

Mila Lehto

# DESIGN THINKING TOOLKIT FOR UAT



UNIVERSITY OF JYVÄSKYLÄ  
FACULTY OF INFORMATION TECHNOLOGY

2024

# TIIVISTELMÄ

Lehto, Mila

Design thinking toolkit for UAT

Jyväskylä: Jyväskylän yliopisto, 2024, 61 s.

Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaaja: Kyppö, Jorma

Tutkielma syventyy muotoiluajattelun, käyttäjien osallistamisen ja hyväksyntätestauksen teemoihin. Muotoiluajattelun ollessa ajattelutapa, tutkimusta katsotaan muotoilijan näkökulmasta läpi prosessin. Teoriaosuudessa pyritään perehdyttämään lukija muotoilijan ajattelumalleihin ja näkökulmiin muotoiluajattelun suunnittelun ja prosessien kautta. Hyväksyntätestausta tarkastellaan sen määritelmän ja suunnittelun näkökulmasta, käyttäjän rooli huomioiden. Myöhemmin kirjallisuuskatsauksessa tehdään syväluotaus käyttäjän rooliin ja osallistamiseen hyväksyntätestauksessa ja sovelluskehityksessä yleisesti. Tavoitteena on löytää ongelmakohtat ja osoittaa millaisia haasteita ja hyötyjä käyttäjien osallistaminen teknologian kehityksessä voi aiheuttaa. Näihin haasteisiin vastataksaan ja hyödyt varmentaa tutkimus tutkailee käyttäjien osallistamista muotoiluajattelun keinoin ja työkalujen merkitystä tarvittavan tiedon löytämisessä. Tutkielma esittää ratkaisuna hyväksyntätestauksessa kohdattuihin haasteisiin joukon muotoiluajattelun työkaluja, joilla luodaan lisäarvoa hyväksyntätestauksen prosesseihin.

Avainsanat: hyväksyntätestaus, uat, muotoiluajattelu, käyttäjän osallistaminen

## **ABSTRACT**

Lehto, Mila

Design thinking toolkit for UAT

Jyväskylä: University of Jyväskylä, 2024, 61 pp.

Information Systems, Master's Thesis

Supervisor: Kyppö, Jorma

This study emerges in the themes of design thinking, user involvement and acceptance testing in software development. Since design thinking is a mindset it has worked as a lens throughout the process of writing this thesis. The theory presents the principles and processes of design thinking to make readers familiar with the perspective of design, which is very different from the usual in the technology world. User acceptance testing is also viewed through the definition and its design process, noting the role of the user. Later in the literature review, the user's role and involvement in UAT and software development in general is examined thoroughly, pointing out the issues and challenges that are raised, as well as the benefits it can cause when properly implemented. To answer the challenges and to ensure the benefits, the research explores user involvement in design thinking and significance of giving users tools to provide necessary information. The study presents a solution for the issues that are faced in user acceptance testing by providing a set of design thinking tools, that provide value in UAT process.

Keywords: user acceptance testing, uat, design thinking, user involvement

## FIGURES

Figure 1 Double diamond.....	14
Figure 2 Design thinking toolkit for UAT .....	43
Figure 3 Five whys .....	44
Figure 4 Brainstorming .....	45
Figure 5 Personas.....	46
Figure 6 Journey mapping.....	48
Figure 7 Thinking hats .....	49
Figure 8 Roleplay .....	51

## TABLE OF CONTENTS

TIIVISTELMÄ .....	2
ABSTRACT .....	3
FIGURES .....	4
TABLE OF CONTENTS .....	5
1 INTRODUCTION .....	7
1.1 Background for the research .....	7
1.2 Research problem .....	9
2 DESIGN THINKING .....	10
2.1 Mindset.....	10
2.2 Principles.....	11
2.2.1 Human-centric approach .....	12
2.2.2 Collaboration .....	13
2.2.3 Frequent prototyping.....	13
2.3 Process .....	14
2.3.1 Discover .....	16
2.3.2 Define .....	16
2.3.3 Develop.....	17
2.3.4 Deliver.....	18
3 USER ACCEPTANCE TESTING .....	19
3.1 Software testing.....	19
3.2 Defining UAT .....	20
3.3 Designing UAT .....	21
3.3.1 Planning.....	21
3.3.2 Acceptance criteria .....	24
3.3.3 User stories & use cases.....	25
3.3.4 Reporting .....	26
3.4 User involvement.....	26
4 RESEARCH METHODOLOGY .....	28
4.1 Data collection and literature review.....	28
4.2 Research method.....	29
4.3 Analysis.....	31
4.4 Reliability and validity in development research.....	32

5	FINDINGS .....	33
5.1	User involvement in UAT .....	33
5.1.1	The impact of user involvement in software development process.....	33
5.1.2	AgileUAT: A Framework for User Acceptance Testing based on User Stories and Acceptance Criteria .....	34
5.1.3	Problems and Challenges of User involvement in Software Development: an Empirical Study .....	35
5.1.4	User involvement: A review of the benefits and challenges.....	36
5.2	User involvement in design thinking .....	39
5.3	Design thinking toolkit for UAT .....	40
5.3.1	Root cause analysis .....	43
5.3.2	Five whys.....	44
5.3.3	Brainstorming and its variants .....	45
5.3.4	Personas .....	46
5.3.5	Journey mapping.....	47
5.3.6	Storytelling and storyboarding .....	48
5.3.7	Thinking hats .....	49
5.3.8	Roleplay .....	50
6	CONCLUSIONS .....	52
6.1	Research implications and limitations.....	52
6.2	Research recommendations.....	54
	REFERENCES .....	56

# 1 INTRODUCTION

This research ventures within the topics of software testing, design thinking and user involvement. Taking a closer look into user acceptance testing and the possible challenges it creates, the research aims to provide a solution in a form of a design thinking toolkit. The goal is to create a possibility to engage a designer's mindset into the development team and customer's end users to collaborate, creating the most suitable requirements and user acceptance tests for the project.

## 1.1 Background for the research

Testing is a quality assurance measure that provides information about the quality and risks. The goal is to gain a level of confidence that the tested software will provide the desired business value. Testing is a must have continuous act during the whole development process and preferably continues after, with maintaining the software. Testing combines verification, validation and exploration of the software. All these activities answer for different, significant question quality in mind. Verification tells the tester if the system is built right. Validation considers if the system in hand is the right system to build. This action is all about the requirements and business use. Exploration is an act that seeks if the system can be used, not to exclude if the system can be misused. These activities are a base for well-founded decision if the system is ready to go to production and taken in use or not. (Marselis, Geurts, van Veenendaal & Ruigrok, 2020.)

Software testing encompasses various stages, including unit testing, integration testing, system testing, and user acceptance testing (Leung & Wong, 1997). However, despite the efforts of the development team and software testers to ensure functionality, the final determination of whether the application meets its intended purpose lies with the customer. This critical evaluation occurs during the User Acceptance Testing (UAT) phase of the development pro-

cess. (Otaduy & Diaz, 2017.) UAT serves as a means to validate whether the developed software aligns with the business requirements defined by the customer and is suitable for their intended use (Ganesh, Mohapatra, Anbuudayasankar, & Sivakumar, 2014).

Embracing a designer's mindset is a way for businesses to improve their products and services, as well as practice in way of human-centred and business-oriented methodologies. Design thinking emerges as a problem-solving approach with a strong focus on fostering innovation. (Combelles, 2020; Micheli, Wilner, Bhatti, Mura & Beverland, 2019.) As for its highly collaborative nature, involving stakeholders and end users is significant to the approach (Combelles, 2020). Including the user's perspective is critical, as their preferences often diverge from conventional business decisions (Chasanidou, Gasparini & Lee, 2015). While gaining popularity, designer's ways of thinking and working has been acknowledged as a benefit for businesses and organisations. This shift has opened a possibility to widen designers' horizons and contribute professional insights to new fields and business innovations. Not only does it offer possibilities for designers but revitalise creative processes across various sorts of organisations by providing new processes and toolkits. (Tschimmel, 2012.)

Design thinking presents a comprehensive and adaptable framework for addressing complex problems allowing space for experimentation and learning. This is enabled by mindsets and tools that guide the process into finding a way to focus on the needs that require solutions (Plattner, Meinel & Leifer, 2014). What makes design thinking challenging is identifying not just change but the specific nature of the required changes (McCausland, 2020). The problem is viewed from the perspectives of different stakeholders and seen from new angles to create advanced interpretations. A crucial aspect of design thinking is providing tools for flexible, dynamic, and focused processes that drive progress in a constructive direction (Plattner et al., 2014).

Design thinking has been used as a user-centred design approach to engage users in the creation of innovative software solutions. It helps software teams understand users, though it is important to understand how design thinking can be incorporated into software development, along with the techniques and models that can enhance the process. (Lang, Spies, Trieflinger & Münch, 2021; de Villiers, 2022; Parizi, Prestes, Marczak & Conte, 2022.) Research has shown that design thinking is a key factor in achieving success in software development. It is crucial to explore how design thinking can be utilised in software development and identify resources that aid in delivering solutions that meet user needs (Canedo & Parente de Costa, 2018; Parizi et al., 2022). Furthermore, though design thinking to some degree has been utilised in software development, its benefits for UAT have been overlooked and this research aims to seek results.



## 1.2 Research problem

There are a major number of studies that showcase the problems, challenges and issues that are encountered when customer's end users take part in software development (Kujala, 2003; Zowghi, da Rimini & Bano, 2015; Padmini, Perera & Bandara, 2016). Some of these studies are examined later in this study, to overview the issues that are the most critical and/or the most often occurring. On the other hand, design thinking is praised for its ability to provide useful user insights through its collaborative and human-centric nature (Tschimmel, 2012; Norman, 2013; Chasanidou et al., 2015). The study continues the research by exploring user involvement in design thinking and different tools used in design thinking process leading to analyse if those would provide value in the process of user acceptance testing.

Through the analysis of user's role and user involvement in user acceptance testing and design thinking, the study seeks answers to following research questions:

- Can design thinking mindset provide value for UAT process?
- Which design thinking tools are useful when planning UAT?

To gain sufficient understanding of the topics to answer these questions, user involvement must be familiarised thoroughly, in the context of both user acceptance testing and design thinking. The user acceptance testing and its procedures need to be comprehended. Design thinking is a mindset, hence known as a fuzzy concept. This means the study must provide an extensive imagery of its principles and processes, to ensure respectable sense of the entity. The hypothesis in this research is that design thinking tools can bring value in user acceptance testing through the use of evaluated toolkit.

## 2 DESIGN THINKING

In this part of the study the main principles and processes of design thinking will be described. The goal is to provide reader with an understanding of a vague subject, a mindset instead of clearly defined scientific terminology. With this mindset, should the reader view the rest of the study.

### 2.1 Mindset

Design thinking is a mindset. It brings market opportunities. User's needs and customer value are met with the business strategy combined with the tools and methods of a designer. The most convenient use is to create ideas to match users' needs and desires rather than improve existing ones to attract them (Brown, 2008). While gaining popularity, business media has acknowledged designer's ways of thinking and working as a benefit for businesses and organisations. For designers though, this has opened a possibility to widen their horizons and provide professional insight to new fields and business innovations. Not only does it offer possibilities for designers, but new processes and toolkits for any kind of organisation to improve creative processes (Tschimmel, 2012).

Thinking like a designer is a way for businesses to improve their products, services and practice in ways of human- and business-centred methods. Including the user's perspective to this is still critical, as often, want it or not, their preferred path differs from business decisions (Chasanidou et al., 2015). As an approach, design thinking is used in several design methodologies including user interface design, user experience design, experience design and service design. These methodologies apply design thinking in different ways, from holistic service design to detailed user interface design. (Lake, 2016; Hämäläinen, 2022.) Since it is a mindset, there are possibly as many views on design thinking as there are design thinkers, in addition to critics and other academics. (Micheli et al., 2019)

Even though design thinking can be seen as a fuzzy concept (Abrahamson, 1996), it should be viewed through its processes. (Micheli et al., 2019) The processes of design thinking emphasise observation, collaboration, fast learning, visualisation, rapid prototyping and concurrent business analysis by involving end users, designers and business people (Lockwood, 2010). Design thinking can be seen as a problem-solving methodology with high emphasis on innovation (Combelles, 2020; Micheli et al., 2019). At its core is an iterative use of creativity and design methods, to instance problem reframing and early prototyping, where involving stakeholders is important (Combelles, 2020). What makes design thinking challenging is that it is more than managing change, it is discovering what kind of change is needed (McCausland, 2020). Summarised, design thinking enables a participatory, creative process, where solution prototypes are developed and validated as part of the exploration of the solution space (Piras, Dellagiacomma, Perini, Susi, Giorgini & Mylopoulos, 2019).

## 2.2 Principles

Today design thinking can be found in information technology and software development, as a part of agile development. It includes innovation, understanding problem and solution exploration, which makes it suitable for early phases of software development. These functions are a basis for its holistic nature and by bringing stakeholders to the table it defines the value creation on a uniquely high level. Taking stakeholders along into the design thinking process, it aims to attain the best possible end result by continuously exploring various solutions against the identified needs. (Combelles, 2020.)

Design thinking is not a linear process that can be carried out quickly before starting the development process (McCausland, 2020). Chasanidou et al. (2015) as many others have recognised different stages of the design thinking process that investigate the problem through divergent and convergent thinking. Design thinking offers a comprehensive and liberal approach to solve wicked problems in a way that allows failure and experimentation (Plattner et al., 2014). It should be a commitment on all levels of the participating organisation, and it needs to be performed with time and care. It is presented to create empathy due to its human-centric core. (McCausland, 2020.)

For the purpose of design, design thinking can be defined to create meaning and make sense of the service or product to be created (Chasanidou et al., 2015). Choosing the correct tools for the process in each unique situation is important, to ensure effective decision making, communication and accurate results (Chasanidou et al., 2015). This is enabled by mindsets that guide the process into finding a way to focus on the needs that require solutions (Plattner et al., 2014). Tools in this sense can be anything from pen and paper to an application that supports the process, the purpose is to help people to understand and visualise concepts but also gain new perspectives (Chasanidou et al., 2015).

Using design thinking provides a designer's mindset to the problem-solving process (Lockwood, 2010). The problem is viewed from the perspectives of different stakeholders and seen from new angles to create advanced interpretations (Plattner et al., 2014). It is in its core a methodology for innovation and enablement. The main characteristics involve deep understanding of the user, close involvement of people and collaboration, failing quickly and non-linear behaviour (Lockwood, 2010). A significant part of design thinking is to provide tools for flexible, spirited and unstrained processes that aim in a productive direction (Plattner et al., 2014).

In essence, Lockwood (2010) summarises, design thinking involves aligning the designer's mindset with the problem-solving process, serving as a methodology for innovation and enablement. It emphasizes a deep understanding of the consumer/user, fostering empathy in the design process. This approach encourages designers to engage directly with people to discover insights and perspectives. Collaboration with users is essential, accelerating learning through visualization, prototyping, and other methods. Embracing quick failure is advocated, alongside concurrent business analysis to ensure alignment with organizational goals. Design thinking often operates in a non-linear manner, embracing the dynamic and iterative nature of problem-solving.

### **2.2.1 Human-centric approach**

Design Thinking is characterised by its human-centred approach, which translates into its collaborative approach to work and its participatory methods of co-creation (Tschimmel, 2012). It is a human-centric process aiming to solve problems for users with deep empathy (Woolery, 2019; Micheli et al., 2019). In order to design in a human-centred way, it is essential to observe with all your senses and empathise with the user and their experiences (Tschimmel, 2012).

Emotion is at the heart of experiencing (Forlizzi & Battarbee, 2004; Hassenzahl, 2010; Norman, 2013), it is what captivates the user to the product in the first place (Brown, 2008). Emotional system is managing if the experience is positive or negative (Norman, 2013). It enables users to communicate their experiences and gives designers a possibility to understand the needs and desires of users. Emotions create the language of experiencing and fill the gap between the people and the material world. (Forlizzi & Battarbee, 2004; Hassenzahl, 2010.) This means, a good understanding of people and their desires is needed when creating a product, and it can be gained through design thinking (Norman, 2013). In order to get empathy and clarify the project task, observation and in-depth interviews with users, as well as photographs and other visual records, are essential (Tschimmel, 2012). This leads to strong understanding of the needs and desires of users and can also help as inspiration later when the ideation process begins.

With great design, both emotional and functional needs can be affected. Not by simple products, but by creating a complex combination of products, services and information using the means of design thinking (Brown, 2008).

Empathy is seen as a core value of human-centred design (Micheli et al., 2019). Empathizing comes from the core of interaction, it creates understanding of the people, the users of the product (Chasanidou et al., 2015). Design thinkers are perceived as empathic by definition and capable of confronting the world from different perspectives (Micheli et al., 2019). Empathising with a user can be a source of inspiration (Lockwood, 2010). With a human-centric “people first” approach, it is possible for design thinkers to imagine solutions. By taking another perspective of a possible user, their physical and emotional wants and needs can be understood as well as what they find meaningful (Micheli et al., 2019). Designers try to get as much information about their future product's users as possible in order to better understand the underlying problem or task (Tschimmel, 2012).

### **2.2.2 Collaboration**

Besides human-centred thinking, in the core of design thinking is collaboration (Clouin & Janhagen, 2019; Liedtka, 2018). Collaboration covers actions related to people, when the design team and the users work together (Lockwood, 2010). Design thinking does not necessarily require a designer, it is a way to work that encourages anyone to act like a designer and aim to understand human needs (Clouin & Janhagen, 2019). The collaboration and diversity of team members creates a number of functions, perspectives and experiences that are in a great role in the process (Liedtka, 2013). The process thrives within a cross-functional team that embraces different kinds of skills and competences (Clouin & Janhagen, 2019). It allows divergent thinking to explode and convergent thinking to be diminished from multiple different points of view (Liedtka, 2013).

In addition to seeking added value, involving users from the beginning makes it possible to gain user feedback in the early stages of the development process (Lockwood, 2010). Customer co-creation provides a way of engaging users to new business offerings by means of providing prototypes for users to try and test, observing their actions and improving the product or service according to their reactions. It is useful to have several rounds of cooperation to ensure a wide and holistic picture of users' impressions on the product overall as well as its perks and flaws. (Liedtka 2011.) This orientation toward co-creation introduces a distinctly social focus, and emphasis on collaboration that is unique for design thinking (Liedtka, 2013). Cooperative process creates meaning between users and the product, which translates into financial and psychological investment and thus is an undervalued method of creating value (Liedtka 2011).

### **2.2.3 Frequent prototyping**

Utilising human-centred thinking and collaboration, design thinking thrives in early, frequent prototyping (Clouin & Janhagen, 2019; Liedtka, 2018). Through this practice, it is possible to create solutions directly for people's

needs and desires by using designer's methods and tools (Brown, 2008). While designing solutions for the users, it must be noted that each user has their own individual ways of experiencing (Newbery & Farnham, 2013).

Prototypes are quickly created solutions for testing purposes that can be anything from sketches and mock-ups to wireframes or simple codes when the design in process is for software (Piras et al., 2019). Early and frequent prototyping leads to user feedback in the design process, which enables improving the possible solution before it is developed further (Newbery & Farnham, 2013). All elements collected in the design thinking process can be utilised in the prototypes (Piras et al., 2019).

### 2.3 Process

Often the design thinking process is visualised as a double diamond (Fig. 1) according to the British Design Council's model (2015), fitting for reflecting its nature of divergent and convergent thinking. It presents the common activities of the general design process: Discover, Define, Develop and Deliver where the two first phases create a diamond and two latter ones another. Two diamonds refer to the problem and the solution, where by discovery and definition the true problem is unravelled and by development and delivery phases the solution to the problem is generated and refined for delivery (Norman, 2013). The double diamond is nonetheless slightly deceptive in its incremental flow. Design thinking is not a step by step process, but an iterative process changing continuously (Brown, 2008; Rowe, 1990; Woolery, 2019).

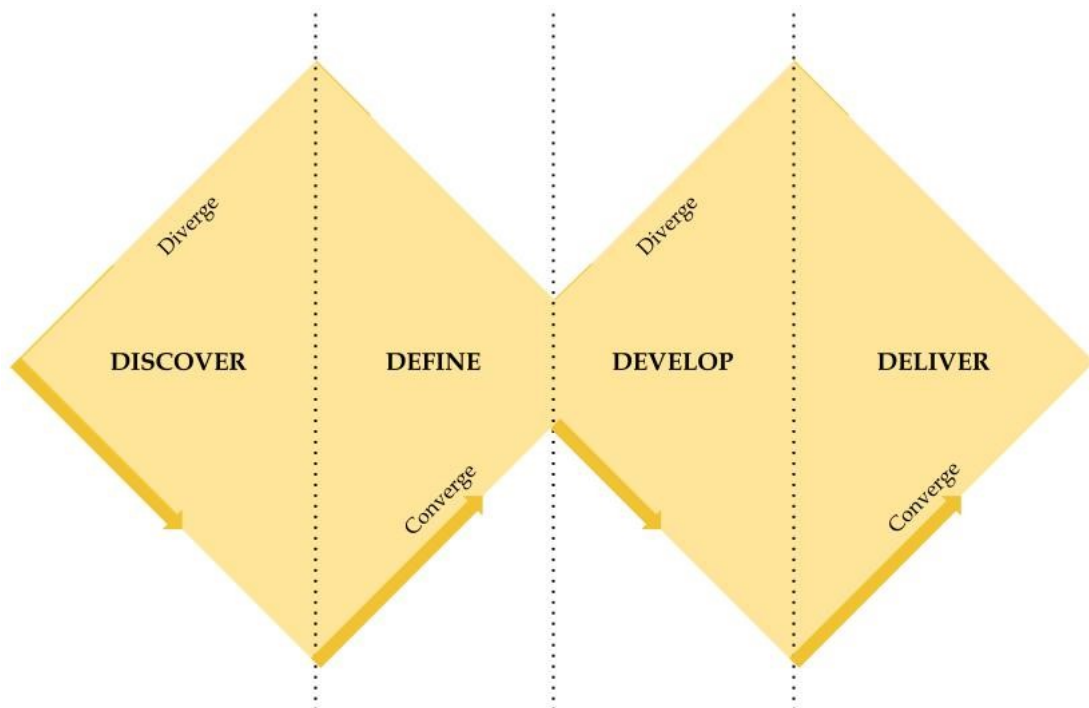


Figure 1 Double diamond

There is not one comprehensive solution to define the continuity and iterativeness of the design thinking process. Every designer has a different kind of specialisation and many organisations have built their own systems that fit the needs of the client and their designers (British Design Council, 2015). Idean (Clouin & Janhagen, 2019) has taken a step further from the Double Diamond and continued with an iterative refine phase to continue the development process, whereas IBM (2018) defines the process as an infinity loop. There is not an all-inclusive correct solution to implement design thinking and the correct tools and methods vary depending on the project and the challenge (Liedtka, 2013). Design thinking is not a linear process that can be carried out in a few days, it requires commitment from all participants (McCausland, 2020). Some studies present design thinking not always sequential, but rather a random process that loops back to earlier stages if necessary. Service design is a nonlinear, iterative process, even though an outline structure can be given, and it is appointed that designing the process is the first step in any process, as it varies from project to project according to the context. (Tschimmel 2012).

The process goes iteratively through the phases of prototyping, testing and refinement (Brown, 2008; Vendraminelli, Macchion, Nosella & Vinelli, 2023). Multiple solutions are tested and improved in the cycle, that actively considers different possibilities, perspectives and constraints aiming for the best solution. (Liedtka, 2013). The process of creating and applying new knowledge is the essence of creativity and innovation, and the composition of a team has a direct impact on this process (Chasanidou et al., 2015). Design thinking exploits stakeholders and aligns for an innovative state of mind through collaboration, to generate a variety of ideas and concepts that in the end lead to the best possible solution (Piras et al., 2019). Design thinking supports innovation through allowing a team to broaden its thinking, allowing multiple inputs for their problem area (Chasanidou et al., 2015). Divergent and convergent thinking may contribute to team collaboration as a result of the design thinking process. A correct definition of the real problem usually results from this creative part of the innovation process.

A relevant aspect to take into account when fostering and stimulating creative inputs in divergent thinking is multidisciplinary of the team (Chasanidou et al., 2015). The team should have an open mind for empathising and feeling the needs and wishes of the end users throughout the process (Piras et al., 2019). When a team lacks open, designer-like experience, searching for relevant information and creating new ideas about the task will give them a better understanding and also balance the lack of designerial experience they may have (Chasanidou et al., 2015). The issue in hand should be opened up from the viewpoint of the user, considering their usual activities and feelings. The ideas will arise from these offsets, from the actual user and the prototypes. The actual end result should be tested by the end user or at least by having the end user mindset on. (Piras et al., 2019.)

Tschimmel (2012) notes that several studies prove there is still a strong emphasis on problem solving within the design thinking movement, but it is

done in a non-linear, holistic way. As a final step in the innovation process, ideas are put into action, convergent thinking is adopted, and innovative solutions are employed (Chasanidou et al., 2015). Nonetheless, the following four phases of the design thinking process are agreed by many researchers and studies. Methods of divergent and convergent thinking are used to expand perspectives, define the issue and innovate solutions.

### **2.3.1 Discover**

Identifying the design problem and elaborating the design framework as well as following the behaviour of a target users in their everyday life create the inspiration for the discovery phase of the Design thinking model (Tschimmel, 2021). The most important goal of phase Discover is to understand the needs of a user. Empathising with users will lead to well understood problem statement that can be solved. Humans are evolved to have a high sense of empathy, which is considered as one of the reasons why we are as successful as species as we are. In the heart of great design is the ability to wield this sense as a strength. (De Waal, 2005.) Within the idea of human-centred design lies the fact that the original idea of a business challenge may differ from what the problem really is. Finding out a stable interpretation of the business problem, as stakeholders might have quite volatile and incoherent views of the problem (Plattner et al., 2014). If this part of the process is overviewed, following through with the solution creation might get tacky as stakeholders can't see the solution fitting the issue that's in their mind.

Whilst working on the discovery phase, the problem should be stated, and users' feelings, wants and needs understood, correctly (Clouin & Janhagen, 2019). When the view on the problem changes, so does the wanted solution and the criteria for the proposals. The design process has to take into account all the different views, especially the main stakeholders, and still leave room to explore and try out totally different and new kinds of solutions that haven't been thought of before. (Plattner et al., 2014.)

When the discovery process is ongoing, it is all about innovation, inspiration, insights - identifying user needs, their feelings and wishes. The issue in hand should be inspected from different perspectives, to see all the sides of it and fully understand it. The methods and tools used for this phase embrace divergent thinking, market research, user research and user stories (Tschimmel, 2012; British Design Council, 2015). The usage of the software in design will be examined focusing on users' desires, perceiving its use, the goal of its use, social contexts and communication (Piras et al., 2019). Within the process a rich knowledge of the issue in hand is gathered (British Design Council, 2015).

### **2.3.2 Define**

All information gained this far is contemplated when the process turns to convergent thinking. As soon as there is an understanding of the users' needs, the



issue should be identified and clearly defined (Liedtka, 2018). When defining the problem, it is time to consider all possible perspectives and try to identify what matters the most. In definition all discovered information should be considered and findings should be reduced to opportunities. (British Design Council, 2015.)

In the phase Define the design team goes through a synthesis process to truly understand what they have found out and learned about the wanted solution (Tschimmel, 2012). The goal is to achieve a framing, clear brief of the fundamental design challenge. Here should be considered the basics, project development, management and sign-off. (British Design Council, 2015.) This way the team can reform the ideas and opportunities to tackle the issue (Tschimmel, 2012). The goal is to have a clear, specific definition of the problem, that is formed by the analysis of ideas and findings in the discovery phase. (British Design Council, 2015.) Visualisation is strongly encouraged within this phase of the design process to paint an easy to understand -picture that all participants can claim (Tschimmel, 2012).

### **2.3.3 Develop**

Design thinking maximises the time spent on the ideation process (Combelles, 2020). When the problem is stated, criticism is banned. There must be a safe space for creativity and an accepting environment for bad ideas to pursue the best solutions. The development phase results in multiple different solutions to solve the problem (Clouin & Janhagen, 2019). Trial and error is a significant part of the process (British Design Council, 2015). The best solutions to solve the problem will be explored further and prototyped. The first models do not need to be perfect, the idea is to fail fast and develop the prototypes (Clouin & Janhagen, 2019). By means of trial and error it is possible to refine and improve ideas. Development process is acted out by brainstorming, prototyping, multidisciplinary working, visual management, development methods and testing. With cycles of processes executed in the development phase, the actual end result is found and specified. (British Design Council, 2015.)

The best ideas are turned into an action plan within the development phase. Among others, Tschimmel (2012) presents prototyping as the core of the implementation, to bring the ideas into reality. New material solutions and ideas will be created into actual products and tested, iterated and improved. This way the design team is able to get a good picture of which of the ideas should be elaborated further and into a final product. Then this will be turned into a communication strategy to make sure everybody is on board with the actual solution.

When moving into the prototyping phase, users are an irreplaceable asset. Their feedback marks a way from prototyping to actual development of the end product (Combelles, 2020). While prototyping ideas, there is no need to try to achieve something close to the end result. The prototypes are carried out iteratively on far-from-finish products, meaning even radical changes might happen

here, redesigns are not excluded (Liedtka, 2018). Prototypes should take only as much time and investment as is crucially needed to get feedback and evolve an idea. Designing the prototypes too far will take back from getting useful feedback. Perfected product is not what prototyping is about, it is pursued for strengths and weaknesses of the idea and possibly see a direction the idea and prototype should be improved on. (Brown, 2008.) Liedtka (2013) summarises it:

*“The function of prototyping in design thinking is to drive real world experimentation in service to learning rather than to display, persuade, or sell.”*

### **2.3.4 Deliver**

Finally, it is time to deliver. The last phase of the design thinking process is utilising convergent thinking and release activities are executed (Tschimmel, 2012). These activities include final testing, approvals and launching, setting targets and evaluating the process. The feedback of the process is collected and shared within the organisation to hand around information and learn. The main objective is to get the designed product to launch. (British Design Council, 2015.) The new product and gained ideas are communicated to the customer (Tschimmel, 2012).

The end result that is going to production should answer to the user's needs, wishes and expectations that were discovered in the beginning of the design thinking process (British Design Council, 2015). To conclude, the delivery phase is dedicated to actions that will ensure the quality of the final solution. The product goes through the acceptance testing and it is signed off to the production. Here it is also important to collect feedback and gain information about the delivered product, but also about the whole design process.

### 3 USER ACCEPTANCE TESTING

The study next observes the theory of software testing, revolving into user acceptance testing. Aiming to provide a comprehension of what is the meaning of testing in general, how user acceptance testing is focused and how it is designed in general. This part of the study provides a brief look into the roles of the user and the development team in the UAT process.

#### 3.1 Software testing

*“Testing is the process of executing a program with the intent of finding errors.” Myers et al. (2011)*

Software testing is a process that aims to ensure the software does what it is designed to do and has no unintended consequences. Software should always be consistent, predictable and reliable to its users. Testing is a way to add value to the software, as it improves the reliability and quality of it. Being able to find and remove errors makes the quality of the software immediately higher. (Myers, Sandler & Badgett, 2011.)

In the past, in the era of traditional software development processes like waterfall and before agile movement, testing was focused at the end of the development process. Today the industry has acknowledged the importance of testing and it is more of a continuous part of software development, accompanying the development from the beginning. This way the highest possible quality at all phases of the process can be ensured. (Tripathy & Naik, 2011.) It could be expected that within noted significance testing would be a well refined science. Though it seems that it is the least known aspect of software development, as Myers et al. defines it, the **“dark arts”** of software development.

The ultimate ideal would be to test all possible scenarios, but it just is not possible. Even the simplest software can have thousands of viable input and output combinations, and it would be very time consuming, uneconomical and

impractical (Myers et al., 2011). In today's world where our whole lives are managed by technology, expectations for software are continuously getting higher and higher. This means more complex software and from a testing point of view, more test scenarios. To have a successful and efficient testing process, tester raises into an elevated role. A good tester has an attitude, a vision that drives them and relies on the assumption that all the software has errors and their responsibility is to find as many as possible (Myers et al., 2011).

The definition of Myers et al. (2011) presented above has numerous implications. It can be read that testing is in a way a mean or even sadistic process that monitors mistakes of others, usually developers. This does not only make testing hard for many as human nature may reach for a more constructive outlook on life, but also has an effect on the relationship between tester and the rest of the developing team.

Software testing can be divided into several phases: unit testing, integration testing, system testing and user acceptance testing (Leung & Wong, 1997). However, as much as the development team and especially software testers ensure the developed features work, it is the customer who makes the last call if the application works as intended (Otaduy & Diaz, 2017). This is where the development process runs into the User Acceptance Testing (UAT) phase. UAT is an essential part of software development, but some may ignore its significance due to lack of understanding. The UAT phase is in the latter part of development and has the power to fail software that has been built for a long time, if the user requirements do not match the developed product (Padmini et al., 2016).

### 3.2 Defining UAT

User acceptance testing comes topical once all system level tests are satisfactorily performed (Tripathy & Naik, 2011; Ganesh et al. 2014; Poston, Sajja, & Calvert, 2014). Here the end user has the possibility to do trial runs in the software with strong support of the developing team and see if it works as planned based on the documentation and acceptance criteria (Otaduy & Diaz, 2017; Tripathy & Naik, 2011). Acceptance criteria is the first glimpse of testing in the whole software development process, as it should be a part of the contract between the customer and the supplier and define clearly the customer needs and expectations for the software (Tripathy & Naik, 2011). It is beneficial to use the software as a means of communication and basis of discussion, so that unnecessary misunderstandings can be avoided (Otaduy & Diaz, 2017). In UAT, key test areas cover the major functions, user interface, and capabilities in handling invalid input and exceptions in operation (Leung & Wong, 1997).

User acceptance testing is a formal part of delivery that aims to determine whether the system meets its acceptance criteria and gives customer a validation on if they should or should not accept the software (Tripathy & Naik, 2011; Otaduy & Diaz, 2017; Pandit & Tahiliani, 2015; Mohd & Shahbodin, 2015). User

acceptance testing is used to verify the software has all the functionalities required (Ganesh et al. 2014) and evaluate the readiness of the operational use of the software (Leung & Wong, 1997). It establishes confidence in the user that the purpose of the software is matched (Pandit & Tahiliani, 2015) and identifies what the software does and how it benefits the user before going live (Mohd & Shahbodin, 2015).

UAT is used to validate if the developed software meets the business needs the customer has set and is fit for their use (Ganesh et al. 2014). In contrast to system level tests that are executed in a laboratory environment, user acceptance tests are performed in the operational environment (Tripathy & Naik, 2011). It is recommended for the development team to have one or more dry runs through the software before moving forward to the UAT to confirm the environment is up and running and to reduce the issue resolution time (Ganesh et al. 2014).

### **3.3 Designing UAT**

Test design is a process within UAT where general objectives of testing are turned into actual test conditions and test cases to form a plan on how testing is performed and accepted (Hambling & Van Goethem, 2013). In a big picture user acceptance testing has three major goals: confirming the software matches the given acceptance criteria, identifying discrepancies and determining the software's readiness to live action (Tripathy & Naik, 2011). It is crucial to understand what the current state of the systems in the customer's organisation is, whether the new software involves significant changes to the way users work, or if it is the first IT system in the organisation. This will allow the development team to make a smooth transition to the new system if they are not familiar with the way users work in the organisation (Hambling & Van Goethem, 2013).

UAT must ensure that the system is capable of handling the way users are expecting to work. The process has to be planned meticulously to describe exactly what is needed to do, when, how and by whom. This plan needs to be communicated to the people involved so that their contribution is ensured. Plan should also include an analysis of things that might go wrong and how these situations are handled to avoid possible delays and uncertainties. It should answer the questions regarding requirements and what kind of testing should be done. (Hambling & Van Goethem, 2013.) UAT plan should be considered in an early stage of software development, to ensure that the business requirements are considered during the development.

#### **3.3.1 Planning**

The importance of the UAT plan lies in the accuracy: with a complete plan it can be ensured that right things are tested in a correct way. Level of detail and

necessary UAT planning activities are decided based on the project and its complexity (Hambling & Van Goethem, 2013). The user acceptance testing differs from the system level testing by its definition, it is done by users. These users are customer representatives and often not from technical backgrounds. This in mind, it is noteworthy to recognise the skillsets of the testers, thereby the user acceptance test plan needs to be very simple and easy to understand. The acceptance test plan stands in the corner of the customer, it is created to confirm customer's requirements are met instead of passing a heavy set of system tests. (Tripathy & Naik, 2011.)

The main goal of test planning is to get ready for the test execution. The plan should give tester a framework to work within, and present the scope, resources, effort, schedule and budget. The framework draws a picture of the ideas, facts and circumstances that will then be conducted into the test scenarios and test cases. (Tripathy & Naik, 2011.) In the UAT plan it should be determined what is the timeline for the execution and who are the customer's representatives running the execution. Test scripts and test cases are created by the identified representatives, whereupon mentioned should be discussed on the managerial level (Ganesh et al. 2014). With well refined scope defined with the customer, the managerial outlines are established for the testing (Tripathy & Naik, 2011).

Test design requires a deep dive into the requirements, identifying the system features to be tested and defining the objectives and behaviour of the test cases. It extends beyond crafting individual instances of test conditions, cases, and scripts; it encompasses orchestrating the entire process of generating these elements in a manner that is both efficient and effective (Hambling & Van Goethem, 2013). Test objectives are drawn from the requirements and the functional specifications, and these objectives will function as frames or goals for test cases (Tripathy & Naik, 2011). Objectives of testing may vary depending on the tester's point of view. It can be something from its ability to work or not to work, reducing risk of failure or reducing the costs (Tripathy & Naik, 2011). Tests formulated solely by others are likely to lack this distinctive end-user insight, undermining the efficacy of UAT as a risk reduction tool (Hambling & Van Goethem, 2013).

The best source for accurate requirements and objectives is naturally the users of the system. Usually expectations are gathered by a semi-structured interview, where the set of questions remains the same for all, but the user has freedom to expand their answers and express their concerns and wishes (Hambling & Van Goethem, 2013). User acceptance test plan is made by end users or third-party testing professionals with thorough understanding of the case to ensure the software matches the customer defined requirements and acceptance criteria (Tripathy & Naik, 2011). Either way, in the test design phase, especially during the formulation of test conditions and cases, collaboration with end-users is essential (Tripathy & Naik, 2011; Hambling & Van Goethem, 2013). Hence it is essential for a business representative to approve the requirements, test conditions, and cases as complete and accurate, ensuring the involvement

of the unique end-user perspective crucial for effective UAT (Hambling & Van Goethem, 2013). For each objective there may be one or more test cases that are clearly specified and easy to understand and reused (Tripathy & Naik, 2011).

Once acceptance criteria are set, an early version of acceptance test plan can be produced. In this point, the user acceptance test plan is a holistic tool to see a full picture of the final product (Tripathy & Naik, 2011). UAT strategy that defines the objectives of the testing and on what level tests should be performed is used as a base for UAT plan (Hambling & Van Goethem, 2013). The user acceptance test plan documents the overall plan for user acceptance testing and description of specific tests.

Tripathy & Naik (2011) present a structure of an acceptance test plan that consists of four parts: introduction, acceptance test category, schedule and human resources. In the introduction there is typically presented the name of the project, revision history, terminology, approvers and approval date, an overview and references. It is a section where the basic facts of the test plan are visible. The parts of schedule and human resources count in how much resources the user acceptance testing process requires from the customer. User acceptance testing is not supposed to be a heavy and exhaustive process, and it might be implemented relatively fast. Human resources should name the users that will perform the user acceptance testing and their specific roles, noting some of the resources are required from the development team. Some of the responsibilities like preparing the test environment, upgrading the software and setting up the networks need know-how in the operational environment and business operations. Acceptance criteria are shown in the second part of the acceptance test plan, which introduces the operational environment, as in the preparations for the environment to carry out the testing, and test case specifications for each acceptance criteria. The specification identifies the test case ID, test title, test objectives and test procedure.

When entering the UAT phase, as Hambling & Van Goethem (2013) note, the original requirements and objectives from the UAT plan should be paid another visit and adjust them if necessary, since at this point of the development process the expectations can be matured and the first version of the requirements might not be realistic or practical in every case. If expectations are updated with new ones that were not originally part of set requirements, the development team should point at this moment if those will be feasible and achievable in a given time frame. This should not be seen as an opportunity to rewrite the requirements from the start, but to tweak and direct the focus. Asking support from users and giving them an opportunity to raise their views can be an important driver to commitment in the process. By cooperating with the customer and the actual users of the system, there should be a reasonably comprehensive understanding of what is needed and what are the expectations. This information should be recorded as user stories and use cases, and that will work as a basis for building UAT. If the original requirements do not align with the contained information at this point, it should be considered to convert all knowledge into a common format.

By the time the project is entering UAT, the users from the customer's side should be well prepared and aware of what is going to happen after the system is operating. Training should have begun and to ensure that transition to UAT happens smoothly, but if the customer's side is not prepared, there is a bigger risk for hiccups. (Hambling & Van Goethem, 2013.)

### 3.3.2 Acceptance criteria

What criteria the software must meet in order to be acceptable? Acceptance criteria is a set of requirements that define if the software has met the acceptable criteria (Tripathy & Naik, 2011), and hence testing can be stopped, and system released (Hambling & Van Goethem, 2013). The UAT acceptance criteria are used to verify the software is ready to be taken into the production. The development team can not make decisions, but it is crucial to provide expertise and support to define a proper set of criteria. Agreed by the customer's representatives and development team together, the software must meet this pre-set requirement to be defined as ready. (Leung & Wong, 1997; Hambling & Van Goethem, 2013).

Acceptance criteria is to be thought about long before entering the UAT, so that it is commonly understood what the importance is and meaning of each criterion (Hambling & Van Goethem, 2013). Rather than treating the requirements document as a separate entity, it should instead serve as the driver influencing all aspects of software development, including progress tracking, task assignment, quality assurance, and information management (Park & Maurer 2008). This enables data collection during the development mirroring the acceptance criteria to be met (Hambling & Van Goethem, 2013). The tests establish the criteria for an acceptable system from the perspective of the customer, potentially serving as a contractual agreement between the development team and the business stakeholders (Park & Maurer 2008).

Because development is always a unique case, there is no simple formula to create perfect criteria, nor is there a guaranteed outcome (Hambling & Van Goethem, 2013). It is preferable for acceptance criteria to be measurable and even quantifiable. The basic principle is to create acceptance criteria that aim at acceptable software quality (Tripathy & Naik, 2011). Acceptance criteria establish the scale and the scope of user acceptance testing. It should be set based on the desired functionality of the software. While setting the acceptance criteria, it should be considered what is the user's expectation on the software and what it is meant to do. At the end of the software development process it will be decided if the software is ready to be deployed based on the user acceptance testing and specifically if the acceptance criteria is met. (Ganesh, 2014.)

Quality is not a simple concept. It is a complex feature that every person has their own take on. In acceptance criteria, it is the customer's opinion that leads the way. There are, however, views proposed to present characteristics of quality. Transcendental view expresses quality as something that can be recognised, but not defined. The view of the user has a satisfactory element on the



quality, while the manufacturing view focuses on conforming the specification. The product view conjoins the quality with inherent characteristics and the view of the value, as assumed, sees numbers and the cost of the development. (Tripathy & Naik, 2011.)

### 3.3.3 User stories & use cases

User stories and use cases can provide an easy way to understand the requirements that might end up being more technical and complex (Pandit & Tahiliani, 2015; Ferreira, da Silva & Paiva, 2022). User story is a compact sentencing of everyday language that covers the user, what they want from the system and why (Hambling & Goethem, 2013). Nevertheless, it contains the description of the technical information on how the software should work for the development team to create the solution for the software. User stories provide the basis for quality delivery (Ferreira et al., 2022).

Hambling & Van Goethem (2013) define user stories and use cases in the context of user acceptance testing. Contrary to what could be expected, user stories are not actual requirements, but a way to explain some of the user expectations. Capturing user stories through actual users of the developed system is a relatively fast and consistent way to figure out the intended use of the system. Using time with as many users as possible results in better understanding and improved UAT process. User stories are most commonly known to be used with agile methodologies to gather requirements but are valuable means of expressing wanted outcomes while designing development projects and its testing phases.

Use cases capture current processes and work as a bridge between requirements stated by users and a more technical overview of the issue (Hambling & Van Goethem, 2013). Use case describes the interactions between the user as an actor and the system aiming at a particular outcome and developers aim to provide a code to match these requirements (Pandit & Tahiliani, 2015).

When working with use cases, edge cases, negative outcomes and other rainy-day scenarios have to be taken into account. These will tell if something goes wrong or unexpected variables turn out. It may not be necessary to write totally new cases for each, it is possible to present alternative paths of action as separate steps below the primary use case. In good practice, sunny days, the desired use case should be reported first, and edge cases after that. Also, the most significant cases should be written first, importance decreasing in order. In order to determine whether the most likely important scenarios have been covered, the use cases and edge cases can be circulated around the UAT team. The business then decides how much time and money should be spent on testing scenarios that are less likely to occur. (Hambling & Van Goethem, 2013.)

### 3.3.4 Reporting

The final outcome of the user acceptance testing should be a test report that states the acceptance conditions (Tripathy & Naik, 2011; Ganesh et al. 2014). There are three possible results for the user acceptance testing: system is accepted as delivered, system is accepted after modifications or system is not accepted at all (Tripathy & Naik, 2011).

The acceptance testing progress and status of test cases is reported on a daily basis at the status report, which will be concluded in the acceptance test summary report. This summary is used as a tool to communicate the progress of testing with relevant parties, like the management. The summary should state the progress as it is reported in the status report, differences between the planned and actual performed testing and overall evaluation of the situation. In the end, there should be stated the user acceptance testing point of view on the acceptance of the software. (Tripathy & Naik, 2011.) Reporting defects is crucial for the success of the release. Those can be identified already in system testing before moving to the UAT and during it by the customer's users. Fixing defects means more changes to the codes, which leads to more tests. Unsolved defects will affect decision making when it is time to either accept the system and go forward with the release or not. (Hambling & Van Goethem, 2013.)

Ganesh et al. (2014) list similar content for the report. In their research it is stated that the UAT report should contain a summary of the process, actions and assignments related to changes in procedures, highlights, action plan and schedule for completion, test scripts and a list of open technical issues. When all of the components pass the user acceptance testing, there should be a formal certification from the client collected. The software can be released for production when the results meet the acceptance criteria (Leung & Wong, 1997).

## 3.4 User involvement

Many studies add a great emphasis on the performing party of the user acceptance tests (Ganesh et al. 2014; Tripathy & Naik, 2011; Otaduy & Diaz, 2017; Pandit & Tahiliani, 2015). The supplier's development team is in a responsible position when it comes to user acceptance testing. It is on their plate to make sure the customer understands the usage of the built software and they coach the customer's representatives on it (Tripathy & Naik, 2011). And still, it should be noted that running forces are dedicated representatives from the customer, the actual users of the software (Ganesh et al. 2014; Leung & Wong, 1997). Even though testing is performed by the customer's representatives, often there is a developer or testing professional from the supplier providing technical advice on the process and test case creation (Leung & Wong, 1997). If and when there are issues, errors, defects, faults, discrepancies, you name it, in the course of user acceptance testing, they work side by side with the customer to coordinate

the fix and resolve the problem. (Tripathy & Naik, 2011; Ganesh et al. 2014) User acceptance testing is executed by the customer's users, but in reality, it is a cooperation between the development team and customer's testing resources (Tripathy & Naik, 2011; Leung & Wong, 1997).

During UAT it is necessary to gather information from the end users to notice possible bugs that might appear in the developed software. As a collaborative process, it depends on the close collaboration between the developing team and the customer's representatives, even if the UAT phase is processed internationally. Real end users bring knowledge and experience about process flows and daily use of the system, and that is an irreplaceable asset. They know where potential risks hide. (Poston et al., 2014.)

## 4 RESEARCH METHODOLOGY

This research is formed in the shape of development research. The first part of this chapter is to introduce the data collection methods and shortly provide a description of the literature review. Then, the development research is portrayed, and it is explained in detail how the study is built up, how its trustworthiness can be assayed and how the results will be analysed.

It should be acknowledged that the research process, though it might seem like a straightforward continuum, could throw some curveballs. It should be noted that there are risks involved, it might be that not always the data collection goes as planned, nor is the hypothesis correct. The challenges encountered during this research are discussed later in the conclusions.

### 4.1 Data collection and literature review

After workshopping through ideas for this thesis, the actual research was started with data collection. There were two main topics to familiarise with: design thinking and testing. By gaining more knowledge of the topics it was pretty obvious that the original topic was too wide and should be defined more condensed. With the researcher's gained expertise, it was decided to confine the research on user acceptance testing, instead of testing in general. The main sources of information in the data collection were Google Scholar and the library database of University of Jyväskylä. As the researcher has professional resources for both testing and design thinking, also literature and research published by Capgemini (referred as business units Idean and Sogeti in references) were part of the source materials.

As mentioned above, the research began with researching testing and design thinking. In the beginning search keys were testing, software testing, design thinking, service design, user experience design, but was later refined to include user acceptance testing, acceptance testing and UAT. The search results from these searches were mainly used to provide context in the theory, explain-

ing the main topics of the study. In the research-aimed literature review, more defined search phrases were used, including 'user involvement in software development', 'user involvement in user acceptance testing' and 'user involvement design thinking'. For design thinking tools keywords 'design thinking tools', 'service design tools' and direct names of some tools were used. With these keywords the research aimed to create a comprehensive understanding of the topics that later in this study will be examined in depth. The goal of the research was to elevate the process of user acceptance testing by means of design thinking.

Literature review is a significant part of any research. In this research literature is divided clearly in two phases. First the study builds a strong foundation by introducing the topics, providing insight into their nature and use in action. Later in the study there is more specified review into the literature, focusing on the user involvement in provided topics. In this part it is explored how involving users in the software development process and UAT is affecting the project, what possible challenges it raises and what should be done. On the other hand, the literature dives into design thinking, how users are involved in those processes and what kind of tools are used in the process. Based on this knowledge the research suggests a set of design thinking tools that could be useful in the context of UAT.

## 4.2 Research method

Development study can not be compared to qualitative or quantitative research by their age, it is rather young to be a research method. It has started to root in the scientific world in the early 2000s. For its young nature it is only natural that there have been a multitude of different names for it. Studies have referred to similar kinds of practical studies as development research, design research, design-based research, design experiment, developmental research and user-design research (Juuti & Lavonen, 2006). For clarity, this study has used development research as a terminology, not to mix it up with the topic of the study, design thinking.

The purpose of development research is not to create generalisations or a simple understanding of the phenomenon under study, but rather to create a change in the studied phenomenon (Kananen, 2012; Vaughan, 2017; Joore, 2022). It reaches to unify the methods of design and research to create new information (Joore, 2022). This study is executed as a development study to create a concrete design thinking toolbox for UAT professionals to benefit from during user acceptance testing. Thus, for example, qualitative or quantitative research were not possibilities, as those methods aim to understand the phenomenon instead of creating a change. The literature review will study the reasons to improve the UAT process and provide a motive for creating a ready-set design thinking toolkit intended just for UAT.

Development research is a complex research methodology that is lacking a singular, straightforward definition. Edelson (2002) explains that the integration of development and research characterises development research: in this approach, the development process is regarded as an opportunity for forming theory and the design of practical products. So, to be said, through a cyclic process where development and research progress iteratively, both theoretical constructs and the developed product can be refined. Edelson (2002) argues that developmental research is grounded in problem analyses related to domain-specific knowledge, the development process, and product design. Consequently, development research produces three categories of insights: domain-specific knowledge, knowledge of the developmental process, and understanding of the resulting product. Juuti and Lavonen (2006), however, argue development research can be characterised by its inherently iterative nature that stems from the need for change and its ambition to reach a widely applicable outcome, generating useful information related to the topic.

As development research is also known as design research, it has been studied from the point of view of designers. Design research is not only to solve problems, but find the correct problems (Vaughan, 2017). Joore (2022) compares the viewpoints of researchers and designers, stating that researchers focus on current knowledge when designers direct in developing new possible futures. By design research these goals are combined, and current challenges are met with desired outcomes. Vaughan (2017) notes that through design, research is capable of lending ideas, concepts and methods for providing more accurate results. They also suggest that by stretching theories, it is possible to gain further information to resolve the research problem.

As stated by Barab and Squire (2004), development research isn't merely an approach but rather a collection of methodologies aimed at generating new theories, products, and practices that can influence in authentic contexts. They argue that developmental research diverges from simple experiential development and formative evaluation due to three main factors: 1. development interventions are consistently tied to theory, 2. developmental research generates new theory rather than solely testing existing theory, and 3. the ability to address certain research questions uniquely within a development research framework, which may not be feasible through traditional laboratory setups.

Development research consists of two processes: research and development work. Even at the beginning of development research, the researcher must gain an understanding of the phenomenon under investigation and its problems, and until then, development research is qualitative research. (Kananen, 2012.) In this thesis, the researcher gained an understanding of the phenomenon through a comprehensive literature review. The development work consists of defining and investigating the problem, analysing the reasons that led to it, and proposing solutions (Kananen, 2012). The research turns into development research when the knowledge gained from literature review is taken into a deeper level and the design thinking methods are considered as part of the UAT phase.

### 4.3 Analysis

This research is based on literature review and development research, thereafter the chosen method for analysing the collected information is content analysis. Content analysis is the applicable option for analysis, when there is a need to gain a meaning for the collected data (Schreier, 2012). Content analysis can be seen as a high-level construct for analysing multiple varying written, heard or seen subjects or finely defined research methods (Tuomi & Sarajärvi, 2009). Content analysis is a methodical, unbiased, and quantitative examination of data insights (Neuendorf, 2016; Krippendorff, 2018). In this study content analysis is understood as a defined research method to support the literature review in this development study. Based on the results of the analysis, further deductions are drawn, and the study carries forward on presenting solutions on how to respond to those.

It is an analysis method that has existed in research literature for ages, originally used to make well-founded deductions on the basis of various literary materials (Weber, R. P., 1990). In content analysis, instead of taking all provided information into account, only the relevant data is processed to answer the research questions (Schreier, 2012). Content analysis in its existence has resulted in eclectic and even surprising results and due to its combinable features, its application is only limited to researcher's imagination (Neuendorf, 2016). Content analysis lies entirely on the way the information is conceived (Krippendorff, 2018).

The content analysis in this research is compiled by optimising the chosen literature in data collection and analysing the studies that were chosen. In the data collection the analysis prioritised the studies that were closest to the topics of this research, peer evaluated and valid by age. The topic and offerings of the research were prioritised over the age, enabling more significant input as part of the study.

The literature review was analysed by making conclusions based on the studies, highlighting the information that was meaningful in the scope of this research. Conclusions were drawn from the results of the chosen studies, providing a basis for the hypothesis of this study. Recurring themes were identified, and the end result of the study was built based on those. This is a qualitative content analysis with some characteristics of a quantitative one, as the repeated themes were taken into account and thematic patterns were looked into. The analysis combines different techniques of content analysis, narrative analysis is used to understand the relations in UAT and design thinking, when semi-otic analysis is utilised to gain comprehension on the structures and meanings behind the words.

#### **4.4 Reliability and validity in development research**

Reliability refers to the extent to which measurements are free from error and consistently produce the same results (Thanasegaran, 2009). In development research, almost the same reliability criteria apply as in qualitative research work. There are no differences between the two studies when it comes to evaluability. The only difference appears if quantitative questions are used in the development study. At that time, the reliability assessment of the quantitative research must be taken into account. (Kananen, 2012.) Reliability is another of the quality verification methods used in quantitative research. This means that if the same study is repeated later, its results will match the results of the previous study. The problem with reliability in development research is that development research aims at change. In this case, the results between the two studies cannot match.

Validity is the other way of verification. Validity aims to evaluate if the research implemented responses to the set research problems (Thanasegaran, 2009). This means researching the right things and correctness of interpretation. Validity takes into account the scope of the sample and measures the generalizability of the research results. (Kananen, 2008; Kananen, 2012.) Effectively valid studies require the integration of multiple sources that cover time on a wide scale. Content validity yields insight if the studied material is meaningful to the research. (Thanasegaran, 2009.)

In this research some sensitive spots in terms of reliability and validity included the data collection and search engines, but also the novelty and neutrality of the source materials. True to the nature of development research, the study aims to change, hence the usual credibility issues from qualitative research were not faced. Further analysis of the credibility was contributed to the conclusions of this study.



## 5 FINDINGS

At this part of the study, the findings of the development research are provided. The first part of this chapter dives into the UAT literature, aiming to understand what are the challenges and issues of user involvement that are faced in the development process. Few key studies that research the subject are presented to highlight where there could be need for new solutions. Then, the study looks into the user involvement in design thinking and provides a light outlook on how design thinking has been used in software development thus far. After these topics are processed, the study moves forward into the analysis, and based on the results an intended design thinking toolkit for UAT is presented.

### 5.1 User involvement in UAT

While the technology develops and gets more complex day in day out, UAT gets more complicated to perform. Software development theory generally suggests including users in the development process since the beginning, throughout the whole cycle. Yet, it is not that simple and is often found difficult. Research highlights the importance of user involvement and notes the fact that there is a lack of understanding on how to maximise the potential of engaging users in the development process and UAT. (Poston et al., 2014). Not to mention, often the development team and the customer's representatives, users, don't really speak the same language. Even if they speak the same language, the professional slang might differ.

#### 5.1.1 The impact of user involvement in software development process

The main challenge Saif, Almohawes and Jamail (2021) recognise in software development is when software developers do not understand the business needs, so forth the actual desires of the customers are not met. Accordingly, user involvement brings significant value and breaks barriers between the tech-

nical side of the software and business requirements. With proper communication with users it is possible to gain accurate information for designing the software, as they are experts in their area of work. This means, they are the best to know what is needed from the solution and by cooperation with the development team, it is possible to create better software that suits the business needs. Involving the user is an effective approach that results in achievable and clear system objectives. When these objectives are developed, they positively impact the user's environment and deliver the requested solution.

In software development people come from different backgrounds to achieve a common goal. Saif et al. (2021) suggests that all parties should be involved and required to carry responsibility to collaborate, share knowledge and expertise. Successful software development necessitates teamwork, effective communication skills, and the ability to obtain information from diverse sources. It also requires continuous support and collaboration among all parties to identify issues and make adjustments until the objective is met. The effectiveness of a software solution largely depends on the user's interests, input, and the existing problem domain. The software development process involves sharing and unifying ideas, and sometimes prioritising them, making user involvement essential throughout the software development life cycle.

If the development team is not used to cooperating with business people, it might demand emotional and physical effort which might lead to low energy levels and worsened relationships with customers. Therefore, user involvement might lead to a situation where the user is involved in the process, but they complicate the development process which affects negatively in the end result. From the customer's point of view, successful user involvement requires commitment. The required tasks might not be familiar to the user and it can appear difficult and time consuming. The user requirements are supposed to be discussed with the whole team to gain a common understanding between technology and business. Collaboration results in addition to better products, in enhanced team spirits and improved processes. (Saif et al., 2021.)

### **5.1.2 AgileUAT: A Framework for User Acceptance Testing based on User Stories and Acceptance Criteria**

There are various aspects that may form as challenges in user acceptance testing. Padmini et al. (2016) acknowledge business, people and process, governance, and tools and automation as such features. Their study had business users engaged in the UAT execution. Since customer's representatives were business experts, there was a lack of IT skills and understanding of UAT. Delays were occurring, and they were not achieving the set targets. As a result of missing IT understanding, reading system test cases was difficult for customer representatives and they did not get any value from existing ones and were to write their own scenarios from the beginning. They also ran into resistance and incompetence by the customer's side when using shared location within the whole project team.

It was noticed by the test management during the UAT phase that there is a need for an update in the current UAT process. Their research suggests that to enhance the coverage and efficiency, UAT work should be done in cooperation between the development team, testing experts and customers. Where the customer has business and domain understanding but lacks IT skills, the development team has a strong understanding of the software but limited domain expertise. Customer's presence is seen as necessary from the early stages of the software development process and needed also through the lifecycle to reduce failures. (Padmini et al., 2016.)

Padmini et al. (2016) worked through the problems by implementing workshops, surveys and interviews to track down the existing issues and to solve these. They improved team capabilities, confidence and knowledge about testing and defects. Teams were made aware of their roles and responsibilities and value and importance of their contribution. They received purely positive feedback about this kind of approach. Making such changes increased the attendance rate and positive attitude towards the UAT, even though it was initially hard to convince them, especially senior employees, of the new way of working. It took some time and clear communication to get there.

### **5.1.3 Problems and Challenges of User involvement in Software Development: an Empirical Study**

Zowghi et al. (2015) too recognise it is not direct or simple to draw a connection between user involvement and a successful system, as there are so many influencing factors. Though they have shown in previous studies that 68% of empirical studies show positive results of user involvement, whereas 32% state negative or uncertain results. This explains that user involvement is in a way a double-edged sword which without proper structure and management may cause more harm than good. They noted challenges in user-related aspects, communication, management and project-related issues. In their study there were recognised the most weighty problems to be communication and misunderstandings between users and development team, which lead to a number of issues in wide variety. Furthermore, lack of motivation seemed to be a top challenge that raised problems concerning attitudes and behaviour of users, such as not taking given tasks seriously, ignoring development team's requests, reluctance to participate in activities.

Users' commitment to software development is seen as a deeply influencing problem in user involvement (Zowghi et al., 2015; Padmini et al., 2016). Often in software development projects users' unrealistic expectations cause issues, especially when decisions are made before or without user involvement. To avoid unmet expectations and desires, effective communication and building common understanding lead the solution. This should begin from early stages of the process, when the requirement specification documents are built. These can be later used and projected in the UAT. As Zowghi et al. (2015) proved in their research, when requirement specifications are not done with

proper commitment from customer's business users, later in the UAT phase they are keen to raise issues that are never agreed in the scope of the project. They are not involved when the planning takes place, but when it is time to deliver, they are expecting things that were not planned and hence not delivered. How to get key users motivated in developing systems is a research question to be raised from these discoveries.

When there are issues about people, those are always in some way connected to feelings. Not having specific IT expertise might cause feelings of inadequate or intimidated, which shows up in resistance to commit to the project. This has an effect on UAT as well, users found it difficult to test the software because it was new to them. As a solution they mentioned a more agile way of working could work and reduce frustration, mistrust and fear. (Zowghi et al., 2015.) There should be a way for development team and business users to communicate and build a common ground of trust. This would ease the feelings of deficiency and improve the means of communication, resulting in a more committed ensemble in UAT.

#### **5.1.4 User involvement: A review of the benefits and challenges**

User involvement is seen as a principle in development of usable systems, when user-centred design and users' needs are set in focus. This means user involvement in the development process to create better understanding of the requirements. (Kujala, 2003.) Kujala (2003) highlights that in their research outlook to literature indicates that user engagement typically yields positive outcomes, particularly in terms of user satisfaction. Moreover, there is some indication that prioritising users as the primary source of information is an effective strategy for capturing necessary requirements. Nevertheless, there are challenges in it as well. Choosing the correct user roles to involve in research must be considered carefully. Also figuring out more cost-efficient ways to gather the implicit needs of the users is a raising concern.

In the study Kujala (2003) aimed to enhance comprehension of early user involvement and its practical significance. User involvement encompasses a broad spectrum of approaches, each with its own set of potential benefits and challenges. Their research endeavours to identify the principal approaches and examine their respective advantages and obstacles. Understanding users and their requirements is crucial in the initial stages of system development, but how can user involvement reveal informal and non-verbal cues from users? They propose that the primary methodologies encompass user-centred design, participatory design, ethnography, and contextual design. Each of these methodologies includes an explanation for involving users and a set methodology for doing so. User-centred design aims at creating products that are both useful and usable. Although there is no universally agreed-upon definition or process for it, the principles outlined by Gould and Lewis back in 1985 are still widely accepted, which emphasise: early emphasis on understanding users and their tasks, employment of empirical measurement, and iterative design processes.

The principles encompass the outlook of user engagement. Some studies advocate for direct interaction between the design team and potential users, emphasising personal contact over mediated information. The second principle underscores the importance of early involvement of intended users in utilising simulations and prototypes for actual tasks, with their performance and feedback being closely observed, documented, and analysed. Usability engineering often intersects with user-centred design, with the terms being used interchangeably. (Kujala, 2003.)

Kujala (2003) highlights the methodology of participatory or cooperative design involving collaboration between designers and workers in understanding user needs and tasks for planning and developing new business practices and interfaces. Users engage in analysing organisational requirements and devising appropriate social and technical structures to support both individual and organisational needs. Key elements of participatory design include democratic participation and skill enhancement. The methodology typically involves observing and conversing with users individually, often one-at-a-time, as they perform their tasks in their natural environment. The objective is to study work processes, describe them, and redesign them by altering role structures, supporting tasks, automating processes, and eliminating unnecessary steps. Central to this approach is a philosophy of regularly visiting users.

To truly grasp the significance of user involvement, it is essential to understand the benefits it brings. Kujala (2003) highlights studies that demonstrate that effective involvement in system design yields the following benefits: improved system quality due to more precise user requirements, prevention of costly system features that users neither want nor can utilise, heightened acceptance of the system, enhanced user understanding resulting in more efficient utilisation, and increased involvement in organisational decision-making.

Kujala (2003) in their study accentuates some early studies from the field. Some of the insights from these pioneering studies are that effort put in user-centred design in the early phases leads to much less effort later. There has been reportedly a positive effect in the quality as well, meaning good systems in the end. Studies mention that users have been asking questions developers could not even think of, leading to a better understanding of the actual use of the system, opposed to a developers' assumptions of the use.

The research Kujala (2003) presents identified major benefits including enhanced understanding, realignment of efforts, improved comprehension of downstream technology, and enhanced mutual understanding and work relationships among stakeholders. Additionally, customers often perceived the visits as gestures of respect and appreciation, while software developers' decisions were more likely to align with user needs. Whereas challenges encountered during field studies were also reported. These include the potential overwhelm of raw data collected, difficulty in influencing design without established field-oriented methods within the development process, and obstacles in gaining direct access to customers.

In Kujala's (2003) study the development team perceive the benefit gained as the capacity to collect precise and thorough information regarding their users' needs. One user even remarked on the impressiveness of the resulting requirements. In essence, the field studies on user involvement yielded positive outcomes. The authors believed they acquired invaluable insights from users, which aided in comprehending the needs of both customers and users and elicited positive responses from customers and users alike. Conversely, they noted various challenges and costs, including the overwhelming volume of raw data collected, difficulties in influencing design, obstacles in gaining direct access to customers, the time investment in studies, communication, and management of large data sets, and users' requests for system changes.

The research by Kujala (2003) identified that the time required for conducting the actual studies, communication, and managing extensive data remained their primary challenges. Additionally, it was observed that while users were generally satisfied, some began to request changes to their system. Overall, employing field studies provided a more comprehensive understanding. The study notes that challenges encountered, including users' reluctance to be observed during work, prolonged efforts to find users willing to participate in observational sessions, and the infrequency of users engaging in real work during observations. These challenges stemmed from difficulties in communication between users and developers. The authors concluded that ideally, all stakeholders should be motivated, and users should be educated about the entire design process. These issues with user involvement appear to arise from the participative nature of user involvement in these projects, without the utilisation of specialised techniques.

Even in situations where the need for user participation is minimal, it was found to have a positive impact on user satisfaction. However, it becomes apparent that in cases of high task or system complexity, greater user involvement is essential. When complexity is low, only the core user participation behaviours contribute to increased satisfaction, while other aspects such as project definition show no significant correlation. In projects where users and documentation are the primary sources of information, the requirements capture process typically requires only one or two iterations for completion. Conversely, in projects where users have a secondary role in providing information, the number of iterations increases, often exceeding three. The benefits of user involvement are evident, as it consistently leads to both improved system success and user satisfaction. (Kujala, 2003.)

The overall interpretation suggests that involving users in the design process poses challenges for developers. Early user involvement holds promise, provided that user involvement methods are refined, and the roles of users and developers are carefully considered. Developers should actively engage users in the process, recognizing users as experts in their respective fields but not necessarily in design. Field studies offer a particularly effective approach for uncovering users' implicit and non-verbal needs, going beyond simply asking about needs to understand behaviour and future context of use. While users may

struggle to articulate precise requirements, they can elucidate their goals and task approaches, allowing designers to develop solutions on their behalf. Contextual inquiry and ethnographic methods show promise yet grappling with the vast amount of raw data collected presents challenges. (Kujala, 2003.)

## 5.2 User involvement in design thinking

In the heart of design thinking are its principles of collaboration and human-centric design. Tuunanen, Myers & Cassab (2010) illustrate the customer value that is created by the user's requirements, value propositions and drivers. The value creation acknowledges the context and the social nature of use, including service production, experience of the process and outcomes of using the service. Reason, Løvlie & Flu (2015), as well, highlight that the best approach to engage users is to profoundly understand the user's needs and experiences. Based on their research analysing the user's actions and observing the intended use provide the most accurate information of what should be done. To get the optimal level of engagement from the user involvement comes from designing the most suitable ways to collaborate, including the tools used in the process (Reason et al., 2015). Design thinking aims at investigating, understanding, reflecting, establishing, developing and supporting mutual learning between multidisciplinary teams, yielding insights on users' needs and desires (Antonini, 2021).

Benefits of using design thinking methods stir around user involvement and its consequences. Users are essential components of the design process, heightening effectiveness of the process, resulting in a product that is more compatible with user needs and business requirements. Knowledge exists within a complex network of artefacts, practices, and interactions that involve people and their social, cultural, economic, political, and physical environments. Users are the experts of their domain, whereas designers have different kind of skillset. It is necessary to lose one's personal viewpoints and aim for collective thinking in order to create new knowledge. (Antonini, 2021.)

Antonini (2021) suggests several benefits that can be gained from utilising design thinking, such as customer loyalty and satisfaction, alongside considerations like the number of fresh concepts for products or services and their originality, value, and feasibility. Additionally, it examines the duration for idea development and marketing efforts. Their study also highlights advantages that are generation of shared knowledge that prioritises access to user experiences, including heightened customer satisfaction and loyalty, accelerated time to market, enhanced product quality, and reduced costs.

Antonini (2021) found improvements in business level too, such as cost reductions, revenue, profitability, and market share gains from new or enhanced offerings, as well as the time required to break even on new introductions. Additionally, collaboration has been credited with numerous positive effects at the organisational level, such as enhancing innovation practices and processes, fostering greater creativity at both individual and group levels, and

expediting decision-making regarding idea development and filtering. Efficiency undergoes a boost when users' input blends harmoniously with designers' perspectives, fostering ongoing improvements in products and services and reducing the likelihood of failures. Meanwhile, effectiveness blossoms through cooperation, nurturing the development of products that resonate more deeply with customers' needs. This cultivates positive perceptions among customers regarding products and services, ultimately strengthening the bond between the organisation and its customer base.

As a multidisciplinary field, design thinking loans methodology from other fields as art, anthropology, engineering, psychology (Tschimmel, 2012). The design thinking process is moulded around user involvement and cooperation with the designing team. The methods can be shaped around the needs and expectations of the user, providing a platform for the most accurate requirements (Antonini, 2021). Design thinking is a mindset, but it is accomplished by adapting a suited set of tools in the process and utilising them to gather as much as valid information as possible. Tools transform unfinished or immature ideas into something more concrete that can be explained and discussed within the designing team (Tschimmel, 2012). The tools and methods of design thinking need to cover both convergent and divergent types of thinking that come from the holistic nature of the methodology (Chasanidou et al. 2015). Design thinking tools enable value co-creation and consideration of user needs to be regarded in a multitude of ways, even without including users directly into the process (Micheli et al. 2019). Using tools help designers to think quicker and more freely. It allows more effective internal dialogue and communication with stakeholders or users. (Tschimmel, 2012.) These tools should never be isolated elements but seen as a linked ensemble. Singular tool might identify design thinking attributes, but links between the set is valuable. (Micheli et al. 2019.)

Sketching and prototyping are acknowledged by the professionals to help in envisioning and exploring solutions, understand the current situation and evaluate the possible futures. Similarly, personas are seen as a tool to visualise situations for stakeholders, provide empathy and ensure the designed problem in hand is understood (Micheli et al. 2019). Tools like sketching, mapping and prototyping are essential, because they stretch into the future, to the future solutions and by tools with visual characteristics create understanding between interdisciplinary teams and enable them to create together (Tschimmel, 2012).

### **5.3 Design thinking toolkit for UAT**

To conclude the challenges that user involvement causes in UAT are mainly humane issues that can be framed by communication and engagement. In the research following challenges were raised:



- Communication between users and development team
- Professional language
- Engaging customer's representatives
- Lack of competence in technology
- Resistance to new technologies
- Reluctance to learn
- Fear of failure
- Access to the customer's representatives
- Lack of motivation to participate
- Empathising with users

When studied projects decided to change the way of working and implement ways like surveys, interviews, walkthroughs and workshops, positive results were reached. Including users in the planning activities is significant for a successful software development project and absolutely necessary to gain the information that is needed in the development and testing. Furthermore, customers frequently interpreted these visits as demonstrations of respect and appreciation, whereas the decisions made by software developers were more inclined to meet user needs.

The advantages of involving users in the initial stages of planning UAT extend beyond improved system quality through precise user requirements. They also encompass averting expensive system features that users neither desire nor can effectively utilise, a greater acceptance of the system, a deeper understanding among users leading to enhanced efficiency in utilisation, and increased participation in organisational decision-making. To mention, the most notable negative effects that were concluded in the research are in relation to the schedules and time required in user involvement and the costs it brings in.

The way to implement user involvement in UAT is to make it more effective and less of a burden. The activities related to UAT should start early in the process of software development and including users in the planning stage is important to ensure the best results. Incorporating design practices is a beneficial way to proceed. Studies have encountered highly positive results of utilising different forms of design practices in technology. Aligning design with user needs and uncovering potential new product or business avenues are pivotal in supporting financial strategies. The benefits of well-designed user involvement include:

- Customer satisfaction
- Appreciation within the team
- Improved requirements
- Better end results
- Heightened team spirits
- Motivated representatives
- Focus on delivery

Based on the research it is clear that user involvement is essential to system acceptance and it naturally runs down with user acceptance testing. Users should be included in the planning processes in the early stages of the development and participate in the creation of requirements, acceptance criteria and user acceptance testing design. Participatory design advocates the foundational principle of user involvement philosophy and advocates for users' rights, presenting the direct collaboration between end users and designers. With the combined intelligence of the development team and business users harnessed by design thinking practices is ensured to provide the best solutions and end results.

Design thinking practices have been adapted to software development to some extent. However, research has overlooked the opportunities to apply the methodology in UAT. Nevertheless, an insight on design thinking tools in action in software development highlights that the mindset brings value in software development and can be utilised also in UAT. Design thinking, as a mindset, views innovation as human-centric, combining divergent and convergent thinking, encouraging frequent and early failure, and promoting the development of prototypes for experiential testing, thus aligning well with common software development practice Agile methodology (Parizi et al., 2022).

Design thinking emphasises understanding the problem and finding solutions that align with user needs, whereas Agile method prioritises speed, simplicity, continuous and fast deliveries, frequent feedback, and rapid responses to changes. The multidisciplinary structure of software development teams, combined with insufficient training in design and the number of available design thinking models and techniques, makes its implementation challenging. (Lang et al., 2021; Parizi et al., 2022.) Potential risks include a lack of participant commitment and the significant effort required for design thinking activities, which can sometimes be perceived as inefficient (Lang et al., 2021). Therefore, IT professionals must be mindful of these issues to effectively leverage design thinking in software development (Parizi et al., 2022) and not assign too many resources on it (Lang et al., 2021).

Findings indicate that design thinking is flexible, allowing adaptation based on the problem's context, without a prescribed order for its working phases. The techniques aid development teams in innovating the final product. (Parizi et al., 2022.) It can be used to change a perspective and shift a way of thinking (de Villiers, 2022). The integration of design thinking with Agile aims to better capture user needs in the early stages and ensure software usability (Parizi et al., 2022), prevents undesirable development alignment (Lang et al., 2021) and contributes to project success (Canedo & Parente de Costa, 2018).

Design thinking has been utilised as a user-centred design approach to involve users in creating innovative software solutions. It aids software teams in understanding users, fostering creativity, and practising both convergent and divergent thinking. Thus, it is essential to understand how design thinking can be integrated into software development and the techniques and models that can enhance the software development process. (Lang et al., 2021; de Villiers,



cause more negative effects and delay of the real solution, naturally affecting many business aspects. Root cause analysis aims to reveal the original problems that cause the challenge that is faced. The root cause is the most harmful thing that creates challenges further in the product or service. (Wieczerniak, Cyplik & Milczarek, 2017.) Root cause analysis can benefit from other design thinking techniques such as five whys (Chapter 5.3.2), brainstorming (Chapter 5.3.3) or thinking hats (Chapter 5.3.7).

The root cause analysis in its core is to improve processes. When the problem is analysed thoroughly, there opens up a possibility to create value. This will result in more logical and systematic products, providing operational excellence, output quality and business performance. (Reid & Smyth-Renshaw, 2012.) In the UAT process, the problem is the use of the system and its desired functions. Root cause analysis can provide an outlook to the requirement setting, acceptance criteria and test case creation. By making an in-depth analysis of the base of the problem it can be ensured that the software will provide value to the customer. Through its comprehensive view to the issue, possible action points can be surfaced, and these are useful data for the test cases. Once the right problem is uncovered, the development team together with the users can begin the process of discovering solutions.

### 5.3.2 Five whys

When faced with a problem, ask why. And ask it five times. This simple mantra-like repetitive act is a known design thinking tool five whys (Fig. 3), that is often connected with root cause analysis. By reiterating the question, the nature of the problem slowly comes to surface. It can wake an 'aha moment', when the original problem is distanced, hidden influences are revealed from deeper on a causal pathway. (Card, 2017.) The process begins with the end result and reflects on the reasoning behind it. This is an effective method for problem solving which ignites doubts and deep thinking. Due to its simplicity, the five whys can be asked in various situations and it is quick to implement. (Serrat, 2017.)

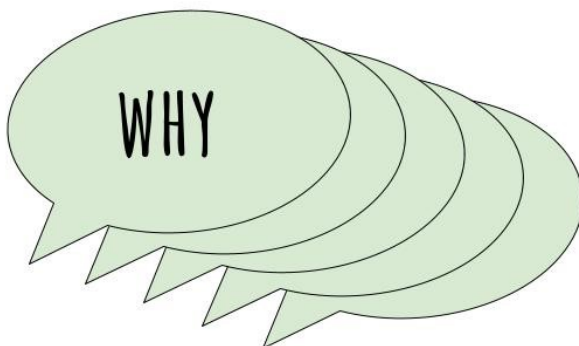


Figure 3 Five whys

Despite its simple nature, the method is powerful in troubleshooting and exploring cause-and-effect relationships. It leans into systematic problem solving. Things that should be considered when applying five whys are accurate and complete statements, honesty and determination to resolve the problem in hand. The questions will most likely raise several follow up problems that need to be solved, and on each round the most significant of these should be selected. In the next rounds, each of these new statements will be questioned again with a new why. The process is carried out, when asking why does not yield further information. Within the statements that have appeared, are the systematic causes of the problem. Now the team should be able to move forward on the design process with accurate problems. (Serrat, 2017.) This tool, as well as root cause analysis is a feasible solution in finding the correct things to work on. In UAT the tool can be used in the beginning of the process, when the business users and developers work on the requirements together. By figuring out the right problem, developing an appropriate software solution comes to great value. The tool can also be utilised in a later phase of the process to gain deeper understanding of the possible options and eliminate cases that are not needed.

### 5.3.3 Brainstorming and its variants

Brainstorming (Fig. 4) and its variants brainwriting and brainsketching are collective processes that encourage more flexible and radical thinking. The process urges to unfold ideas that are not yet discussed out loud or thought through. Goal of this tool is to come up with a large number of ideas in a relatively short time that come from intuition and emotion instead of rational thinking. (Tschimmel, 2012.)



Figure 4 Brainstorming

Brainstorming as a traditional tool might not be the right one in all situations and for all teams. It is great for people who like to think on their own and process ideas in their minds. Some people are more comfortable and creative

through verbal communication and thinking together as a team. Thoughts of others influence them and provide inspiration to more creative ideas, especially within a group with different levels of expertise. These things have induced into variants of brainstorming; brainsketching and brainwriting. In these processes instead of just generating ideas each on their own, the team adds new ideas on the board visible to all members enabling those to inspire and recreate further ideas by others. The process facilitates organising and categorising ideas from the beginning without a separate moderator. In the end the team starts an evaluation process where the ideas should be arranged in a way where the ideas with most potential are in the middle and the least potential ones on the outer sides. Each member can participate in the organising, during a limited timeframe. (Tschimmel, 2012.)

Brainstorming is seen as a particularly useful tool in the ideation process. It is a collaborative action that aims to find new solutions that might not arise through individual ideation. (Micheli et al., 2019.) When planning the UAT, this tool is exceptionally advantageous when discovering feasible use cases and requirements due to its divergent character. In the beginning of the software development process, when setting the acceptance criteria and business requirements, brainstorming methods can provide value in discovering the usage of the software. Combining the brains of the development team and end users, this method can broaden the perspectives the software is perceived. Brainstorming is quite a well-known tool and often familiar to people, which makes it easier to apply in interdisciplinary teams.

### 5.3.4 Personas

Personas (Fig. 5) are fictional characters that represent the key user groups of the service or product (Piras et al., 2019; Tschimmel, 2012). In the design thinking process it is possible to consider user needs without necessarily having direct contact with the users, which of personas is a good example (Micheli et al. 2019). Through comprehensive personas it is possible to understand the actions, needs, experiences and goals of the people that will be using the service (Chasanidou et al. 2015; Piras et al., 2019; Tschimmel, 2012).

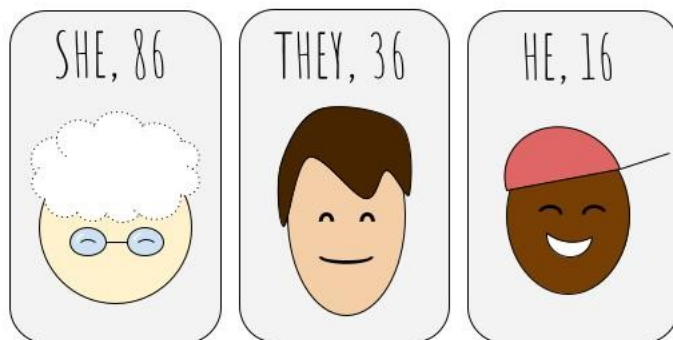


Figure 5 Personas

In addition to service design, personas are useful in other business purposes as well, such as engaging with a target audience, developing marketing products and communication. In design thinking they are used while empathising with target groups in the discovery phase and defining the possible new solutions. (Chasanidou et al. 2015.) Rather than just representing the target group, persona is used to reveal deeper insights to the different kinds of experiences users might have, with an objective to improve said experiences. Persona aims to humanise and exemplify the user and their attributes. (Tschimmel, 2012.) Designers gain further information on the different perspectives of how the service is viewed, users and their psychological factors and strategies that are beneficial to take into account when planning improvements to the service (Piras et al., 2019).

Personas as a tool for software development and UAT have major potential. The tool is viable for discovering possible use cases of the software. Usually personas are generalised imaginary customers that are created based on the research material. However, when considering software development, it is possible to utilise customer's representatives for personas. Creating in-depth personas of actual users is a way to establish accurate information of real users' desires and the way they are going to use the software. This however, requires trust and cooperation within the team. It can be applied as an inspiration for improved acceptance criteria and setting requirements for UAT.

### 5.3.5 Journey mapping

Journey mapping (Fig. 6) is commonly used by design and consultation. It is a suited technique for empathising with customers. A customer journey map is originated from a wider business planning technique service blueprint. (Chasanidou et al. 2015.) Journey mapping is a visual representation of a customer's experience and their interactions with a company when they receive a product or service (Liedtka, 2011; Tschimmel, 2012). It describes a collection of touchpoints from start to finish of the service delivery from the customer's point of view. Touchpoint can be defined as a point of communication or interaction between a customer and a service provider. (Chasanidou et al. 2015.)

Journey map helps to identify the chances for service innovation and possible problematic areas where service improvement could be beneficial (Chasanidou et al. 2015). The map can be created to present the actual or the ideal journey, but either way it forces the focus into the customer. Mapping the journey puts the design team into the shoes of the customer. It highlights the emotional highs and lows and the meaning the customer experience holds for the customer. (Liedtka, 2011.) Using visual means such as colours, sticky notes, forms and pictures can help designers to stimulate the designed affair in a more meaningful way and resonate into new perceptions (Tschimmel, 2012).

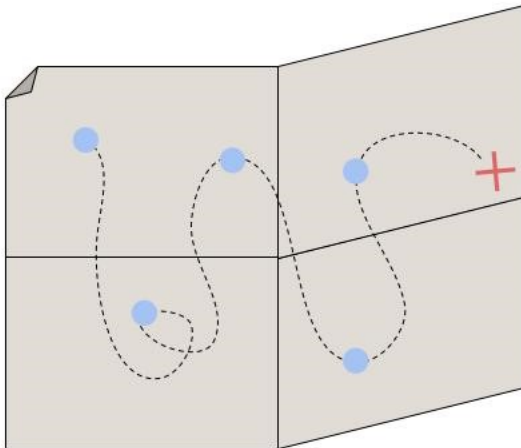


Figure 6 Journey mapping

Journey mapping in the context of UAT can provide valuable and detailed information about the real use of the system in hand. After the main use cases are selected and defined by the means of discovery and definition, these can be developed further. With journey mapping the possible bottlenecks and significant touchpoints can be refined, meaning more accurate requirements and acceptance criteria. Journey maps can even be converted directly to test cases for UAT. This tool enables the development team and customer's representatives together to highlight what is important in the integral everyday use of the system, contributing to better planning and outcomes in software development projects.

### 5.3.6 Storytelling and storyboarding

Storytelling is an ancient form of communication and visualisation. It has been used to communicate feelings, dreams and theirs since the beginning of time. In design thinking it is a tool that helps teams to share new concepts, situate products and services in the reality or narrative context. Stories are by nature affected by and affecting emotions. They are illustrative, symbolic and memorable ways to bond with the audience. Storytelling by its characteristics allows a more detailed viewpoint to the design. (Tschimmel, 2012.)

Storyboarding is a visualised form of storytelling. It can be executed through different forms of visual presentation, like drawings, illustrations or photographs that are displayed sequentially to represent a storyline. Storyboarding permits the team to revisit the sequence and consider different kinds of solutions. The visualisation of the story encourages dialogue between the participants, providing more fulfilling results. (Tschimmel, 2012.) These tools, for added value, can be used together and combined with other tools like roleplay (Chapter 5.3.8) or journey mapping.



For UAT storytelling provides prospects to go through the use of software and consider the touchpoints user meets during the usage. It can be useful when starting to figure out the actual need that the software is built on, to give a platform for users to put into words what they would expect. The tool can also be applied once the divergent discovery is finished and there are some options on how to proceed. Storytelling then gives input for convergent thinking and defining the set use cases. Creating a story of the usage and visualising it on a way that is easy to understand connects people with different competences. The simplicity of the tool improves the communication between the end users and development team, creating a common language for them.

### 5.3.7 Thinking hats

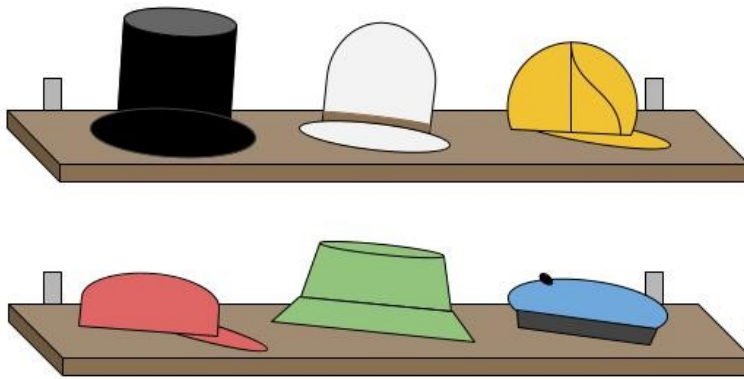


Figure 7 Thinking hats

Six thinking hats (Fig. 7) is a methodology that rests upon an ideology, that using different perspectives challenges can be monitored through more effectively, innovation can be stimulated, and opportunities can be found. These perspectives are visualised as hats, which all have their own viewpoint. (De Bono, 2021.) The original thinking hat theory roots from De Bono in 1985, when the researcher published the theory as an essential approach for business management. Since then, thinking hats have become a viable tool in design (Göçmen & Coşkun, 2019; Chen, Liu, Hu, Du, Bai, Ren, Lan & Yu, 2023).

All hats have their appointed point of view and can be worn by either one or all of the team members, and in singular use or sequentially. The black hat is one to look at the challenges, risks and difficulties that the project might face. This perspective renders a possibility for finding the problems that can arise during the project. To perceive this viewpoint, the team should ask questions like “what might go wrong”, “why this doesn’t work” and “where are the most sensitive spots”. This way, it is possible to find solutions to the problems before they come alive. The white hat ideology is simple. This perspective is to look at the facts. What is the information that is available and necessary for success? With yellow hat, the team seeks optimism. The value that the project provides is

evaluated and the benefits can be underlined. The red hat is for intuition and feelings. With this point of view, it is possible to share the desires, fears, loves and hates, likes and dislikes. Here is an opportunity to prioritise what is important and what might come as an extra-value. Creativity is cherished by the green hat. This one inspects the possibilities, new ideas and alternatives. Here the team can explore new concepts and perceptions for the project. The blue hat is in control and its role is to manage the thinking process and observe the guidelines for this method. (De Bono, 2021.)

Thinking hats -practices enable discovery outside the box. It creates dynamic within the team and permits people to see all sides of the problem by using parallel thinking and making collaborative thinking clearer. (De Bono, 2021.) Thinking hats is a multipurpose activity that can be combined with other design thinking methods, like brainstorming (Göçmen & Coşkun, 2019) or storytelling (Chen et al., 2023). In UAT, this design thinking tool can be used in combination with other tools to make the process more efficient and exhaustive, or on its own. It forces people out of their own shoes, to see the world from another perspective. This does not only show a different outlook on the problem in hand, but also moulds the team together as they all go through the process with different perspectives than their own. Seeing things from another perspective requires trust. It is a viable option when making discoveries through brainstorming or just playing along with the perspectives or deepening the understanding of the issue with storytelling. Thinking hats is a very versatile tool, that yields results in various ways.

### **5.3.8 Roleplay**

Roleplay (Fig. 8) is a prototype of sorts, which has a strong user-centred orientation. As a design thinking method, it is quite easy to build, and it produces ideas and experiences of the designed products or services. The goal of roleplay is to try out an idea and offer it a space where it can grow with responses and reactions. (IDEO, 2024.) It is a way to engage the team in a playful way, presenting the concepts that might require communication and further insights. Roleplay can improve communication within the team members, raise empathy and attach into the complexity of technology. (Boess, 2006.) Roleplay can be carried out as a teamplay where each team member has their own role, or as a solo where one of the team members presents their idea to others. Costumes and props are highly effective tools in roleplay, but not absolutely necessary. These can make the play more realistic and vivify the concept, even with just a few details. (IDEO, 2024.)



Figure 8 Roleplay

Often projects might have new concepts and requirements that are difficult to bring to life verbally. This is why roleplay is feasible, it allows people to perform nuances that might otherwise stay hidden. For this reason, the method holds so high potential in design, due to surprises, inadvertent actions and reflective learning it nurtures. Roleplaying also forces the team to strip off their own perspectives and act out the concepts objectively. It deconstructs all presumptions and explores foreign viewpoints. (Boess, 2006.) In UAT, this enables the development team and business users to get closer to each other. They remove their own professional masks and step in another's shoes.

In UAT roleplay has a place when the use cases have been discovered and designed, to provide a confirmation that the software development aims to build right solutions. By acting out the scenarios, the requirements come to life and it is easy to see possible faults and if there is a need to make last-minute changes. Similar to the thinking hats, also roleplaying ties the team together as it requires trust to play along. The tool acts by indulging one to the process. To use of the roleplay provides valuable information of the software that is developed. In addition, it gives the development team a chance to step into the shoes of the users, and this way builds a common ground.

## 6 CONCLUSIONS

In conclusions the key contents and observations from this research are presented and discussed what was learned. The section includes an analysis of the implications raised by the research, limitations it faces and recommendations for the future research

It is clear that involving users in early phases of testing design and software development processes in general has major benefits. In this study there are some of the design thinking tools analysed in addition to find best matches for the purpose.

### 6.1 Research implications and limitations

This study emerges in the themes of design thinking, user involvement and user acceptance testing in software development. Since design thinking is a mindset, it has worked as a lens throughout the process of writing this thesis. The theory presents the principles and processes of design thinking to make readers familiar with the perspective of designing, which is very different from the usual in the technology world. User acceptance testing is also viewed through the definition and its design process, noting the role of the user. Later in the literature review, the user's role and involvement in UAT and software development in general is examined thoroughly, pointing out the issues and challenges that are raised. The study also examines what kind of benefits there is of user involvement and how it should be approached. Research questions observe the possibilities to benefit from design thinking tools in UAT:

- Can design thinking mindset provide value for the UAT process?
- Which design thinking tools are useful when planning UAT?

To answer these questions, the research explores user involvement in design thinking and significance of giving users tools to provide necessary information.

To summarise the results, it is proved by the literature review that the user involvement and methods of design thinking can serve as a major asset in user acceptance testing. Numerous studies present that user involvement is a necessity in software development, however implementing it is a complex task. Often, when user involvement is overlooked and not planned properly, it might present negative effects in the whole development project. This means, it is highly important to have a user-centric standpoint in the process. Design thinking again, is by nature a fit for user involvement. Research on design thinking shows great results in user involvement, which impacts on improvements in several areas such as end result, team spirit, user commitment, efficiency and even profitability. It has been taken into consideration to some extent in software development, but there is no research that would suggest design thinking to be exploited in user acceptance testing. Considering user acceptance testing starts from the beginning when the business requirements and acceptance criteria are set, it is only natural to advance from the end users from the beginning.

Regardless of its user-centric and collaborative nature, the design thinking process also has its pitfalls on user involvement, hence the tools and techniques should be well planned to suit the case. This study assembles a toolkit built of design thinking tools that fit for user acceptance testing and provides an outlook on how those are supposed to be used, with a view of the benefits the tools provide in the UAT process. The research aims to present tools that can be used in several ways and in different phases of the design thinking process. This way, the development team together with the users gain the most benefits and lead to the best accuracy on what comes to the requirement setting, acceptance criteria and test cases.

The research is based on very precise literature from the field, aligning into software testing, user involvement and means of design thinking. Software development as an entirety is definitely well studied but leaning into the two main topics of this research, both software testing and design thinking have been described as vague terms, a 'fuzzy' concept or 'dark arts'. Neither of the topics have one definitive description, rather they have a number of different definitions. Both of the concepts are relatively young in the scientific world, having risen to the tables of researchers around the early 2000s. There has been a couple of decades to research these topics, but neither of them raises to their deserved recognition. This study provides an insight into these ambiguous concepts, combining them in a special way creating value by a new method of working.

Examining this research in the viewpoint of reliability and validity, there are some issues. Validity in mind, the study confronts some challenges. The data collection has focused on the literature confirming the hypothesis of the study, nevertheless more accurate assumptions could have been made if also opposing phrases would have been used in the data collection phase. The research also relies on just a couple of search engines, in the main use being Google Scholar and the library database of University of Jyväskylä. To provide more fulfilled analysis on the topic, the research could have used more specific

databases from the field. The method for the analysis in this research was content analysis, which has raised concerns in its quality standards due to its weight on researcher's views. Formal standards have not yet been defined for this method and it is seen as quite simple and easy to act out with a low amount of experience (Krippendorff, 2018). Another thing that might affect the credibility of this study is that even though there are fresh studies within, some of the source materials are dated in early 2000s and some even 1990s. In technology, it is ages, but still some of the original theories stay valid. However, the research material is large and extensive, and having theory covered in several eras might have positive effect on the credibility of the study.

When evaluating the reliability of this research, there is especially one thing that should be highlighted. This research has been created in a light of design mindset and roots for user involvement. The chosen studies have been concluded to reassure the means of this research, meaning there might be limited information about opposing standpoint. The search phrases did not include ones to go against the research hypothesis. If only the literature review is studied, even with different references, the assumption should stay similar: user involvement is a double-edged sword that might, if not properly designed, raise issues rotating around commitment, proficiency and communication. However, in a development study, evaluating reliability is not a simple task. Development study aims for a change, an improvement, meaning that the results should be different in the end. In this case, as the study merely provides a toolkit, and does not inspect a singular project, the change could only be perceived when taking the toolkit in use. For this kind of study, the development research is the only possible option, and it has enabled the study to provide new information and fresh, practical methods for the field.

There is a very high number of positive outcomes on the hypothesis, which has led the researcher to trust in their original assumption. This does not mean there is no scientific proof fighting the results, for example Kujala (2003) in their study has raised one study amongst many that was competing against the presented norm. If gone further in the literature, it is possible there would be other studies as well, that would not be on the same side with this research. The study mentioned was by Heinbokel, Sonnentag, Frese, Stolte & Brodbeck (1996) and it claimed that user participation disturbs the process of software development. In the IT world this statement can be strongly agreed on, but that can also be seen as a reasoning for this study. If the user involvement is done with proper structure and planning, it will be beneficial for the process and especially for the end result.

## **6.2 Research recommendations**

However, studies explored during the process of this research, few concerns are raised in terms of user involvement. It was seen as a disturbance for the development team, which alone is a reasonable thing to explore. In this study some

of the issues have been raised through a literature review, but it would be intriguing to go into the process and observe what are the main problems and how to avoid them without ignoring the users. Challenges that arise in software development from human-related issues could also be studied in terms of psychology, providing an additional angle on understanding the root causes.

Another direction for future research could also be drawn from the concerns noted while making the study. Various studies recognised that involving users in the development process can be a costly feature. It is suggested that in future the financial side of working with customers' representatives is studied. There is a major improvement possible in this field, if more cost-efficient ways to provide insight is created. This study aims at proper preparations by early involvement of users, battling against costs in later phases and striving for better quality to avoid extra costs on defects. By means of proper financial examination it would be possible to focus the user involvement where it is most efficient and discard from where it is not.

For the last and the most obvious direction for future research should be underlined: the usage of the provided toolkit. Seeing the output of the study become reality and being used in actual software projects would be a perfect continuation of this theme. Through multitude of studies it is proved that design thinking can provide value to the software development and UAT, and it is to be desired this research provides a launching point for its implementation.

## REFERENCES

- Abrahamson, E. (1996). Management fashion. *Academy of management review*, 21(1), 254-285.
- Antonini, M. (2021). An overview of co-design: advantages, challenges and perspectives of users' involvement in the design process. *Journal of Design Thinking*, 2(1), 45-60.
- Barab, S., & Squire, K. (2004). Design-based research: Putting a stake in the ground. *The Journal of the Learning Sciences*, 13(1), 1-14.
- British Design Council. (2015). Design methods for developing services.
- Boess, S. (2006) Rationales for role playing in design, in Friedman, K., Love, T., Côte-Real, E. and Rust, C. (eds.), *Wonderground - DRS International Conference 2006*, 1-4 November, Lisbon, Portugal.
- Brown, T. (2008). Design thinking. *Harvard business review*, 86(6), 84.
- Canedo, E. D., & Parente da Costa, R. (2018). The use of design thinking in agile software requirements survey: A case study. In *Design, User Experience, and Usability: Theory and Practice: 7th International Conference, DUXU 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15-20, 2018, Proceedings, Part I 7* (pp. 642-657). Springer International Publishing.
- Card, A. J. (2017). The problem with '5 whys'. *BMJ quality & safety*, 26(8), 671-677.
- Chasanidou, D., Gasparini, A. A., & Lee, E. (2015). Design thinking methods and tools for innovation. In *Design, User Experience, and Usability: Design Discourse: 4th International Conference, DUXU 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings, Part I* (pp. 12-23). Springer International Publishing.
- Chen, D., Liu, F., Hu, Y., Du, X., Bai, Y., Ren, Z., Lan, L. & Yu, W. (2023). Service design from the perspective of "six thinking hats": A comparison of storytelling strategies of experts and novices. *Thinking Skills and Creativity*, 47, 101219.
- Clouin, P. H., Janhagen, V. (2019). Long live design thinking. *Idean*.
- Combelles, A. (2020). Design thinking.



- De Bono, E. (1985). *Six Thinking Hats: An Essential Approach to Business Management*. Little, Brown, & Company (Ed).
- De Bono, E. (2021). *Six thinking hats*.
- De Waal, F. (2005) *The evolution of empathy*.
- de Villiers, R. (2022). *Design Thinking as a Problem Solving Tool*. In *The Handbook of Creativity & Innovation in Business: A Comprehensive Toolkit of Theory and Practice for Developing Creative Thinking Skills* (pp. 223-242). Singapore: Springer Nature Singapore.
- Edelson, D. C. (2002). *Design research: What we learn when we engage in design*. *The Journal of the Learning Sciences*, 11(1), 105-121.
- Ferreira, A. M., da Silva, A. R., & Paiva, A. C. (2022, April). *Towards the Art of Writing Agile Requirements with User Stories, Acceptance Criteria, and Related Constructs*. In *ENASE* (pp. 477-484).
- Forlizzi, J., & Battarbee, K. (2004, August). *Understanding experience in interactive systems*. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques* (pp. 261-268).
- Ganesh, K., Mohapatra, S., Anbuudayasankar, S. P., & Sivakumar, P. (2014). *User Acceptance Test*. *Enterprise Resource Planning*, 123-127.
- Gould, J. D., & Lewis, C. (1985). *Designing for usability: key principles and what designers think*. *Communications of the ACM*, 28(3), 300-311.
- Göçmen, Ö., & Coşkun, H. (2019). *The effects of the six thinking hats and speed on creativity in brainstorming*. *Thinking Skills and creativity*, 31, 284-295.
- Hambling, B., & Goethem, P. van. (2013). *User Acceptance Testing* ([edition unavailable]). BCS Learning & Development Limited.
- Hassenzahl, M. (2010). *Experience design: Technology for all the right reasons*. *Synthesis lectures on human-centered informatics*, 3(1), 1-95.
- Hämäläinen, H. (2022). *Utilizing Design Thinking in the product development of early-stage corporate startups*.
- Heinbokel, T., Sonnentag, S., Frese, M., Stolte, W., & Brodbeck, F. C. (1996). *Don't underestimate the problems of user centredness in software development projects there are many!*. *Behaviour & information technology*, 15(4), 226-236.

- IDEO. (2024, May 25). Design kit, Methods, Roleplay.  
<https://www.designkit.org/methods/role-play.html>
- Joore, P. (2022). Applied Design Research. CRC Press (Unlimited).
- Juuti, K., & Lavonen, J. (2006). Design-based research in science education: One step towards methodology. *Nordic Studies in Science Education*, 2(4), 54-68.
- Juuti, K., & Lavonen, J. (2013). Design-tutkimukseen osallistuvien opettajien rooli tutkimuksen eri vaiheissa. Teoksessa J. Perna (toim.), *Kehittämistutkimus opetusalaalla* (s. 45-67). Juva: PS-Kustannus.
- Kananen, J. (2008). *Kvali: Kvalitatiivisen tutkimuksen teoria ja käytänteet*. Jyväskylän ammattikorkeakoulu.
- Kananen, J. (2012). *Kehittämistutkimus opinnäytetyönä: Kehittämistutkimuksen kirjoittamisen käytännön opas*. Jyväskylän ammattikorkeakoulu.
- Krippendorff, K. (2018). *Content analysis: An introduction to its methodology*. Sage publications.
- Kujala, S. (2003). User involvement: a review of the benefits and challenges. *Behaviour & information technology*, 22(1), 1-16.
- Lake, F. (2016). 'Human Centred Design vs Design Thinking vs Service Design vs UX . . . . What do they all mean? – LinkedIn'.
- Lang, D., Spies, S., Trieflinger, S., & Münch, J. (2021). Tailored design thinking approach-a shortcut for agile teams. In *Software Business: 12th International Conference, ICSOB 2021, Drammen, Norway, December 2-3, 2021, Proceedings 12* (pp. 37-49). Springer International Publishing.
- Leung, H. K., & Wong, P. W. (1997). A study of user acceptance tests. *Software quality journal*, 6(2), 137-149.
- Liedtka, J. (2011). Learning to use design thinking tools for successful innovation. *Strategy & Leadership*, 39(5), 13-19.
- Liedtka, J. (2013). *Design Thinking: What it is and Why it Works*. Design at Darden.
- Liedtka, J. (2018). Why design thinking works. *Harvard Business Review*, 96(5), 72-79.
- Lockwood, T. (2010). *Design thinking: Integrating innovation, customer experience, and brand value*. Simon and Schuster.

- Marselis, R., Geurts, D., van Veenendaal, B., Ruigrok, W. (2020). Quality for DevOps teams. Sogeti.
- McCausland, T. (2020). Design thinking revisited.
- Micheli, P., Wilner, S. J., Bhatti, S. H., Mura, M., & Beverland, M. B. (2019). Doing design thinking: Conceptual review, synthesis, and research agenda. *Journal of Product Innovation Management*, 36(2), 124-148.
- Mohd, C. K. N. C. K., & Shahbodin, F. (2015). Personalized learning environment: alpha testing, beta testing & user acceptance test. *Procedia-Social and Behavioral Sciences*, 195, 837-843.
- Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing*. John Wiley & Sons.
- Neuendorf. (2016). *The Content Analysis Guidebook (Second Edition)*. SAGE Publications, Inc.
- Norman, D. (2013). *Design of everyday things*.
- Newbery, P., & Farnham, K. (2013). *Experience design: A framework for integrating brand, experience, and value*. John Wiley & Sons.
- Otaduy, I., & Díaz, O. (2017). User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps. *Journal of Systems and Software*, 133, 212-229.
- Padmini, K. J., Perera, I., & Bandara, H. D. (2016, April). Applying agile practices to avoid chaos in User Acceptance Testing: A case study. In 2016 Moratuwa Engineering Research Conference (MERCon) (pp. 96-101). IEEE.
- Pandit, P., & Tahiliani, S. (2015). AgileUAT: A framework for user acceptance testing based on user stories and acceptance criteria. *International Journal of Computer Applications*, 120(10).
- Parizi, R., Prestes, M., Marczak, S., & Conte, T. (2022). How has design thinking being used and integrated into software development activities? A systematic mapping. *Journal of Systems and Software*, 187, 111217.
- Plattner, H., Meinel, C., & Leifer, L. (Eds.). (2014). *Design thinking research: Building innovators*. Springer.
- Piras, L., Dellagiacomma, D., Perini, A., Susi, A., Giorgini, P., & Mylopoulos, J. (2019). Design Thinking and Acceptance Requirements for Designing Gamified Software. 2019 13th International Conference on Research Challenges in Information Science (RCIS).

- Poston, R., Sajja, K., & Calvert, A. (2014). Managing user acceptance testing of business applications. In *HCI in Business: First International Conference, HCIB 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings 1* (pp. 92-102). Springer International Publishing.
- Reason, B., Løvlie, L., & Flu, M. B. (2015). *Service design for business: A practical guide to optimizing the customer experience*. John Wiley & Sons.
- Reid, I., & Smyth-Renshaw, J. (2012). Exploring the fundamentals of root cause analysis: are we asking the right questions in defining the problem?. *Quality and Reliability Engineering International*, 28(5), 535-545.
- Rowe, P. G. (1991). *Design thinking*. MIT press.
- Saif, N. B., Almohawes, M., & Jamail, N. M. (2021). The impact of user involvement in software development process. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(1), 354-359.
- Schreier, M. (2012). *Qualitative Content Analysis in Practice*. SAGE Publications Ltd.
- Serrat, O. (2017). The five whys technique. *Knowledge solutions: Tools, methods, and approaches to drive organizational performance*, 307-310.
- Thanasegaran, G. (2009). *Reliability and Validity Issues in Research. Integration & Dissemination*, 4.
- Tripathy, P., & Naik, K. (2011). *Software testing and quality assurance: theory and practice*. John Wiley & Sons.
- Tschimmel, K. (2012). Design Thinking as an effective Toolkit for Innovation. In *ISPIM Conference Proceedings* (p. 1). The International Society for Professional Innovation Management (ISPIM).
- Tuomi, J., & Sarajärvi, A. (2009). *Laadullinen tutkimus ja sisällönanalyysi* (6., uud. laitos. ed.). Helsinki: Tammi.
- Tuunanen, T., Myers, M. D., & Cassab, H. (2010). A conceptual framework for consumer information systems development. *Pacific Asia Journal of the Association for Information Systems*, 2(1), 5.
- Vaughan, L. (2017). *Practice-based Design Research*. Bloomsbury Publishing.
- Vendraminelli, L., Macchion, L., Nosella, A., & Vinelli, A. (2023). Design thinking: strategy for digital transformation. *Journal of Business Strategy*, 44(4), 200-210.

Weber, R. P. (1990). *Basic content analysis* (Vol. 49). Sage.

Wieczerniak, S., Cyplik, P., & Milczarek, J. (2017). Root cause analysis methods as a tool of effective change. *Business Logistics in Modern Management*.

Woolery, E. (2019). *Design thinking handbook*. InVision.

Zowghi, D., da Rimini, F., & Bano, M. (2015, April). Problems and challenges of user involvement in software development: an empirical study. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering* (pp. 1-10).