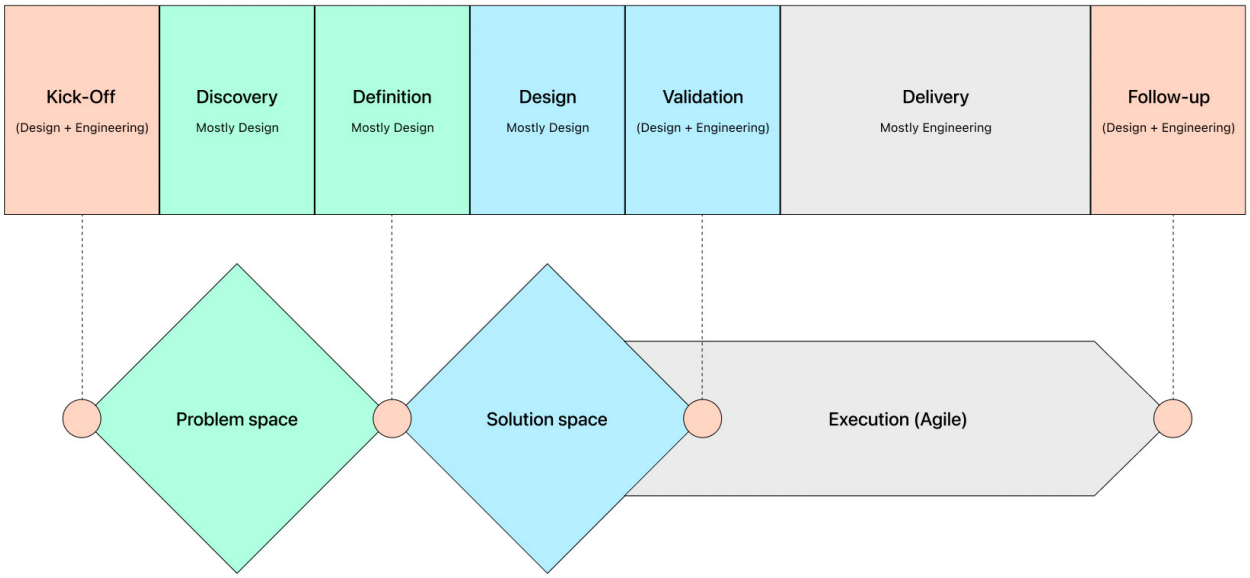
With the given that processes, collaboration, and the way they are established being to some extent unique for each organisational product development context, including each team, and each and every project, the author now reflects deeper on the sub-research question 2. Are covered below the nature of the process, the process phases, the practices and operations, the collaboration, and the tools.

<u>Nature of the process:</u>
The overall nature of the product development process in the industry shares common trends across organisations, but also has specificities when looking at given project contexts. The project timeline seems to be a major component of the process, and its nature inherently influences this latest. As detailed in the previous part already, Agile methodologies are widely adopted in a big proportion of the industry, where they are a major driver for product development processes. At the same time, design specific practices seem to be less considered and less carefully defined. Based on the fact that given projects will comport unique sets of tasks and objectives, an important characteristic of the product development process is its flexibility and ability to provide teams with room for freedom and experimentations along the way. The research shows that this aspects is not equal everywhere, not only between different organisations, but also between two projects or teams of the same organisation. Overall, a small half of the organisations involved in this research embrace flexibility in the process. Few organisations allow flexibility under more justified conditions, for example if trust is established, or without urge in the timeline. Finally, a small third acknowledges being very tighted to Agile methodologies, and doesn't seem to embrace flexibility. The nature of the product development process is also impacted by the type of project. The research highlights two major types, the continuous projects, mainly intrisict to in-house organisations, and the finite projects, usually in agencies. Between these two types, the timeline and pace fluctuation impacts on multiple aspects of the processes and collaboration, such as room for flexibility, space for failure and iterations, goals alignement, and team spirit, among others.

Process phases:

When it comes to the phases of the product development process, there are trends that correlate across organisations. The research allows to map a high-level pattern that seems applicable to most of the product development contexts. Seven high-level phases reflect the industry's conception of the product development process. These phases are the Kick-Off, the Discovery, the Definition, the Design, the Validation, the Delivery, and the Follow-up. The Kick-off represents the initiation of the project, where the team aligns on the goals and objectives, as well as the timeline of the project. The Discovery constitutes the research of the project, where designers, and potentially other roles, aim at empathising with end-users and understanding their context. The Definition phase is the problem defintion, or problem re-framing phase. In this phase, project objectives are re-visited. The Design phase represents the solution ideation phase, where ideas and concepts are formulated, developed, tested and iterated. The Validation constitutes the solution alignement phase, and is the ultimate phase before the implementation of the designed solution. The Delivery includes the design hand-offs to engineering as well as the technical execution of the solution. The Follow-up phase is probably the broader phase, following the release of the solution, and mostly aims at measuring the success of the solution by tracking pre-defined metrics and KPIs.

Combined together, the Discovery and the Definition phases consitute the problem space of the project, and the Design and Validation phases constitute the solution space. The diagram below visualises this high-level product development pattern drafted based on the research findings.



The seven high-level phases of the product development process, based on the findings

The level of consideration of these phases and how they are practically conducted varies from one organisation to another. It is also good to remember the iterative nature of product development, and the fact that these phases are not always done in a linear way. Regardless of that, according the respondents, these high-level phases help teams maintaining alignement on the process.

It is to be highlighted that these phases focus on design and engineering disciplines. A more complete model, including business, marketing, and any other discipline involved in product development, would embed this formulated model, and include additional phases.

For more details about the phases from the research respondents' perspectives, refer back to Chapter 3, 3.3.3 "Sub-cluster 4: Processes and collaboration; Process phases".

Practices and operations:

Regarding product development practices and operations in the industry, this research also highlights variations and diversity. A trend repeating in a fairly big proportion of the companies involved is the lack of established and solid design operations (DesignOps), as opposed to usually more considered development operations (DevOps). Similarly as previous described aspects, the design and engineering practices can vary within the same company, specifically in cases of companies whose organisational structure clusters people per area, as opposed to discipline related departments. Indeed, different product areas include projects and tasks with varying objectives, consequently defining their own set of practices to adopt throughout the process. There are fluctuations with the practices between the continuous and finite types of projects. For example, finite projects are more subject to team forming processes than continuous projects, where people often know each other already, and team shuffles are more seldom. Overall, kick-off practices seem to be shorter in continuous project environments compare to finite. The research also highlights growing considerations when it comes to standardising product development, engineering and design processes and practices. Some organisations for instance dedicate time to focus on elaborating handbooks, guidelines, project templates, or internal research aiming at enhancing processes. Nonetheless, particularly well considered initiatives on that regard only occur in a small proportion of companies.

Collaboration:

The way collaboration is established varies in the industry. In over half of the organisations, the design-engineering relationship is considered to be synergic. The research highlights common activities, mutual involvement, and good communication between design and engineering. Some common activities are the timeline drafting, feasibility checks, or design handoffs. Overall, collaborative initiatives seem to be a good mix of pre-defined rituals, such as sprint plannings, and on-the-go initiatives, such as spontaneous alignement checks. Two big elements play an influence on the collaboration in product development, the organisation hierarchical model, when it comes to decision-making and validation rounds, and people's physical location, mostly impacting on the communication and frequence of interactions. Based on the timeline and length of a given project, refering back to the distinction between continuous and finite projects, the familiarity and the ease that people have to work as a team can be impacted. In finite projects, people usually have less time to getting used to work together. Lastly, a good way to foster an optimal collaboration and enhancing team spirit, trust, and psychlogical safety is the aspect of informality. Informal initiatives of any form, from weekly casual chats to afterworks, from relax office spaces to company events, seem to have a positive impact on product development's overall atmosphere and quality of work. Nonetheless, this aspects is unequal in the industry, due to varying levels of consideration, or due to budget.

Tools:

The choice of the tools is probably not the most important concern, in comparison to aspects such as the team mindset or vision alignement, but they definitely are huge enablers in the process. Overall, based on the findings, practitioners in the industry use two major types of tools, being the discipline related tools on one hand, and the collaboration tools on the other. For discipline specific tools, the design tool widely adopted is Figma. Regarding engineering, GitHub was the most mentioned in this research. For collaboration tools, work and task management tools, such as Jira or Trello, were mentioned. For communicating, practitioners seem to highly rely on Slack, combined with video call tools, such as Zoom. For project documentation and storage, Notion, Drive, and Confluence were mentioned. For specific collaborative activities, Miro and FigJam seem to be the go-to options. Based on the insights, tools are not always freely choosen by the practitioners, potentially affecting their ease to use them. Overall, tools with the holistic and sharable attributes seem to gain in popularity, in the quest of continuously reducing their number.

SUB-RESEARCH QUESTION 3:
How will product development processes and collaboration models be affected by the evolution of the design and engineering disciplines, as well as the technologies, in the future?

As participant's insights highlighted in this research, numerous sides of the design and engineering disciplines, as well as product development and digital products in general, are suspected to evolve in a near and far future. This is particularly hard to provide an accurate answer to this sub-research question 3, not only considering the limitations of this research, and the fact that the insights were partially based on speculations, but also due to the part of unknown and unpredictabilities of the future trends. Nonetheless, the findings of this empirical research provide an interesting reflection around some lines that worth paying attention to.
The author adresses this sub-research question 3 in two parts. Firstly, the insights provided seem to be asserting the relevance and the "future-proof" nature of the Agile UX approach to product development. On a second level, the insights open up a broader reflection on the evolution of product development, design, and engineering, and its potential impact on Agile UX.

When being questioned on the evolution of their respective discipline, and product development, the practitioners expressed trends that correlate with the Agile UX model. A first well shared idea is the fact that design and engineering disciplines will blend more and more with one another. The aspect of systematisation and standardisation of design, and product development processes also rises thoughts in the industry, while predictions of consequent evolutions of the product development tools are made. These trends directly connect with the Agile UX cores 6, 7, and 8, about collaboration, communication, and unified processes. Many respondents highlighted design maturity and the rise of UX design as a near future trend as well. They confidently see the design discipline at a more strategic level in more and more organisations. This correlates with the Agile UX core 1: "A high design maturity level". From the engineering's side, respondents also perceive their role to be closer and closer to the strategy in the future. Overall, the insights seem to align with the idea that both designer and developer's roles will be more holistic and strategic, as opposed to somehow more specified today. Along with this idea, designers wish to gain more technical knowledge and understanding, which links to the Agile UX core 3: "Mutual knowledge and understanding between design and engineering". Software engineer becoming a more holistic role potentially contributes in favoring this aspect of design-engineering awareness as well. Among the additional skills rising in importance for design roles, the ability to lead, advocate, and facilitate design processes and philosophy seems to be another well foreseen idea, aligning with nothing but the Agile UX core 2: "UX designers showing leadership". With both designers and developers gaining more skills and becoming more diversified, it is probably justified to think that collaboration in product development will be more seamless (Agile UX core 6). Within product development processes, the rising aspect of co-creation with end-users seems to trigger attention in people's minds. This goes in pair with the fact that users will become more and more experts and well aware of their own needs in the future. End-users being more involved in the product development process is a great sign of a high-level design maturity (Agile UX core 1), great collaboration (Agile UX core 6), and overall successfully adopted Agile UX approach.
Regarding digital products more generally, the insights make it clear that their number will keep growing exponentially. Therefore, it appears even more crucial for these products, embedding people's lives, to have strong and well defined purposes, in addition to providing quality experiences to end-users. Furthermore, practitioners seem to share the idea that product lifecycles are becoming shorter and shorter, impacting on a rising need for continuous improvements. All of that emphasizes the importance of enhancing product development processes and collaboration, and how beneficial Agile UX will be in this journey.

Following the overall idea that the Agile UX approach fits in the future landscape, other trends highlighted in this research feed a broader reflection on the evolution of digital product development, design, and engineering, and inherently the impact on the Agile UX approach and its adoption. How many opportunities will the evolution of digital technologies unlock for how many potential challenges on the way? Will the industry, and approaches such as Agile UX, keep up with the ever growing demand for purpose-driven, business centric, and user friendly digital products?

A first topic bringing a lot of questions and contributing to the evolution of product development is the tendency of processe automation, spreading both in design and engineering's disciplines. This rises the question of what in product development will UX designers and software developers not take care off anymore, and potentially, what are new tasks they will perform, or skills they should gain? Automation will surely impact the nature of the processes and to some extent how people collaborate. Some phases of the process are maybe suspected to become shorter, for example the design phase with ready-made solutions or templates that designers would only have to tweak, some of it already happening today. In terms of collaboration, automation might as well impact on the interactions. One could wonder what proportion of interactions across disciplines can be supressed, and are there any that will stay vital no matter what? Another example, with the rise of data and evolution of data collection methods, is about the research phase and whether research can be automated at a point that interacting with end-users would not be needed anymore. Despite the fact that many aspects of product development, design, and engineering seem to be subject to automation, the insights manifest that a human touch will always be needed. Some designers think that qualitative research as well as creating concept solutions to complex problems cannot be automated. Developers seem to have the same reflection in regards of technical back-end infrastructures. On the contrary, when purely entering into the territory of project execution, as UI design and front-end development, this is where automation will maybe have the biggest impact.

Another trend highlighted in this research, and very connected to the first one, is the evolution of product development tools. It appears that tools constantly evolve in the direction of making processes faster and collaboration easier, but what will be the limit to that? Up until which point can the tool empower digital practitioners, and how much time and effort can they realistically save? It also worth keeping in mind that constantly fastering and easing processes, as one of the Agile's goals, brings up its part of challenges.

A facet of product development also triggering of lot of reflection is data and analytics. With the constant rise of the amount of data on people, products, and markets, as well as the evolution of the role of data in product development, impact on processes and collaboration will most likely be consequent. One could wonder how this will affect the user research process for example. In connection to that, insights highlighted the particular importance that "Thick", or qualitative data should take over "Big" or quantitative data. While the first pragraph seemed to inform about potential automation in the research process, will this rise of "Thick", data only push designers to be more specialised? Will the field of design research actually bloom based on that? Regarding processes, it is likely that the rise of data, but especially their quality, will allow better and quicker decisions on products to be made. It is also to be born in mind that data transparency and ethical use of data are highly rising trends, therefore, data collection and management will likely to see regulations and more standard practices. Finally, the aspect of data bridges directly to another growing topic, influencing product development, being artificial intelligence (AI) and machine learning. Hopefully, this field of technology will also bring contributions into making digital processes faster and collaboration easier, but again, with which proportion of compromises and potential challenges on the way?

Another big evolving trend in digital products nowadays highlighted in this research is accessibility and inclusiveness. There is an ever growing concern and need for creating better products, but also better products for everyone, regardless of culture, gender, origin, habit, and handicap. This topic definitely emphasizes the need for more design recognition in the industry. Accessibility and inclusiveness are still evolving fields, and they will most likely model future aspects of user research as well. Co-creation, and the rising need to involving end-users into the product development process, connects with that as well.

Few other points have been touched upon during this research, which will certainly bring their contribution to the evolution of product development and digital products. The aspect of security seems to be an ever growing concern, due to many reasons, such as the increase of data, amount of digital products, and hacking threats, among others. Few respondents also reflected on the evolution of the project manager and think that this evolving role will be become more crucial in bringing design and engineering together, and facilitating product development in the future. The insights also vision the apparition of potential new fields and disciplines in product development, such as sociology, psychology, and anthropology. The evolution of the working settings also generates lots of thoughts. All in all, practitioners are concerned and skeptical about the industry operating in a fully remote way. They think that offices and human live interactions will always be needed, at least partially. Finally, there is a big reflection around emerging technologies, potentially modeling the future of digital experiences. Examples of these technologies are voice and sounds interactions, virtual reality, 3D, and sensorial technologies in general. With these emerging opportunities, the evolution of digital experiences beyond the 2D screens brings up many questions around the nature of product development, and the design and engineering roles.

It is worth emphasizing that above are only few aspects out of most likely many others not uncovered in this research. Overall, a lot of opportunities for better product development processes and collaboration are foreseen, while Agile UX seems to align with the future trends. Nonetheless, many aspects and unknowns open up discussions and trigger questions, demanding practioners and researchers to keep their eye constantly open in order to anticipating the future.

MAIN RESEARCH QUESTION:
How to contextually apply an Agile UX approach to product development, today, and tomorrow?

Following the answers to the three sub-research questions, the author tackles here the main research question of the thesis. All the answers above partially contributed into answering this main question already, but the author's intention here is to provide the reader with a more practical perspective, and actionable solution to be taken out of this research. After all, the title of this chapter is "Research actionables". The author proposes below a three phases formula as the draft base of a potential deeper methodology aiming at formulating and implementing an Agile UX approach to product development contextually. The idea of this methodology is to enable any organisation and team in any context of the industry to being able to generate and put into practice their own Agile UX framework.
As a reminder, the main research question addresses the two Agile UX gaps formulated in the background research, therefore is the proposed solution as well. The first Agile UX gap depicts a lack of attention and consideration of current Agile UX frameworks on industry's contextuality, and a lack of actionables to apply Agile UX. The second gap describes a lack of anticipation of current Agile UX frameworks to the evolution of product development, design and engineering's disciplines, and digital technologies in general.

The three phases detailed below present an idea and proposition of structure for a more detailed methology to be designed. With a deeper and more targeted research, and after testing and validation rounds, this more complete model could be the proper solution for teams and organisation encountering challenges between design and engineering, and eager to enhance their product development processes through Agile UX. This more detailed methodology could take multiple forms, such as a toolkit or handbook. It could as well stay a simple and high-level three phases model, which would already provide teams and organisations with a clear direction line and good support to plan and facilitate their Agile UX adoption. Activities and materials could be developed to support the methodology, such as workshops, mapping exercices, diagrams, and canvases. For each of the three phases detailed below, the author provides the reader with ideas and suggestions of such activities and materials. Not to go too far in describing what the methodology could be, the nature of the current three phases draft is like its end goal, it is meant to be supportive, flexible, and customisable in order to fit in a given context, organisation, team, just like the Agile UX approach itself should be.

Before jumping into the three phases, the author highlights here few pre-conditions that better be met in a given context before considering following the methodology and adopting the Agile UX approach. Firstly, there needs to be some level of buy-in and willingness to improve product development processes and collaboration. If this is not the case, it will difficult to facilitate the methodology and maintain Agile UX over time. A solution to that could be to advocate Agile UX and highlight the need for it. Practitioners initiating the movement could for example report potential challenges in processes and collaboration within their team or organisation, and demonstrate how adopting Agile UX could solve these. Another condition, the Agile UX approach and its goals should be understood, at least on a high-level. The benefits of the approach should be clear; this will actually help people, especially designers and developers, to better work together, and enhance product development processes. To help in creating this initial understanding, a small handbook aiming at describing Agile UX could be a good material support to the methodology at that point. Finally, although the methodology aims at staying lightweight and tailored to a given context's needs, there should be a minimum level of time and resources dedicated to it, and people involved should be available for the journey. As an example, it might not be the priority for busy startups, or organisations facing financial issues, with the need to focus all their energy on keeping their business up and running.

Phase 1: Understanding and evaluating:

Once a clear need for enhancing product development processes and collaboration is recognized, and the aforementioned pre-conditions are met, the first phase of the methodology focuses on two major aspects. On one hand, it aims at forming a clear understanding and awareness of the organisational and team context. On the other hand, it aims at evaluating the status-quo of the current processes and Agile UX maturity, and uncovering challenges to be solved.

- Empathising with the context

This first point is essential for people to better picture the context they're working in, even if they are already aware of it. It is also useful for the next phase of the methodology to formulate the tailored Agile UX framework. Again, Agile UX won't be implemented the same way in different organisations, even if similar challenges are to be solved. Empathising with the context in a nutshell consists in highlighting and understanding all aspects related to the organisation or team that have an influence on the ways of working in product development, so that they can be taken into account to formulating the framework. To help teams rapidly mapping and understanding the different characteristics of their context, this step of the methodology could be supported by a properly designed canvas and a context mapping workshop to fill it in. The canvas could be structured following the following high-level structure:

1/ Overall environment characteristics:
- Company type (e.g. in-house, agency...)
- Company stage (e.g. startup, scaleup, established organisation...)
- Business type (e.g. B2B,B2C...)
- Operating field(s)
- Country(ies)
- Description of the client company (if applicable)

2/ Details about the organisation and settings:
- Description of the design maturity
- Description on the state of DesignOps and DevOps
- Hierarchichal model (e.g. flat, seniority distictions, decision-making bodies...)
- Working location (e.g. people collocated, remote, hybrid...)
- Any other factor(s) potentially impacting on processes (e.g. hight growth, business model switch...)

3/ Details about the product(s) and/or project(s)
- Description of the product (e.g. nature, scale, scope...)
- Description of the users
- Nature of the project(s) (e.g. continuous, finite...)
- Description of the timeline(s) and milestones

4/ Details about the team(s) and people
- People profiles (e.g. role, experience, time in the company...)
- Number of teams
- Team structure model (e.g. cross-functional team(s), department related teams...)
- Any other people factors potentially impacting on processes (e.g. shuffling between teams, new teams...)

At this point, challenges can potentially be uncovered already, and be taken for the next step of the phase. Since settings are never set in stone, this context mapping should be revisited over time while the organisation evolves, so that the formulated Agile UX framework is always accurately targeted.

- Evaluating the current product development processes and mapping the challenges

This second step of the first phase is about identifying and recognising what goes well and what does not with the current ways of working. This is where the need for Agile UX should be highlighted. What should be improved in product development varies a lot based on the context, organisation, and team. At this point, it might be that only few minor challenges are uncovered, or on the contrary, the evaluation could reveal numerous weaknesses with the current practices in place. For example, there will be less work to do for organisations where the design-engineering relationship is considered to be synergic, as opposed to those where designers and developers are isolated. Similarly, organisations with a high design culture in place already are steps ahead on their journey towards an Agile UX approach. In any case, big or not, there will always be a room for improvement to be pin-pointed in the current product development processes. To support this step of the methodology, a challenge mapping workshop could be designed and complemented with a challenge map canvas. Interviews or surveys to be run internally could also help identifying the challenges. Examples of questions to be asked to practitioners could be: "How empowered do you feel with the current processes and practices in place?", "How close to the engineers/designers are you?", "What frustrates you currently?", "Describe your team synergy", etc...

Visualised and detailed on the challenge map canvas, the highlighted challenges could be clustered per type. The structure could for example follow the six challenges between design and engineering depicted in the background research of this thesis:

1/ Lack of design support and resources
2/ Mutual misunderstanding between design and engineering
3/ Unshared vision and differing mindsets
4/ Inadequate settings and contexts
5/ Clashing processes and practices
6/ Broken communication and decision-making

All challenges pin-pointed should be detailed by the participants as much as possible. Testimonials and case examples could be a good complement for everybody to empathise with them. At the end of this step, challenges in product development should be clearly defined and acknowledged by the team.

<u>Phase 2: Formulating:</u>

With a good awareness of the organisational context in mind, and clear formulated challenges to be addressed, the second phase of the methodology essentially aims at formulating the right Agile UX product development framework. This goes through two steps; firstly defining areas and aspects to focus on based on the given challenges, then drafting an Agile UX action plan, which will serve as a guide to adopting and sustaining Agile UX within the organisation and team.

- Defining areas of focus based on the identified challenges

Essentially, this first step of the second phase of the methodlogy aims at translating the identified challenges into more tangible actionables. The previous phase should have provided people with a clear understanding of what goes well and what does not with their current approach to product development, including processes and collaboration practices. The idea here is to maintain the good facets, and define what aspects should be paid attention to, and maybe modified, so that the pin-pointed challenges can be alleviated. The definition of the areas of focus will model the action plan formulated in the next step of this second phase.

As for the two steps of the previous phase, this step would highly benefit in being supported by a designed workshop in order to help participants to translating problems into opportunities. Two examples of challenges and their formulated areas of focus are provided by the author below:

Challenge 1:
Software developers and designers highlight difficulties in understanding each other, and often need to go through last minute energy consuming conversations to take decisions and make compromises, generating frustrations from both sides.

Area of focus 1:
Knowledge sharing could be one of the way to alleviate the above challenge. Time could be allocated for software engineers and designers to sit together, physically or remotely, and present aspects of their discipline on a regular basis. Being a more informal and relaxed initiative, this would also enhance psychlogical safety and ease for people to communicate with each other more continuously.

Challenge 2:
Designers highlight challenges in understanding business needs and user problems, as well as iterating design solutions within the fast-paced Agile environment. This challenge relates to clashing processes and practices between design and engineering.

Area of focus 2:
In this case, areas of focus would likely be the rise of design maturity on one side, and change in the product development processe, applying Agile methodologies to rigidly, on the other. Attention should be put into making processes more flexible, and putting desing more ahead of the engineering work.

- Formulating an action plan

With challenges translated into proper areas of focus, now is the time to draft a targeted action plan to alleviate the challenges defined, and ultimately enhancing product development processes and collaboration. The first thing to be done in this step is to analyse the defined areas of focus. It might be that some areas repeat multiple times, or different areas could be merged into one. The more synthetic the areas of focus are, the more actionable they are. A method such as affinity mapping could be a way to make sense of the defined areas. It is also necessary at this point to highlight the level of importance of each area, and prioritise them. Estimations of the efforts and tasks to be done for each area should also be done. Based on the context and available resources dedicated to adopt Agile UX, the effort estimation can potentially impact on the prioritization of the areas of focus. The formulation of the action plan could be supported by an ideation workshop. For each area of focus, participants would be able to generate ideas of tangible tasks and initiatives to be put in place. Additionally, a canvas, or handbook, could be proposed as a template to build the action plan. The eight Agile UX cores defined in this thesis research could be a good inspiration for structuring such a template. Below is a suggestion of action plan structure, with few examples of tangible tasks and initiatives for each category:

1/ DesignOps: Design maturity and practices
- Create design handoff guidelines
- Allocate time to enhance the design system
- Involve design representatives in strategy related meetings

2/ DevOps: Software engineering maturity and practices
- Make the Agile methodologies more flexible
3/ Design-engineering relationship
- Encourage knowledge sharing (monthly informal presentations)
4/ Team
- Initiate regular feedback meetings
- Re-define the team structure to be cross-disciplinary
5/ Vision alignment and goals
- Involve all team members in kick-off meetings
- Set regular get togethers to update on the project timeline and goals
6/ Collaboration
- Create collaboration good practices guide
- Initiate monthly informal gatherings and games
- Reduce the amount of tools and clarify their usage
7/ Communication
- Centralise design and engineering discussions in one channel
8/ Product development process
- Put the project timeline and milestones visible for everyone in one place
- Draft high-level process phases and model and put it available in one place

Above being only few examples of tasks and initiatives, a well formulated action plan would ideally comport more detailed descriptions of the areas of focus and their related initiatives. To be trully actionable, the action plan should also document the attribution of the tasks or initiatives to specific people. Clear goals and success criterias should also be linked to each area of focus, so that proper evolution tracking can be done in the next and last phase of the methodlogy.

Phase 3: Executing, tracking, and sustaining:
Finally, the last phase of the methodology represents the continuous execution of the action plan, in order to implement and sutain the defined Agile UX framework. If successfully undertook, this phase should reveal the results of the methodology and benefits of Agile UX; alleviated challenges, smoothier product development processes, and better overall collaboration between design and engineering. This continuous phase is depicted in three layers: Executing, tracking, and sustaining.

- Executing

Based on the formulation of areas of focus and their related actionables, put together and prioritized in a structured action plan, it is now time for people to execute what has been assigned to them. If the context was well considered during the formulation of the action plan, putting it into practice should no be felt as overwhelming for the team. The plan is supposed to alleviate the specific challenges highlighted with the given time and resources available for it. Before executing the plan, it can be useful to formulate a summary for the people involved on the concrete goals of the plan, and how they contribute into making it happen. At this point, it should be clear to everyone what they should take care of, work on, and/or pay attention to in their ways of working. It is also great to keep the outcomes of each phase and step of the methodology available in one place, so that people can refer back to it if needed; these include the context map, the challenges, the areas of focus, and the action plan itself.

- Tracking

Ideally, the action plan should contain goals and success criterias linked to each area of focus in order to enhance product development processes and collaboration. Some success criterias can stay quite high-level, as for example alleviating a given challenge, but they can also be broken down into more detailed KPIs or metrics. As an example, if one of the high-level success criteria is to enhance psychological safety and synergy within a team, more detailed KPIs could be the amount of informal initiatives undertook, the amount and quality of interactions between people, or the amount of co-creation throughout the process. If another high-level success criteria is to reduce mis-alignement and unshared decision making between designers and engineers, more detailed KPIs could be related to the amount of common meetings or activities undertook between both disciplines. The success criterias and related KPIs defined should be easy enough to track over time. If they require too much effort, the risk is that people will give up on tracking them. Tracking the evolution of product development processes, following the execution of the action plan, could also be done through regular surveys or interviews to be run internally. Again, this aspect of the methodology could be complemented with guidelines to help practitioners following-up with the execution of their action plan. Whether they are quantitative or qualitative by nature, success criterias and KPIs will tell people about the level of success of the Agile UX adoption within their organisation and team, and potentially inform on change or evolution to bring to the action plan.

- Sustaining

The third layer of this ultimate phase of the methodology is about sustaining the action plan and Agile UX enhancement over time. Based on the tracking described above, the people involved should re-visit the action plan, and potentially the context map, challenges, and areas of focus. Again, sustaining the practices and initiatives defined in the action plan should not cost too much effort. The overall idea here is evaluating whether the action plan is up to date after a certain amount of time or not. Multiple evolutions could triger the need for a potential update of the action plan. These evolutions are twofold; the evolutions as changes in the context, and the evolutions brought as the result of the action plan. An example of change in the context can be the growth of the organisation and new people hired. An example of evolution brought as the result of the action plan might be the rise of design maturity. For any of these two examples, it is necessary to re-evaluate and update the challenges and areas of focus, to ensure the constant relevance of the plan at a given point in time. Maybe new challenges have arisen over time, or on the contrary, challenges have been alleviated, redefining in both cases the areas of focus and their priority. Another indicator of the potential need to update the action plan is the thrive of defined challenges over time, which would inform that the current plan was not drafted properly, or that its execution has failed to some extent. Ultimately, considering the case of an organisation where Agile UX would have been successfully adopted, and where almost no challenges would be felt in product development anymore, the attention given to sustaining the action plan and ensuring a high Agile UX level would probably be less needed. Nonetheless, as the findings of this thesis highlighted, the future of product development is to a large extent hard to predict, emphasizing the relevance for any organisation to constantly re-evalute its processes and collaboration practices, even if doing very well in at a given point in time.

2. Suggestions for further research

Before concluding on this extensive thesis research on product development workflows, focusing on the design and engineering disciplines and the Agile UX approach, the author details here three suggestions of areas for further research. These potential further researches could be building on the findings of this thesis, to complement and deepen them, or perhaps to validate or invalidate them.

This research being an academic thesis undertook by a Master' student, the author calls for professionally executed researches to validate or invalidate the findings. They include the findings of the background research, the six formulated sources of challenges between design and engineering and how Agile UX responds to these. The eight Agile UX cores defined by the author could be re-visited. The two Agile UX gaps identified at the end of the background research would need to be verified as well, and the author does not exclude other gaps and limitations of the current Agile UX approach to be uncovered. Following this line, the author encourages deeper researches to address the main research question of the thesis "How to contextually apply an Agile UX approach to product development, today, and tomorrow?". The three phases methodology draft proposed here is only a suggestion to be taken further and developed. Action researches putting it into practice and testing it within different types of organisations and teams would be helpful to develop it. Along with the elaboration of the methodology, the author suggests time and attention in creating side support materials. Some ideas of side materials have been suggested here already, such as workshops, mapping canvases, and handbooks. The author is also keen to hearing from any other practical idea or initiative, other than the three phases methodology proposed, to answering the main research question.

The author encourages further researches to be undertook more specifically on the evolution of product development, design and engineering disciplines, and digital products. This would provide with more accurate perspective, and help practitioners and researchers to better anticipate future trends. This aspect was investigated here in this thesis research, but was only grasped.

Finally, the author feels the need for further researches on product development workflows to be done, while taking into account all the other disciplines and perspectives involved, and not solely design and engineering. Other perspectives to include would be the business and strategy, product management, data science, sales and marketing, and customer success, among others. The author also reminds that the design perspective was dominant here, and researches highlighting more the engineering point of view would already be a good complement.

Conclusion

User-experience (UX) design and software engineering are two major disciplines coming into play in digital product development. Their complementarity is essential for creating quality products that are business centric and user-friendly. Nonetheless, conflicts in the processes and collaboration between design and engineering roles are many. The background research of this thesis emphasized this fact by presenting six main sources of challenges. In respond to those challenges, the Agile UX modern approach to product development appears to be an ideal solution. This thesis provides a deep understanding of this approach, and lays dow eight Agile UX cores as a way to synthesizing its values, principles, and key aspects.
The problematization of this thesis research originates from the fact that, althought Agile UX represents a solution to reconciling design and engineering, the current Agile UX frameworks have limitations. Two gaps were framed in the background research, the main one being a lack of attention to contextuality and a lack of instructions to put Agile UX into practice. The secondary gap is the lack of anticipation to the evolution of product development, design, engineering, and digital products in general. The second part of this thesis research, relying on empirical insights from industry practitioners, aimed at adressing these Agile UX gaps. One main objective was the driver; understanding how may an Agile UX approach to product development be contextually applied, today, and tomorrow.

The extensive findings of the empirical research have highlighted the diversity of the industry, and the uniqueness of each product development context. Differing levels of synergy between designers and developers have been depicted. Many contextual facets, at an organisation, people, or project level, impacting on the design-engineering collaboration and product development processes, have also been emphasized. The fluctuation of the state of the workflows within different organisations or teams across the industry appears evident. Some signs of the industry evolving towards an Agile UX approach to product development can be observed, such as initiatives to standardising processes or a rising design maturity, but this evolution is very unequal. Some organisations still represent challenging environments for any design and engineering reconciliation. More practically, there are multiple divergent points in the ways processes and collaboration practices are put into place, some major examples being the room for flexibility, the hierarchical model, the level of Agile commitment, or the level of structure in design and engineering operations. On the contrary, some general correlations are also observed, such as the process phases schema, key practices, or the fact that design overall is given less attention than engineering. Looking at the evolution of product development, design and engineering disciplines, and digital products in general, Agile UX seems to align with future trends and to assert its relevance. Additionally, many opportunities for better processes and collaboration between design and engineering can be foreseen. Nonetheless, a big part of unknown and unpredictability with the evolution of product development opens up discussions and questions, and triggers the need for practitioners and researchers to continuously let their eyes open.

Based on the findings, and as an attempt to providing a practical contribution, the author proposes a three phases methodology draft to adopting Agile UX. In a nutshell, this methodology aims at supporting practitioners in understanding and evaluating their context, formulating their tailored Agile UX framework, and sustaining it over time. This draft may be considered as a first step towards a more concrete solution for teams and organisations to reconciling their design and engineering ways of working, and ultimately optimising their product development workflows.

Glossary and acronyms

The following list aims at clarifying the meaning of the key words and core terms used in this research. Each explanation is formulated with the author's words, based on the literature and Web.

Agile user-experience design (Agile UX): Modern approach to product development aiming at reconciling UX design and software engineering, focusing on unifying UX processes and practices with Agile software development methodologies.

User-experience design (UX) (also called product design or UI/UX design): Design discipline of the field of human-computer interaction (HCI), focusing on the conceptualization and design of digital products.

Software engineering (SE) & Software development (SD): These terms are used similarly in this thesis, and often are mixed. Nonetheless, it worth precising that SD is a sub-discipline of SE, and focuses on building, releasing, and maintaining softwares. SE has a broader perspective on engineering technologies, and includes additional areas such at analytics, data structures, system management, and algorithms, among others.

Agile software development (ASD, Agile): Model and philosophy, dominant standard in software development, aiming at building applications in a structured and efficient way. Agile defines more practical frameworks, some major ones being Scrum, Kanban, and Extreme programming (XP).

Product development: All the steps, including all processes and practices, embedding different disciplines, bringing new products to market, sustaining, and renewing them. In this thesis, product development is reffered to as digital product development.

Digital product: Referred here as any software product, from mobile to web application, platform to website.

Project: Referred here as the highest level of sub-division in internal work, varying in scope, ecompassing given objectives (e.g. improving a feature, creating new product from scratch ...).

Workflow: Referred here as mix of processes and collaboration practices in a given project.

Process: Referred here as all the tasks, practices, and activities, successively undertook in a given timeline.

Collaboration: Referred here as all types of interactions between people in a given project.

Philosophy & mindset: Set of high-level values and principles, guiding practioners in a given discipline or sets of disciplines (e.g. Human-centered design, Agile).

Model or approach: High level guidelines, reflecting a given philosophy & mindset, informing on ways of working (e.g. Agile UX, Design Thinking).

Framework: More practical and detailed than a model or approach, a framework supports processes with a given structure and practices, including certain methods or activities to run. Some detailed frameworks also have clear roles definition, with clear sets of tasks (e.g. Dual-track, InnoDev, Scrum, Kanban...).

Literature references

Alsaqqa, Samar & Sawalha, Samer & Abdel-Nabi, Hiba. (2020). Agile Software Development: Methodologies and Trends. International Journal of Interactive Mobile Technologies (iJIM).

Beck K et al (2001) Agile alliance. Principles behind the Agile Manifesto. Available at: http:// agilemanifesto.org/principles.html

Kashfi, Pariya & Feldt, Robert & Nilsson, Agneta & Berntsson Svensson, Richard. (2014). Models for Integrating UX into Software Engineering Practice: an Industrial Validation.

WHAT IS HUMAN-CENTERED DESIGN? 2015 [Online]. Available: Designkit.org

ISO25010, "Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems," Geneva, Swiss, 2019. [Online]. Available: https://www.iso.org/standard/77520.html

Dina Salah, Richard F. Paige, and Paul Cairns. 2014. A systematic literature review for agile development processes and user centred design integration. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14). Association for Computing Machinery, New York, NY, USA, Article 5, 1–10. DOI:https://doi.org/10.1145/2601248.2601276

Tiago Silva Da Silva, Milene Selbach Silveira, Frank Maurer, Fábio Fagundes Silveira, The evolution of agile UXD, Information and Software Technology, Volume 102, 2018, Pages 1-5, ISSN 0950-5849, https://doi.org/10.1016/j.infsof.2018.04.008.

Göransson B., Lif M., Gulliksen J. (2003) Usability Design-Extending Rational Unified Process with a New Discipline. In: Jorge J.A., Jardim Nunes N., Falcão e Cunha J. (eds) Interactive Systems. Design, Specification, and Verification. DSV-IS 2003. Lecture Notes in Computer Science, vol 2844. Springer, Berlin, Heidelberg. https://doi-org.libproxy.aalto.fi/10.1007/978-3-540-39929-2_22

Hartson, R., & Pyla, P. S. (2019). The UX book: Agile UX design for a quality user experience.

Sedano, Todd & Ralph, Paul & Péraire, Cécile. (2020). Dual-Track Development. IEEE Software. 10.1109/MS.2020.3013274.

Schwartz, Lou. (2013). Agile-User Experience Design: an Agile and User-Centered Process?.

Anders Bruun, Marta Kristin Larusdottir, Lene Nielsen, Peter Axel Nielsen, and John Stouby Persson. 2018. The role of UX professionals in agile development: a case study from industry. In Proceedings of the 10th Nordic Conference on Human-Computer Interaction (NordiCHI '18). Association for Computing Machinery, New York, NY, USA, 352–363. DOI:https://doi.org/10.1145/3240167.3240213

Navarro, P.L.M., Pérez, G.M. & Ruiz, D.S. A Script-Based Prototyping Framework to Boost Agile-UX Developments. J. Comput. Sci. Technol. 31, 1246–1261 (2016). https://doi-org.libproxy.aalto.fi/10.1007/s11390-016-1695-6

Kuusinen K. (2015) Task Allocation Between UX Specialists and Developers in Agile Software Development Projects. In: Abascal J., Barbosa S., Fetter M., Gross T., Palanque P., Winckler M. (eds) Human-Computer Interaction – INTERACT 2015. INTERACT 2015. Lecture Notes in Computer Science, vol 9298. Springer, Cham. https://doi-org.libproxy.aalto.fi/10.1007/978-3-319-22698-9_3

Khaled Al Ghanmi, N., & Shahida Mohd Jamail, N. (2020). Integrating Scrum development process with UX design flow. Bulletin of Electrical Engineering and Informatics, 9(6), 2630-2636.

Dobrigkeit F., de Paula D., Uflacker M. (2019) InnoDev: A Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup. In: Meinel C., Leifer L. (eds) Design Thinking Research. Understanding Innovation. Springer, Cham.

Braz R..S., Merlin J.R., Freitas Guilhermino Trindade D., Eduardo Ribeiro C., Sgarbi E.M., Junior F..S. (2019) Design Thinking and Scrum in Software Requirements Elicitation: A Case Study. In: Marcus A., Wang W. (eds) Design, User Experience, and Usability. Design Philosophy and Theory. HCII 2019. Lecture Notes in Computer Science, vol 11583. Springer, Cham.

Lassi A. Liikkanen, Harri Kilpiö, Lauri Svan, and Miko Hiltunen. 2014. Lean UX: the next generation of user-centered agile development? In Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational (NordiCHI '14). Association for Computing Machinery, New York, NY, USA, 1095–1100. DOI:https://doi.org/10.1145/2639189.2670285

Patterson, Ella & Erturk, Emre. (2015). An Inquiry into Agile and Innovative User Experience (UX) Design.

Kuusinen K. (2016) BoB: A Framework for Organizing Within-Iteration UX Work in Agile Development. In: Cockton G., Lárusdóttir M., Gregory P., Cajander Å. (eds) Integrating User-Centred Design in Agile Development. Human–Computer Interaction Series. Springer, Cham. https://doi-org.libproxy.aalto.fi/10.1007/978-3-319-32165-3_9

Alisa Ananjeva, John Stouby Persson, Anders Bruun, Integrating UX work with agile development through user stories: An action research study in a small software company, Journal of Systems and Software, Volume 170, 2020, 110785, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2020.110785.

Julio Cesar Pereira, Rosaria de F.S.M. Russo, Design Thinking Integrated in Agile Software Development: A Systematic Literature Review, Procedia Computer Science, Volume 138, 2018, Pages 775-782, ISSN 1877-0509.

Alexander Jones & Volker Thoma (2019) Determinants for Successful Agile Collaboration between UX Designers and Software Developers in a Complex Organisation, International Journal of Human–Computer Interaction.

Ferreira J. (2012) Agile Development and UX Design: Towards Understanding Work Cultures to Support Integration. In: Bajec M., Eder J. (eds) Advanced Information Systems Engineering Workshops. CAiSE 2012. Lecture Notes in Business Information Processing, vol 112. Springer, Berlin, Heidelberg. https://doi-org.libproxy.aalto.fi/10.1007/978-3-642-31069-0_51

Manuel Brhel, Hendrik Meth, Alexander Maedche, Karl Werder, Exploring principles of user-centered agile software development: A literature review, Information and Software Technology, Volume 61, 2015, Pages 163-181, ISSN 0950-5849, https://doi.org/10.1016/j.infsof.2015.01.004.

Gothelf, J., & Seiden, J. (2016). Lean UX: Designing great products with Agile teams.

I. Signoretti, S. Marczak, L. Salerno, A. d. Lara and R. Bastos, «Boosting Agile by Using User-Centered Design and Lean Startup: A Case Study of the Adoption of the Combined Approach in Software Development,» 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Porto de Galinhas, Recife, Brazil, 2019, pp. 1-6, doi: 10.1109/ESEM.2019.8870154.

Emma J. Rose, Cynthia Putnam, and Craig M. MacDonald. 2020. Preparing Future UX Professionals: Human Skills, Technical Skills, and Dispositions. In Proceedings of the 38th ACM International Conference on Design of Com- munication (SIGDOC '20), October 03, 04, 2020, Denton, TX, USA. ACM, New York, NY, USA

Web references

Blog posts/Web articles

[1] Ben Ralph “Agile & Lean UX“ medium.com 2017

[2] Page Laubheimer “Agile Is not Easy for UX: (How to) Deal with It“ nngroup.com 2017

[3] Kate Kaplan “DesignOps Maturity: Low in Most Organizations“ nngroup.com 2020

[4] Kate Kaplan “DesignOps 101“ nngroup.com 2019

[5] Rachel Krause “Collaborative Agile Activities Reduce Silos and Align Perspectives“ nngroup.com 2018

[6] Nick Babich “The Designer’s Guide to Lean and Agile UX“ xd.adobe.com 2018

[7] Marcin Treder “Lean UX vs. Agile UX — is there a difference?“ uxpin.medium.com 2016

[8] Kate Kaplan “The State of Design Teams: Structure, Alignment and Impact“ nngroup.com 2021

[9] Kaja Toczyska “How UX design fits into Agile methodologies“ withintent.com 2020

[10] Janet M. Six “User Experience and Agile“ uxmatters.com 2017

[11] Interaction Design Foundation “Design Principles“ interaction-design.org

[12] Interaction Design Foundation “Agile Development“ interaction-design.org

[13] Interaction Design Foundation “A Simple Introduction to Lean UX“ interaction-design.org 2021

[14] Erkan Öğümsöğütlü “How do UX practices fit with agile development practices?“ uxdesign.cc 2018

[15] Emily Grace Adiseshiah “Choosing a UX Strategy that suits you and your users: Agile Vs Lean UX“ justinmind.com 2016

[16] Deepak Arasu “Choosing the Right UX Process for Your Software-Development Model“ uxmatters.com 2019

[17] Debbie Levitt “Agile UX: How to Incorporate UX and Product Design into Agile“ toptal.com 2019

[18] Danny Bluestone “How to Combine User-Centered Design and Agile Development“ uxmatters.com 2015

[19] Andrew Turner “Make Your UX Design Process Agile Using Google’s Methodology“ interaction-design.org 2020

[20] Anders Ramsay “Agile UX vs Lean UX – How they’re different and why it matters for UX designers“ coderchronicles.org 2012

[21] McKinsey & Company “The Business Value of Design“ mckinsey.com 2018

[22] https://thedesignsquiggle.com/

Podcasts

[23] Design Sytem Podcast “Embracingintentionality and creating trust“ Spotify 2021

[24] Tech Done Right “Diverse agile teams“ Spotify 2018

[25] What is wrong with UX “Advocating for UX“ Spotify 2019

[26] Conversations with Hans van Dam “The hurdle that conversation designers and developers face: client team alignement and how a conversational bot can fulfil their needs“ Spotify 2021

[27] Design Details “The designer-developer handoff“ Spotify 2020

[28] Design Huddle “What is a full stack designers and ways to become a better UXer“ Spotify 2019

[29] UI breakfast “Lean, agile and design thinking“ Spotify 2019

[30] UI Breakfast “Optimizing processes in a UX team“ Spotify 2018

[31] Honnest UX talks “UX design processes“ Spotify 2021

[32] User Defenders “Should developers design?“ Spotify 2017

[33] UI narrative “Designers working with software engineers“ Spotify 2020

[34] Design Systems Podcast “Building shared culture and better collaboration between designers and developers“ Spotify 2020

[35] Village Global’s Venture Stories “Reimagining collaboration for designers and developers“ Spotify 2020

[36] UI breakfast “Bridging the gap between designers and developers“ Spotify 2016

[37] Tech Done Right “Agile UX product design“ Spotify 2017

[38] Delta CX Live Streamed Podcasts “UX and Agile Software Development“ Spotify 2020

[39] User defenders “The lines between designers and developers are starting to blur“ Spotify 2015

[40] Developer tea “3 guidelines for happier work with designers“ Spotify 2020

[41] Design Better Podcast “How designers and engineers unite the possible with the probable“ Spotify 2020

[42] Design Life “Tackling implementation issues between design and development“ Spotify 2020

[43] Bloc Thinking “UX and Agile: Finding the right balance“ Spotify 2018

[44] UI narrative, ep 18 “How UX designers work well with agile methods“ Spotify 2020

[45] UX Podcast #224 “The business value of design“ with Jeanne Liedtka, Spotify 2019

Youtube videos

[46] Google Developers “Designer & developer communication - Google I/O 2016“ Youtube 2016

[47] React Conferences by GitNation “A Common Design Language: Let designers and developers talk to each other - Andrey Okonetchnikov“ Youtube 2019

[48] Career Foundry “The UX Design Process (Explained By Experts)“ Youtube 2018

[49] Flux “Designers & Developers Workflows“ Youtube 2019

[50] Mike Locke “Companies and the Product People Are Still Confused About UX Design“ Youtube 2019

[51] Malewicz “Designers vs developers - how to work better together “ Youtube 2020

[52] NNgroup “10 UX Challenges for the Next 25 Years (Jakob Nielsen Keynote)“ Youtube 2018

[53] ChariotSolutions “15 Minutes With: Tom Purcell on Designers and Developers“ Youtube 2020

[54] NNgroup “Does Agile Destroy UX?“ Youtube 2017

[55] NNgroup “Why is UX so difficult?“ Youtube 2018

[56] NNgroup “Design Thinking is Like Cooking“ Youtube 2019

[57] NNgroup “Design Principles 101“ Youtube 2020

[58] vaexperience “Bridging the Gap Between Design Thinking, UX and Agile - UX Chat“ Youtube 2020

[59] Ben Taylor “UX Tutorials | UX and Agile Dev Process“ Youtube 2017

[60] AJ&Smart “WHAT’S THE DIFFERENCE BETWEEN AGILE SPRINTS & DESIGN SPRINTS? - Aj&Smart“ Youtube 2017

[61] Jesse Showalter “Full Stack Designer... What’s that?“ Youtube 2018

[62] blondiebytes “What Does a Software Developer Do?“ Youtube 2020

[63] NN Group “Attributes of Effective Agile UX“ Youtube 2018

[64] NN Group “Clarifying the UX Role for Your Team“ Youtube 2018

[65] Codeacademy “Agile software Development Methodology: What is Agile?“ Youtube 2019

[66] Jesse Showalter “Working with Developers (5 Tips for Designers) | Design workflow tutorial “ Youtube 2017

[67] Design Gal “How to Work With Developers as a Designer“ Youtube 2018

[68] Delta CX “Should UX completely redefine our design process to fit better into Agile?“ Youtube 2020

[69] NN group “Building Strong Designer-Developer Relationships“ Youtube 2017

[70] Google chrome developers “Creating a Collaborative Environment -- Designer vs. Developer #1“ Youtube 2017

[71] Designlab “Meet the Team How Designers and Developers Work Together“ Youtube 2020

[72] Mike Locke “Agile Scrum Development Process and How UI/UX Design Fit In“ Youtube 2017

[73] NNGroup “How UX Professionals Define Design Thinking in Practice“ Youtube 2019

[74] NN group “How Can UX Professionals Balance a Range of Skills as They Build Their Careers“ Youtube 2020

[75] NN group “Design Thinking & Agile“ Youtube 2019

Other references

Marsh (2018). User Research: A practical guide to designing better products and services.

Appendix

Guide and questions for the semi-structured interviews

Overall objective: Understanding how to contextually apply an Agile UX approach to product development, today, and tomorrow.

Welcoming:
Hi! Thank you for accepting to share insights!
Introducing myself
Telling about the purpose of the interview and research context, topic and objectives
Have you heard of Agile UX? (quickly discuss around it, explain if needed)
Explaining the use of data, will be anonymized as well
Asking for recording
It's gonna take about 40 min
Explain the structure
Any question before we start?

Intro questions:
Could you briefly introduce yourself and your background?
Could you briefly describe your organisation and role?

Objective 1: Understanding the state of Agile UX (the state of the relationship between UX design and SE) in the industry.
Remember the participant to refer to one organisation only

1) Can you describe a bit your company's work environment, settings and culture? Structure? How roles/ teams are organised? What is your view on it?

2) What type(s) of project are you working on?

3) What does successful project mean to you? In terms of results? In terms of process and collaboration?

4) How do you collaborate and what process(es) are you employing? What model? What nature? What phases? What activities? What tools? What communication? Decision-making? What timeline? Facilitation?

5) Reflecting on that, how well is your organisation embeding UX and Agile (or other SE model) together? Can you describe how it is done? Design maturity? What culture? What mindset?

6) Can you describe the relationship you and other UX designers have with software developers? What do you think of this relationship?

7) What is the space for informality and casual times within the organisation?

Objective 2: Understanding how processes and collaboration models are being defined in contextualized projects.

8) Once there is a new project (or feature), can you describe how things are organised? What rituals? How is the team formed? What are these first steps? Does it differ from one project to another?

9) How do you design the process and define the collaboration model for each project? Is it different from one project to another? What is your view on it? What is the method/framework? Do you put it visually somewhere?

10) From a general perspective, what are the contextualized aspects of a project that might influence the way of working and collaborating? E.g. B2B vs. B2C, Consultant vs. InHouse.

11) How carefully the processes and collaboration models are being followed along a project? What room for flexibility? What might be redefined later?

12) Personally, what do you wish was different/better in terms of processes and collaboration? How could the company improve the ways of working?

Objective 3: Reflecting on how software project processes and collaboration models will be affected by the evolution of the UX an SE disciplines, as well as the technologies.

13) How do you see digital products evolve in the future? What technologies? What trends? What roles in the society?

14) About UX and SE; how do you see the evolution of these disciplines (tools, activities, methods, knowledge etc...) changing the way digital products are created? What new and future aspects can impact?

15) How do you see yourself working in 10 years? How would you collaborate? What process? What settings?

Wrapping-up:
That's it! Thank you very much!
Anything else you would like to share on this topic?
Explain what's next.
Would you like to receive the final thesis publication?
Thank you again for your time, have a great day!


Thank you!