Martina Wolna

XDD - Experience Driven Design

Impactful product design through optimised development process

Master's thesis in Interaction Design
January 2022

Master's thesis

NTNU
Norwegian University of Science and Technology

NTNU
Norwegian University of
Science and Technology

Martina Wolna

XDD - Experience Driven Design.

Impactful product design through optimised development process

Master in Interaction Design

Gjøvik, July 2022

Supervisor: Frode Volden

Norwegian University of Science and Technology
Faculty of Architecture and Design
Department of Design

NTNU
Norwegian University of
Science and Technology

Abstract

The predominance of developers in the IT industry results in the devaluation of design into a mere sub-field of product development. This disparity is also the reason for the unconscious separation from the actual end-user and focusing on creating new features the users do not necessarily need. At the same time, the huge number of digital products available on the market provides users with virtually infinite number of options to choose from. As a result, they pick applications that are functional and convenient to use at the moment, but their needs are constantly evolving. Therefore, there is a need for an optimal process that would include the user-centred research. Explaining the gains of such an approach and comparing it to current product development processes will make this need even more apparent and display the gap between the product team and the actual user. These different perspectives can be successfully combined in an interdisciplinary UX process, which will further help to balance the dynamics of the IT industry.

Abstrakt

Overvekt av utviklere i IT-bransjen resulterer i devaluering av design til et rent underfelt av produktutviklingen. Denne forskjellen er også årsaken til den ubevisste separasjonen av sluttbrukeren i utviklingen, og fokus på å skape nye funksjoner brukerne ikke nødvendigvis trenger. Samtidig gjør det enorme antallet digitale produkter som er tilgjengelige på markedet at brukerne har praktisk talt uendelig mange alternativer å velge mellom. Som et resultat velger de applikasjoner som er funksjonelle og praktiske å bruke for øyeblikket, men behovene deres er i stadig utvikling. Derfor er det behov for en optimal prosess som vil inkludere brukersentrert forskning. Å forklare gevinstene ved en slik tilnærming og sammenligne den med gjeldende produktutviklingsprosesser vil gjøre dette behovet enda tydeligere og vise gapet mellom produktutviklingsteamet og den faktiske brukeren. Disse ulike perspektivene kan med hell kombineres i en tverrfaglig UX prosess, som ytterligere vil bidra til å balansere dynamikken i IT-bransjen.

Table of Contents

1. Introduction

The aim of this dissertation is to propose an optimal working environment that combines theoretical knowledge with practical professional experience in the IT sector, with particular emphasis put on UX Design in product development processes. The outline of the issues related to the cooperation between designers and programmers, supplemented with applied frameworks, contemporary design principles and recommendations of specialists clearly indicate the need for changes in the strategies of software development. The main area for improvement lies in concentrating product design processes around users with utilising the UX research methodology in each phase of the design process, including after implementation.

Originally, design was seen as an element of product development or even as its subcategory. Designers were not in charge of the innovation work itself, their main task being rather to take care of the visual aspect of the idea. Such an approach stimulated market growth in many areas, with increasingly attractive and aesthetically pleasing products being created that connected technology with its users. All that translated into growing consumers’ interest and strengthening of brands. Today, the role of designers in product development processes is becoming much more substantive. Rather than focusing on the visual appeal of the product, the main focus of their work is to respond to the evolving and changing needs of users and – based on research – support and facilitate their experience with the product. The visual dimension is of tactical importance and builds product value only in a limited way. The utility dimension, on the other hand, is of strategic importance and leads to a significant increase in value (Brown, 2008). The utility dimension is a priority. Only when the product fulfils its function can it be supplemented with a visual layer.

Today, more and more digital products are being created, with applications replacing traditional printed forms and services. The transition of design to a digital platform has influenced significant developments in such areas as product usability, user problem solving or even product strategy planning. This has created not only new design fields like Interaction Design (IXD), User Experience (UX) Design or User Interface (UI) Design, but also the need to develop new processes to take a digital product from idea to realisation.

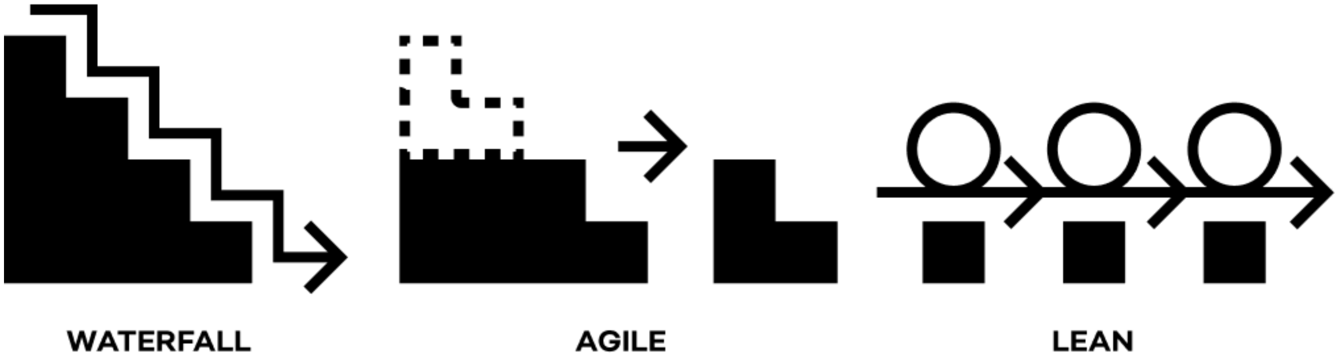
Compared to other fields of economy, the digital technology industry is exceptionally young. However, it is characterised by unrivalled innovation, which is still gaining momentum. If we take the first Apple Mackintosh released in 1984, displayed in the Teknisk Museum in Oslo, as its beginning, the industry is not even 40 years old. During this time, the sector has developed its own distinct procedures and work patterns, which need constant updating. Nowadays, digital technology is ubiquitous, which leads to the oversaturation of the market and high competitiveness of the proposed solutions. For that reason, the processes of implementing new products should be embedded in the multidimensional context of already functioning technologies. This means that the designed implementations should not only be focused on innovative features, but also update previous solutions and directions. Digital product design is increasingly a redesign, as its production starts with an analysis of the previous products the team has produced and the features that can be reused. A new product usually means replacing a current one that has become so obsolete that it is not worth optimising. The design process is therefore a continuous redesign.

The choice of the topic of this paper resulted from the author's several years of professional experience in UX design, the experience that revealed weaknesses within product creation processes. In many cases, the User Experience design team was incorrectly embedded in processes or carried out their tasks too superficially. To improve my mode of work, I looked for answers to my questions at workshops and conferences. An interdisciplinary approach with the use of new design disciplines significantly increases the usable value of the product, as well as the return on investment, as confirmed by the cited industry research. The methods presented are also helpful in reducing the risk associated with launching a new product and increasing team productivity. This thesis aims to summarise that professional and educational experience in order to find an optimal way of working that combines theoretical knowledge with a practical experience of working in a dynamic and chaotic environment. I intend to build a product development process that I believe would work best for design teams and fit the IT sector market, taking into account its dynamism and volatility.

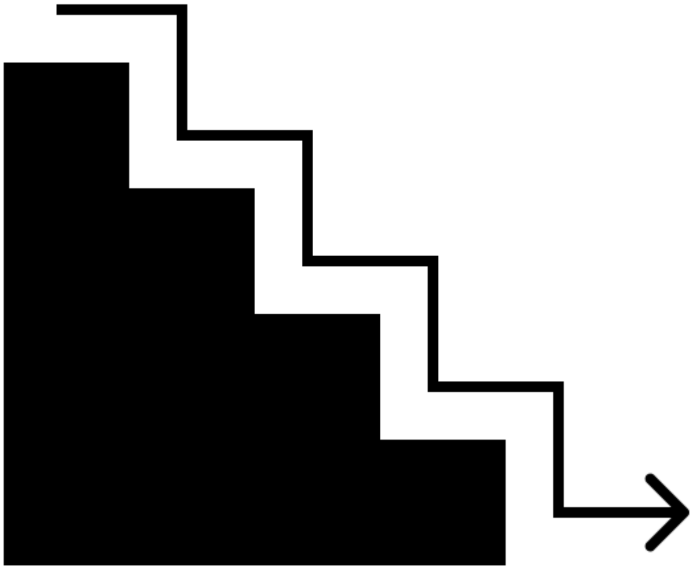
1.2. Product tests case study

When taking on the task of designing new functionality to an existing application, I began the process by understanding how the product works and why the new feature was needed. The product being tested is an application that assists Sales Representatives in doing their job, in short, of visiting shops, inspecting the range and selling the products of the brand they represent. The product turned out to be very complicated, or at least it looked that way from the interface. Having the opportunity to contact the Service Owner, I conducted a series of in-depth interviews to understand not only the application itself, but also the context around it, its users, what their work is about and at what moments they use the product. While visiting the shop, SR would 'start the visit' by clicking a button in the app that started counting time and provided 'activities' for SE to complete in the form of tasks. One of these tasks was to take a picture of the shelves on which the products were placed so that the wholesaler could see whether the space in the shop for which they had paid (shelves in shops are "rented" by wholesalers) was filled with their products according to the Planogram - a visual representation of the arrangement of products on the shelf. The company responsible for the application under test has an extensive database, which it presents to users through a number of coupled products for viewing and organising data. SR performs their tasks in the app because the Sales Manager adds them in another app created for managers so they can see the results of SR's work. My task was to design a solution that would enable the review of photos taken by SR. Meanwhile, during company meetings, I learned about two competing solutions that the company had lost its customers to - app Y and app Z. I asked the SO for the opportunity to interact with the user to conduct Field Studies, which involved accompanying the user in their natural environment and gathering information. In this case, I spent a day at work with the user, visiting sales points, inspecting the stock and taking photos after it had been replenished. It turned out that the user - in addition to the tablet on which he used our app - had also a smartphone on which he used app Y. I asked why he was not using the photo-taking function of our app, visible in the interface. While testing this functionality, we found that it does not work on the user's device, so he was forced to use a competing app to supplement the non-functioning camera feature. Optimising the photo-taking function for new devices and systems would solve the problem and make the researched user's job easier, as he would no longer have to use two applications at once. This revealed a problem in the process of designing a functionality without user research and product testing, without updating it for further improvements and adapting the application to constantly changing technology. This made me realise that the company for which I conducted the tests is not focusing on user problems and user needs, and that their products are being evaluated incorrectly. In addition, competition problems are caused by the rapid obsolescence of the proposed functionalities.

2. Modes of working in the IT sector



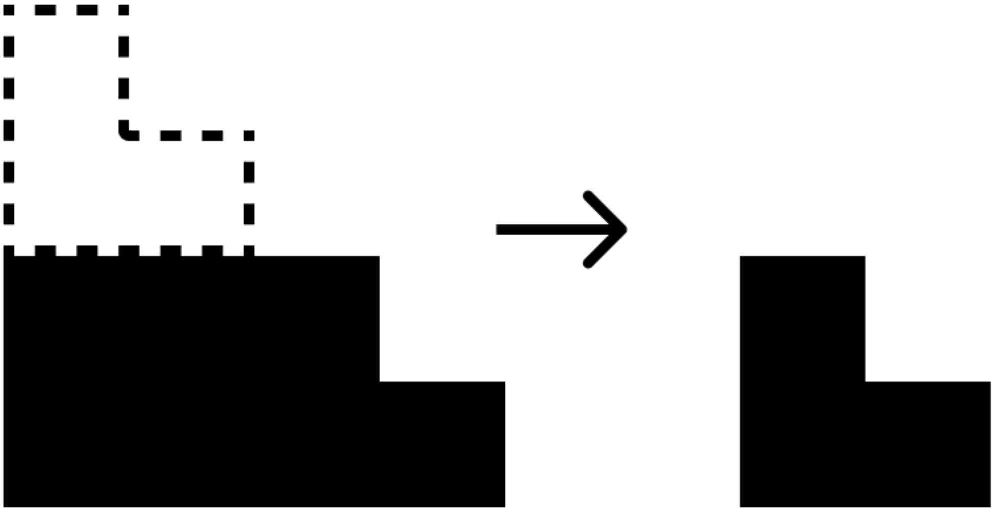
2.1. Waterfall



The waterfall model of creating a product was first defined in 1970 by Winston W. Royce, in his article *Managing the Development of Large Software System* (Royce, 1970). It is based on the initial determination of an overall plan related to what the customer requires to be the end result, which phase is followed by the plan being implemented without breaking it down into smaller parts and without prioritising the mandatory features of the product, called MVP (Minimum Viable Product). In this model, the individual design phases follow one another: planning, analysis, design, implementation, testing, operations. If any phase results in an unsatisfactory outcome, the process returns to its previous element to repeat it.

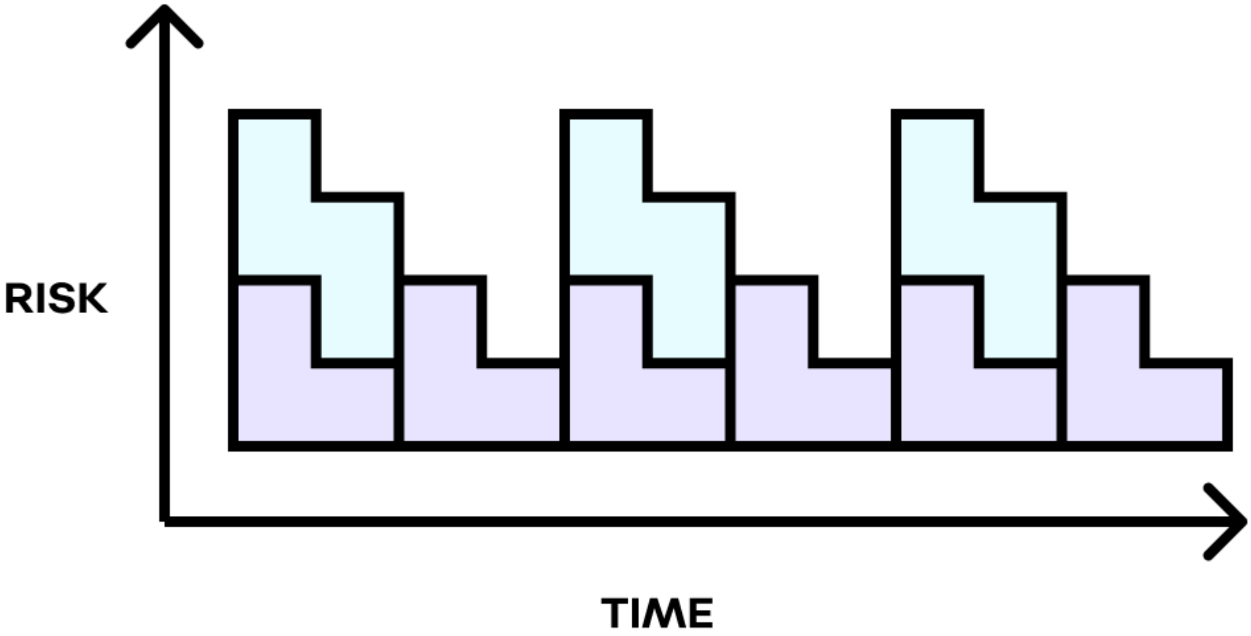
Due to the inflexible division into phases and with testing carried our only at the end of the development cycle, this model has not worked well for product design processes. The design review took place only after the implementation. At that time, another list of changes was planned, according to which the project would be improved. As the tests took place on an already finished product, the optimisation of the detected bugs required further investment. A disproportionately long list of changes may have forced the project team to regress fully and go back to the very beginning of the process. This type of strategy proved not only demotivating for those involved in the process, but also excessively expensive due to the time-consuming iterations.

2.2. Agile



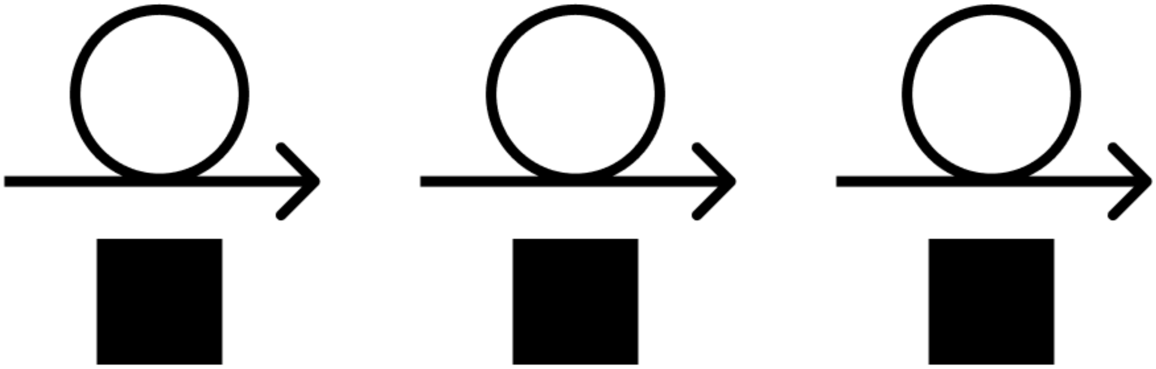
In 2001, the frustration of the 1990s led to the creation of a manifesto authored by a group of several passionate programmers, involved in the development of the industry, whose text enumerated the main desirable principles and values (Asmo, 2018). The manifesto had a huge impact on project development within the IT industry The Agile Manifesto emphasises collaboration between teams, flexibility in the project delivery process, continuous improvement, versioning, testing and high quality results (*Principles behind the Agile Manifesto,* 2001).

Agile was mainly aimed at developers and such phrases as UX research, Human-Center Design and the like do not appear in it at all (Kaley, 2020). Nowadays, the importance of UX designers is increasing and they are starting to be included in the process, but they are still underestimated due to the focus on the biggest team, formed by developers. In the 1990s, there were no people in the IT working cross-disciplinarily, across a few teams, such as the Product Owner responsible for gathering requirements, or the User Experience Designer engaged in research with users, etc. It was then decided that instead of working on a large project in the Waterfall mode, the team would work on its small parts according to a list of priorities, starting with the mandatory ones that the product has to possess if it is to give the user the minimum number of necessary functionalities (MVP). In addition, they had to verify their work at each stage, evaluating the progress, testing the product and implementing the improvements. The new methodology was hugely successful at the start of the second millennium and became extremely popular with developers who began to refuse to work in Waterfall mode in favour of Agile. The main advantage of Agile was to be the reduction of risk with simultaneously being able to meet the required deadline for the product. Programmers were no longer overwhelmed by gigantic documentation and could check the progress of their work in real time, implementing the required corrections. To this day, every IT company claims to work in Agile, even if they actually do not.



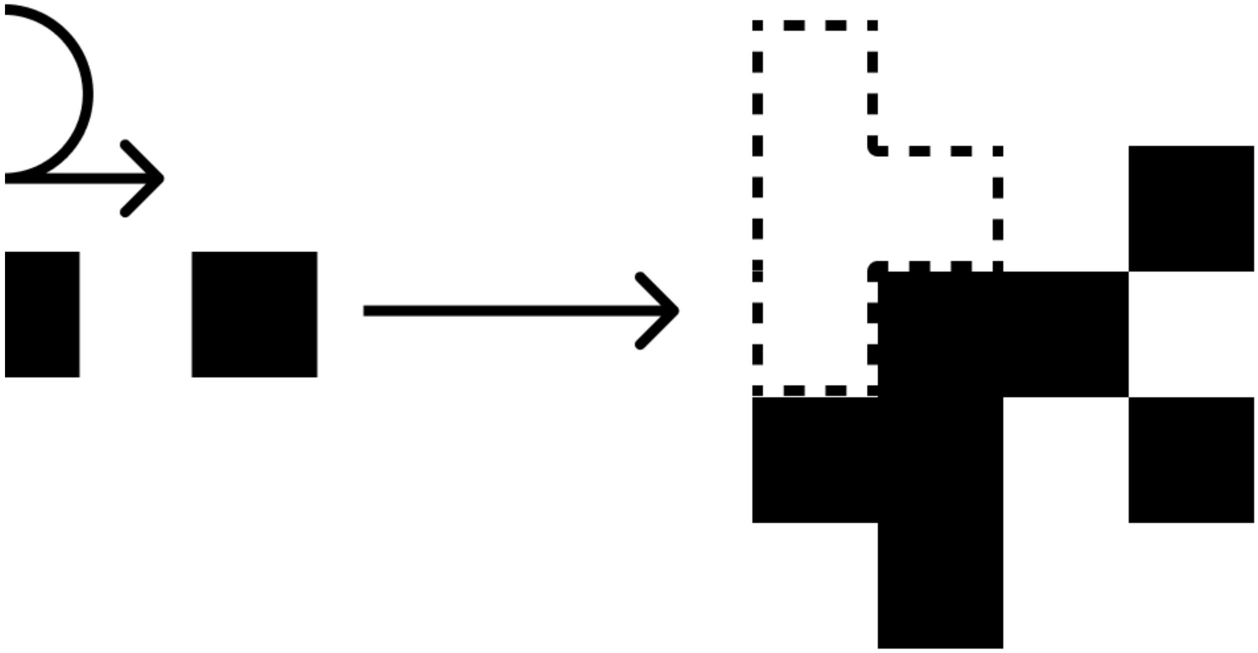
With Agile came the practice of putting designers into Scrums to work in 2-week sprints. During the product design phase there was still not enough time to carry out the required tests, so it was advisable to divide the designers' work. Before this happened, however, attempts were made to combine the processes in the Design Sprint and Rapid Research methods, where time was the main factor and results were visible after a week. Therefore, idea of combining the processes became an issue. Agile is focused on developers and Product Owners. UCD is focused on processes and users. They do not work well together in a single framework, as a large group of developers is able to advance their opinions, which do not have much to do with research data obtained from actual users.

2.3. Lean UX



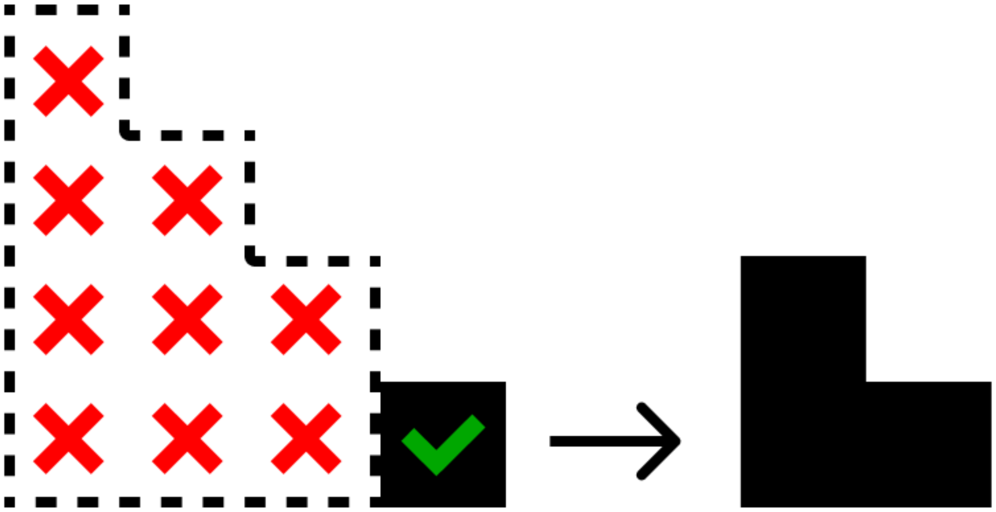
"(...) the success or failure of your product isn't the team's decision-it's the customer's/ They will need to click that >>Buy Now<< button you designed. The sooner you give them a voice, the sooner you'll learn whether you've got an idea that works" (Gothelf and Seiden, 2016).

Lean UX is a style of working focused on creating a product in small portions, tailored according to what is known about the users. It is based on testing the product as quickly as possible, implementing improvements and repeating the process ad infinitum. Each successive test brings new findings that enable the product to be tailored to users. In this way, the team learns from its own mistakes, and puts small portions of the product into production that are easier to improve as there are no complex solutions to be dealt with. Instead, only a small part of the product is implemented, which is also preceded by researching or testing the concept before its development. Concept testing or other research methods that take place without the use of a product - for example, when it has not yet been developed - focus on identifying the problem that users face every day when performing certain activities. Having definitely determined the problem, one can focus on finding the solution. Trials involve creating something that can be tested again involving the user and it does not have to be a digitally encoded product, but for example a paper imitation in the form of wireframes that are discussed during the trial. This allows one to quickly spot errors and verify if the product is being created as expected. Then, when putting the product into use, i.e. coding it, it is necessary to create its MVP version trimmed to the minimum functionality. To do so, you do not build all the features approved in the product specification, but only those most necessary for the product to exist on its own. Unlike Agile, Lean allows you to test a concept, a paper prototype, and not necessarily the coded, final form. It is important to focus more on functionality and User Experience than on its appearance and User Interface. After the next test, more features are added to the priority list.



Lean UX is not only about processes, but also about culture. To adapt Lean means to promote a culture of learning, development and curiosity. This is why there are no celebrities called 'gurus' or 'ninjas' in Lean teams - members focus on sharing knowledge rather than building individual prestige. The group tries to reduce unnecessary and time-consuming activities such as detailed documentation or other activities that do not lead to the main goal and take them out of the process. This type of action strategy aims at replacing the doubt about whether a part of the product is needed at all with the confidence that it will work and fulfil its function.

Unlike Agile, in which a product is broken down into smaller parts delegated to be completed in successive Sprints, Lean does not involve creating a project and dividing it up. Instead, it involves starting with one of the smaller parts, testing and producing it, testing it again, implementing improvements and moving on to the next element after learning from the previous creation phase. Therefore, the process of building a product is dynamic and one does not assume an outcome at the beginning, one acts without expectations.



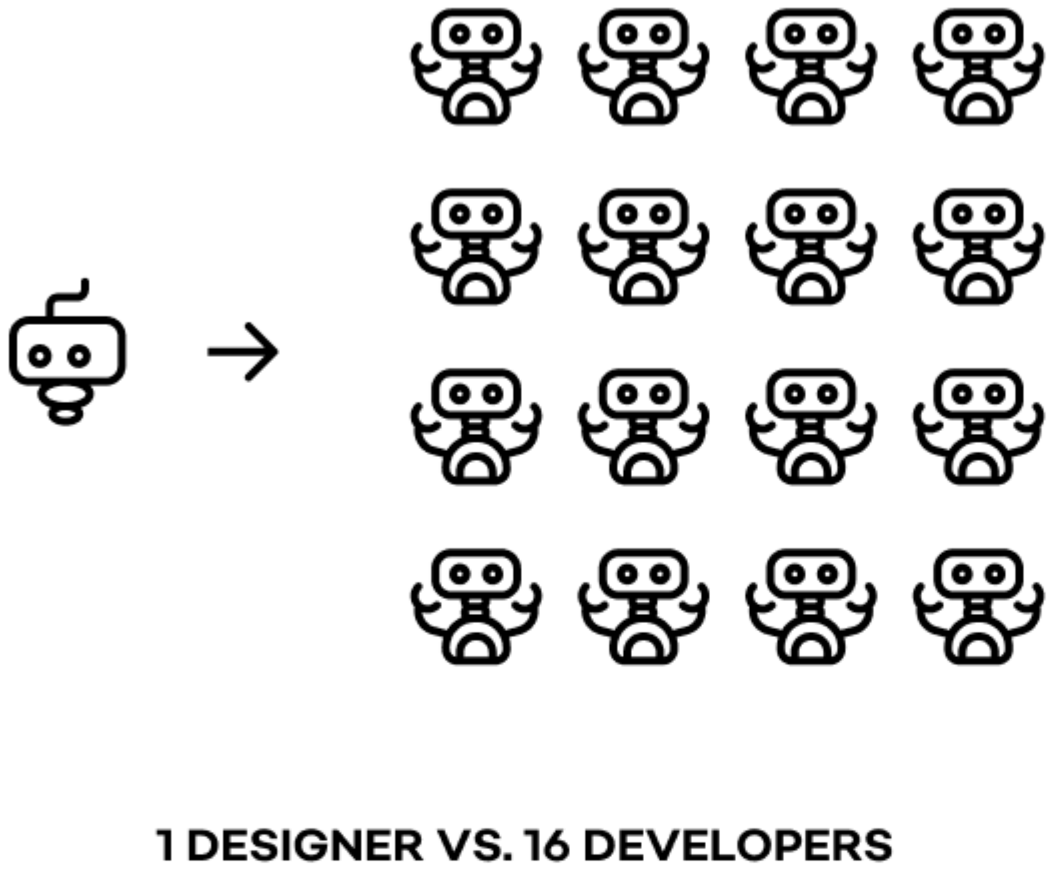
Proposing assumptions for a larger project requires prioritisation of product functionality. *“The biggest lie in software is Phase Two. (...) You set aside features and ideas for the next phase of work and then they are gone – never to be heard from again. As designers, we’ve had hundreds, if not thousands, of wireframes and workflows end up in this same bucket.”* (Gothelf and Seiden, 2016). Those features that are not mandatory for the MVP go into the so-called Backlog, a place for ideas that do not need to be done in the first place. Backlog tends to be overloaded with various ideas that will never be implemented, because during the testing of the first version the tests and research findings suggest the right direction, which does not always coincide with the previous assumptions. Therefore, in Lean UX, iterations are frequent and short in order to quickly determine the direction of development without assuming future product features in advance, but to act on an ongoing basis according to the results achieved.

3. UX Design

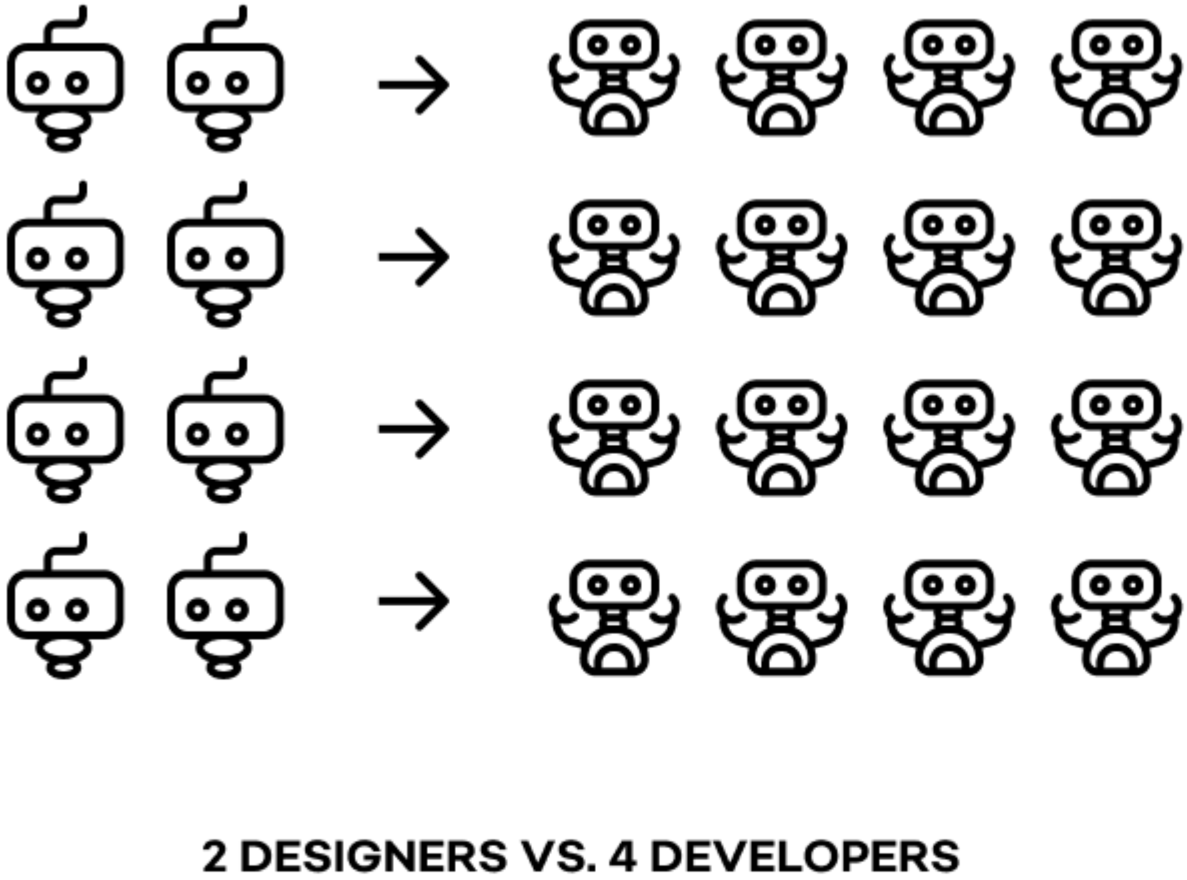
3.1. Design versus development. Outline of the issues

In the 1980s the fields of User Experience and Human-Centered Design, which did not yet use such names, began to develop parallel to the IT framework. The guru and father of UX is generally considered to be Don Norman, who introduced, among other things, the concept of affordances (Norman, 2018). In the 1990s, designers began to emerge, tasked with making sure that the product being designed would reach its target user group and work properly. They designed their products by creating, testing and analysing them in the form of iterations. Design branched out into several individual fields. Affordances were related to non-verbal information about the possibilities to interact with the product. This does not only apply to digital products, Norman himself describes it using the example of a door whose handle should be designed in such a way that the user knows without writing on the door whether to push or pull it. In the same way, in a mobile app, the user should understand that the element in question is a button and is clickable. This section of design is called Interaction Design and is related to whether the user knows how to use the product. Graphic design and interface design is focused on providing graphic components, not necessarily tested with users, in order to implement graphic principles of order and harmony or to give character to a product and make it fit with the brand it represents. In response to research that discovered that the vast majority of users just "scan" a text and only a small minority read it word for word (Nielsen, 1997), the field of UX Writing was born, where the focus is on writing and creating the language needed for today's users who spend less time reading. UX Design consists of many disciplines based on performing research and analysis to understand users. Unfortunately, despite the dynamic development of the IT sector, there is still a clear disproportion between the role of designers and programmers in product development processes. Even today, in a typical IT company, the team of programmers is much larger than that of experience designers. Designers are also disadvantaged by the fact that the effects of their work are primarily visual, and this aspect is extremely subject to external judgments and tastes. Design is therefore often reduced to designing what pleases the HiPPO (Highest Paid Person's Opinion) or the majority, i.e. developers with no design background (Zabawa, 2016). In a typical IT company, the team of developers is much larger than that of experience designers. The usual ratio is 1:16 or even higher (Kaley, 2020). It is extremely difficult for the knowledge of a designer to prevail over the opinions of the stakeholders.

An additional obstacle to the cooperation between designers and developers is the divergence of priorities and the resulting communication problems between teams. They do not work well together in one framework because developers' decisions are not consistent with the results of research on actual users. The issue has been aptly identified by Ben Nadel, who metaphorically presents the handover of a prototype as a beginning of a one-off event that represents the end of the design process and the start of the engineering process (Nadel, 2020). The problem with this way of thinking is that creating a product is not an assembly line process, but a collaboration. When engineers and designers do not have the opportunity to continually present their different skills and perspectives, it puts the product at risk. While designers and engineers may work in different teams with different daily goals, they must not forget about the ultimate beneficiary of the process - the user. Performance is as important to the user as any prototype feature, and engineers need to remember that their insights are just as important as the designers'. Nadel suggests flexibility and refraining from treating the designed interface as an 'absolute truth' that must be reflected in one-to-one code. The design should be a suggestion as to the direction, but it must be discussed with the developers and Product Owner to make sure it is feasible and does not hinder the performance of the product. Perhaps time or technological constraints will require certain simplifications of the interface. The designer should be informed of such limits as early as possible so that he can prepare amendments in advance. At the same time, it is the engineers who do not stick to the design that need to explain the necessary changes to the designers. Inefficient collaboration translates into missed elements, new bugs and unnecessary or impossible features that will affect the final product (Nadel, 2020).

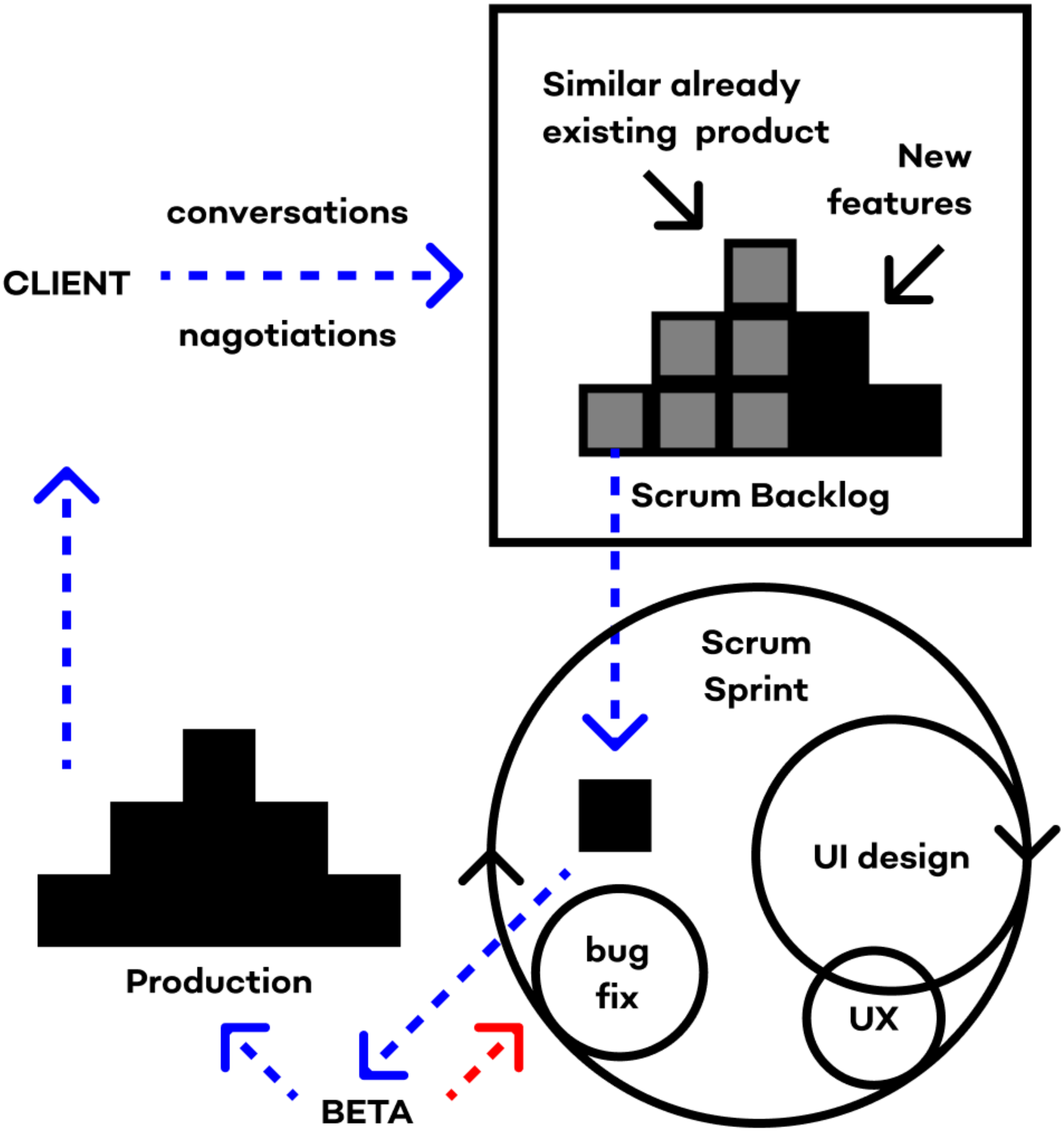


Mike Jones in his article differentiates UX designers into Digital Designers and Service Designers (Jones, 2021). Service Designers need to understand the wider context they offer and create sustainable solutions and optimal experiences for both customers in their unique contexts and service providers. They divide services into sections and tailor refined solutions to hand them over to UX designers, focused on specific users in specific contexts. Jones also recommends multi-person teams with a minimum of 2 designers for a maximum of 4 developers. This is to ensure a broader scope in terms of strategy and the ability to focus on the research process, discovering and defining the problem and confirming the hypothesis rather than proposing immediate graphic solutions, which are often the only requirement for a designer. Thanks to such proportions, designers become more important, they are full members of the product team and their tasks are not confined to the strictly defined design. At the same time, he notes that every designer has a specialisation in which he feels comfortable and, despite the need for general knowledge in design and technological fields, there are areas in which he excels. That is why he does not leave the designers alone in his teams, but brings them together in sparring-pairs. For example, - service designers are more skilled at the Discovery part, namely – using Double Diamond terminology – at holistic perspective, user paths, information architecture, insight into different types of users, integration with business logic and making sure the team is working on the right problem (*What is the framework for innovation? Design Council's evolved Double Diamond,* 2019). The Digital Designer focuses on a particular aspect of the service and its user and designs and tests a suitable interface to finally hand it over to development. According to Jones, both roles represent the interests of the user, which is why the tasks of the Service Design and Digital Design are included in those of UX. Service Design, however, seems more relevant through its holistic approach to the service being designed, where different users appear along the path and the UX designer, by designing specific solutions, represents only one of type of user. That's why several UX designers can draw separate paths for different types of users, which will all contribute to the overall service. For example, sending a parcel in the context of a service includes both the courier and the process of placing the parcels on the conveyor belt, scanning them and distributing them to the different business location zones. Consumers of this type of service can also be divided into several categories: the sender, the orderer and the physical recipient of the delivery. This can be illustrated and described by the Service Designer, and the drawing of specific user paths should be carried out by individual UX designers.

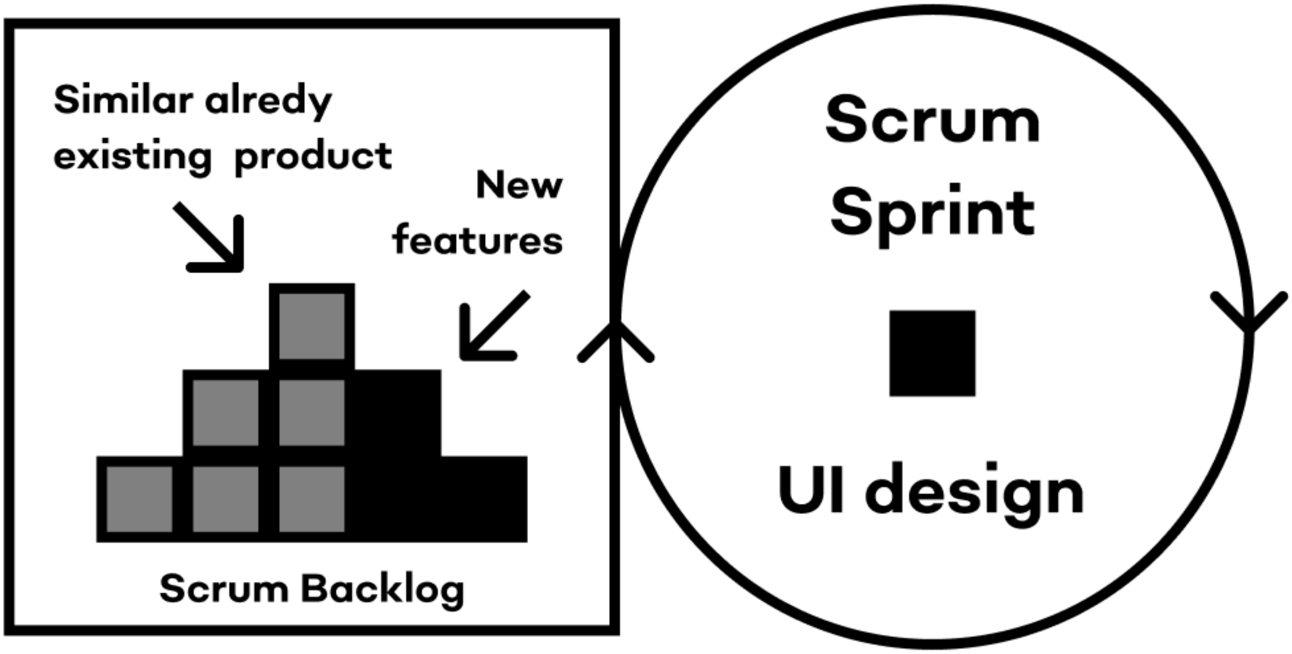


In an ideally balanced team, one UX designer would be employed for one aspect of the job. The whole service should be supervised by a Service Designer. UX researcher would do the research and write reports, UX designer would prepare wireframes, information architecture and measure the layout of individual pages and subpages of the project and prepare connection paths. While Interaction Designer would check if the user performs the designed actions according to the design assumptions, test the clickability of the elements and the accessibility of the interface, measure the contrast and size of fonts and the understanding of icons, UI designer would work on the basis of approved wireframes and refine every detail, design icons and choose colours consistent with the brand image, etc. Unfortunately, with such a detailed division of responsibilities, additional designers would have to be employed.

3.2. The current product development process

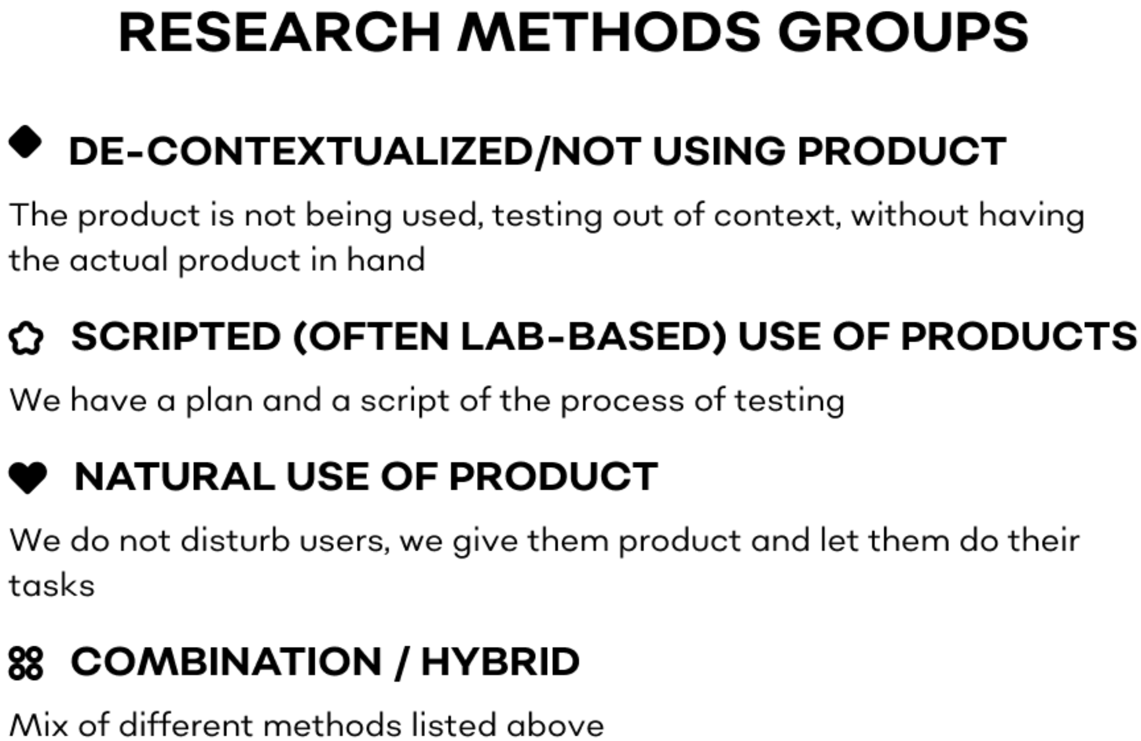


Today, the process of designing a new product starts with the Business Developer, who works on the contract with the customer. This process usually involves the CEO or an employee responsible for sales. They talk to the customer and build a contract that will specify the product deadline and guidelines. Contracts are usually for large sums. Because of the significant profit to the company, the design process prioritises the needs of the client above all else, neglecting the future users. The difference between customer and user lies in the fact that the customer will not necessarily use the final product. For example, if a customer orders the design of a dog toy, the customer will be a human, but the user will be a dog. In this case, methods would need to be found to carry out research covering the needs of the dog, its owner, perhaps also tenants sharing the home space, etc. Focusing solely on the customer and a specific deadline makes the team focused on quick solutions. Given the highly competitive and saturated market, it must be assumed that similar products are already on the market. In addition, it is usual for companies to produce similar products, drawing on previous developments and supplementing them with possible additional functionality. This type of replication does not solve potential problems for new users. Previous solutions are not confronted with innovative methods and technologies. Also omitted is the process of reliable research on users whose needs develop or change over time.



When the replication plan with the new functionality is ready, it is implemented via Scrum into production. Design in this scheme is treated as a subcategory of product development, i.e. graphic solutions for the appearance of UI elements and the provision of mockups. Design activities are carried out in the Just In Time framework, which means a mode of working on an ongoing basis according to the needs of the developers (Kaley, 2020). In this mode, there is a lack of time to think through a course of action and the word 'design' is often replaced by the acronym 'UX' misunderstood as the delivery of digital components. User research does not fit into a time-limited process. The product is made in BETA version, which is tested by QA testers (programmers who do not have the users' needs in mind, but check whether the product works from the technical side). Detected bugs go into the Backlog, and if the product works, it is deployed to production. After a sufficient number of such Scrum iterations, the product is ready and presented to the customer. Increasingly, the room for UX is appearing in Sprint although it is confused with interface or graphic design. Sprint is time-limited and gives no space to do solid research with users.

3.3. Methodology of user research



RESEARCH METHODS GROUPS

◆ **DE-CONTEXTUALIZED/NOT USING PRODUCT**

The product is not being used, testing out of context, without having the actual product in hand

✿ **SCRIPTED (OFTEN LAB-BASED) USE OF PRODUCTS**

We have a plan and a script of the process of testing

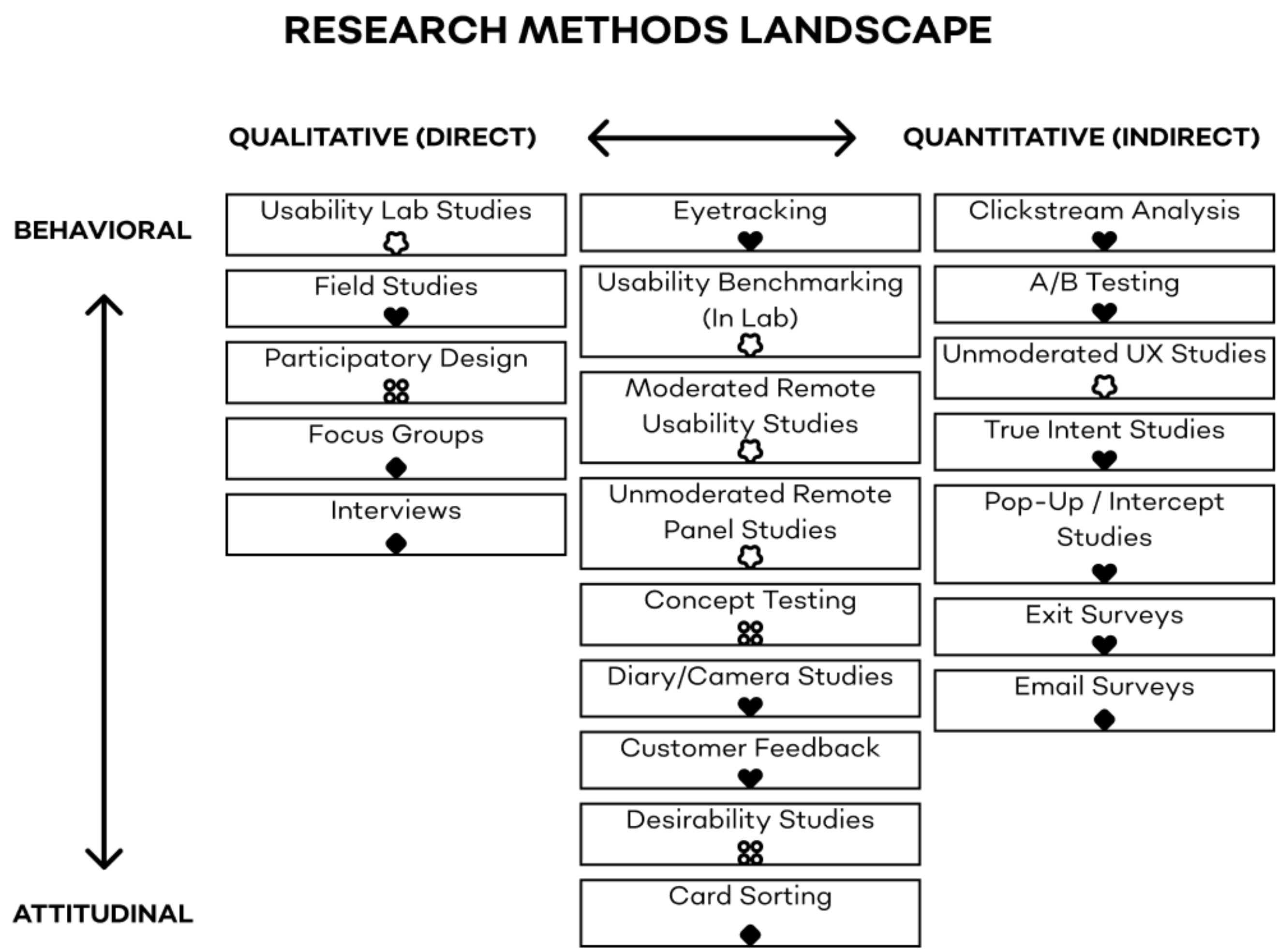
♥ **NATURAL USE OF PRODUCT**

We do not disturb users, we give them product and let them do their tasks

⌘ **COMBINATION / HYBRID**

Mix of different methods listed above

Christian Rohrer delineating the landscape of research methods divided them into four groups (Rohrer, 2021). The first is 'De-contextualised/not using product' - they are methods during which the product is not directly used, with the activities taking place outside its context. Such an approach is useful before creating a product and to test whether the plans to create it are sound. It includes Focus Groups, a method by which we can explore the general opinion of a brand or existing brands with similar solutions, for example we can ask how the brand as a whole is interpreted by customers, whether they use already existing products with similar applications, what shortcomings they have noticed and what the still non-existing product could be better at. Often, work on a product starts with a Competitor Analysis, i.e. a comparison of existing products in order to determine what the new one should differ in. Unfortunately, this study is interpreted as an attempt to make a cheaper copy of a product with similar functionalities and a stable position on the market. Such issues as recognition of challenges and problems or understanding the user are then overlooked, and methods are sought to copy another existing product. This approach is certainly wrong – by copying someone else's solutions we solve someone else's problems and leave out our own users. Moreover, the result is a copy, not an innovative product. Another method from this group is Card Sorting. It involves the user intuitively sorting cards with phrases or words, indicating how menus can be sorted, what categories and subcategories can be created to enable the user to find what he or she is looking for in the application. The last methods in this group are interviews and surveys, which can be used to study users of an already existing product or before the start of its production, when the team needs guidance on how it should be built. The Business Developer's conversations and negotiations with the customer during the initial product planning phase, could be performed by a UX researcher to identify customer needs through interviews, which can be conducted remotely or in person. It is recommended that interviews include open-ended questions so that the researcher can elaborate on them if the answers are unclear. Surveys, on the other hand, should contain closed questions in which the respondent has the option of choosing one of predefined answers. Care must therefore be taken to ensure that they cover various possibilities. The key is to find the user's need/problem and use it as guidance in subsequent research methods.



The remaining groups of tests are related to the direct use of the product. The "Natural use of product" is a type of research on an existing product with the aim to verify its usefulness. Some of the methods in this group can also be used without a product. They include Diary/Camera Studies, which involve the user writing a diary documenting their daily routine in a specific context, for example during a bus journey, to gain data on possible improvements to the ticketing service. It is then necessary to recognise the patterns with which the person being tested moves when performing this action, to identify the processes that are problematic and to propose facilitations. This study can also be carried out on an existing product and examine its daily usability to detect problems with interactions. In the past, the survey consisted of writing down a diary, but today users are more likely to document their daily life with photos and video - as they do on social networks - which is why the camera is mentioned in the name. Other methods focusing on an existing product include Eyetracking, which involves placing a device on the user's head to track their eye movements. Thanks to this research, we know, for example, that users are more likely to 'scan' a text than read it (Moran, 2020), and that they start on the top left. Any UI element placed within the text knocks the user out of their reading rhythm and makes them start ‘scanning' the text again. Another research method belonging to this group is A/B testing, which consists of preparing an A version and a B version of the product in the form of a prototype and testing both versions on the user. Another method is the collection of user feedback, which can take place in different versions, including: Clickstream Analysis, Customer Feedback, True Intent Studies, Pop-up/Incept Studies or Exit Surveys.However, the method that gives the most extensive understanding of users and their needs is Field Studies. Rohrer calls it Ethnographic Field Studies, referring to the discipline that deals with the description and analysis of human cultures, their fields and their products in historical contexts, keeping in mind the habits of users, their mental models, rituals and customs that define the way they use a product. The study takes place in the user's natural environment, where his or her actions are subjected to detailed observation without an imposed scenario. It involves detecting patterns of activity, understanding personal needs when performing daily operations on the product and drawing conclusions to suggest further improvements to the process of use. The researcher in this format is a passive observer and focuses on the documentary material, which is then subjected to in-depth analysis.

In contrast to the methods associated with observing the natural use of a product, there are also methods for which the researcher uses a pre-written scenario. It is a set of tasks to be performed by the user, which the researcher observes remotely or in person. This set of studies belongs to 'Scripted use of product'. Due to budget constraints, they rarely take place in a dedicated laboratory, where the user is observed through one-way mirrors, and the screen on which they work and their movements are monitored and recorded. A solution could be a temporary laboratory in the office, where the user sits with the researcher in one room and the stakeholders can observe everything from another room - during the pandemic, many companies invested in digital meetings and, for hygienic reasons equipped, offices with additional screens, cameras and microphones, which can also be used to create such a laboratory. You can also conduct an online survey with design tools that have the option to track cursor and clicks or share a view of the user's screen. In addition to the Usability Studies described above, there are also methods that allow the product to be examined in a broader perspective, in the context of a large time frame and by repeating certain activities in sessions separated by a large time interval. They allow you to assess the user's progress and how the product is used, which happens after a period of time called Panel Studies. By tracking the user's movements over the long term, even for several years, data is obtained on how the usability of an app changes as it is used (Laurie, 2013). An additional method is Usability Benchmarking, which serves as a market comparison of applied solutions in order to establish universal standards (Joyce, 2020).

The last group of studies is 'Combination/Hybrid', which is a set of studies involving the use and combination of methods, so they are more difficult to categorise and assign to one of the above groups. These methods can be applied at different levels of product development, starting with Concept Testing – the presentation of a sketch, a model, a short advertising film or a solution that is already in the production process – to explain the operation and gather feedback, or when applied to a product that does not yet exist in can help obtain funding or to explore the potential of a product. The term Desirability Studies means researching user opinions and desires, finding out what the user wants and likes. The result should be confronted with another method from the natural product use group to assess whether what the user thinks about himself agrees with what he really does. Rohrer also includes Participatory Design as a research method in the hybrid research group, but in other sources it is interpreted as a way/approach to design through collaborative design/co-design that aims to actively involve all stakeholders (e.g. customers, employees, partners, clients, consumers) in the design process understood as a way to better understand, meet and sometimes anticipate their needs (Elizarova and Dowd, 2017). As a research method, it is interpreted by Rohrer as a collaborative performance of workshop tasks, but because of the many tools on the market for this, a separate group of workshop or co-design methods could be created.

In addition to Participatory Design, the "Co-design" group can include Design Thinking and Design Sprint methods, Experiments and all kinds of workshops aimed at taking a common direction for a product. They have many common features and often use the same tools, such as all kinds of games and tasks involving markers and sticky notes. The aim is to engage stakeholders at the very beginning of the process, without overloading them with too much analytical effort, but to focus on creativity and shaping the vision in a broader perspective before a concrete direction is set. These methods differ slightly from each other. Design Thinking aims to explore multiple ideas simultaneously. The predominant approach is to be user-centred, which is not always possible if it is the client, originator and developer who is involved in the process, while the potential user is not. Beyond business and technological considerations, innovation should take into account user behaviours, needs and preferences. The aim is to capture and find unexpected insights and create innovations that reflect consumer expectations more accurately (Brown, 2008). Design Thinking corresponds to the workshop design methods used by Google and referred to as Design Sprint, which concludes in 5 days, during which members define a problem by choosing a point to focus on during that particular sprint. On the following days they sketch competing solutions on paper, discuss and evaluate them, looking for the best way to make a hypothesis that can be tested by designing a quick prototype the next day and testing it on the final day (Knapp, 2016). In short, it is a proven methodology for solving problems by designing, prototyping and testing ideas with users. Design sprints quickly align teams to a shared vision with clearly defined goals and outcomes. Ultimately, it is a tool for developing hypotheses, prototyping an idea and quickly testing it with as little investment as possible in as real an environment as possible. Unlike weekly sprints, you can undertake Experiments, which can be done on the same day, without a pre-defined plan. Experiments involve brainstorming session participated by several people. The results are used to create a prototype, which is then tested on random users. This agrees with the original idea of Participatory Design, having its roots in the work of Kristen Nygaard and Olav-Terje Bergo, active in Norway in the 1970s. It is a long-standing tradition of designing systems with active user participation, centred on a design methodology that emphasises active collaboration between users and designers; an approach that has also attracted considerable attention outside Scandinavia, notably in Silicon Valley in the United States known for its innovative startups (Bødker and Pekkola, 2010).

3.4. Main principles of Lean UX design

3.4.1. Risk reduction

Innovation entails risk. It can be reduced by testing concepts and conducting research before implementation. The cost of doing an informal study called an experiment (Firlit, 2022) is much lower than optimising a version of a programme already in production. This means that a designer performing a UX study of a new idea tests its potential in order to confirm a hypothesis and is thus able to determine whether it is worth investing in the concept. De-contextualised testing methods can be used for this purpose (Rohrer, 2021). These are research methods that allow to study the product before its creation. One of them is Focus Groups – a meeting of a facilitator with representatives of the target group of users – held to determine their needs and associations as well as understand their previous habits related to products of a similar type. The objective is to avoid repeating the already existing mistakes and propose innovative solutions before the product’s implementation. Another method may be Concept Testing, i.e. creating a story in the form of a several-minute film, story or drawings about the planned product and confronting the idea with potential users, gathering their opinions and perspectives on the presented solution on an abstract level.

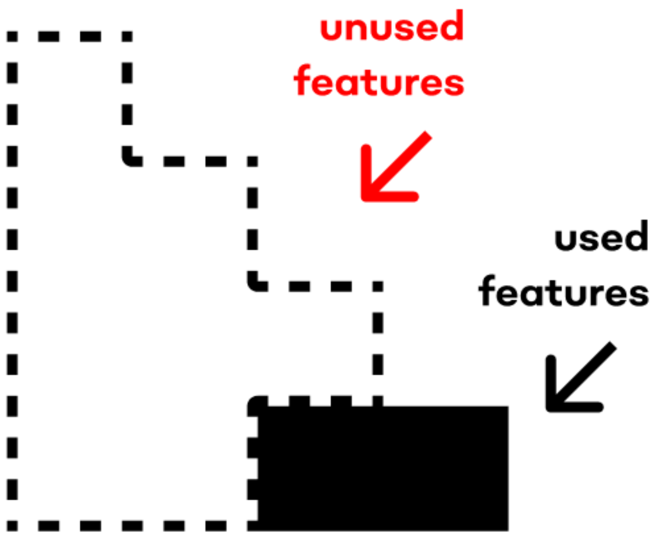
3.4.2. Quick failure

The team should suffer failures as soon as possible (Gothelf and Seiden, 2016). This will help them understand whether the ideas are worth pursuing and whether it is worth going in the direction they want to go. The sooner the idea is a failure, the easier it will be not to make unnecessary expenses. Failure is carried out with collective responsibility. Despite the pejorative connotation of the word, the team does not feel personally responsible for the failure, but treats it as a discovery and a tool to establish improvements or to change the concept. The sooner we find out what is wrong, the sooner we can fix it. The cost of conducting pre-investment research is lower than the cost of making mistakes and implementing and maintaining the existing features. The same applies to products during the design process. If the team was not able to spot errors early enough and started investing in uncertain solutions, it is necessary to do an evaluation and find a turning point to change the direction of the product creation and find something that makes the project worth investing in again.

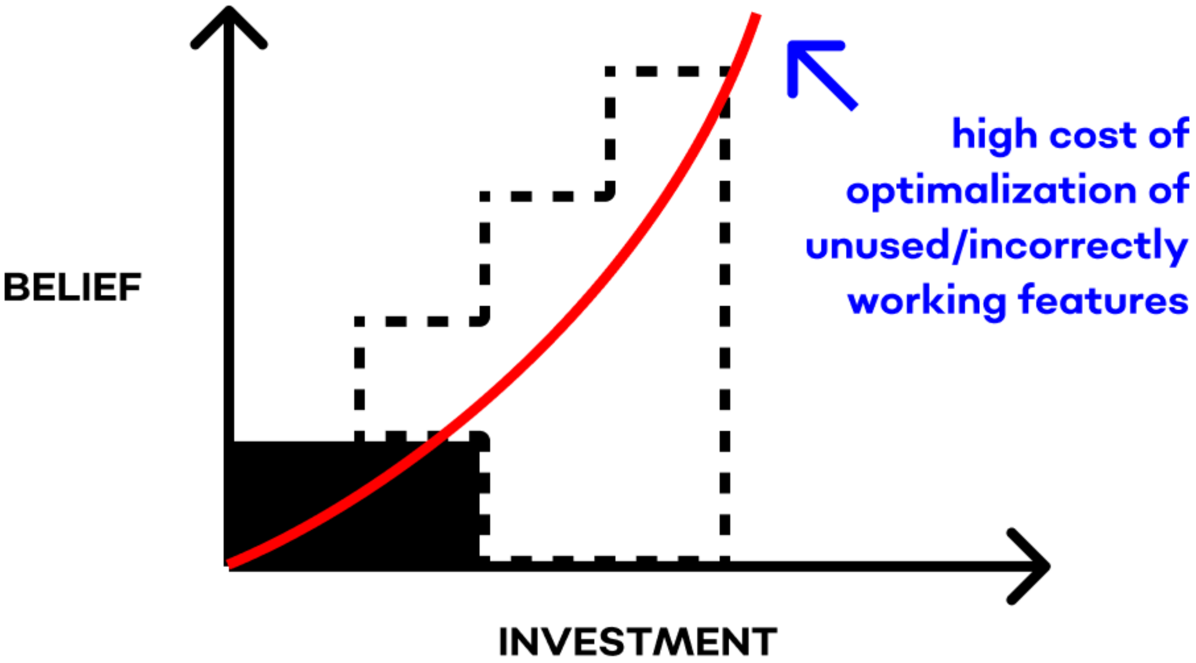
3.4.3. Reducing the amount of waste

If the item is not an MVP, it should be omitted. Unnecessary meetings, documentation, additional features, without which we can build the product, should be reduced. We do what is 'enough'. Adherence to processes involves many repeated rituals. In Scrum, daily meetings and ceremonies are aimed at guiding the product and are closely linked to the framework and methodology. At the same time, they take time, so it is important to consider whether all recommendations are necessary for our process or represent an additional burden. Additionally, Lean UX recommends skipping documentation and paperwork if it is not necessary. The selection principle can also be applied to reducing MVPs to only necessary functionalities. Reduction will save time and warn against building a product whose functions will not be used.

3.5. Benefits of using UX in the design process



Huge sums of money are invested in the IT industry, and contracts often set out assumptions in advance, supplemented by anticipated solutions. The amount of capital allocated for building a product increases with the belief that it will work. Research from 2020 shows that 80% of product features are not used (Firlit, 2022). By comparison - in 2015 this figure was 64% (Rice, 2015). This clearly indicates an over-investment in features that the team deems necessary, regardless of the real needs of the user. The number of redundant features at such a high level sounds unbelievable, but just analyse the first app you see, such as Facebook, used for commenting, liking and posting. Its interface is equipped with a number of additional functions such as: Ads Manager, Climate Science Centre, COVID-19 Information Centre, Community Help, Dating, Fundraisers, Marketplay, Memories, Gaming video, Play games, Watch, as well as Facebook Pay and a weather forecast. As Magdalena Firlit put it in her workshop – there is a pervasive "featureosis", i.e. an exaggerated tendency to keep adding new features without a clear demand (Firlit, 2022). While committed programmers add new features to showcase their own advanced abilities, those completely uncommitted copy codes from available sources and use ready-made templates of various available features to create the impression that they are moving forward. In both cases, there is a visible trend to equip products with additional functions. Just because something is doable does not mean you should invest in it. An investment mistake can cost several million because of the subsequent need for costly optimisation. Building a prototype in a design program or even drawing it on paper and consulting with potential users through an interview or survey is much cheaper than blindly investing in features hoping for a lucky strike. Firstly, the work of a team of developers is much more expensive and involves more people than the task of a designer. Secondly, the work of developers is more time-consuming, so you wait longer for the final result. Thirdly, backing out of a coded feature or optimising it increases the risk and cost, so it is better to invest in the first phase of design and spend more time understanding user needs.



A report on the impact of Design Thinking implemented at IBM (2018) revealed the following effects, achieved after three years of its being in place: USD 20.6 million in revenue, the ROI of 304%, twice as fast market implementation, 75% less design and testing time, and a 33% reduction in the product development phase (*The Total Economic Impact™ Of IBM's Design Thinking Practice,* 2018).

The Design Value Index study (2016) showed that 16 companies considered design-centric made 2.11 times more profit than the overall S&P 500 stock index (Kałamarz, 2019). According to the report "Parts Without A Whole?" Design Thinking furthermore translates into at least two important business aspects. 71% of respondents said that the introduction of DT had improved their work culture on a team level and 69% perceived the company's innovation process to be more effective (Schmiedgen. et al., 2015).

3.6. Product obsolescence - optimisation of existing applications

Another important issue is designing for existing products. Built in specific conditions and contexts, they are prone to ageing. If they have not been updated and tested over the years that passed since their implementation, one should assume that they have not kept up with changes in technology, user habits, an evolving market, etc. A product has its useful life, after which it becomes significantly less functional. The frameworks described earlier were created to build a new product, identify user needs and propose a solution. With technology constantly evolving and the market being saturated with digital products, building new products takes place in a specific context and a competitive environment. Users do not rely on just one solution. They have many similar applications to choose from and select a particular one based on their own preferences. Therefore, it makes sense to define a target group and understand the specific user instead of trying to please everyone.

There are also many products on the market that need repair and optimisation. For their modernisation, a design process is needed that defines a different strategy of action. Some recommend closing the application and designing a new one in its place, which is not always a feasible solution. It is then necessary to examine the existing project and identify and prioritise its errors. It is worth verifying and clarifying the target audience, researching how they use the product and drawing conclusions based on the findings. It is necessary to separate the operations that can be achieved in a relatively short time from those that require a very long repair process and segregation in such a way that they can be implemented within the selected process.

In some cases, building a new product on top of a currently existing one requires a separate process. Where a company has a functioning project that has become obsolete and at the same time is building a new one to operate on similar principles, a process where both products can be analysed will be useful. A detailed comparison focused on finding common characteristics should make it possible to export data from the old product to the new version.

4. Models of leading a project team

4.1. Scrum - focus on developers, designers must adapt

The dominant framework in the IT world is Scrum, which involves solving complex problems by breaking them down into smaller parts executed in what are called Sprints, in an iterative format. Scrum is described as a productive and creative process, delivering implementations of the highest possible value (*A Better Way Of Building Products*, 2020). Its creators, Ken Schwaber and Jeff Sutherland, have written a Guide in which they define member roles and describe processes consisting of events, artefacts and the rules that link them. In the model Scrum, all members of the project team are referred to as developers, irrespective of whether they work as frontend, backend or UX. The Scrum trainer Magdalena Firlit refers to the etymology of the word developer, which means to create, to build something. She draws attention to a broader interpretation of the term, which is not limited to a person working with code. To emphasise egalitarianism in the team, and to dispel the usual dichotomies, she advocates the universal phrase "team member", which refers back to Scrum's original theory (Firlit, 2022).

Anna Kaley points out that the Scrum framework has been created for developers and has no place for a designer. The dominant group of developers is given the priority and can impose their conditions. Today, developers earn more than designers. They are mostly male, which leads to the "Boys club” phenomenon, an enclave made up of people with similar characteristics, able to influence those outside it. While studying Interaction Design at NTNU in Norway, I am in a group consisting mostly of women, which might suggest that the IT environment is gradually attracting more female employees. However, the 2021 Women in Tech report states that the majority of women in the tech industry have already had to deal with 'Bro culture', the industry's version of the corporate men's club, the operation of which can adversely affect women's comfort at work and even lead to exclusion or the feeling of insecurity. The report also found that women in the tech industry find it harder to find mentors and role models. Many of the women interviewed said they had never had a mentor in their industry and often felt ill-adjusted or less qualified compared to their colleagues. 45% of them believe that discrimination and prejudice are the cause of the wage gap in the IT industry. Men are 3 times more likely than women to think that the wage gap is due to a difference in productivity. In addition, women feel more pressure to engage in their work than their colleagues of the opposite sex (Sullivan-Hasson, 2021).

However, the problem does not necessarily lie in the gender difference but can rather be traced to the division between a group of developers and, usually, just one designer. Code programming is not evaluated by people from outside the field, while design trends to be judged by everyone, including people without the actual qualifications. The role of design in the design process itself is also commonly generalised. The lack of distinction between Industrial Design, Graphic Design, Product Design, Service Design, UI or UX Design makes it difficult for designers to use their specialised qualifications correctly. The effects of their work are often assessed on the basis of opinions rather than a careful analysis of the data collected during user surveys.

While working in a company where the CTO was a licensed Scrum Master (Chief Technology Officer, in this case a person trained in this framework, with the competence to manage Scrum Sprints) I was referred to not as a designer, but a UI developer. At this position, I was expected to design packaging, logos, t-shirts and other elements that are associated with graphic design or Industrial Design. This revealed a problem in properly understanding what User Experience really is. I also had the opportunity to work in a team where the developers were grateful for the opportunity to work with me, as they had previously worked without a designer. Initially, the collaboration consisted in familiarising the team with the possibilities of various design fields and identifying those areas where Design could be applied. What matters in such a process of establishing a new position is a clear division of responsibilities. Design tasks should be carried out by the designer and their usability checked, based on the research of the developed product’s target group.

Respect, mutual understanding and openness to different competences constitute an important part of the process, so in addition to nurturing hard skills such as knowledge of methodology and software, developing soft skills within the team should also not be neglected. Both parties should keep certain distance from the results of their work and evaluate them externally, through user feedback.

4.2. User Driven Development: Build – Measure – Learn

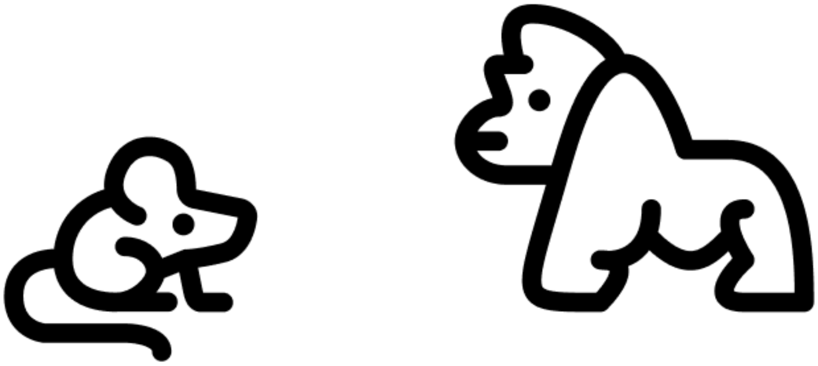
In the design community, User Centred Design is now the main paradigm. Due to the increasingly popular focus on the user in product development processes, the term – User Driven Development – has also emerged in the software development community. It is not just a set of methodologies, but a philosophy that should guide the team (Psychogios, 2014). The most important factor that will make a product or service successful is the satisfaction of the end user. Product development depends on the personal needs of users, i.e. the replication of the Lean principles, according to which one starts with a BETA version of a new product, reduced to the minimum required features, confirmed by user research and confronted with technological and budgetary constraints. The functional prototype should be presented to the customer as soon as possible to receive feedback. The process should be repeated after the implementation of each new product feature. This ensures that the team constantly receives user feedback for each subsequent implementation, including when new features are introduced. This continuous process is also called build - measure - learn. It is a concept inspired by Eric Ries's "Lean Startup" (TechTarget Contributor, 2017). The method attempts to apply science-like practices to start-up companies. BML aims to avoid less effective methods, mainly excessive haste and the "before and often" approach that results in a lack of time for thoughtful selection of needed software revisions. Whenever a number of extensive requirements for security, usability, maintainability and buildability is implemented according to internally set priorities to the exclusion of real user needs, the budget is at risk. The perspective of the target group is an extremely valuable source of data in the product development process. In addition, once the product is built, customer feedback and metrics are used to verify hypotheses and validate customer and market knowledge. This allows energy to be focused on the right features, with efficient use of time and less risk of error.

5. Optimal mode of operation while using UX Design

5.1. Design Leadership

The traditional approach to leadership separates the leader from the team, places them higher in the hierarchy of the company's employees and allows for the management of human resources. In the modern model, the manager does not question the authority of the subordinate, knows his or her weaknesses and is able to assemble a team with the right qualifications. It is therefore the person who sets the direction, while maintaining the autonomy of the other employees. Setting direction also means going ahead and paving the way for the other team members, taking risks, discovering obstacles and overcoming them.

An important quality of a leader is to recognise their weaknesses. An accurate diagnosis will make it possible to fill in the revealed gaps during the selection of the project team members. The leader’s conviction that he or she does everything best and that the team members should copy his or her actions will limit the opportunities for growth within the team, because, as a result, everyone will reproduce just what remains within the leader's limit of knowledge. To create something innovative, you need specialists with different skill sets. Being aware of this, the team can act on knowledge sharing principle and spot gaps in the process.

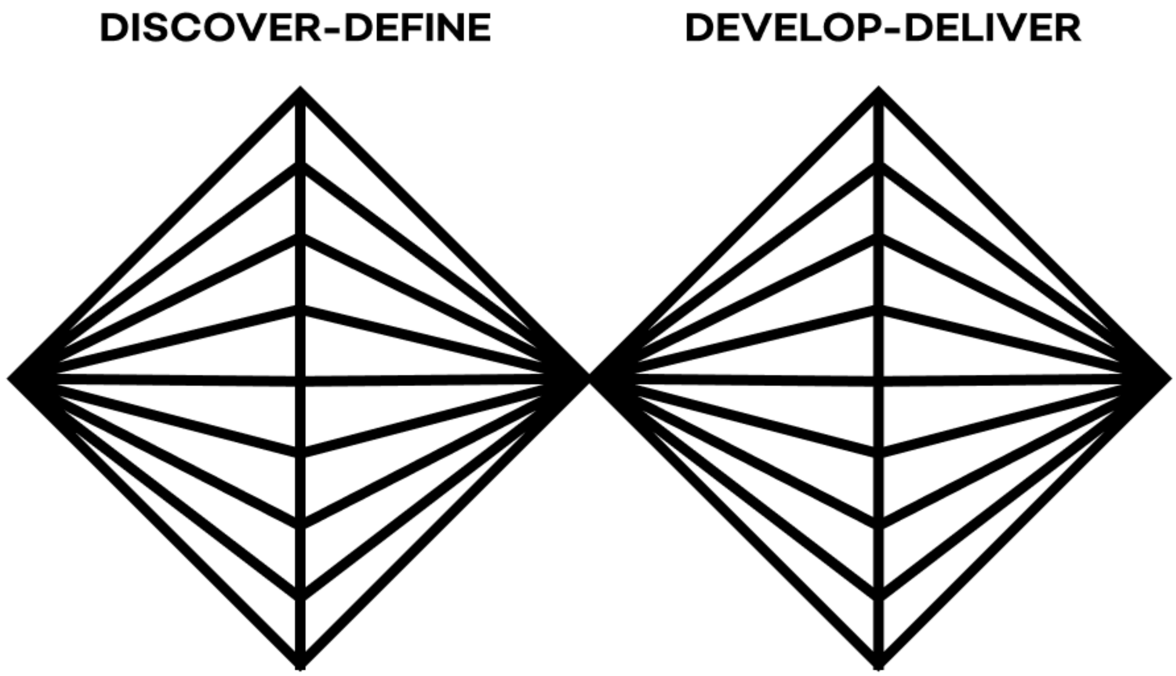


JUNIOR VS. SENIOR

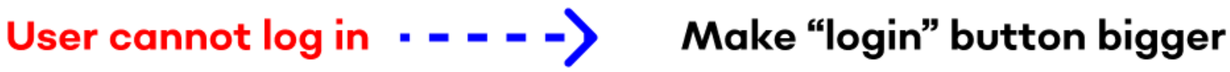
Another aspect is motivating team members. Juniors and Seniors need different motivation (Vetrov, 2021). The junior is keen to look at the work of more experienced people and needs direction and feedback on their tasks. Seniors are more likely to work without much control and are motivated by having a real impact on the project. The leader should give them a free hand and observe their progress. The number of tasks for a particular team member should be controlled in order to avoid excessive workload that can lead to burnout. UX designers are divided into UX Generalists and UX Specialists (Ross, 2015). Ideally, a company has enough budget to hire specialists and build a team in which everyone is responsible for a chosen sub-field of design. The team is supported by UX Researchers who do research, Interaction Designers who study how users react to the presented interface, UI Designers who oversee the Design System, etc. However, with the current disparity, the UX designer must fulfil the work of multiple roles, i.e. be a UX Generalist. Every employee has his or her own strengths and weaknesses, and multitasking does not allow them to train to be a specialist in a particular field. This large disparity also means that design is not seen as something important. Then, it is much easier for the large group of developers to promote their visions, while designers who remain in the minority find it difficult to advance their ideas and win support. UX Lead should step in such situations and ensure that the company is aware of the importance of UX in terms of profit and product success. According to Mike Jones, a sufficient ratio of designers to programmers is at least 1:4 or better 1:3. (Jones, 2021).

5.2. Scrum + UX

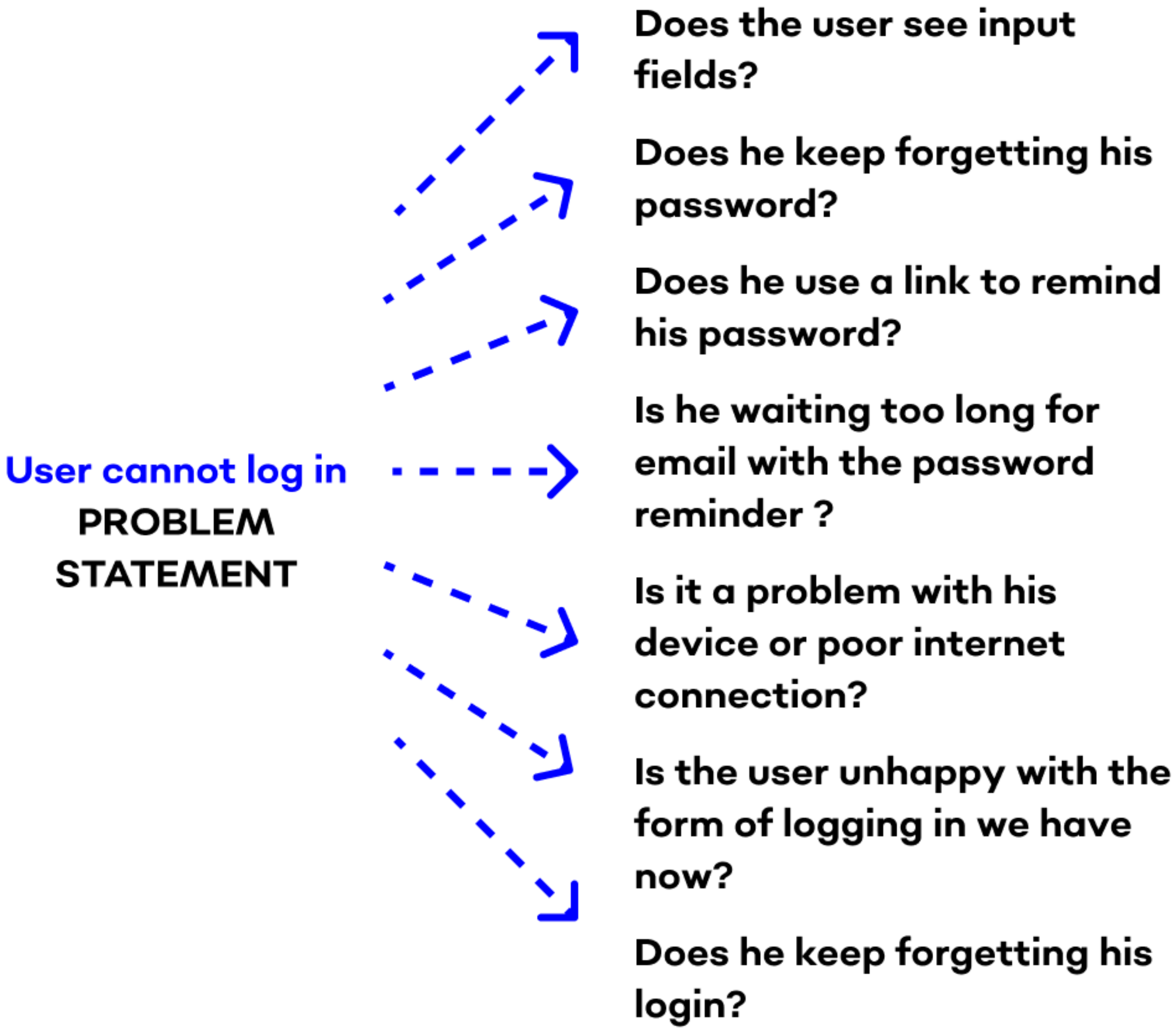
One of the most common frameworks among UX designers is the Double Diamond, officially presented by the Design Council to visualise the design process in four phases divided into two diamonds: Discover-Define and Develop-Deliver. These two diamonds represent the process of examining a problem widely and deeply (divergent thinking) and then taking focused action (convergent thinking) (*What is the framework for innovation? Design Council's evolved Double Diamond,* 2019).



The visualisation of this method takes the shape of diamonds – rhombi: the form starts on the left from a single point, expands to the two outermost extremities and narrows back to one point (Pervaiz, 2018). The method does not employ the simple question – answer format. For example, in the case of a user's login problem, in the straightforward question – answer mode, the team will unanimously choose the obvious solution, which might be to increase the visibility of the log in button. Although the problem can indeed be much more complicated, in many cases such decisions are made for the user, without conducting proper research. Especially, if the user has no representation in a team consisting only of engineers and no UX Designer.



In Double Diamond, however, a question may be asked of why the user is having trouble logging in, followed by further questions – that will recreate the rhombus shape: can the user see the input field? Has he forgotten his password? Does he use the password reminder link? How long does he wait for feedback? Is it a problem with his device or a poor internet connection? Is it rather that the service is too slow? Does the user like this form of login?



User cannot log in
PROBLEM STATEMENT

Does the user see input fields?

Does he keep forgetting his password?

Does he use a link to remind his password?

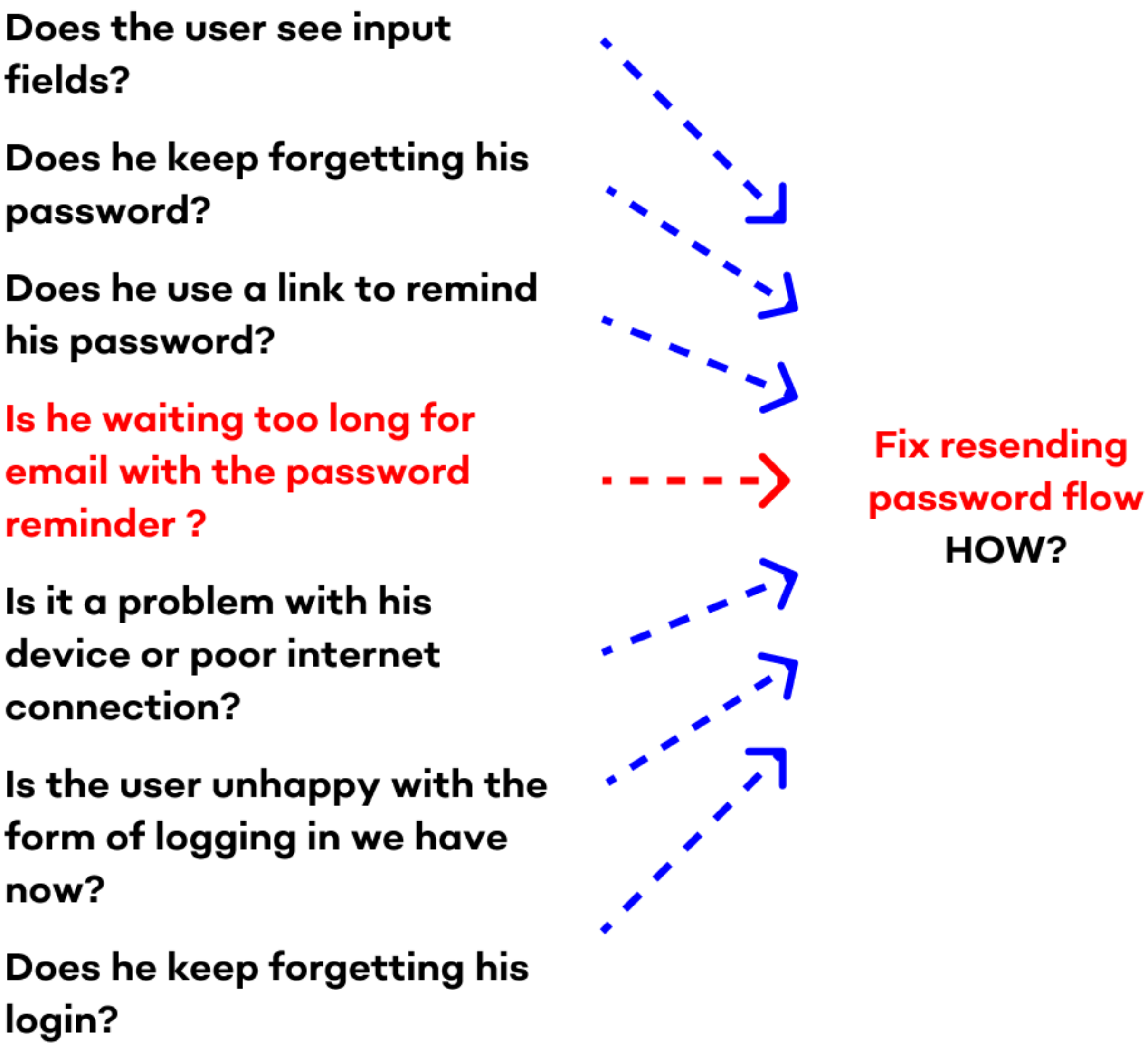
Is he waiting too long for email with the password reminder ?

Is it a problem with his device or poor internet connection?

Is the user unhappy with the form of logging in we have now?

Does he keep forgetting his login?

I In order to define the problem and understand if we are looking for an error in the right place, having completed the broadening stage (divergent thinking), we need to start narrowing until the problem is identified. For example, user research may indicate that the login problem is due to the fact that the reminder link arrives after too long, making the user unable to log in and stop using the application.



Does the user see input fields?

Does he keep forgetting his password?

Does he use a link to remind his password?

Is he waiting too long for email with the password reminder ?

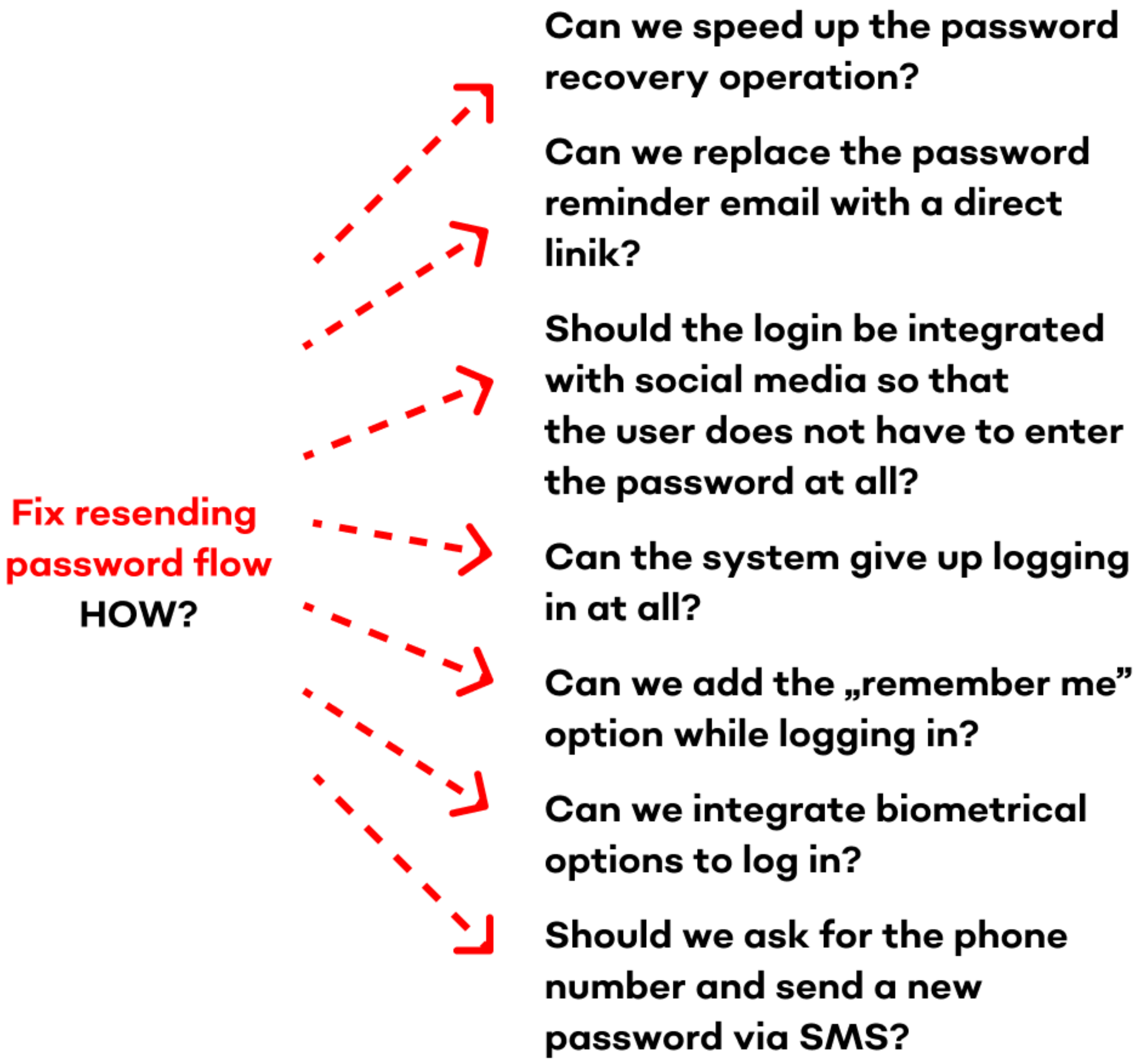
Is it a problem with his device or poor internet connection?

Is the user unhappy with the form of logging in we have now?

Does he keep forgetting his login?

Fix resending password flow

HOW?



Fix resending password flow HOW?

Can we speed up the password recovery operation?

Can we replace the password reminder email with a direct linik?

Should the login be integrated with social media so that the user does not have to enter the password at all?

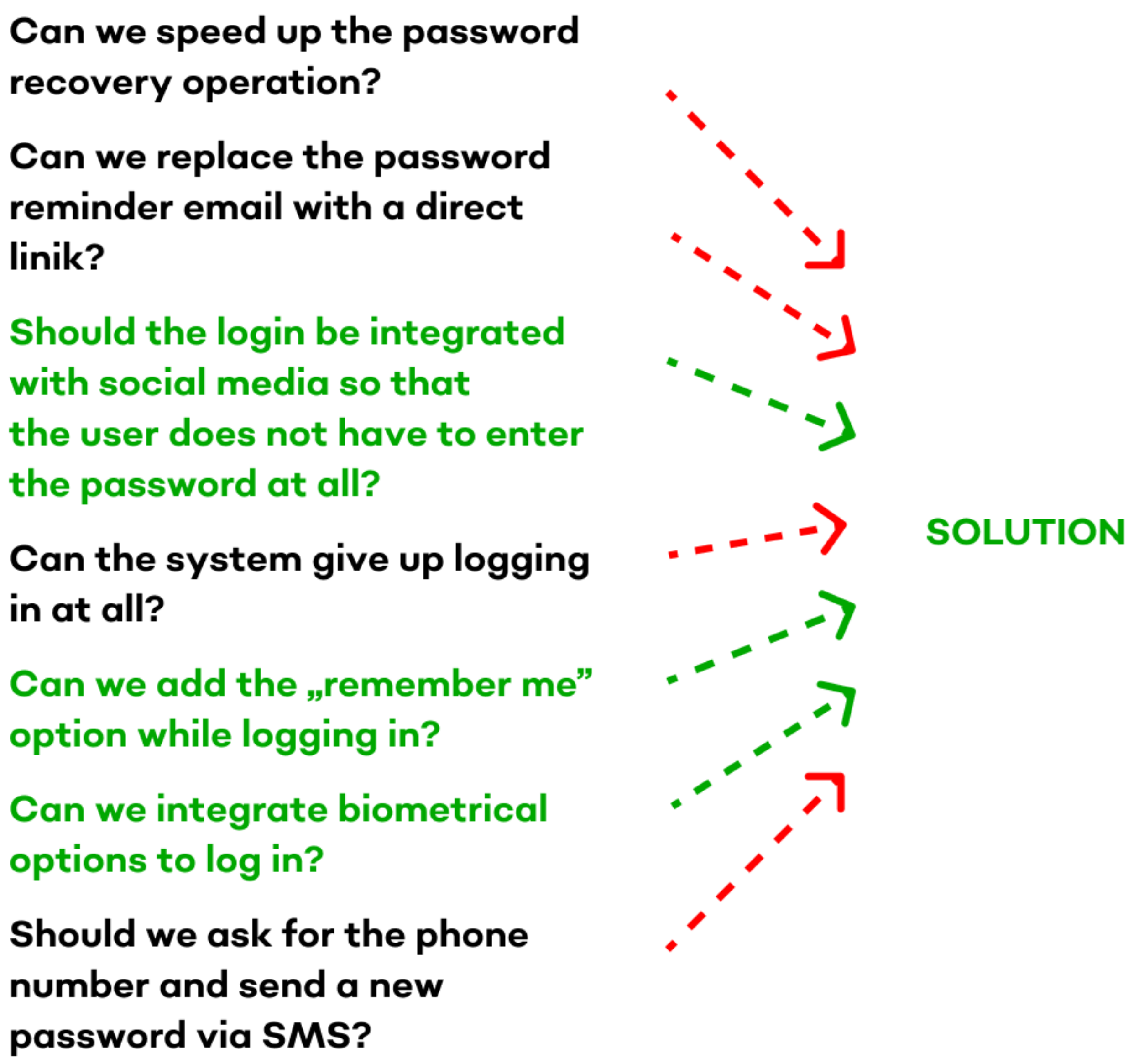
Can the system give up logging in at all?

Can we add the „remember me" option while logging in?

Can we integrate biometrical options to log in?

Should we ask for the phone number and send a new password via SMS?

Then, the range of solutions is broadened again: can we speed up the password recovery operation? What can the password reminder link be replaced with? Do we integrate login with social networks so that the user does not have to enter a password at all? Can the system dispense with logging in, or is it necessary in this application? Then, once a suitable question and solution have been selected, a specific hypothesis can be made. The selection is made through various research, understanding of technological limits, analysing the budget and available resources. In the above-discussed case, the application of the DD method might help to determine that the user needs a "remember me" option when they first log in so they do not repeat the action the next time they use the app. It would also be advisable to add the log in via social media option, which eliminates the need to type in a password or to enable the user to log in via biometrics if they are using a phone.



Can we speed up the password recovery operation?

Can we replace the password reminder email with a direct linik?

Should the login be integrated with social media so that the user does not have to enter the password at all?

Can the system give up logging in at all?

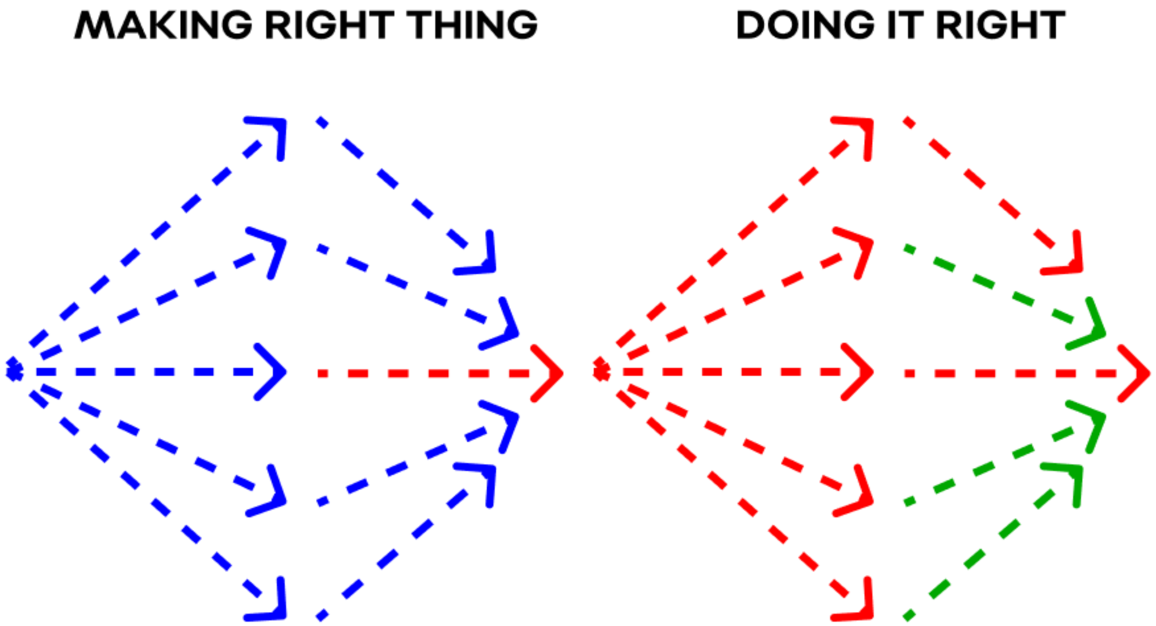
Can we add the „remember me" option while logging in?

Can we integrate biometrical options to log in?

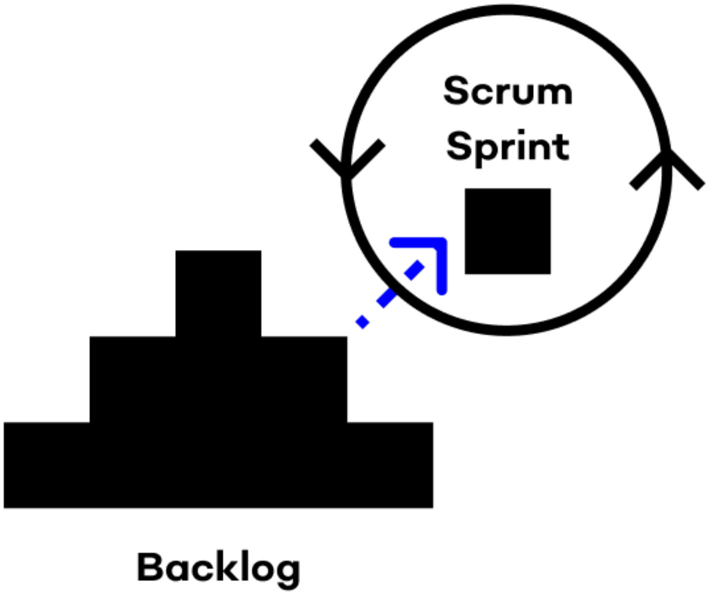
Should we ask for the phone number and send a new password via SMS?

SOLUTION

Of course it can be debated further whether this is a valid hypothesis – the log in rules will be different for a banking app with enhanced security and for an app showing bus timetables. Selecting appropriate test methods and performing the analysis will resolve these and other concerns. The role of the presented example is only to demonstrate why this method is called a double diamond.

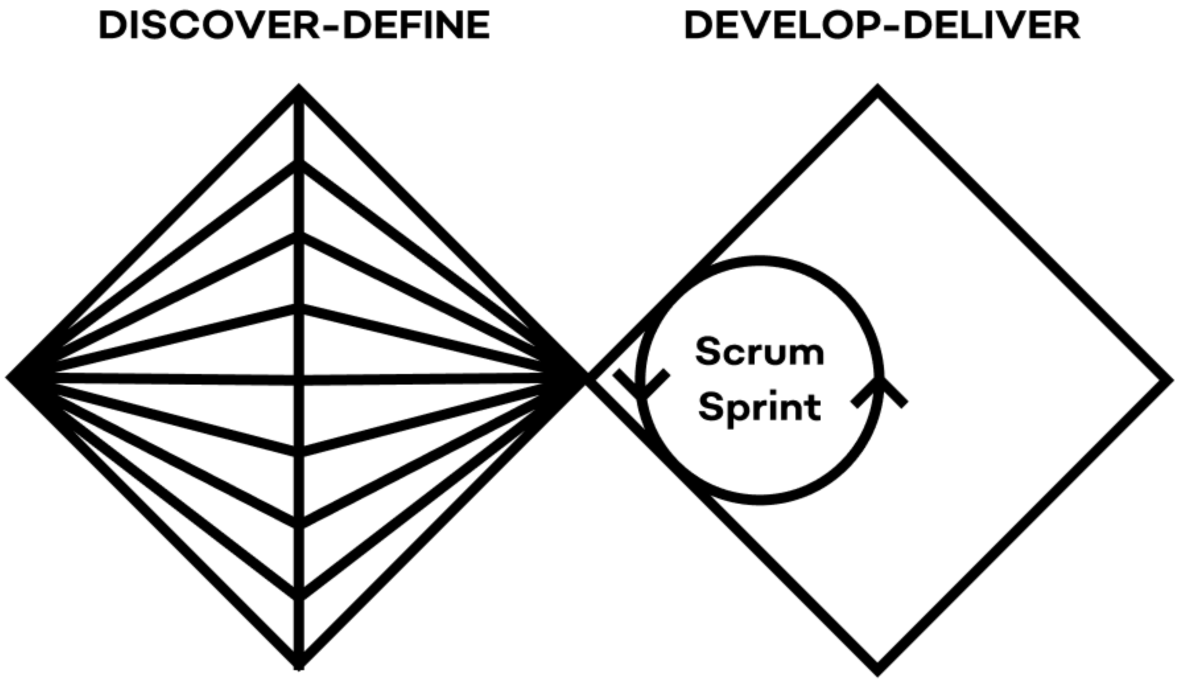


Scrum has its own rules and principles: firstly, everything that needs to be done is kept in one place – in the Product Backlog.

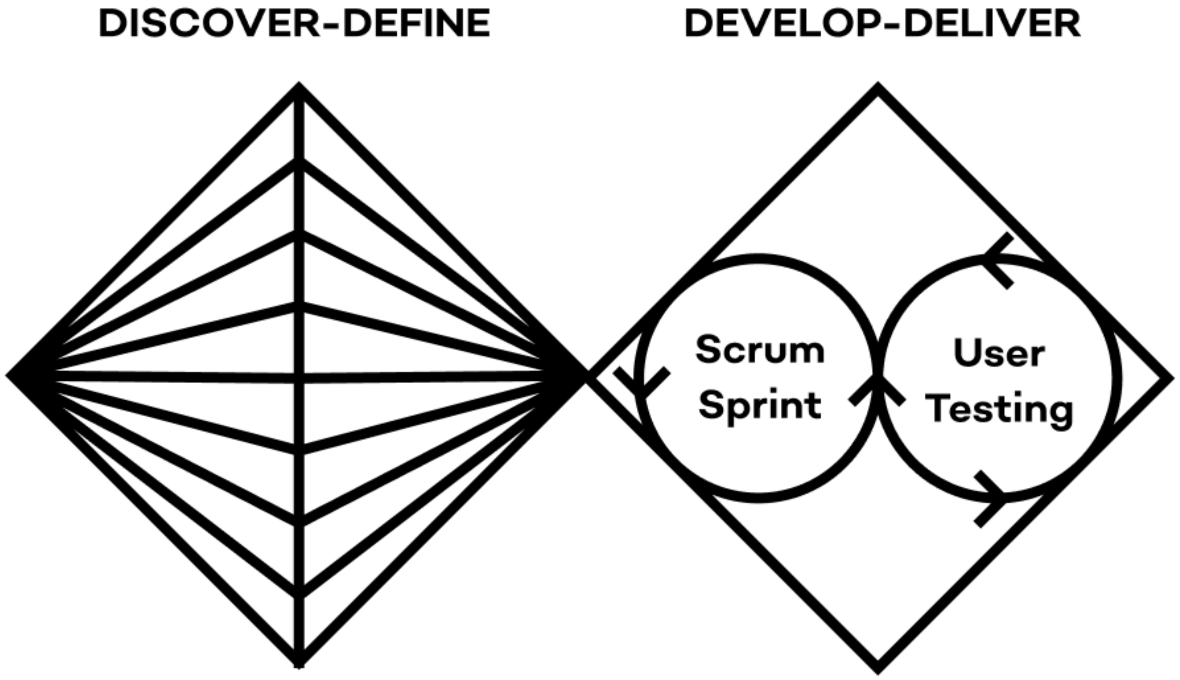


This to-do list is constantly updated and tidied up – this is the responsibility of the Product Owner. The team then makes a detailed plan for a short period of time called Sprint. This happens during Sprint Planning and the detailed action plan goes into the Sprint Backlog. Both Product Backlog and Sprint

Backlog are available to the team and any interested person (Transparency). The team starts working on tasks and meets for a while every day (this meeting is called Daily Scrum) to check if the plan is still valid and takes action if necessary (Inspection and Adaptation). The culmination is the last day of the Sprint. The team presents the results of their work to the people concerned and, based on the feedback, decides on the next steps of the work (Inspection and Adaptation of the completed work). This meeting is called Sprint Review. It is followed by the all-important Sprint Retrospective, where the team looks back at how they worked and makes improvements with which to achieve higher efficiency (Inspection and Adaptation of the way they work). They are assisted by a Scrum Master, who is also responsible within the organisation for implementing the entire process as a facilitator (Borowiec, 2021). As most IT companies work with this framework, the designed interface will be included in Sprint and the ideas that emerged from the UX research go to the Backlog.

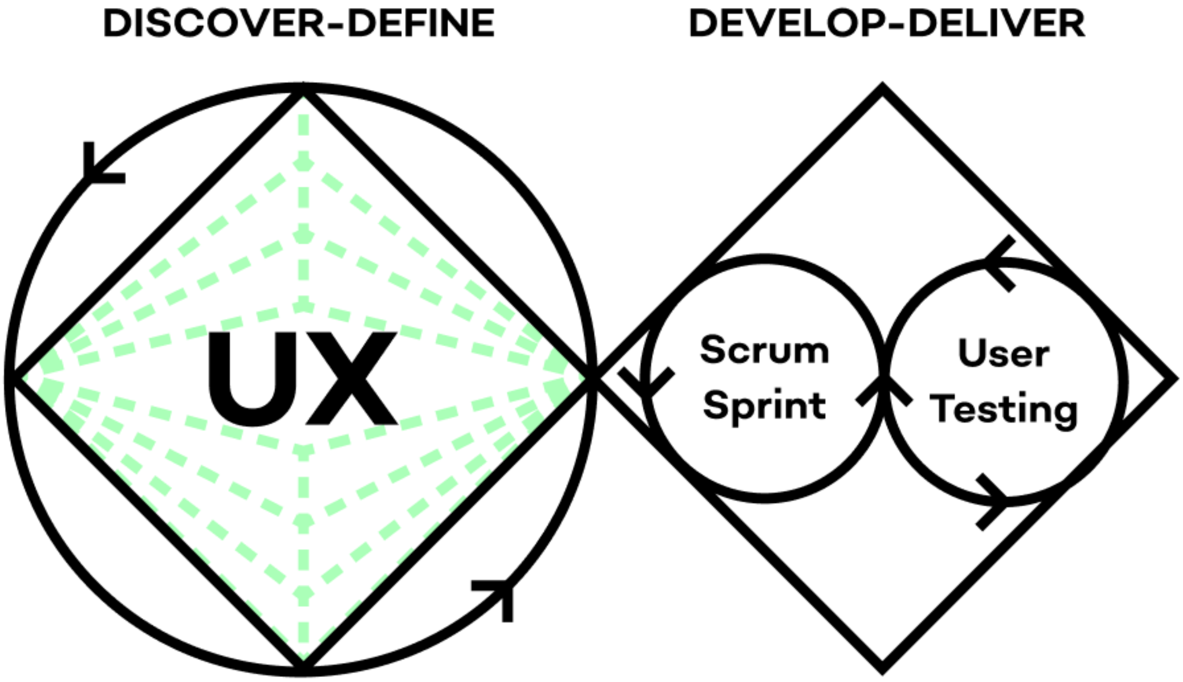


Product testing also takes place iteratively after each Sprint.



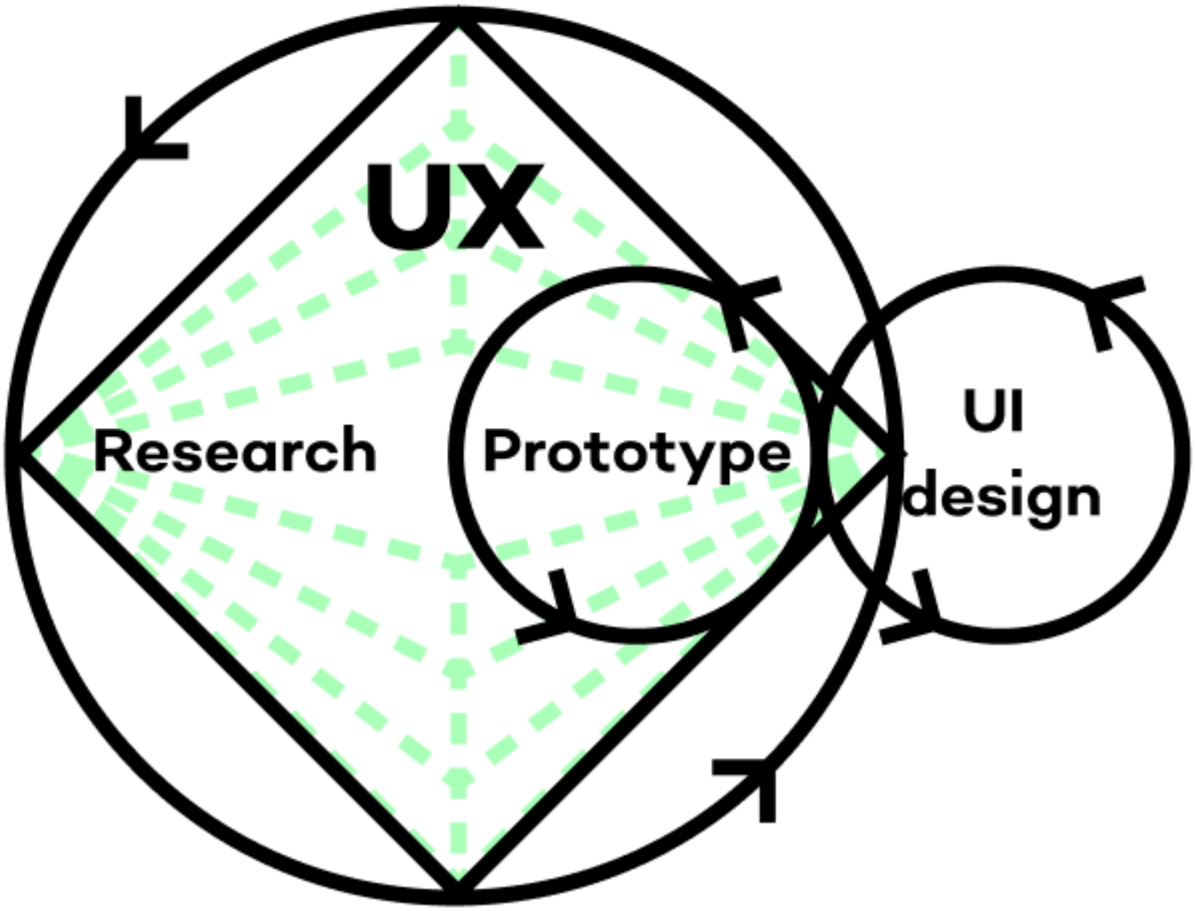
However, before we take on the costly job of coding a product, UX research should be done to make sure we are on the right task. This may be achieved by changing the perspective of the framework. In fact, Agile is not only about Sprints, but also the Envisioning phase that should precede them. This is a product-level envisioning and planning phase (Rubin, 2012) that helps organisations describe an idea and create a rough plan for how to approach its creation. It is in this phase that time for research and Design should be found. Therefore, developers' Scrum Sprint should be preceded by designers' Sprint, so that the latter have enough time to prepare changes and corrections resulting from the ongoing research. This would give designers time to prepare the design for developers for the next Sprint. Unfortunately, designers are usually required to work in a Just in Time framework, i.e. fill in missing elements on the go in the same Sprint as the other teams, without the opportunity to do research and think whether implementing a particular element is a sensible solution.

The research phase is also iterative, as building a product is never finished. Technology, people's habits and ways of using products are constantly changing, and to keep up with these changes we need continuous improvements based on ongoing research.

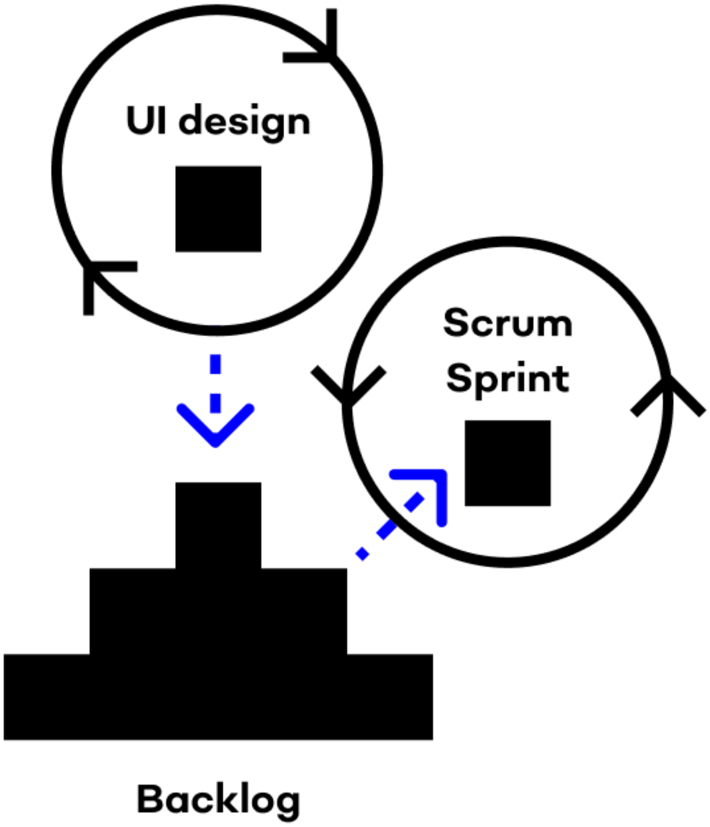


5.3. Agile + Lean

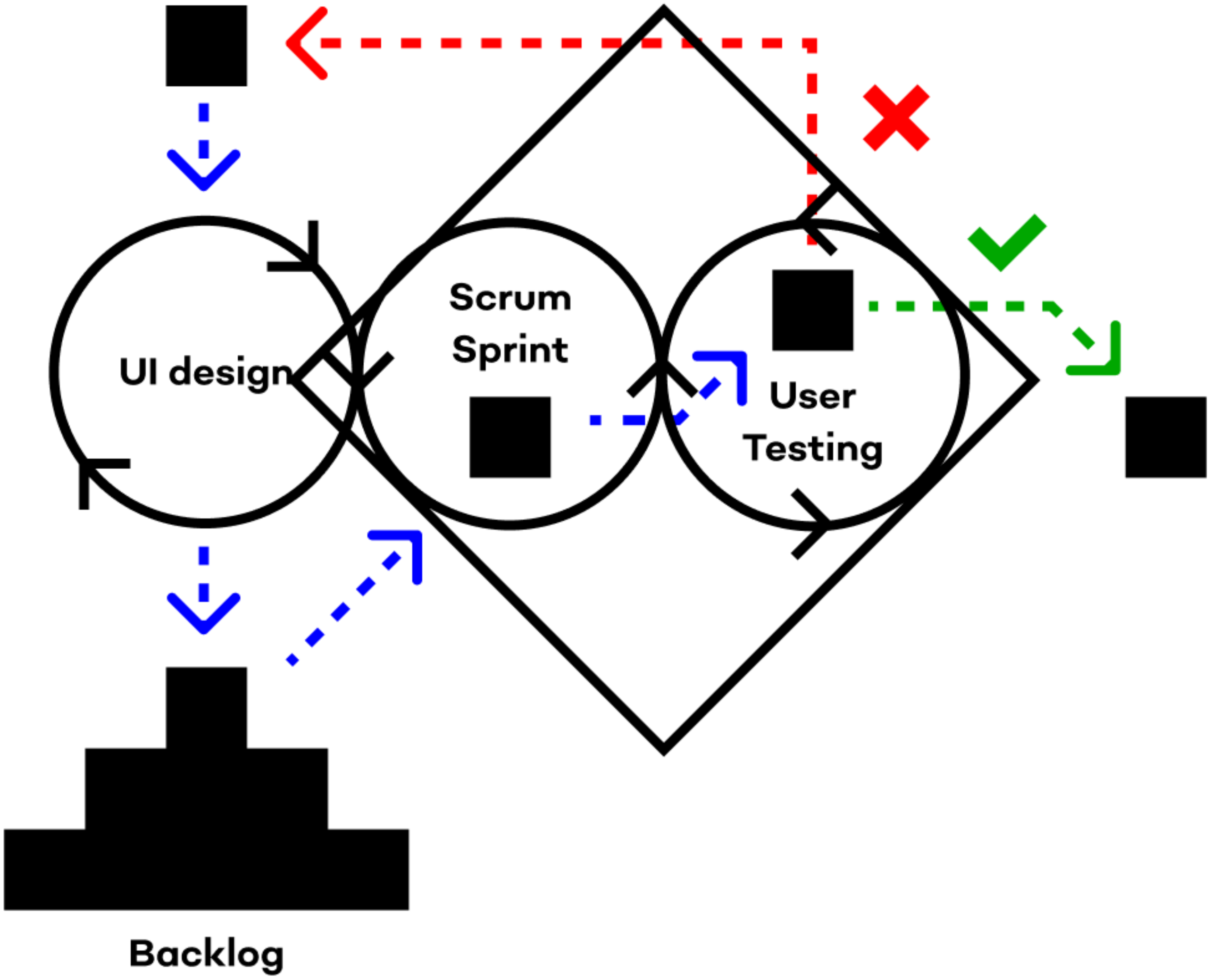
Lean, which involves adjusting and shaping the product according to user feedback, and Agile, which involves delivering the product in small parts, come together in the design process. The first stage is testing without using the product. Through interviews, focus groups, surveys and analysis of current user processes, it is possible to define the problem and confirm the chosen hypothesis. The next step is to create an abstract prototype in the form of a sketch, story, short video or other form that can be validated by users. Once the capabilities have been extended and the first tests have been carried out, prototyping can continue, either as a low-fidelity or a high-fidelity prototype, with the use of mock-ups and depending on whether the designer has a design system with ready-made components to build user paths. Some of these ideas and prototypes will be verified by further user tests and forwarded to the interface designer, who will refine their appearance so that they can be presented to developers and consulted by the Product Owner, who will decide what the first priority in the Backlog will be.



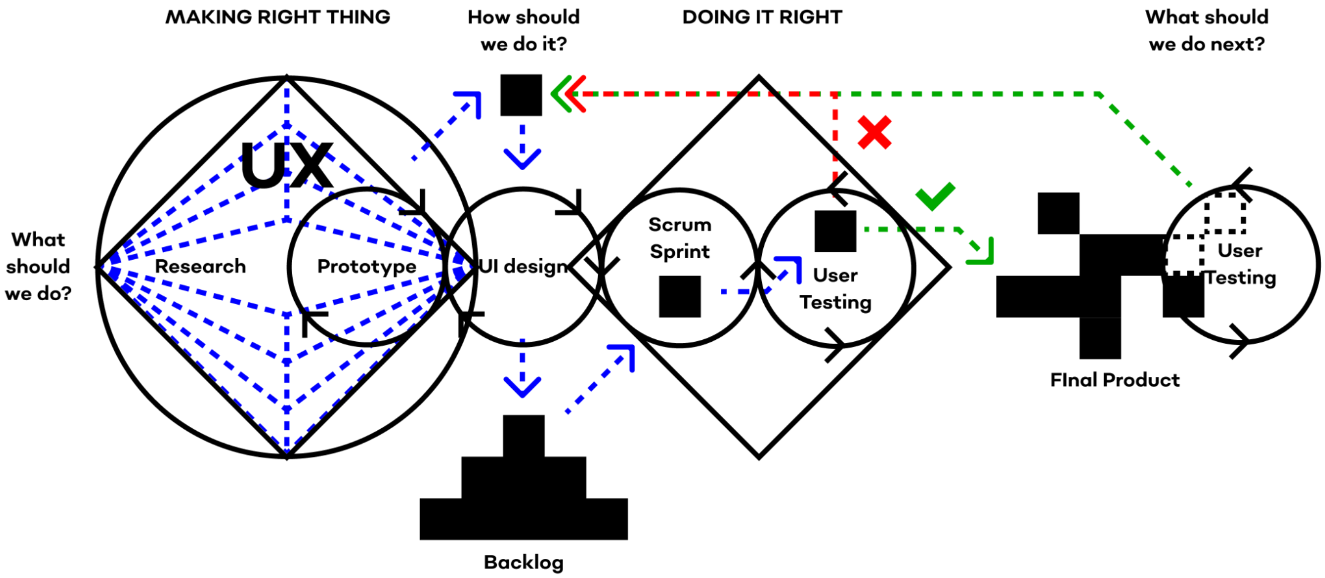
In the proposed model, prototyping is the culmination of the first diamond. Its narrowing can take place through testing the designers' prototype to get the idea of what the design might look like. The conclusions are then collected, which allows for preparing the specification and scope project. It is when UI Design is beginning to emerge. It represents partly the tested prototype and partly the new ideas and solutions gathered during the Discovery phase, as well as technical constraints and guidelines associated with the brand. UI Design does not cover the entire product at this stage, but its basic features. Design work should precede coding so that developers can take a broader view of the product and the direction of its development. This will improve the planning of the next tasks, as well as thinking several Sprints ahead. The most important features, however, are those without which the product cannot work. These basic elements make up the MVP, which should be placed in the Backlog along with other ideas. All the data is arranged according to the priorities of the Service Owner or Product Owner.



Working according to Lean methodology, the Scrum team builds the product from the first small part. While not being sure what the final product will look like, the team works according to a plan for the next few stages of the process. Testing each of these will set and confirm the course of action. When talking about testing, we do not mean QA Scrum team testing, but the UX Designer confronting the product with the user. The second diamond phase is to reassure the team that we are doing "it" right.



The confirmation of the product part with the user's needs determines the further direction of the work. If the tests go well, the created part lands in production; if not, it goes back to the UX team, which may re-examine and improve the design, or is eliminated if PO or SO decides so. In this process, the product is built until it meets the requirements agreed on with the customer in the contract. To get the best user rating you need to keep testing the product. It is worth collecting the documentation of the tests in order to pass them on to the customer should the need to justify the changes, improvements or optimisations appear. The user research report will also be helpful when negotiating changes to the contract or concluding further orders.



7. Summary

Creating and testing ideas at a very early stage can be part of the discovery, and in an ever-changing and digital world, no idea is 'finished'. We continually receive feedback on the performance of products and services and constantly improve them. Undertaking research at an early stage of product development, during the development and after the implementation, makes the market more dynamic and influences the development of digital products and services. Innovation requires courage and risk-taking. The risk itself, however, can be reduced by using an appropriate process. Early detection and anticipation of possible errors builds confidence within the project team. Principles should be adopted by problem solvers so that they can work as effectively as possible. Putting the user first and understanding the needs and aspirations of the service users should characterise each phase of the team’s work. Concentrating the process of building a product around the usability is the main task of designers in project teams. With a clear division of roles, the team will be able to collaborate effectively and be inspired by the work of others, and then just iterate.

The presented methods for developing software for modern digital devices are not ideal. In this work I have tried to show the positives and negatives of each strategy. However, there are a few principles that are common to them all:

1. the need to know the real needs of the user
2. the need for teamwork
3. the need to clearly define the competences and tasks of individual team members

The discussed methods also help to reduce risk when bringing a new product to the market and boost team productivity. They increase the likelihood that working on a new product will be successful and that the planned budget will not fall apart before implementation.

The field of economy dealing with software development is relatively young. The mechanical machines that were the prototypes of modern computers were developed during the Second World War. Unlike construction, which has been developing since the dawn of humanity, software development has been with us just for a short while, but it has already become an integral part of our civilisation and deserves more attention. The modern methods that improve the work of teams creating new projects can also be successfully implemented in other sectors. This work is intended to serve as a source of knowledge and inspiration for people in the IT industry. It is intended to help them systematise what they already know and facilitate the software building process, making it efficient and effective – to the benefit of the user and the producing company.

6. References

A Better Way Of Building Products (2020) Available at: https://www.scrum.org/resources/what-is-scrum (Accessed: 24 May 2022).

Asmo (2018) Uncovering the Agile Manifesto: Finding better ways for developing software, *Zenkit.com*. Available at: https://zenkit.com/en/blog/uncovering-the-agile-manifesto (Accessed: 22 March 2022).

Borowiec, T. (2021) Co to jest Scrum?, *Agileforce.pl.* Available at: https://agileforce.pl/blog/co-to-jest-scrum (Accessed: 24 May 2022).

Bødker, S. and Pekkola, S. (2010) A short review to the past and present of participatory design, *Scandinavian Journal of Information Systems,* 22(1), pp. 45–48.

Brown, T. (2008) Design Thinking, *Harvard Business Review.* Available at: https://readings.design/PDF/Tim%20Brown,%20Design%20Thinking.pdf (Accessed: 20 March 2022).

Elizarova, O. and Dowd, K. (2017) Participatory Design in Practice*, Uxmag.com.* Available at: https://uxmag.com/articles/participatory-design-in-practice (Accessed: 24 May 2022).

Firlit, M. (2022) Professional Scrum with User Experience, *Scrum.org Workshops*, April 28-29.

Gothelf, J. and Seiden J. (2016) *Lean UX: Designing Great Products with Agile Teams.* 2nd edn. Sebastopol: O'Reilly Media Inc.

Jones, M. (2021) Principles for building the Oda UX design team, *Medium.com*. Available at: https://medium.com/oda-product-tech/principles-for-building-the-oda-ux-design-team-905a4808ee87 (Accessed: 27 May 2022).

Joyce, A. (2020) 7 Steps to Benchmark Your Product's UX, *Nngroup.com.* Available at: https://www.nngroup.com/articles/product-ux-benchmarks/#:~:text=Share%20this%20article%3A,testing%2C%20analytics%2C%20or%20surveys (Accessed: 24 May 2022).

Kaley, A. (2020) Lean UX & Agile, *NNG Virtual UX Conference*, 6 December 2020.

Kałamarz, M. (2019) 7 pułapek Design Thinking, *UX Magazyn,* no. 2/2019, p. 58.

Knapp, J. and Kowitz, B. (2016) *Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days*, 1st edn. New York: Simon & Schuster.

Laurie, H. (2013) Panel Studies. Available at: https://www.oxfordbibliographies.com/view/document/obo-9780199756384/obo-9780199756384-0108.xml (Accessed: 24 May 2022).

Moran, K. (2020) How People Read Online: New and Old Findings, *Nngroup.com.* Available at: https://www.nngroup.com/articles/how-people-read-online

Nadel, B. (2020) The most common mistake engineers make during the designer-developer handoff, *Invisionapp.com*. Available at: https://www.invisionapp.com/inside-design/developer-engineer-handoff-mistakes/?fbclid=IwAR3Gp74SVYLaScLrUd6h8z6z4ZOJaU_kmKwbGn12LiwtVKMH49p1ryKvGYU (Accessed: 24 May 2022).

Nielsen, J. (1997) How Users Read on The Web, *Nngroup.com.* Available at: https://www.nngroup.com/articles/how-users-read-on-the-web (Accessed: 24 May 2022).

Norman, D. (2018) *Dizajn na co dzień.* Translated from The Design of Everyday Things by D. Malina. Cracow: Wydawnictwo Karakter.

Pervaiz, M.A. (2018) UX Design for Beginners - The Essentials of UX + Usability, *Udemy.com Online Course*.

Principles behind the Agile Manifesto (2001). Available at: http://agilemanifesto.org/principles.html (Accessed: 22 March 2022).

Psychogios, S. (2014) User Driven Development: The Lean startup principles implemented in enterprise software development, *Blog.inf.ed.ac.uk*. Available at: https://blog.inf.ed.ac.uk/sapm/2014/03/12/user-driven-development (Accessed: 24 May 2022).

Rice, D. (2015) Why 45% of all software features in production are NEVER Used, *Linked.com*. Available at: https://www.linkedin.com/pulse/why-45-all-software-features-production-never-used-david-rice (Accessed: 24 May 2022).

Rohrer, Ch. (2021) User Research Methods: From Strategy to Requirements to Design, *NNG Virtual UX Conference*, 21 February 2021.

Ross, J. (2015) UX Generalists or Specialists?, *Uxmatters.com*. Available at: https://www.uxmatters.com/mt/archives/2015/09/ux-generalists-or-specialists.php (Accessed: 24 May 2022).

Royce, W.W. (1970) Managing the development of large software systems: concepts and techniques. Available at: https://www.praxisframework.org/files/royce1970.pdf (Accessed: 22 March 2022).

Rubin, K. (2012) Envisioning (Planning) Products in Scrum, *Innolution.com*. Available at: https://innolution.com/essential-scrum/table-of-contents/chapter-17-envisioning-product-planning (Accessed: 24 May 2022).

Schmiedgen, J., Rhinow, H., Köppen, E. and Meinel Ch. (2015) *Parts Without a Whole?: The Current State of Design Thinking Practice in Organizations*. Potsdam: Universitätsverlag Potsdam, pp. 100-105. Available at: https://publishup.uni-potsdam.de/opus4-ubp/frontdoor/deliver/index/docId/7996/file/tbhpi97.pdf (Accessed: 24 May 2022).

Sullivan-Hasson, E. (2021) TrustRadius 2021 Women in Tech Report. Available at: https://www.trustradius.com/buyer-blog/women-in-tech-report?fbclid=IwAR0kwLXlH0gUmmtjasbknF25ygOO_9kZGj3gFlGa8IXzhIfWcJmCbs-FdO4 (Accessed: 24 May 2022).

TechTarget Contributor (2017) *Definition: build-measure-learn (BML)*. Available at: https://www.techtarget.com/whatis/definition/build-measure-learn-BML#:~:text=Build%2Dmeasure%2Dlearn%20(BML)%20is%20a%20process%20of,ultimate%20sustainability%20of%20the%20company (Accessed: 24 May 2022).

The Total Economic Impact™ Of IBM's Design Thinking Practice (2018) A Forrester Total Economic Impact™ Study 2018, p. 4. Available at: https://www.ibm.com/design/thinking/static/Enterprise-Design-Thinking-Report-8ab1e9e1622899654844a5fe1d760ed5.pdf (Accessed: 24 May 2022).

Vetrov, Y. (2021) Design Management Masterclass, *Smashing Conference*, Dec 1–15 2021.

What is the framework for innovation? Design Council's evolved Double Diamond (2019) Available at: https://www.designcouncil.org.uk/news-opinion/what-framework-innovation-design-councils-evolved-double-diamond (Accessed: 24 May 2022).

Zabawa, A. (2016) Jak stać się jednorożcem, *Element Talks Conference.* Available at: https://www.youtube.com/watch?v=urhrX17QCpU (Accessed: 24 May 2022).

NTNU

Norwegian University of
Science and Technology