Leveraging Design Thinking to Support Internal Agile Software Development

- An Opportunity for Nike Technology

Master Thesis
Strategic Product Design

Celine Hoogendijk

TU Delft

Nike, Inc.

Leveraging Design Thinking to Support Internal Agile Software Development

- An Opportunity for Nike Technology

Author

Celine Hoogendijk

Master thesis

September, 2021

MSc. Strategic Product Design
Faculty of Industrial Design Engineering
Delft University of Technology

Supervisors

Chair - Deborah Nas
Faculty of Industrial Design - Product Innovation Management (?)

Mentor - Jimena García Mateo
Faculty of Industrial Design - Human Centered Design

In collaboration with

Nike, Inc. - EMEA Technology

Master Thesis
Strategic Product Design
Celine Hoogendijk

Preface

The graduation project report in front of you marks the end of my master Strategic Product Design at the Delft University of Technology. It embodies my years as a student in business and design and captures my interests in those areas combined with technology.
The observation and realization that Design Thinking is a contextual concept motivated me to get clarity around many questions I had and opportunities I saw in the internal agile software development context during my internship. Being able to immerse myself in this topic has been a joy, especially in the context of Nike.

I would like to express my gratitude to those who supported me on this journey.
First of all, I would like to thank Akanksha for being a great mentor throughout this project. It has been a journey! I'm grateful for your trust, critical eye and continuous support; this project would not have been possible without you.

To all my Nike team members and other colleagues involved, thank you for your support, open-mindedness, time and feedback - I enjoyed working with all of you! The safe space you created for me to explore, to learn and to develop myself is invaluable to me.

I would like to thank Deborah and Jimena for the trust and feedback you provided throughout this project and the conversations we had - I learned a great deal from working with you. You challenged me to keep simplifying my thinking and output, forcing me to understand things even better, which has been a great learning experience.

A special thank you to my friends. Thank you for always being patient, for inspiring me, and for your enthusiasm and support.

Finally, I would like to thank my family. Remon and Annick, thanks for your laughs at any time of the day while working from home, for providing me with procrastination opportunities and with popcorn when I'm busy, and for always being there for me - your support means a lot to me!
Mom and dad, thank you for all the opportunities you have given me throughout my studies. Seven years, and two BSc, two MSc and a great internship later, I can truly say that this would not have been possible without your support. I have no idea how to express how much you mean to me.

Dear reader, I am excited to share my graduation project with you!

Cheers, Celine
Vlaardingen, September 28, 2021

Executive Summary

As agile practices lack a focus on understanding the actual problem, and Design Thinking is assumed to be a promising approach to complement agile practices regarding this lack, this graduation project aims to identify opportunity areas to leverage the Design Thinking methodology in the process of agile software development. The context of focus was a specific technology unit within Nike, Inc.

The main research question is formulated as follows:
'How might we use Design Thinking to our advantage in the agile software development context of the targeted Nike Technology unit?'

Recognizing that Design Thinking is a contextual concept that needs further adaptation to contextual user needs, literature research and context analysis are done towards Design Thinking, agile software development, and related opportunities and boundaries.

Research findings following the interviews indicate three main areas of concern that form boundaries to problem exploration: having a solution-oriented rather than a problem-oriented mindset, organizational structures that limit the space for problem exploration in terms of time, processes, and the role of technology in the problem exploration phase, and the need and importance of having a clear and aligned vision.

Literature and exploratory research findings are integrated, answering the research question through a conceptual model covering three key principles: problem-oriented and human-centered thinking, dynamic alignment towards strategic fit, and divergent thinking to consider more fit-for-purpose alternatives.

Subsequently, the conceptual model is translated into a usable artifact: a Problem Deep Dive Canvas accompanied by a Problem Deep Dive Tool Guide. The product aims to support product managers and product owners to put the key principles of the conceptual model into practice in collaboration with agile software development teams and business stakeholders.

The threshold to use the product is low as there are no significant conflicts with current structures and processes. Initial validation results are promising towards feasibility, desirability, and viability of the product. Using the canvas on actual requests already showed that the outcomes of the canvas potentially significantly impact the further trajectory of the intended projects.

Table of Contents

Chapter 1

Introduction

Project background, objective and approach

In this chapter the project background, objective and approach will be briefly discussed to create common ground prior to exploring related literature and diving further into the context.

1.1 Project background

Using agile practices, the goal of software development teams is to produce products in a cost-efficient way with minimum errors (e.g. Gurusamy et al., 2016). The danger however is that the approach to problem-solving tends to focus on the technical and analytical perspective, and perfecting functional requirements, rather than understanding and meeting actual user needs (Lindberg et al., 2011).

Research and practice show a possibility to align the goal of development projects by leveraging the Design Thinking methodology (e.g. Adikari, 2013). Design Thinking is a human-centered problem-solving approach that focuses on 'building the right thing', while the main focus of agile practices is to 'build the thing right' (Hildenbrand & Meyer, 2012) (fig. 1a).
As Design Thinking is often viewed as a 'Silver Bullet', the focus of research and organizations is often towards implementing generic Design Thinking, which might misfit the current mindset and processes within an organization. Recognizing the need to adapt the concept to a specific context and therefore deliberately investigating contextual needs is required to effectively apply Design Thinking as a means to an end.

Picking up on the assumed opportunity to complement agile software development processes with Design Thinking, an opportunity arises for Nike Technology to potentially innovate and advance current development processes. Therefore, in this graduation project, a case study will be done in one of the technology units (see §3.1.4 for more information about the unit of focus).

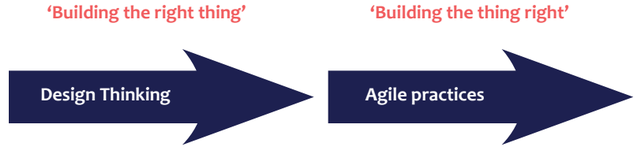


Fig. 1a: The assumed opportunity to complement Agile practices with Design Thinking

1.2 Project objective

Currently little is known about how to support internal agile software development teams with clear structure and tools in the problem exploration process. To find out how to advance Nike Technology's development process in agile software development, an opportunity arises to explore the role of the Design Thinking methodology to support the teams in exploring and understanding the problem and solution space, broadening problem-solving capabilities.

Viewing Design Thinking as a contextual concept (Johansson-Sköldberg, Woodilla, and Çetinkaya, 2013), see § 2.1.1, there is a need to investigate the context of agile software development teams and implications that might hinder and/or create opportunities to leverage Design Thinking, i.e. what aspects of Design Thinking do or do not apply and in which situations would the application be valuable? And how would it fit in current agile processes?

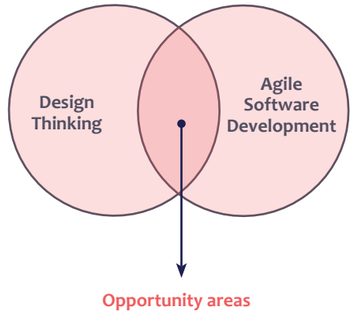


Fig. 1b: Diagram representing the aim of the project to identify opportunity areas to leverage Design Thinking in the process of agile software development

Thus, *the main aim of this project is to identify opportunity area(s) to leverage the Design Thinking methodology in the process of agile software development, and how we can tailor the process and relevant tools to fit the needs of this particular target group.*

1.3 Research questions and project approach

1.3.1 Main research question and sub-questions

The main research question is formulated as follows:

How might we use Design Thinking to our advantage (according to theory & practice) in this specific context?

Three main components arise that need further investigation to answer this question.

1) Design Thinking - method
2) This specific context: including the agile software development aspect
3) Advantage - the opportunities that arise

1) Design Thinking
First of all, the Design Thinking methodology will be explored.

Subquestions:
- What is Design Thinking?
- What is (or could be) the value of DT within this or similar (Agile) contexts according to literature?
- What can we learn from other companies applying DT?

2) The specific context
The next step is to understand the context of 'Agile Software Development'.

Subquestions:
- What is the current development process followed within the team?
- Target group - who are the key players (/ users) in this process?
- What type of products are made, and what are the specific attributes?
- What are gaps, pain points, challenges faced within the development process related to getting to the right thing to build?

These questions will provide themes, needs, and insights into opportunities for Design Thinking to complement the current process, narrowing the scope of the project.

3) Advantage
Thirdly, in the concluding 'using to our advantage' part insights of both components come together to determine the main research question.

Sub-questions:
- What are the main opportunity areas for ASD process-advancement with DT?
- How to tailor this to the specific context?

1.3.2 Project approach

In order to achieve the objective of the project and to answer the questions stated in §1.3.1, a literature review about Design Thinking and Agile Software Development will be done. Next to that, case studies of similar companies will be explored and exploratory interviews will be done regarding current boundaries to problem exploration (a more detailed description of the approach will be given in §4.1). In this way, the main opportunity areas will be uncovered. Insights will be translated into a conceptual model, representing the direction in which Design Thinking can be most effectively leveraged to support agile software development in the context of focus. Subsequently, the conceptual model will be translated into a usable artifact to support the target group in experimenting with and adopting the changes proposed.

Chapter 2

Theoretical Foundation

Problem Exploration in Agile Software Development

This chapter focuses on understanding and exploring the project topic of 'problem exploration in agile software development' from a theoretical perspective. Theories about the concepts of User-Centered Design and Design Thinking will be introduced to define a framework through which the current situation and opportunities can be analyzed. Next to that, literature and successful case studies are explored and analyzed to get insight into context and success factors. Main insights are presented at the end of this chapter and form the foundation for the context analysis afterward.

2.1 What is Design Thinking?

2.1.1 Defining Design Thinking

Even though Design Thinking gained popularity as a problem-solving and innovation approach, there is no universally accepted definition. Understandable, as the concept originated in many different fields with different perspectives and discourses: engineering, political and economic science (Simon, 1969); architecture and design (Cross, 1982; Lawson, 1979); arts and humanities (Buchanan, 1992); innovation (Tim Brown, 2008; Wyatt & Brown, 2010); cognitive science (Martin, 2009); and management science (Boland & Collopy, 2004; Kimbell, 2011; Lockwood, 2010; Liedtka & Ogilvie, 2011; Owen, 2005). Johansson-Sköldberg, Woodilla, and Çetinkaya (2013) did an extensive literature review of Design Thinking and argued that, even though a consistent definition might seem more appealing and clear, for the development of the concept it might actually be harmful as the concept presents a range of opportunities for application. Therefore, in this perspective, Design Thinking is seen as a contextual concept that has to be defined by practitioners and researchers based on the context of use.

In general, the Design Thinking approach focuses on understanding the user, challenging assumptions, and redefining problems. Central to the approach is the concept of empathy with the target user, which involves diving into emotions, needs, motivations, drivers of behavior. By digging deeper into the problem and user needs, one tries to identify and consider alternative strategies and solutions that might not be immediately apparent in the initial level of understanding.

Carlgren, Rauth & Elmquist (2016) define the following five core elements of Design Thinking:

Human focus - empathizing with users to discover and understand latent needs

Problem framing - challenging and reframing the initial problem, expanding the initial problem and solution space

Visualization - externalizing knowledge by visualizing insights and ideas – thinking by doing

Diversity-seeking different perspectives, creating teams and climates in which every opinion counts

Experimentation - failing often and fast in order to learn more about the problem, through quick prototyping and testing solutions with users, iteratively diverge and converge

To understand and capture the right discourses in the working definition in this thesis, the main discourses in the context of management will be explored next.

2.1.2 Two main discourses of Design Thinking in organizations: innovation and problem solving

Within the management science area, research distinguishes two main discourses of Design Thinking (based on Johansson-Sköldberg et al., 2013):

1) as a (/IDEO's) way of working with design and innovation (Kelley, 2001; Brown, 2009)

Initially, Design Thinking was introduced in the business world in the context of innovation by the design firm IDEO and COO Tim Brown. The IDEO's approach to innovation is described through five steps, with direct end-user engagement as an essential factor. Brown defines Design Thinking as *"a human-centered approach to innovation that draws from the designer's toolkit to integrate the needs of people, the possibilities of technology, and the requirements for business success"* (IDEO', 2021, 'how we work', para. 2). It aims to capture human needs and desires in a technologically feasible and strategically viable way, converting them into consumer and business value (Brown, 2009) through products, services, or processes. In this view, it is often driven by the search for innovation; the integration of human/user needs (desirability), technological possibilities (feasibility) and business value (viability) is seen as the sweet spot of innovation (see fig. 1).

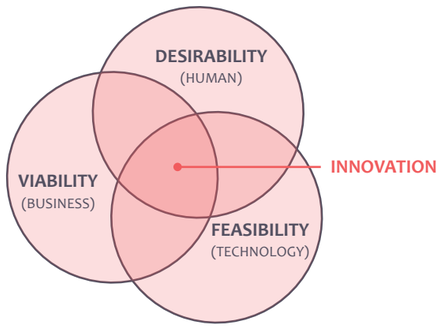


Fig. 1: Sweet spot of innovation integrating desirability, viability and feasibility (Brown, 2009)

With a vision that 'everybody could do it' and the intention to make the practices of designers accessible to the mainstream (specifically to managers), Tim Brown brought Design Thinking to a broad audience through his design company IDEO. The books, courses, and information provided by the company focus on creativity, detailed steps of the Design Thinking iterative process, tools to use, and many success stories from innovators and businesses (Brown & Wyatt, 2007).

Studies have found a relationship between Design Thinking and innovation. Bicen & Johnson (2015) found that companies that adopt Design Thinking are better at innovating, even when resources are scarce, showing the accessibility to Design Thinking to ordinary firms. Hobday, Boddington, and Grantham (2012), discuss that design is a primary driver of innovation and essential to bring innovative ideas into fruition. Next to that, researchers explore ways to use Design Thinking in the context of innovation to better understand the successful adoption of the approach (e.g. Chang et al., 2013)

2) as 'a way to approach indeterminate organizational problems, and a necessary skill for practicing managers' (Dunne & Martin, 2006; Martin, 2009)

Next to the application to innovation, many practitioners and researchers call for a broader use beyond innovation enabling companies to address adaptive challenges in the business world. Seen as a more generic approach to problem-solving compared to it being a means of innovation (first perspective), in this perspective, the approach is seen as a continuous cycle of generating ideas (abduction), predicting consequences (deduction), testing, and generalizing (induction), and is applied to a wide range of disciplines.

According to Martin (2009), managers focus too much on deductive reasoning, while they fail in exploring alternative options. Boland and Collopy (2004) argue that managers are able to create better outcomes when they adopt Design Thinking. They point less towards a way of working or process (e.g. like IDEO), but more towards individual cognitive characteristics.

Next to managers being engaged in decision making, i.e. selecting the best option from a list of existing options, Design Thinking allows to develop and consider new alternative options. In the complex and uncertain business environment we are in today, e.g. in the context of digital transformation, Boland and Collopy (2004) argue the need for both decision making and design for managers to be able to successfully address current challenges in the business environment. More often, problems can be described as indeterminate or ill-defined; they do not have a pre-defined list of solution options and require an exploration of both the problem and the solution space. However, according to the authors, managers are too focused on decision making and not enough on idea generation.

In line with this, Lafley and Charan (2010) state that there is too little focus on 'imagining what could be possible' (Lafley & Charan, 2010, p.

106) and too much focus on what is already existing and past evidence in the teachings of business schools.
Owen (2005) argues for the need for Design Thinking next to scientific thinking as well. According to him, combining both creates a comprehensive toolset to address ill-defined problems (Owen, 2005).
This is in line with other researchers who argue that business professionals need Design Thinking to complement their toolbox, making it suitable to solve the adaptive challenges that organizations face today (e.g. Kimbell, 2011; Liedtka & Ogilvie, 2011; Lockwood, 2010).

In this thesis, the definition described by Liedtka & Ogilvie (2011) will be taken as a foundation, which captures the perspective of Design Thinking as an approach to both innovation and indeterminate organizational problem-solving, the discourses described in 2.1.2:

> *"Design Thinking is a style of thinking that combines empathy for the users and immersion in the context of a problem, creativity in the generation of insights and [alternative] solutions and a data-based experimental approach to assessing the quality [and fit] of solutions"*
> (Liedtka, 2013, 'Design Thinking', para. 1).

Within this approach, dynamic thinking is required to both generate and evaluate ideas (Dunne & Martin, 2006).

2.1.3 Design Thinking as means of problem and solution exploration

As Design Thinking is a means to an end, it is useful to understand the core concepts of 'problem and solution exploration' (see fig. 2), and 'divergent and convergent thinking' behind it, which are especially relevant to understand the fit of the concept prior to actual solution development – the context of this graduation project.

The problem and solution space
Generally in science, and analytical problem-solving, the focus lies on exploring the solution of a given problem. Contrary to this, the approach of Design Thinking treats both the problem and the solution as a space to be explored and to learn about (Lindberg et al., 2011).

- Exploring the problem space - developing a fundamental understanding, observing cases, synthesizing to point of view or reframed problem.
- Exploring the solution space - developing multiple ideas in parallel, considering alternatives, sketching and prototyping ideas.

Lindberg et al. (2011) describe that this process is iterative and focused on learning and exploring the problem, solution, and problem-solution fit. The ideas represented through prototypes and sketches are used within the team and with users, customers, experts, and other stakeholders. Making information concrete and tangible in the solution space can be used to understand and further explore the problem space, which provides information

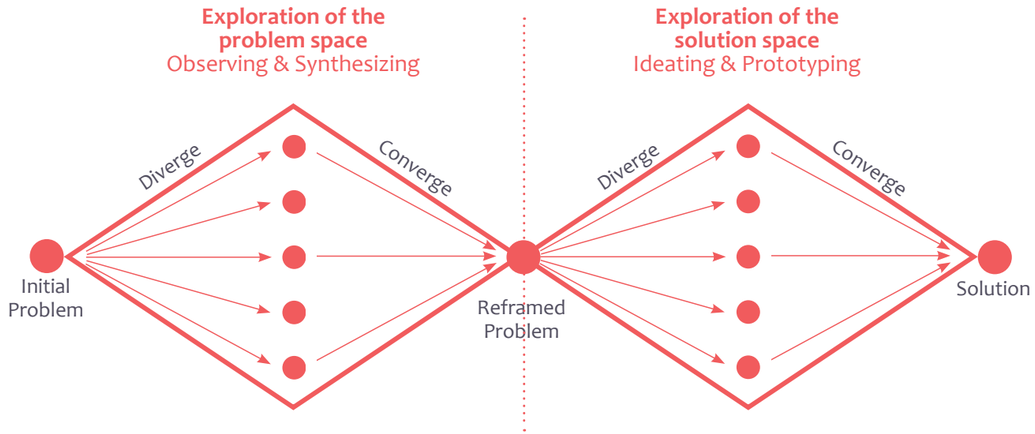


Fig. 2: Problem and Solution Space of the Design Thinking Process (based on Lindberg et al. 2011).

that can, in turn, refine and revise a chosen solution path (Lindberg, 2011). In this way, Design Thinking is not seen as a concept solely used by designers, but as a 'meta-disciplinary concept that broadens disciplinary reasoning and helps, for example, engineers to forget about the 'drawers' for a moment that they have internalized in their academic training - until a problem has been defined precisely enough so that professional rationales and expert knowledge may suitably be applied'. (Lindberg et al., 2011, p. 8)

Divergent and convergent thinking
The approach to explore both spaces can be described through divergent and convergent thinking (Lawson, 2006) - there is an interplay between a diverging phase of exploration of the problem and solution space, and a converging phase of synthesizing and selecting. In learning and broadening understanding and knowledge, possibilities open up that could be means of viable solutions.

2.1.4 Towards a Design Thinking Framework

In order to more holistically understand the concept of Design Thinking, and to build a framework around the different elements of the concept, the different elements will be explored. Design Thinking is described as an organizational resource in the light of three perspectives: as a mindset, a process, and a toolbox (e.g. Carlgren et al., 2016; Wölbling et al., 2012; Brenner, Uebernickel & Abrell, 2016).

Mindset
The mindset view is often described as a set of principles. Chesson (2017) provides a comprehensive overview of ten capabilities of design thinkers, made cursive in the text below, to describe the Design Thinking mindset (see fig. 3 for an overview). He notes that these capabilities are not absolute and can be further developed through practice.
As a mindset, Design Thinking inspires a *human-centered view*: the user is placed at the center of attention (Chesson, 2017) and there is a

strong focus on obvious and hidden needs of customers and users. There is a continuous focus on the fit of potential solutions with the needs of end-users (Owen, 2005). Linked to this, an empathetic mindset (empathy is the core value of human-centeredness) is useful to establish an understanding of the user and user needs and for working in diverse teams (Carlgren, Rauth & Elmquist, 2016). *Empathy* allows for seeing situations from multiple/ another's perspective(s) and for imagining solutions that fit user needs (e.g. Benson & Dresdow, 2015).

Next to this, Design Thinking requires a *dynamic mindset*: an iterative mindset that allows for shifting between inventive thinking (generating ideas, thinking about future possibilities and 'what could be') and analytical thinking (evaluating ideas) (e.g. Martin, 2009). This mindset includes being comfortable with ideas evolving over time through feedback and moving between the different modes of thinking.

Reflection is a related critical aspect, allowing to move between creating and reflecting and implementing feedback and insights to iterations. Viewing the process as iterative, design thinkers use prototyping to make ideas tangible in order to gather feedback. It allows for experimentation with efficient resources, making it relatively safe to explore ideas. The prototype should be tangible in order for people to offer their insights concerning the potential solution (Brown, 2008). The prototyping mindset recognizes the incompleteness of prototypes purely designed for exploration.

Other characteristics are *embracing failure* and *openness to risk-taking.* Design thinkers question the status quo in order to find new opportunities, taking risks by proposing new , unproven ideas for discussion. Related to this, failure is viewed as part of the problem-solving process (e.g. Liedtka, 2011) and information gained is used to iterate on an idea. In exploring the unknown, *comfort with ambiguity* is required. This interplay between what is known and what could be (Fraser, 2007), requires the courage to move forward without being in control or having complete information.

Solutions are co-created through a *collaborative* approach. Knowledge is acquired by engaging with clients, users, stakeholders, peers (Martin, 2009) to create a shared understanding.

Sidebox

Common bottleneck: Lack of mindset

Implementing Design Thinking often fails because companies or teams practice Design Thinking without committing to the Design Thinking mindset (Dunne, 2018). In such cases, Design Thinking is treated as a linear, step-by-step efficiency process and too little time is spent gaining a deep understanding of the problem, thus jumping too quickly to the solution space of the process (Koh et al., 2015). The Design Thinking mindset is argued to be the most crucial element in the Design Thinking approach (Hassi and Laakso, 2011) and practicing Design Thinking without the mindset is doomed to fail (Kimbell, 2012; Schweitzer et al., 2016).

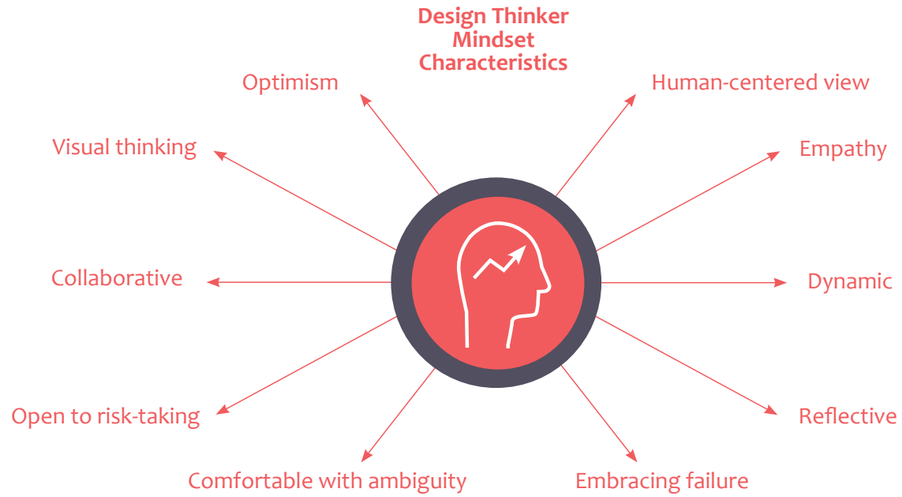


Fig. 3: Overview of eleven capabilities to describe the mindset of a design thinker (Chesson, 2017)

Collaboration is also about being open to and integrating new perspectives.

Another aspect is *visual thinking,* which has two components. First of all, design thinkers imagine solutions and situations of 'what could be' and conceptualize not yet existing solutions. Secondly, to communicate, conceptualizations are brought to life (Junginger, 2007), e.g. through sketching. Note that the purpose of sketching is not to create a beautiful sketch, but rather to roughly visualize to get ideas out of mind. It does not require drawing skills or training, and relies on basic shapes to convey ideas.

Finally, a certain degree of *optimism* is required to commit to finding better alternatives. Problems are approached with an attitude that there is at least one potential solution that will be able to transform a situation into a more desirable one.

Process

As a process, Design Thinking can be seen as a set of defined steps guiding the exploration of the problem and solution space. For the sake of simplicity, the phases are presented consecutively. However, they must be regarded as highly interconnected, iterative and non-sequential, allowing to go back or forth in the process when necessary (e.g. Wölbling et al., 2012). Different variations of the model are presented to define the process (see table 1). While there are some differences between the models, all describe three main stages of the process to varying degrees: understanding, idea generation and experimentation (Chasson, 2017), e.g. compared to Brown's model, other processes differentiate one step into multiple. The terminology of the Stanford d.school model will be taken as the focus of interest, and as a base for further elaboration on the steps. This model has been associated with agile practices (e.g. Pereira & Russo, 2018).

Below, the steps will be briefly described

according to Plattner (2009), see figure 4. A more elaborate description, including why and how to use each phase, can be found in appendix 2.

Empathize - In the 'empathize' phase, the aim is to familiarize oneself with the topic and the context, and identify and understand the main stakeholders in the context of the problem brief. Who are they, what do they do and why? What are their needs, pains and gains, and what is meaningful to them?

Define - This phase is all about sense-making and bringing clarity and focus, based on all the gathered (and scattered) insights and learnings about the users and context, in order to define the challenge. The goal is to scope down to a reframed problem brief, a 'meaningful and actionable problem statement' or as Plattner (2009) calls it: a 'point-of-view', which guides the solution exploration process.

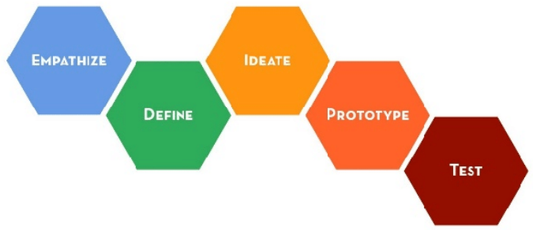


Fig. 4: The Design Thinking Model
-Image by Stanford d-school.

Ideate - This phase is about idea generation. Before converging to a particular solution, the aim here is to diverge in terms of concepts and solutions without judgment (prevent solution fixation and postpone judgment). Understanding of the problem space and users is combined with creativity and imagination to get to innovative solution concepts.

Prototype - *'The Prototype mode is the iterative generation of artifacts intended to answer questions that get you closer to your final solution'* (Plattner, 2009, p. 5). Early in the process, that could be quick prototypes to learn about broader questions through initial feedback from users. Throughout the process, both questions and prototypes may get more refined. The prototype itself can be anything the user can interact with (and ideally experience), e.g. wireframes, a wall of post-its, a sketch, a role-playing activity, a storyboard, etc.

Test – The testing phase is all about getting feedback and learning about the solution and the user. There is a focus on user interaction, and by continuing to ask 'Why?' a great deal can be learned about the problem, as well as potential solutions and the users themselves.

Table 1: Design Thinking models

Design Thinking models	Phases and activities						Source
IDEO, International Design and Consulting Firm	Inspiration		Ideation		Implementation		Brown, 2008
Institute of Design at Stanford (Standford d.school)	Empathize	Define	Ideate		Prototype	Test	Plattner, 2009
International Organization for Standardization (ISO): Human-centered design for interactive systems	Understand	Specify			Produce	Evaluate	DIS, 2010
Google Ventures - Design Sprint	Understand		Sketch	Decide	Prototype	Validate	Google, 2008
Design Council UK: Double Diamond	Understand	Define	Explore		Create		Design Council, 2018
Wölbling Design Thinking process for developing software	Understand	Observe	Point of view	Ideate	Prototype	Test	Wölbling et al., 2012
Brenner et al. Design Thinking micro process	Needfinding	Synthesis		Ideate	Prototype	Test	Brenner et al., 2016

Table 1: Mapping the phases and activities of different Design Thinking models

Toolbox

As a toolbox, Design Thinking refers to a set of techniques and methods that support the defined process steps. This third and 'operational layer' guides the practitioners of Design Thinking towards a solution to a problem by giving multiple ways to facilitate a step in the process. Well-known tools are the persona and user journey map, for example. According to Brenner et al. (2016), it is critical for the success of the DT project to deploy appropriate methods. Well-known examples of available toolkits aimed at practitioners are 'This is Service Design Thinking' (Stickdorn et al., 2018) and '101 design methods: A structured approach for driving innovation in your organization' (Kumar, 2012).

In the framework (see fig. 5), the three perspectives are brought together, including examples of tools for each phase.

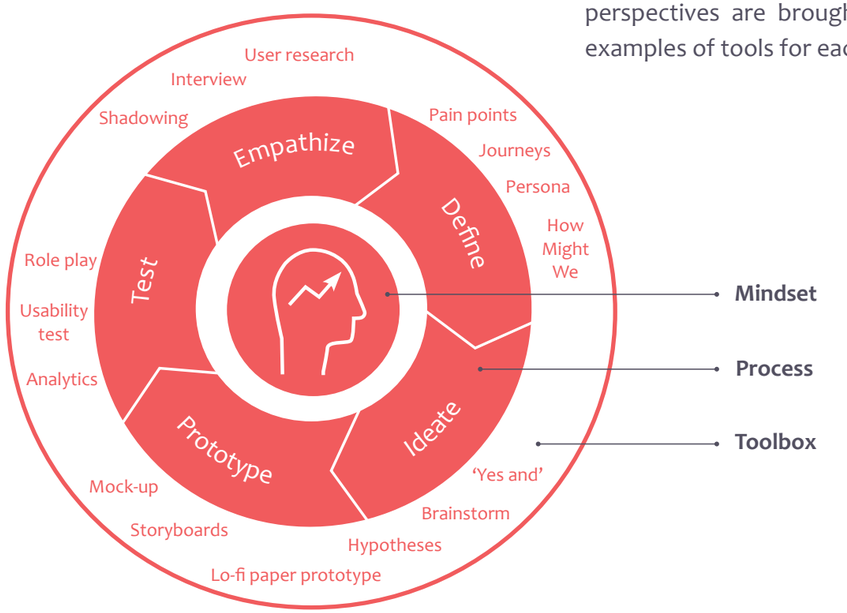


Fig. 5: Design Thinking framework - Design Thinking as a mindset, process and toolbox

2.2 Agile Software Development

In order to understand the team's way of working, this chapter focuses on the software development process. The Software Development Life Cycle will be described first. The development teams implement this cycle in an 'agile' way of working; another focus point in this chapter. In chapter three, typical scenarios of the Operations Tech unit will be placed within this cycle according to their 'maturity' stages.

2.2.1 The Software Development Life Cycle

The Software Development Life Cycle (SDLC) is the process required to build any software see (fig. 6). Whether following a traditional waterfall or iterative model, these phases/stages will be present in the process. There are different models that describe the complex lifecycles in various levels of detail. In this research, the phases described by Nigam & Gupta (2017) will be taken as a foundation to further analyze the SDLC within the unit of focus, and to place common scenarios (see §3.2.2) into the SDLC context.

These phases are:

1. Communication/Initiation: the need for or requirement of the software is initiated by the customer, user, the development team, or another stakeholder

2. Requirement Gathering: the information about customer demand and needs are gathered to find out the customer requirements

3. Feasibility Study: the project is analyzed in terms of technical, practical and financial feasibility, to decide if the project should continue

4. System Analysis: the system is analyzed to find out limitations, impact on the organization and the scope and resources are planned accordingly

5. Software Design: the software is designed 1) in terms of logic of metadata and flowcharts etc. and 2) in terms of the physical look of the software

6. Coding: the logic is implemented and the code is written in a certain programming language

7. Testing: the software is tested using different testing criteria (e.g. structural-, function-, unit-, system-, alpha-, beta-, stress-testing etc.)

8. Integration: different software modules are joined with each other and/or outer resources

9. Implementation: the software is implemented so end-users can use it. This includes adaptability checks and onboarding (user training) as well

10. Operation & Maintenance: the software is checked and maintained over a period of time, errors are updated when found

11. Disposition: the software is of no more use, very old and becomes obsolete. After which a new product lifecycle will start

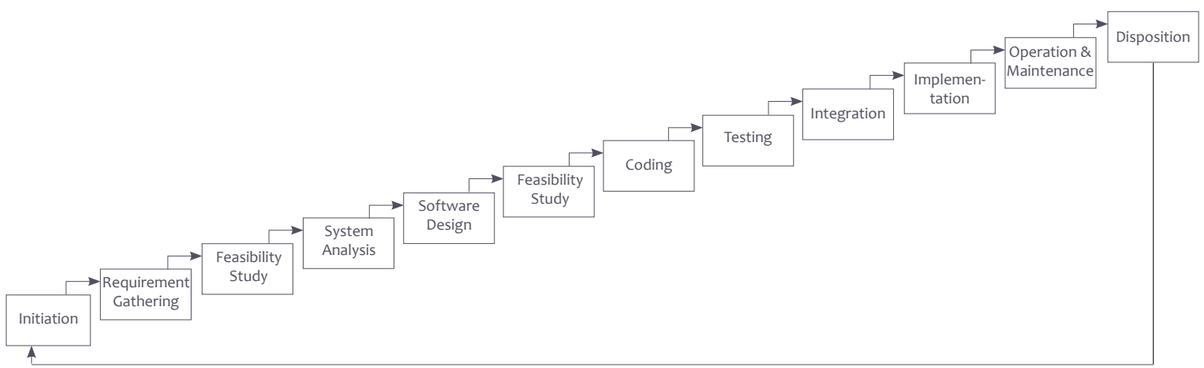


Fig. 6: Software Development Lifecycle

2.2.2 Heavyweight versus lightweight software development methods

According to Cho (2010), software development methods can be classified as either a heavyweight method or a lightweight method. Traditional software development methods, also called plan-driven methods (Girma et al., 2019) or milestone-based methods (Lindberg et al., 2011), are considered heavyweight. Generally, their focus lies on heavy documentation and planning and design is largely or completely done up-front. These methods provide high predictability, stability and assurance (Boehm & Turner, 2003) and there is a strong focus on clarity and fixation of what is going to be developed. Traditional methods take the Software Development Lifecycle as a linear process in which phases are passed through in a predefined order.

A core criterion to complete a phase is to make conditions for the next steps explicit through in-depth documentation (Lindberg et al., 2011). The 'Waterfall model' is an example of this.

Contrary to heavyweight methods, lightweight methods, also called agile methods, recognize the need for adaptability along the way of the project as environments, therefore requirements might change (Mishra & Mishra, 2011). This adaptability is important as 'the ability to take appropriate action in response to a change often determines the success or failure of a software product' (Williams & Cockburn, 2003; Girma et al., 2019, p. 34). Requirements cannot always be known in advance and might not be as stable as heavyweight methods require (Boehm & Turner, 2003); a key difficulty in gathering requirements is the discovery and fulfillment of changing and unarticulated needs of various stakeholders (Hehn et al., 2020).

In the case of lightweight methods, requirement analysis does not precede actual development as is the case with heavyweight methods, but it is a parallel process to development and software increments developed serve to get feedback from experts and users (Lindberg et al., 2011).

A commonality in lightweight methods is that the software development process is seen as empirical, or non-linear; within the Software Development Lifecycle, the activities are executed iteratively, multiple times and in various orders, or simultaneously.

Lightweight methods have been proposed with the aim to address shortcomings of heavyweight models (Highsmith & Cockburn, 2001), which include 'slow adaptation to constantly changing business requirements' and the tendency to be behind schedule, over budget and to meet fewer requirements than specified (Schach & Schach, 2004; Williams & Cockburn, 2003).

2.2.3 Agile Software Development

To find an alternative to the heavyweight methods, practitioners convened and set up the 'Manifesto for Agile Software Development' (Beck et al., 2001). The manifesto emphasized four principles:

- *Individuals and interactions* over process and tools
- *Working software* over comprehensive documentation
- *Customer collaboration* over contract negotiation
- *Responding to change* over following a plan

It comes with twelve 'principles behind the Agile Manifesto' (see appendix 3).

The agile philosophy focuses on simplicity and speed, delivering critical requirements fast, and testing and getting feedback fast and often to react to changes in the business and technology environment (Abrahamsson et al., 2003).
The practices are lightweight and product-oriented, focus on iterations and frequent re-planning, and rely on self-manageable teams (Highsmith, 2002).
Moreover, the emphasis is placed on short development cycles and frequent delivery and customer interaction. The practices are proven to improve the quality and speed of development activities and to increase customer satisfaction (Baseer et al., 2015). Benefits from the perspective of the customer are increased product quality and value and a better relationship with the stakeholders involved (Solinski & Petersen, 2016). For agile practitioners, there is a broader knowledge exchange in which collaborators can learn from each other, bringing more satisfaction, development of social skills, constant feedback and the trust of professionals (Solinski & Petersen, 2016).

Within this philosophy several methods and processes emerged to implement agile practices. Well-known methods are Scrum (Schwaber & Beedle, 2002), eXtreme Programming (Beck & Andres, 2004), Lean Software Development (Poppendieck & Poppendieck, 2003), and Kanban (Anderson, 2010). Predominantly used Agile methods are Scrum, hybrid Scrum and eXtreme Programming (XP) (Sommerville, 2011; Vallon et al., 2018).

Initially, agile practices were aimed at small collected teams (Corral et al., 2015), but because of the success on small scale projects, Agile Scaling frameworks have been introduced such as SAFe and LeSS (Leffingwell, 2007; Kalenda et al., 2018) to support large enterprises to scale agile, and the methods have been proven to be valuable in the context of large enterprises and complex products (Bass, 2015; Lindvall et al., 2004). However a significant amount of research has been done in this direction (e.g. through case studies), scaling factors are still an open research area (Dingsøyr & Moe, 2014; Girma et al., 2019; Thompson, 2013).

2.2.4 Scrum & Kanban

This paragraph describes Scrum and Kanban, as those methods are currently followed in the Operations Tech unit (see §3.1.6).

Scrum

Scrum emphasizes the short development cycles by working in multiple subsequent development sprints in which software is developed iteratively and in increments (see fig. 7). Sprints are time-boxed efforts, usually of two to four weeks, which start with a planning meeting and end with a demo of a software increment. The method is highly dependent on self-organized teams. The Scrum framework focuses on the management of the development process (Mann & Maurer, 2005) and consists of three main pillars, namely roles, ceremonies and artifacts (Schwaber, 2004):

Key roles – Product Owner, Scrum Master, and the Team (Schwaber & Beedle, 2002)
Key ceremonies – backlog refinement, sprint planning meeting, daily stand-up, sprint review (demo) and sprint retrospective
Key artifacts – product backlog, sprint backlog, product increment and a definition of done

The term 'Scrum' is derived from a rugby game, in which a play is restarted and the ball is passed back and forth in a clustered mass of players (Livermore, . The scrum formation was compared to cross-functional teams in the Harvard Business Review in 1986 (Takeuchi and Nonaka, 1986), after which Schwaber and Sutherland used the term for their proposed software development methodology in 1993 (Sutherland & Schwaber, 2007).

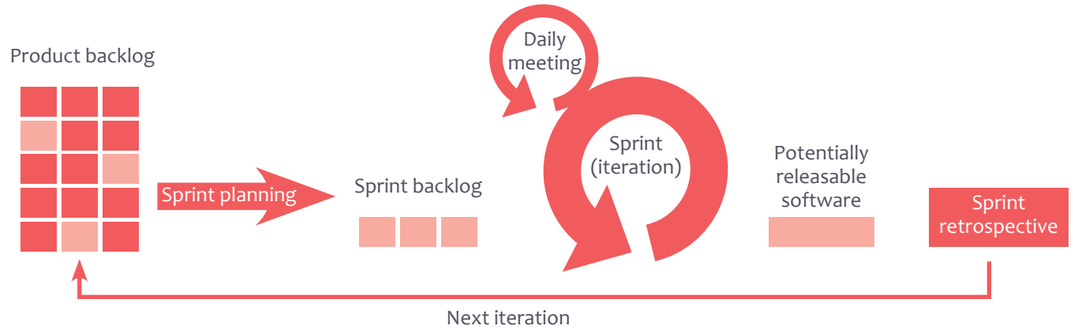


Fig. 7: Scrum process overview (based on Correa, 2008)

Kanban

The Kanban method is a process model that can be implemented to suit each context (Kupiainen et al., 2015). Kanban systems are 'pull systems' (Poppendieck & Poppendieck, 2003): when there is capacity for it, work is pulled from a backlog to development, instead of work being pushed into development. The work is split into pieces and written on physical or digital cards, and the workflow is visualized by assigning cards to the column describing where that item is located in the workflow (e.g. 'To Do', 'Doing', or 'Done', see fig. 8). There is a limit to how many items can be assigned to each work state. (Kniberg & Skarin, 2010)

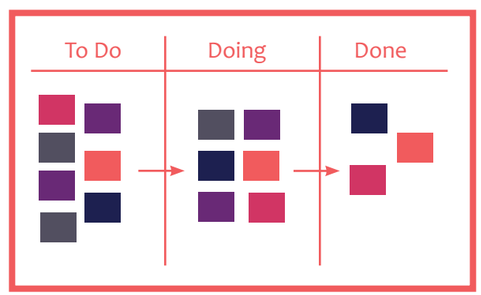


Fig. 8: kanban method visualizing the workflow

2.3 The complementation of Design Thinking in Agile Software Development

2.3.1 Complementary approaches

Hildenbrand & Meyer (2012) argue that Agile practices and Design Thinking complement each other within the same project, as certain goals and values are shared between those methods. Their case study sheds light on how to leverage both Agile and Design Thinking, investigating how to come up with a product vision and requirements in an unknown domain. According to Hildenbrand & Meyer (2012), business software companies not only need a structured framework on how to turn the right ideas into viable products, but also on how to come up with those 'right ideas' in the first place. As discussed before, Design Thinking brings such a framework and aims to increase the likelihood and fit of innovations and ideas by integrating feasibility, viability and desirability (Brown, 2009; Martin, 2009). As Hildenbrand & Meyer (2012) concisely put to words:

> *"While lean thinking and agile practices help organizations to build and ship products right, meaning in time and in quality, Design Thinking focuses on building the right product in the first place"* (Hildenbrand & Meyer, 2012, p. 219).

They write that it is the development of understanding of the problem space (e.g. context, user and relevant stakeholder understanding) that can guide teams in developing a product vision and deriving requirements (Hildenbrand & Meyer, 2012). Agile thinking does focus on customer value, however, the framework does not provide any guidelines or principles on 'how to find out what is actually valuable to the customer' (Womack & Jones, 1990, 2003). Similar to this problem, a 'product vision' is assumed as a starting point in Agile methodologies, though where that vision comes from or how to develop it is not included (Highsmith, 2009; Pichler, 2010). Considering and introducing practices that focus on empathy and in-depth (human-centered) understanding of the problem could therefore be beneficial.

Unlike Agile methodologies, Design Thinking suggests to spending sufficient time in the problem space, observing and researching users to better understand the context, customer and user needs and pains, developing empathy for users and taking a user perspective. Agile methods for requirement engineering (Hildenbrand et al., 2008) usually do not distinguish the problem space (as-in scenario) and solution space (to-be scenario), whereas Design Thinking clearly separates those two (e.g. Meinel and Leifer, 2011).

"While lean thinking and agile practices help organizations to build and ship products right, meaning in time and in quality, Design Thinking focuses on building the right product in the the first place"

In line with this, Lindberg et al. (2011) mention the complementation of Design Thinking in agile IT development with regard to the following aspects:

- **Building on diversity:** Whereas *'strong team-based collaboration is a core feature of agile development'* (Lindberg et al., 2011, p. 12), Design Thinking builds on disciplinary diversity, implementing different styles of thinking as well as Design Thinking on a meta-disciplinary level.

- **Exploring the problem space:** Design Thinking supports in creating an understanding of the problem, user and context prior to actual development, whereas for software engineers, the voice of the user is often presented through technical specifications. Even when these specifications are validated, the broader picture to deepen understanding and to allow ideation is still missing.

- **Exploring the solution Space:** Considering alternatives, ideation and conceptualization to make ideas tangible are not included in the agile development models. The focus on incremental progress of agile methodologies tends to limit divergent thinking, restraining the potential tendency of agile methodologies to explore alternatives.

- **Iterative alignment of both spaces:** Both agile and Design Thinking focus on learning through iterative user feedback based on prototypes. The main difference is the focus on the iterative learning about the problem space in Design Thinking, e.g. quick prototypes are made with the intention to learn about the problem space, allowing for more extensive exploration of both spaces in Design Thinking.

The author elaborates on the divergent thinking part mentioned in the 'exploring the solution space'. In Agile, the goal is known and the focus is on incremental refinement rather than exploring and considering new solution paths. In the research done by Lindberg et al. (2011), one of the interviewees puts this contradiction to words as follows:

> *"In Agile, you downsize the problem so that they're actually small enough that people can deal with it and make progress and don't get lost. But that's a very constraining technology. (…) Agile is always looking to remove options from the table. Design Thinking is always trying to keep options on the table as long as possible."* (Lindberg et al., 2011, p. 11).

In Agile, divergent thinking is rather avoided to sustain efficiency and overview of next tasks. Consequently, "the whole aspect of problem understanding in Design Thinking is limited down to trial and error approach of iterative prototyping. This is why the focal goal of Design Thinking to put divergent options on the table will hardly be achieved." (Lindberg et al., 2011, p. 11). To achieve an actual expansion of thinking styles, the way software development projects are managed will have to be reconfigured. It may include adding design specialists to the software development process.

Note that adding a user-experience (UX)

designer to a development team does not guarantee shared core problem understanding or ideation integrating different perspectives of e.g. engineers, data analysts, architects and other relevant experts. Next to adding design specialists e.g. user-interface design (Mandel, 2009), interaction design (Dix et al., 2003), user-experience design (Buxton, 2007) serving different demands for design in a software development project, which may vary per project, "Design Thinking aims at influencing people meta-disciplinary" (Lindberg et al., 2011, p. 12). While adding designers does seem to support the meta-disciplinary adoption of new ways of thinking (e.g. see the IBM case chapter §2.4.1), it is important to be aware of the different purposes behind adding a UX specialist and aiming for Design Thinking adoption within the team. In this graduation project, the focus is on the meta-disciplinary perspective of Design Thinking besides the needs for specific design specialists.

The software developers are not the users of the final product themselves, which does require empathy to understand the customer and user needs and discover opportunities for innovation. According to Hildenbrand & Meyer (2012), Design Thinking can support to develop and choose the right backlog items and user stories, through building an understanding of and empathy for the customers and users and their context. In this way, time investment upfront and the fast feedback cycles reduce the project risk, which is high in the beginning due to the number of unknowns.
Even while still investigating the problem space, continuous prototyping is suggested to fail early and often (Brown, 2009), in pursuit of learning and iteratively creating a better understanding of the problem and the solution space. Prototypes are build for a specific reason and are often thrown away after they served their purpose of e.g. testing assumptions, clarifying the problem, leveraging certain feedback on possible solutions or solution directions, supporting conversations or inspiring better ideas. Moreover, the process is iterative; Design Thinking recognizes that certain user insights might require iterations, or possibly even a complete restart or what has been done so far (Brown, 2009; Ries, 2011).

From a Design Thinking perspective, Design Thinking does provide tools to build empathy with end-users, to develop a solution idea or vision, and includes prototyping to find the right product to build for the end-user. However, it lacks clarity, tools, and steps to take the vision from a prototype to an actual (scaled) finished product. This shows the complementary value of Design Thinking and Agile practices. Especially for complex software products, and especially in large organizations, structuring the actual realization step is a must.

2.3.2 Implementing Design Thinking prior to Agile Software Development

A core finding of Lindberg et al. (2011) is that Design Thinking is perceived as a risk by those reporting to higher hierarchy levels. Employees are generally evaluated on their scientific reasoning, tight budgets and time schedules, working with scarce resources etc., which conflict with the uncertainty of divergent thinking. Convergent thinking is more secure to employees, even if the outcomes might not be as innovative as they could be. This is why, in established companies, Design Thinking is often integrated as a separate or front-end approach, separate from the Agile Software Development process. The more Design Thinking is related to the fuzzy front-end of the development process, the easier the implementation, as conflicts with existing (e.g. reward, reporting and controlling) processes and structures are limited (Lindberg et al., 2011). The authors suggest companies that pursue an agile approach to start with design-thinking inspired concept development prior to actual agile software development. They mention that a front-end approach might be the ideal form for agile development processes as aligning the problem and solution space up-front would meet the agile logic through team communication, iterative learning, integration of user feedback etc.

When Design Thinking is done seperately, the difference between knowledge flow and communication media used in agile software development and Design Thinking, mainly related to the different vision with which prototyping is used, is found to be challenging (Lindberg et al., 2011). In Design Thinking, prototypes are made with the purpose to learn about the underlying product concept (problem and solution space), often through testing and evaluating mock-ups. These mock-ups, or quick prototypes, can be experimental and made in any quick and cheap way that allows learning about the ideas behind the concept (not about technical specifications yet). Agile software prototypes, on the other hand, are generally already made in the same tool as the final product, and are iteratively improved into the final product. The aim is not so much to learn about the product idea or problem, but to find smooth ways to build the product in the right way.

The ability and way to integrate Design Thinking throughout the development process depend on many factors and still open questions. There is little research addressing these factors and questions.
Organizational structures such as stage-gate models with predefined workflows constrain explorative and creative thinking. The degree to which a company is open to an entrepreneurial approach, opposing the need for controlled processes and resource flows has a significant impact as well. Integrating Design Thinking throughout development would require employees to concentrate on only one project at a time.
Next to that, quality and controlling measures that discourage divergent thinking need to change to encourage Design Thinking. (Lindberg et al., 2011)

2.4 Case studies

To learn about how other companies view (human-centered) problem exploration and, for example, codify Design Thinking to their needs, insights from three cases - Slack, IBM and Spotify - will be provided in this chapter. For more context, see appendix 4.

There is a lack of information and cases about companies adopting Design Thinking focused on internal products. The companies analyzed below, mainly focus on external customers.

In these cases, Design Thinking is often applied to lower the risk to of losing customers to the competition. In the internal software development case, it is not directly about losing the users and internal dependencies play a large role. Though, impact in terms of innovation and effectiveness of products 'building the right thing', and for example user satisfaction do apply. This is why it is still relevant to look at companies that apply Design Thinking to externally focused products.

Image source: IBM

2.4.1 IBM Enterprise Design Thinking

In 2013, global technology company IBM started a large-scale project to develop a design-driven culture, called 'IBM Enterprise Design Thinking'. The project took over three years, involved over 750 designers and affected over 10,000 employees (Azis, 2016). The company has adapted Design Thinking to fit their Agile Software Development process (Lucena, 2016), which makes it an interesting case. To bring the core focus areas within IBM's context to the employees, the tailored design process is visualized in 'The Loop' (see fig. 9), which comes with main steps, principles and keys (see appendix 4 for more in-depth information about this case and the vision of IBM behind design, Design Thinking and their learnings). Key insights from this case for Nike can be found on the next page:

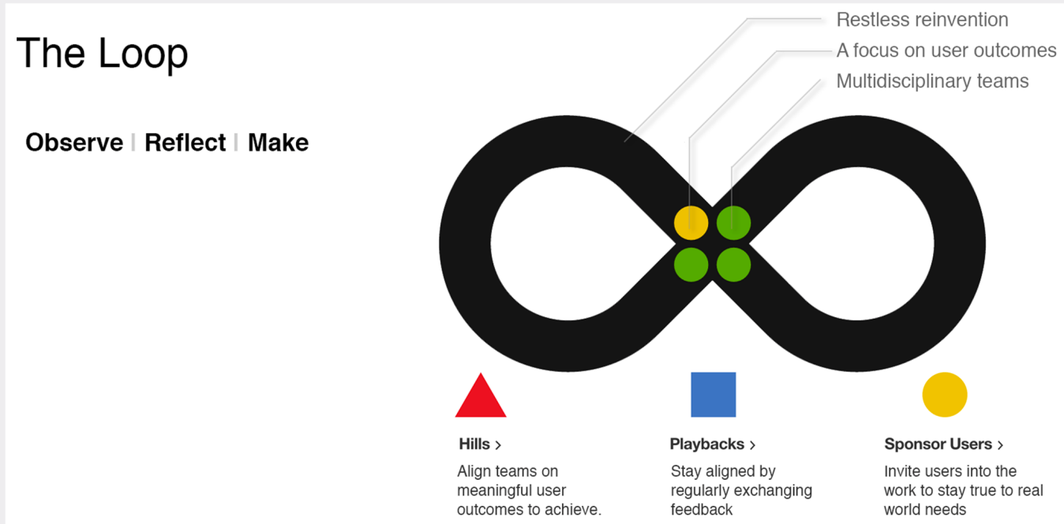


Fig. 9: IBM Enterprise Design Thinking 'The Loop' including steps, principles and keys (source)

IBM - Key Insights

- A focus on user-centered goals is required instead of output-centric goals. To get this right from the start, IBM provides a format to employees to set user needs as project goals, making the offset of a project human-centered at the core.
- The trick is to balance team efforts between discovery and delivery and manage a workflow that integrates both activities.
- IBM recognizes the difficulty to break the agile rhythm and tight delivery schedule in order to try Design Thinking methods. They therefore recommend to start small and to integrate the Design Thinking activities (e.g. user research) into the current sprint plan ('hybrid sprint'). This will help team members to see how Design Thinking can help in connecting user needs to business goals.
- IBM views Design Thinking as a mindset that can be adopted by everyone, and in every stage of the process. Employees need education with regards to this.
- To accommodate the cultural shift towards a more design-driven culture within IBM, more designers are present in the project teams. They went from designer:developer ratio 1:72 in 2012 to 1:8 in 2017 (Chicoria, 2018).

2.4.2 How Spotify builds products

The core philosophy of audio streaming service Spotify is to manage risks by prototyping cheap and early. They divide the development process of their main initiatives into four stages: a 'Think it', 'Build it', 'Ship it' and 'Tweak it' stage, developed with the biggest risk of building the wrong product in mind. (Kniberg, 2012)

Spotify - Key Insights

- In the 'Think it' stage, product risk can be reduced at low costs (see fig. 10). As there currently is no stage in which quick prototypes are made prior to development in the Functional Tech team's process, there is an opportunity to make most out of this.

- Spotify defines the 'wrong product' as 'a product that doesn't delight our user, or doesn't improve our success metrics such as user acquisition, user retention, etc. We call this 'product risk'.' (Kniberg, 2013, p.3). Note that this company targets external consumers, therefore these success metrics are different in comparison with internal consumers (e.g. user acquisition does not apply). It is important to understand how the Nike Functional Tech team does/should define 'the wrong product'?

- After every stage, the product can be rejected if it will never be good enough for users, iterated if it is not yet good enough, or continued to the next stage if it is good enough. The notion that rejection is an option provides freedom to experiment.

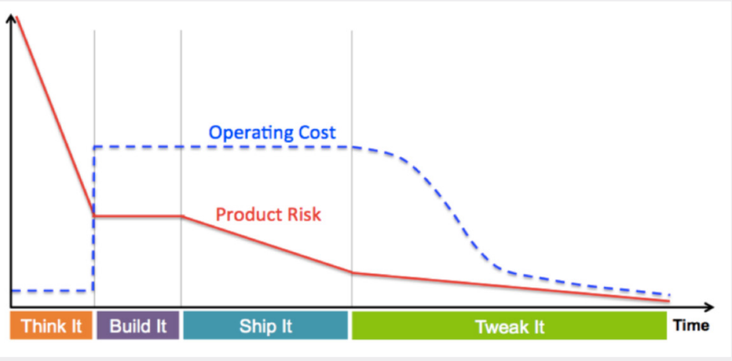


Fig. 10: Spotify's four stages to manage product risk. Note that the first stage significantly reduces the risk at low costs. (Kniberg, 2013)

2.4.3 Human Centricity at Slack

Slack Technologies, Inc. is a software company that develops a worldwide communication platform for teams (Bloomberg, 2021). At The Next Web conference, Tamar Yehoshua (Chief Product Officer at Slack) talked about Human Centricity at Slack. Next to the Design Thinking tools used at slack, she emphasized the role of top-down management and the measurement of impact. My notes of this session can be found in appendix 4.

Slack - Key Insights

- Tools like personas and user journey maps are used by Slack to change the mindset of how a product is developed. Next to that, they give a shared language and bring in the voice of the customer into the process.

- User research can be seen from three perspectives: usability (e.g. user experience studies), foundational (influencing product strategy), validation (done quick and early to improve velocity)

- In large companies, top-down management plays an important role in setting the tone for human centricity and taking accountability for some loss of velocity.
  (Different) metrics apply to incentivize the right human centered change

2.4.4 Sub-conclusion cases

Looking at those case studies, there are two common themes that these companies address. The first theme is the human-centered mindset, which is addressed through the format of articulating the project goal (IBM), recognition of education around the mindset (IBM), teaching different tools IBM, Slack), integrating different user research perspectives (Slack), and the notion that encouragement for human centricity is needed from top-down management (Slack).
The second theme is the clear separation between the exploration or discovery phase and the delivery phase and the focus on implementing both (e.g. through hybrid sprints and conscious workflow management of IBM, and making the most of reducing the product risk at low costs in the exploration phase at Spotify with a gate between exploration and delivery).

2.5 Chapter conclusions

Design Thinking
There is no universally accepted definition of 'Design Thinking'; it is seen as a contextual concept that has to be defined based on the context of use. In this graduation project, Design Thinking is considered from the perspective of two management discourses, namely 1) as a (IDEO's) way of working with design and innovation, and 2) as a way to approach indeterminate organizational problems and a necessary skill for practicing managers. Taking these discourses into account, Design Thinking is described as "*a style of thinking that combines empathy for the users and immersion in the context of a problem, creativity in the generation of insights and [alternative] solutions and a data-based experimental approach to assessing the quality [and fit] of solutions*".
The core concepts behind Design Thinking are problem and solution exploration, and divergent and convergent thinking. Moreover, Design Thinking is described as an organizational resource in terms of three perspectives: as a mindset, a process and a toolbox. In this chapter, a framework is presented integrating those perspectives.

Agile software development
The Software Development Lifecycle can be implemented through heavyweight (traditional, linear) methods with a focus on heavy documentation and fixed planning, or lightweight (agile) methods that recognize the need for adaptability along the way of the project.

In agile software development emphasis is placed on short development cycles (iterations) and frequent delivery and feedback. The Most popular methods to implement agile are Scrum, hybrid-Scrum and eXtreme Programming.

Complementary approaches
Literature shows the complementarity of agile methods and Design Thinking: while agile methods focus on developing a product efficiently and in the right way, Design Thinking focuses on problem understanding and divergent thinking to be able to build the right product in the first place. Design Thinking can support teams in building diversity, exploring the problem space, exploring the solution space and iterative alignment of both spaces.
Implementing Design Thinking can be perceived as a risk as the uncertainty and divergent thinking that come with Design Thinking conflict with the convergent thinking and focus on short-term efficiency in terms of budget and time that employees are generally evaluated on. The ability and best way to integrate Design Thinking depends on different factors that restrain/allow or encourage/ discourage Design Thinking. In established companies, implementation of Design Thinking as a front-end process is found to be easiest as conflicts with current processes and structures are limited as the two purposes can be clearly separated. However, it is still an open question when and how to optimally support agile software development with Design Thinking.

Different companies leverage Design Thinking, adapted to their needs and context. Case studies of IBM, Spotify and Slack show a focus on the user-centered mindset of employees and the organizational aspect of intentionally separating the exploration/discovery and delivery phase.

To create an understanding of the elements of this chapter in practice, the next chapter will focus on the internal context of the Operations Tech team.

Chapter 3

Context Analysis

Context of focus within Nike Inc.

In this chapter, the focus is on understanding and exploring the context of the Nike Operations Tech unit. The company context will be briefly described, as well as the context of the team within the company. Next to that, their agile way of working including common products and problem-solving scenarios will be explored, as well as the current process of 'deciding what to build' prior to actual development.

3.1 Company & Team Context

3.1.1 The company

Nike Inc. is an American multinational company and the largest supplier and manufacturer of athletic footwear, apparel and other sports equipment in the world (Statista, 2021a), with revenue exceeding US$37.4 billion in fiscal year 2020 (Statista, 2021b). Founded in 1964 as 'Blue Ribbon Sports', Nike currently operates a portfolio of three different brands, namely Nike, Jordan and Converse.

With 75,400 employees worldwide, as of 2020, the company operates in well over a thousand brick-and-mortar retail stores (respectively Statista, 2021c and 2021d). The world headquarters are located in Beaverton, Oregon in the Portland metropolitan area in the Unites States and the European headquarters are located in Hilversum, the Netherlands.

Nike store (Source: Shutterstock)

- Nike's Mission -

BRING INSPIRATION AND INNOVATION TO EVERY ATHLETE* IN THE WORLD

*IF YOU HAVE A BODY, YOU ARE AN ATHLETE.

3.1.2 The organizational structure

Two key characteristics of the organizational structure are its matrix structure and hierarchy. In order to understand the context of focus in this graduation project and how the teams function, both will be addressed briefly.

Nike functions as a "*collaborative, matrix organization, where team members often report into two areas, such as a geography and a global function.*" (investors.nike, 2019, 'NIKE, Inc Management', para. 1) A matrix structure is a hybrid organizational structure that includes two or more distinct hierarchies (Davis & Lawrence, 1977; Mee, 1964) and both vertical and horizontal management, e.g. information flow is managed vertically towards departmental directors and horizontally between project and process managers (Radović-Marković & Omolaja, 2011). Organizations employing a matrix structure seek to simultaneously benefit from efficiency and flexibility, which comes with the cost of high internal complexity (Fjeldstad et al., 2012).

Nike is organized in (affiliate) brands, geographies, products and functional areas, managed both vertically and horizontally.

> "*In the NIKE brand, teams work across footwear, apparel and equipment product engines; our core consumer categories - action sports, basketball, football (soccer), men's training, running, sportswear, and women's training; and in our four geographies - North America; Europe, Middle East & Africa (EMEA); Greater China; and Asia Pacific & Latin America (APLA). Our NIKE, Inc. affiliate brands operate in a similarly collaborative way, as well as critical corporate functions.*" (investors.nike, 2019, 'NIKE, Inc Management', para. 1)

Within the matrix structure, hierarchy plays an important role. Hierarchy is used for control and coordination; higher-level employees have a broader view of the organization and its goals and environment and are therefore authorized to resolve conflicts and make decisions concerning lower levels (March & Simon, 1958). However, next to the benefits of control and coordination, managerial hierarchy in which there is a top-down flow of directives is known for its rigidity, which is disadvantageous for dealing with tasks that might require rapid change (Hamel & Breen, 2007) or addressing uncertainty (Lee & Edmondson, 2017).

3.1.3 Nike Technology

> "*Fundamentally, at this stage, Nike is a technology company. It's a technology company that builds upon its historical strengths in footwear design, storytelling and inspiration, and it's able to use those in combination to solve problems that no one else can solve*" - Michael Martin, vice president of Nike Direct products, growth and innovation (Witte, 2019, para. 18)

From the Nike website and mobile apps, to the development of products, management of big data and the engineering and systems that support daily operations, technology is a large driver within the business. In current 'digital transformation' times, in which digital shifts plays out in every company, the role of technology will become more and more influencial. As Microsoft CEO Satya Nadella notes: "The rise of digital IT creation in every organization means developers will increasingly drive and influence every business process and function" (Dignan, 2019, para. 6)

"Nike is a technology company"
(Nike (n.d.), 'Technology')

3.1.4 Context unit of focus

In general, there are four main functional areas of business within an organization, namely marketing, operations management, finance and human resource management (Jiang, 2009). Within the matrix organization, these functional areas intersect with the technology business unit (see fig. 11). In the context of this graduation project, the context of focus is a technology sub-unit located within the 'operations' functionality, i.e. concerning the supply chain, within the EMEA region (Europe, the Middle East and Africa) and is located in Hilversum, the Netherlands. For privacy reasons, the specific name and of the team will be anonymised to 'Operations Tech unit'.

The product management team is directly linked to four software development squads: two front-end application squads and two back-end data engineering squads. Each squad has its own product owner and scrum-master, and next to that the team includes 4 product managers who are involved with local and global teams. In total the team consists of around 30 people, both full-time employees and external temporary workers.

Although this functional technology unit is the main target and scope of this graduation project, this does not preclude that outcomes can not be applied to a more general target group working in software development.

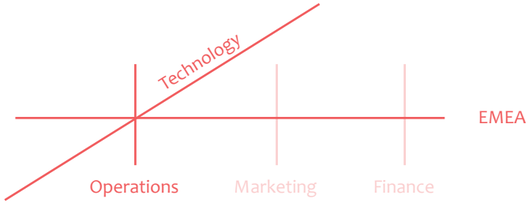


Fig.11: The unit of focus is located on the intersection of technology and the operations (concerning the supply chain) functional area, within the EMEA region

3.1.5 Type of products: enterprise software

Within the Operations Tech team, in general enterprise resource products related to demand forecasting, planning, allocation, pricing and optimization prior to and within the season are developed and maintained.
"*We have a subdivision in support work, tech hygiene which just has to happen, and developing new functionalities*" (P5).
Depending on budget and priority, some products in the suite of demand and supply planning tools/applications are in active development, others have a KLO status (Keep the Lights On), which means that the tools are kept in a maintained state without active further development.

The main users of the software tools are internal demand and supply planners who use the enterprise software to make sure that in the end, the right retail products are in

the right place, at the right time, in the right amount. Other users of the software include management, but also to vendors, accounts and employees working in stores for example.

In general, the tools developed are excel-like grids that include algorithms, logic, data and analytics, and have dashboard and reporting capabilities. A simplified representation can be found in fig. 12.

3.1.6 Agile way of working

The development teams are following the Software Development Lifecycle in an Agile way, iterative and incremental (see chapter 2.2.3 for more information about Agile practices).

When a request comes in to build a product or feature, basic or initial requirements are taken and prioritized to build a Minimum Viable Product (MVP). The MVP is built and tested to learn about more requirements along the way. To better understand the type of requests that the teams work on, chapter 3.2 elaborates on common development scenarios.
The customer is involved early on and the development team gets feedback and insight into errors and barriers throughout the development process, allowing for fast delivery and adaptation (see appendix 6 for a visualization of the process from the perspective of one of the product owners).

The development squads work mainly based on Scrum and have the freedom to adapt their way of working to their own needs. Squads predominantly focusing on one product are able

Products	KPIs	Total ($ or units)
Shoe type 1 Shoe type 2 etc.		

Fig. 12: simplified representation of a tool developed for the operations functional area.

to follow Scrum and work in short development sprints of two weeks. The first sprint is a PI-planning (Product Increment) sprint, an event coming from the SAFe framework (Scaled Agile Framework) which is dedicated to 'planning, building and validating a full system increment' (Scaled Agile, Inc., 2021, 'Program Increment', para. 2), answering what the teams will work on in the coming sprints and aligning them in terms of the overarching mission and vision.

A team working on multiple products, which are in different stages of maturity of the Software Development Lifecycle, is working in a hybrid Scrum methodology 'ScrumBan'. Scrumban combines Kanban with Scrum practices (see §2.2.4 for more information about Scrum and Kanban).

3.1.7 Personas

In this sections, personas will be presented of the main roles present in the functional technology unit (Product Manager, Product Owner and developers) and of the main user of the software developed (a demand/ supply planner). The descriptions are based on the interviews and other observations and conversations done throughout this graduation project.

Alicia
Product Manager

Bio
Alicia is a product manager at Nike for almost a year. Originally from France, she is currently living in Utrecht.

Main goals
Translating customer needs; the launch of successful products; maximizing revenue and minimizing costs; minimize barriers for the technology team and optimize customer satisfaction.

Activities
- Bridging the business and technology side
- Working with management to figure out the future vision and breaking down this vision for the development team
- Getting requirements, expectations, acceptance criteria and other information needed to make decisions
- Outlining and managing the roadmap and corresponding stakeholder communications
- Facilitate stakeholder conversations
- Prioritization

Personality
Introvert — Extravert
Analytical — Creative

Frustrations
- Stakeholder alignment
- Prioritization amongst an overload of business needs
-

More analyst than creative; a little more extrovert.

Kevin
Product Owner

Bio
Kevin is a Dutch Product Owner, living in The Hague with his wife and new-born daughter. Depending on the project, he might combine the tasks of a product manager with his own responsibilities or actively support the product manager.

Main goals
Streamlining and maintaining the product backlog; defining user stories; accomplishing the product goal.

Activities
- Leading the development squad, helping them deliver
- Managing day to day deliverables
- Product backlog creation and management (adding, changing, removing and refining items)
- Generating user stories for the development team to use; turning customer problems and complaints into actionable items
- Set the development priorities so the team known what to focus on
- Making sure the development process follows the product roadmap

Personality
Introvert — Extravert
Analytical — Creative

Frustrations
- Changes in the organization
- Conflicting needs
- Unclarity about the higher level vision and purpose of certain tasks
- Dependencies on other teams
- Team capacity issues / full backlogs
- The traditional business set-up (including deadlines) conflicting with the new way of working agile

Michelle

Software Developer

Bio

Michelle is a 39 year old developer. She has been working for Nike for 5 years as an external temporary worker. She lives in Utrecht with her husband and 2 children. Main goal: Get working product to satisfy stakeholders

Personality

Introvert — Extravert

Analytical — Creative

Main goals

- Satisfy requests of stakeholders, from bug fixes to developing new features
- Efficiently working through the backlog

Activities

- Developing, maintaining and improving software
- Working with the users to onboard them when new tools are introduced
- Meetings with product partners and technical discussions

Frustrations

- Changes in user needs
- Dependencies

Alex

Planner

Bio

Alex is a 28 year old demand planner, ambitious to grow fast within the company. He lives in Amsterdam with his girlfriend.

Personality

Introvert — Extravert

Analytical — Creative

Main goals

- Get the right forecast and plan to meet customer demand

Activities

- Spending a lot of time pulling and consolidating data
- Creation, refinement and monthly submission of plans
- Double checking, tracking and analyzing plans
- Stakeholder management to ensure execution of plans
- Support strategic decision making
- Using various excel, files, and (ERP) applications

Frustrations

- Manual work
- Inconsistent and outdated information
- Changes in the organization and environment
- Different viewpoints on the same number

Persona image sources: Unsplash

3.1.8 Stakeholder mapping

An simplified overview of stakeholders of the Operations Tech unit are presented in figure 13 below. The overview is made with information coming from informal conversations with Product Managers of the team and is presented to provide understanding of how information flows between the main parties involved.
Note that the communication towards the end-user of the software is often intermediated by a respresentative (a capability lead or a plan-lead) as there are a great deal of planners.

Moreover, involvement of the product owner with the development team depends on the project; communication between the development team and the product manager is often intermediated by the product owner.

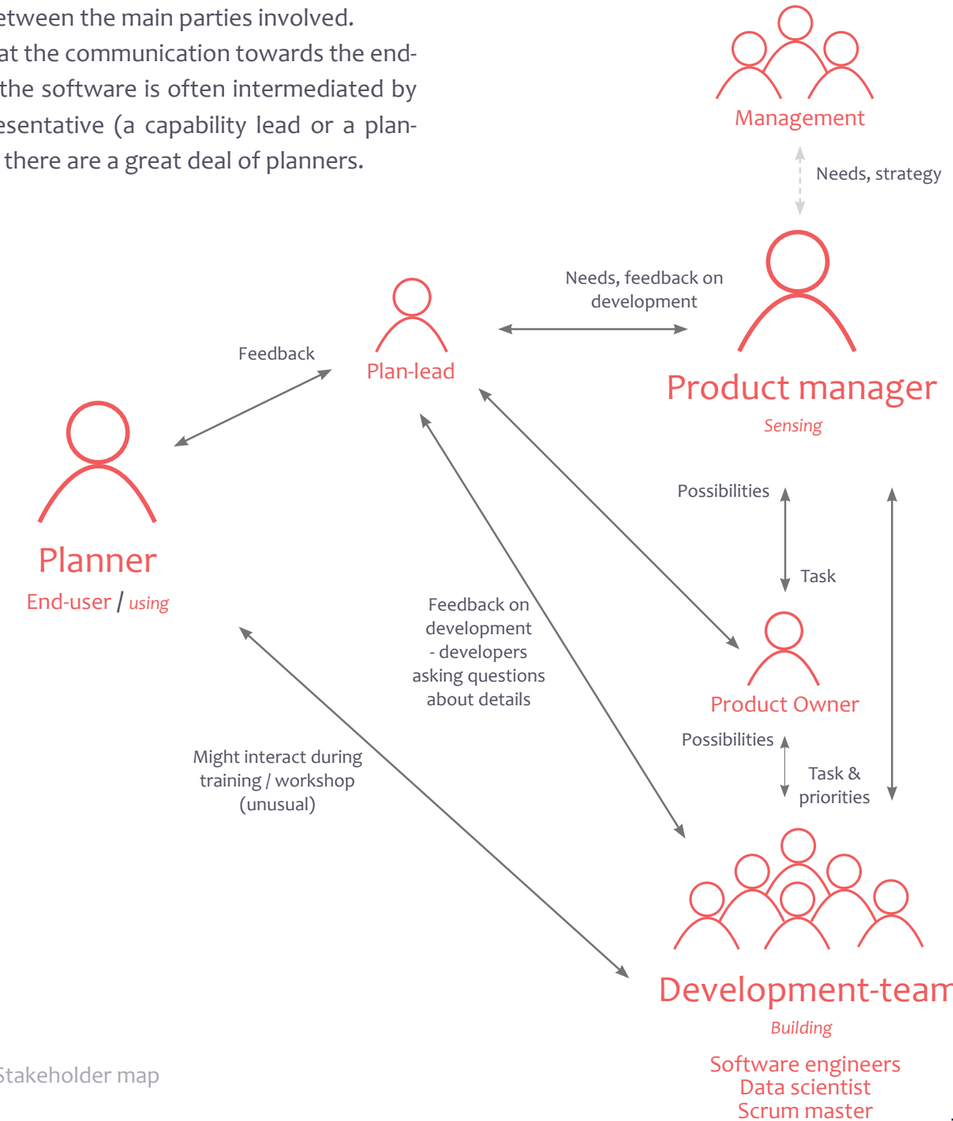


Fig. 13: Stakeholder map

3.2 Development scenarios

3.2.1 Common development scenarios

To better understand the type of requests the team gets and thus the type of problems the team is solving, this chapter elaborates on typical development scenarios. Information comes from interviews, observations, internal documents, and other informal conversations. The four scenarios presented below reflect recurring patterns concerning common request scenarios the squads are or have been working on.

1 Bugs/error requests

A part of the requests concern bugs or errors that have to be fixed. In these cases the problem is often well-defined (and therefore often not applicable for Design Thinking, see §2.1.2) and for example in case of an architectural or security risk, immediate fixing should happen (in other cases the priority might be less critical). The cause of such a problem could be an issue with the code, testing, requirements or the environment. Diving into the problem in hindsight is applicable to find out if there are other 'symptoms' of the root cause that can be prevented and/or if this kind of error can be prevented next time by adapting processes, skills etc.

2 Adding features or capabilities to existing software

For existing tools that have passed the implementation phase, requests from users and business stakeholders are often related to functionalities and features that should be added to existing tools. In this situation the question arises if the request should be prioritized or if other requests or features add more value. This situation of adding features to existing software is often reactive, based on requests that are often solution-focused (see chapter 3.4). In this case there is a need to dive deeper into the problem to understand if the proposed solution is indeed the 'thing' that would be the best option to build and if it should be build at all (P1).

3 Scaling existing software towards other user groups

Existing products can also be scaled to other use-cases. It can be 1) part of the initial strategy to start in one area and scale later on in the roadmap, or 2) that a product is found to be successful in one area and therefore requested by other areas, or 3) that opportunities for exploitation of the product are sought as a great deal of resources have been put into the development.

In case of scaling, requirements are often clear and there is alignment around the vision to roll it out to the wider audience (P1). For the development team this situation comes down to finding out if and how the added channels operate differently than the current channel onboarded in the tool.

The risk here is that an existing product has to be adjusted within boundaries and the new users to whom the product is scaled might feel it doesn't match their needs well enough:

".. but now we are onboarding [channel x], which is completely different and they feel 'you're giving me something created for [channel y] that doesn't match my expectations […]'. It's a difficult discussion because the product is already built. We had to tweak it for them, but they feel they want to change it completely" (P6).

Considering future users early on helps to tackle this problem and to prevent problems further down the line (e.g. low adoption rates) (P6).

4 Development of new software

In this situation the canvas is blank and a new product can be built for existing or new users. In general, the initiation comes via a request from the management, users, or other business stakeholders. Ideally, for innovation initiation comes from multi-disciplinary collaboration in which e.g. the (potential) users, market trends and opportunities, and (new) technological possibilities are researched, understood and integrated (e.g. see the sweet spot of innovation fig. 1 in chapter 2.1.2).

At the moment, finding opportunities is difficult as there is a lack of overview and visibility into the organizational landscape: *"There are so many more processes within the business that could be better supported with technology, but they don't get visibility or attention to be investigated in the way we are organized at the moment."* (P5)

3.2.2 Common development scenarios and the Software Development Lifecycle

These scenarios tend to happen generally at different stages of the Software Development Lifecycle (see chapter 2.2.1 for more information about the SDLC), see figure 14. For example the development of new software (scenario 1) can be placed at the initiation stage at the start of the SDLC. When an MVP or product is established, new features can be added (scenario 2) and at a certain point the software can potentially be scaled to other areas (scenario 3) if relevant. When software is released, bugs and errors

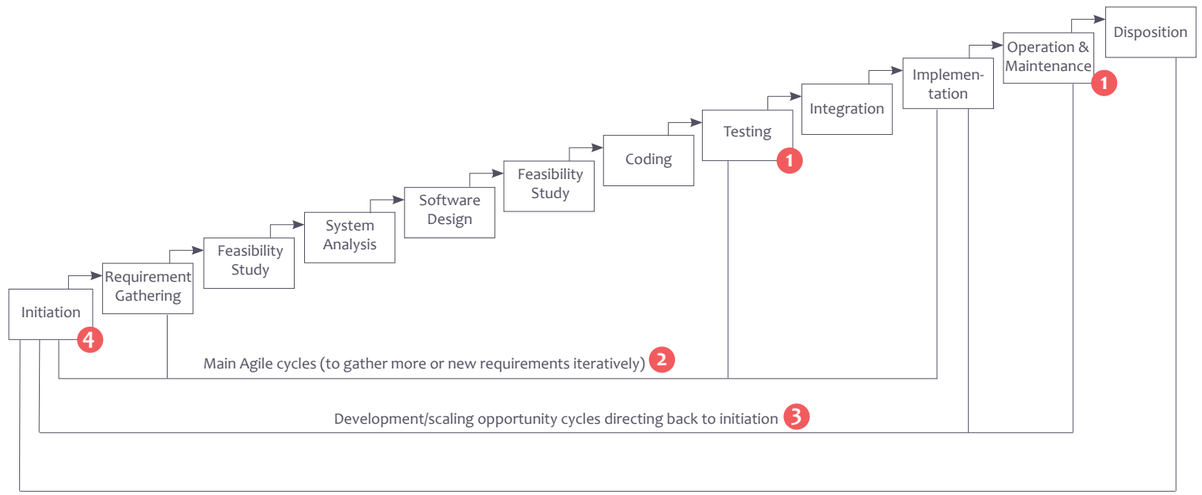


Fig. 14: The software development lifecycle including the typical development scenarios. Note that bug fixes are placed (mainly) at the testing and maintenance phase, though those scenarios might not lead to the initiation phase.

are fixed (scenario 1). Next to continuous maintenance throughout iterations, a product stays within this stage when it is mature or in inactive development for different reasons.

3.2.3 Internal software growth matrix

To be able to analyze and aid the conversation around the relevance of Design Thinking, and to generalize conclusions, it would be helpful to have a framework to classify current and future requests. As internal software development requests are initiated with an (internal) growth strategy in mind concerning existing or new products or features, and existing or new user groups, and project needs differ related to these characteristics, the Ansoff Matrix (Mullins & Walker, 2009; Ansoff, 1957) seems to be a good fit (see fig. 15).

The Ansoff Matrix is a simple framework that depicts the strategic directions an organization can go into to improve revenue or profitability.

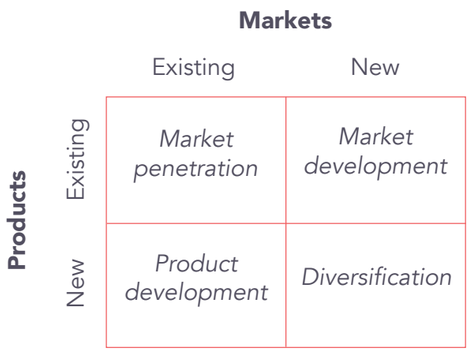


Fig. 15: The Ansoff Matrix for external growth strategies

First described by Igor Ansoff in 'Strategies for Diversifications' in the 1957 Harvard Business Review (Ansoff, 1957), it allows for structuring thinking and for classifying objectives.

In this case, the axes allow for classification of internal software development strategies within the company, see fig. 16. Below as description of each quadrant will be given. For reference, the development scenarios of §3.2.1 are placed in the matrix (see §3.2.3 for more information about the matrix in practice).

A) User group penetration: existing products / existing user-groups

In the top left quadrant, the software is already built and development is either focused on better satisfying existing users through maintenance, support, bug-fixing etc., or in terms of growth it could be onboarding more users of the same user-group focusing on a higher adoption rate of the tool within the onboarded user-group (e.g. a user group might use the tool for 30% and still use excel 70%).

B) User group development: existing products / new user groups

In the top right quadrant, the software is built and the development is focused on either strategically scaling the product to other user groups to roll out the product more broadly throughout the company, or it could be based on finding other use cases for the same product as a the money has already been invested in the product and could possibly be further exploited.

C) Product development: New products / existing user-groups

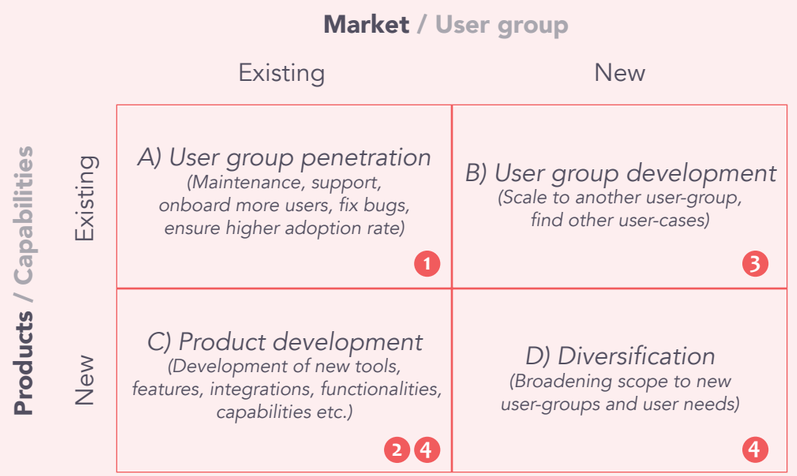
In the bottom left quadrant, new software is built for existing users, which can be from scratch or based on existing software. This quadrant also includes the situations in which new features, new integrations or other new functionalities are added to existing tools. I chose to put these in the new products quadrant instead of the existing products quadrant, as the process of problem exploration and development is more alike in this case. The focus of the top left category is maintaining 'what we have' and ensuring higher adoption, the focus of the bottom left category is on building new capabilities.

D) Diversification: new products / new user-groups

In the bottom right quadrant, the development team broadens its scope to new domains. This initially means unknown users and thus the need to understand the users and their functional context. In this situation there is an opportunity to diversify from current products and structures.

3.2.4 Common development scenarios in the internal software growth matrix

Scenario 1, fixing bugs and errors, fits in user group penetration, improving the existing product for existing users. Scenario 2, adding features fits into product development as it adds new features to an existing tool. Although we are talking about existing software, this scenario requires consideration and validation of a new element, hence the placement as a new product/capability. Scenario 3, scaling, is placed in user group development as it concerns scaling an existing tool to new users. And finally, the development of new software can be either for existing users or for new users, which is why it is placed in both the product development and the diversification quadrant.



Internal software growth matrix

Fig. 16: The Ansoff inspired Matrix for internal software growth strategies including the common development scenarios indicated with the red dots:
1) Bugs/error requests
2) Adding features or capabilities to existing software
3) Scaling existing software towards other user groups
4) Development of new software

3.2.5 The matrix in practice

Putting the matrix in practice, the following assumptions are made: 1) all software development projects can be placed within this matrix, 2) the matrix can support the conversation around Design Thinking needs within projects, and 3) forms of Design Thinking differ in different quadrants of the matrix.

The matrix has been used to aid two informal conversations around Design Thinking relevance with managers, and within a discussion session with product managers, product owners, a director and a Scrum master. Next to that, it has been discussed with a design expert within the company (appendix 12).

Main insights:

- Within the conversations held, people were able to place their current or past projects or requests within the matrix.

- The matrix supported the conversations around Design Thinking needs within projects. People were able to place their situation in one of the quadrants and compare different projects to each other. One participant of the discussion session mentioned that on hindsight a certain project has been wrongly classified as user group penetration as users the software was scaled towards were wrongly assumed to be similar to the existing group.

- User group penetration seems to be the only quadrant in which Design Thinking is not relevant as the problem is well-defined. In certain cases, understanding the core of a problem, e.g. through root-cause analysis, might be relevant to prevent similar problems in the future.

- Product development might be most dangerous in terms of not building the right thing as users are known and opinions and requests might be accepted without questioning as it is assumed that the user knows what he or she wants (which might not be the case looking at latent needs and e.g. new technological possibilities). "Problems might not be actual user issues, but rather an uninvestigated stakeholder wish list item" (P8) Taking a request at face value without questioning it and not thinking outside the box (anymore) is likely to happen when users and their needs are understood well and people have been working together for a long time.

- In the other quadrants, Design Thinking needs are difficult to generalize and it is difficult to form conclusions about the form and focus of Design Thinking within each quadrant. E.g. in the product development quadrant, many factors that can not be generalized such as involvement of the user, history and relationship with the user, expertise about the request within the team, urgency of the project, etc. seem to influence the needs. Furthermore, flexibility to what can be built is often limited in terms of the defined strategy, time and pre-defined technology. The conclusion here is that the goal and (user-) assumptions of every individual project needs to be considered together with what that means in terms of problem exploration and user-centricity.

Note that validating these assumptions have not been the focus of this research project. Further research would have to be done to make stronger generalizable conclusions about the assumptions. However, aiming to create a shared understanding of project goals and corresponding needs concerning problem exploration and Design Thinking, the matrix might be helpful to support the conversation.

3.3 Lack of focus on the problem

As written in chapter 2.3, Agile practices generally do not distinguish the problem space and it is the notion of a lack of problem exploration that gives room to Design Thinking to complement and support agile practices. In line with literature findings, there is indeed a lack of focus on the problem in the current way of working within the Operations Tech team, validating this assumption.

> *"We talk a lot about solutions, tools, about how to fix something, but we don't talk a lot about the problem." (P5)*

> *"I think that there is too much focus, not on thinking about the problem and if that is the problem we want to solve and if we want to solve it at all, but on 'okay there is a task that must be carried out." (P4)*

> *"I think that that [focusing on the problem] is a muscle that we haven't build probably also with the business. They want to move fast, so how can we help them move fast, but also make sure again that we've obsessed the problem and that we have validated it." (P1)*

There is currently no structural step in the process in place that allows for problem exploration. It is up to the teams and business stakeholders to decide if they want to take the step of problem exploration and validation prior to development. Therefore, it is dependent on the skills within a team if the problem validation phase will take place after a request:

> *"if a certain engineer is good at it then [problem validation] will happen and if people have less experience with it then it will not happen" (P5).*

> *"We don't have a validation loop in place, it's more like 'okay, this is what we're going to do'" (P1)*

Hence, the focus is rather on getting the requirements (P1). In terms of time for innovation next to day to day activities, the first sprint of the quarter is meant for planning and innovation. Again, it is up to the teams to use this time for innovation activities. It often happens that time for innovation is limited and the week is used to e.g. finish other work.

3.4 Deciding what to build - business strategy

In a large enterprise context, such as Nike, business strategy strongly influences the decisions made about what to build.
Generally speaking, within the business domain defined, the long-term strategy and the strategy for the year ahead are revised every half a year and budgetting is done accordingly.
Within the strategic priorities set during this strategic planning, business requests and ideas are prioritized according to the business value they promise to deliver.
In this sub-chapter, next to looking at business strategy and strategic fit theoretically, the 'strategy translation' process - from higher level goals to what is set out to be built - will be described in order to understand current 'gates' and limitations in the process prior to actual development.

The main source of information is an in-depth interview about this topic with a product manager (P3), complemented with observations and the interviews done regarding the boundaries to problem exploration (see chapter 4).

3.4.1 Business strategy and strategic fit

Chandler (1962) defines strategy as *"the determination of the basic long-term goals of an organization, and the adoption of courses of action and the allocation of resources necessary for carrying out these goals"* (Chandler, 1962, p.13). A strategy is essential for organizations to adapt to changes in the environment, create and maintain competitive advantages, and to increase chances in terms of growth and/or survival (Pfeffer & Salancik, 1978; Venkatraman & Prescott, 1990). It is found to influence firm performance (Burton, Lauridsen & Obel, 2004), and strategic misalignment or misfit negatively impacts the return on assets (Burton, Lauridsen & Obel, 2002). Strategic planning is done to align the organization with its environment, and includes the allocation of resources to support the alignment (e.g. see Mullins & Walker, 2009, on strategic fit). Strategic *alignment* or *fit* is described as '*matching organizational resources with the corresponding environmental context*' (Ginsberg & Venkatraman, 1985, p.421). As the environmental context is rapidly and constantly changing, the ability to adapt and find strategic fit is ever so relevant and important in the functional and technology departments nowadays.
Building on the definition of strategic fit as the relationship between external and internal factors, Itami (1987) suggests a model that leads to five elements of strategic fit: customer fit, competitive fit, technological fit, resource fit and organizational fit (see fig 17.).

More recently, in their insights paper, global design and consultancy company Frog Design puts business and strategic fit in the context of product design and related success or failure in a 'Problem-Solution Fit' model, which fit the challenge faced in the context of this research concerning the solution-oriented instead of problem-oriented (see §3.3) approach to requests well (Klamann & Shah, 2020). This

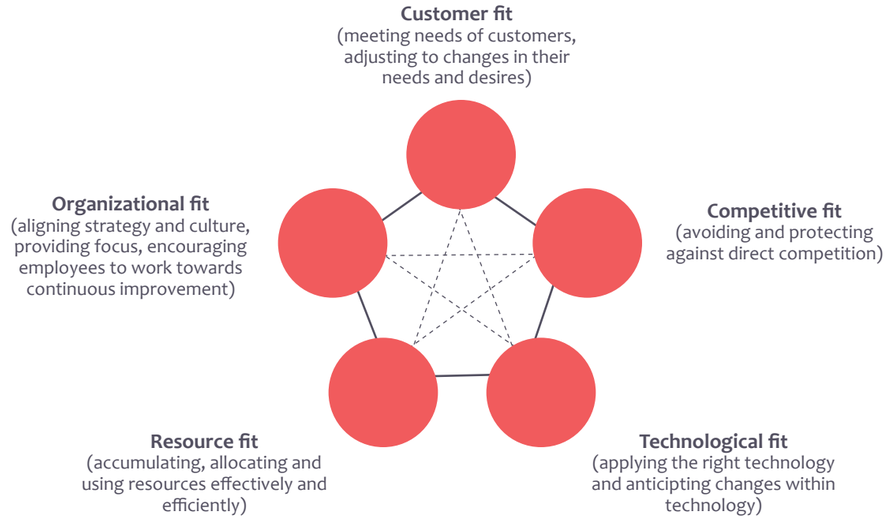


Fig.17: Elements of strategic fit (Itami, 1987)

model includes problem-, solution- and business objectives (see fig. 18).

In either way, the elements of strategic fit lay the foundation for the strategic orientation of the company and/or business unit, which affect the definition of 'the right thing to build'. Hence, we could argue that 'the right thing to build' is the thing that integrates the different elements of strategic fit effectively.

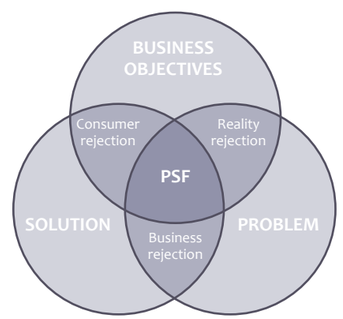


Fig. 18: Product-Solution fit (PSF) model by Frog Design (Frog, 2020)

However, do notice that Moran (2007) argues that strategic alignment is not a state that can be achieved, but rather a process to manage constant change. Aligning elements toward strategic fit might be repeated several times throughout a project life cycle. The outcome of alignment processes can be sets of goals to focus on to accomplish strategic fit.

3.4.2 Strategy translation within the organizational unit

The company is reliant on controlled processes and resource flows linked to the organizational structure. An overview of the steps of business strategy translation that take place before development is started - from higher level strategy to initiatives that end up on the backlogs of the teams - is provided below.

Long-term company level

There is a long-term strategic plan which

includes strategic priorities and the vision for the company. This plan is made by the leadership team (and other internal and external parties) and is revised every half year approximately. Corresponding decisions are also made at leadership team level.
While it would be interesting to investigate what user-centered design means or could mean for this level of strategic planning, this strategic level is out of scope for this thesis.

Functional level
Next to the long-term plan, there is an Annual Operating Plan, which includes the higher level strategic priorities that will be focused on in that year. This is where product management gets involved, as business and technology counterparts need to figure out what to focus on high level to support the functions' strategy.

> *"An example could be that we want to improve the demand for our x tool by 5 percent in the coming years, then one of the tasks would be how would we move from where we are today and improve our focus by 1% more, what should we invest in?" (P3)*

The answer could be to take a certain tool to another area of the business as well, defined on a high level, not in detail about what that would mean in terms of specific features or user needs. Mapping will be done to find out what value is expected in terms of revenue uplift or cost saving or compliance. Then, like all the other teams within the company, a submission of the plan is done to the finance and leadership teams. These teams will make the prioritization of the asks and final decisions on how the budget will be divided.

If stakeholders or users come with requests, this is the point where the requests are compared to the business strategy to see if it fits with the strategic priorities for the year.

> *"If a user is asking for a certain feature or functionality, is that in line with top down? And if not, we have that conversation to make sure we're adding most value, and if so, we'll move forward" (P1)*

The focus of the priorities is in general solution-oriented towards a specific tool, instead of problem-oriented and user-centered, as this allows for better value predictions.

Fig. 19: Levels of strategy translation

Initiative level
When budget is assigned to initiatives, the next question is what in the initiative should actually be build and how should it be build. In the example of the *x tool* above, this would be the moment in which actual tools, features, user-needs etc. are defined and requirement gathering is started. This is the stage in which the development team gets the request to develop, implement or scale a certain tool or feature.
At this point, when a request is placed on the backlog, a commitment to building the item is made and generally speaking it is past the problem exploration and validation phase; the focus will be on how to build the 'thing' in the right way and agile development starts.

The strategy translation process is visualized in fig. 19 on the previous page.

3.4.3 Discussion
First of all, it is clear that there is a high dependency on higher level strategy in this large enterprise context, especially in terms of priority assigned to projects and consequent budgeting. Strategic misalignment negatively impacts return on assets (Burton, Lauridsen & Obel, 2002) and therefore performance. The ability to adapt to the constantly changing environment and dynamically find strategic fit is essential.

Strategic fit and Design Thinking
The elements of strategic fit defined by Itami (1987) do seem to directly fit with the elements that are integrated through Design Thinking,

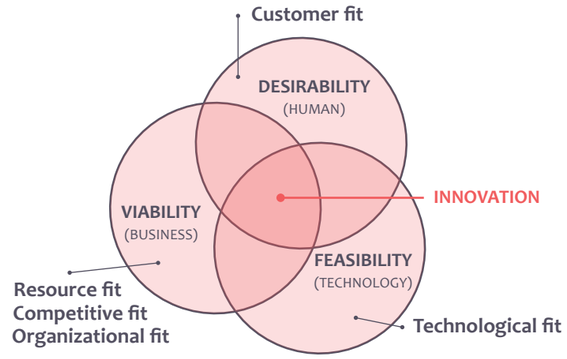


Fig. 20: Suggested fit IDEO's innovation Venn diagram and Itami's elements of strategic fit

see the original Venn diagram in fig. 1 (§2.1.2), an overview of Itami's elements of strategic fit in fig. 17 (§3.4.1) and the suggested fit in fig. 20.

In this internal enterprise context there is no need to win customers over in order to not lose them to competitors and there are significant limitations to ideation because of the strategy and software architecture already in place. However, strategic alignment between the elements is still relevant and essential to keep adapting to changes in the environment, changes in user needs, changes through technological possibilities and advancements, etc.

Strategic planning on a functional and initiative level
Strategic planning is done on different levels, described in §3.4.2., from which potential roles of Design Thinking in the process prior to and throughout Agile Software Development can be obtained.

Functional level - On the functional level, the main focus is often on tools, not on problems or users. Value is mapped to find out priorities between the initiatives, but there is little focus on understand problems (human-centered) and rapidly exploring initiatives to reject, pivot or continue with an idea grounded in initial qualitative and quantitative data. There is no explorative phase in between the idea/request and commitment on the backlog, which is a missed opportunity in terms of cheaply reducing product risk (e.g. which Spotify seems to utilize well with their 'think it' phase, see §2.4.2).

Moreover, the over-focus on existing options and decision making next to idea generation and considering new alternative options, as found in literature (Boland and Collopy, 2004), is relevant and present on this level. Limitations exist concerning what has already been build, current user-processes that need to be aligned and higher level strategic choices, that need to be understood in order to potentially deviate.

There is an opportunity for Design Thinking prior to commitment to building in terms of gaining a better understanding of the different elements making up strategic fit and diving into the core of problems, user and stakeholder needs and practices in order to question and validate a stakeholder request prior to commitment to building.
There is an option to set up a team around a certain problem, goal or idea or to put exploration on the backlog with the option to pivot or reject an assumed direction prior to commitment to building.

Initiative level - On the initiative level, there is a commitment to build, i.e. the task is already on the backlog at this point and budget is assigned. The next step is often requirement gathering to start agile software development, assuming without further questioning that the solution proposed is the right thing to build. The problem and the users are not thoroughly researched and understood, and alternative options of what could be possible are not explored, which are reasons that are in line with literature arguing for the need for Design Thinking to approach indeterminate organizational problems (e.g. Martin, 2009; Lafley & Charan, 2010).

Looking at the Design Thinking literature, and viewing the problem and solution space as spaces that should be explored and aligned iteratively (Lindberg, 2011; Boland & Callopy, 2004), there is an opportunity to consciously view the proposed solution as an 'assumption' of what could be the right thing to build. This would allow for further exploration and validation through rapid prototyping, consideration of alternatives, and thus early and cheaply pivoting/adapting the concept to optimize fit.
In this perspective, as strategic alignment is viewed as a dynamic rather than a static process (Moran, 2007), Design Thinking could complement the agile software development process in iteratively working towards strategic fit and managing constant change.

There is an opportunity for further exploration in order to optimize the solution-fit user-centric and iteratively, and concequently to build and communicate accordingly to improve user-experience, -satisfaction and –adoption.

3.5 Chapter conclusions

In this chapter the context of the Operations Tech team has been explored. A general understanding of the team context, main personas, stakeholder information flows, type of software (enterprise software mainly for demand and supply planners), the agile methods employed (Scrum and Scrumban, influenced by SAFe) and the process of strategy translation have been developed. The type of problems the team commonly solves are analyzed by looking at common development scenarios in terms of projects and requests.

Within the hierarchical matrix structure, the team of focus is located on the intersection of the EMEA geography, the technology domain and the 'operations' functional area. Rigidity of managerial hierarchy might restrain tasks that require rapid change or dealing with uncertainty, such as Design Thinking.

An important validation is that there is, in line with literature on agile practices, indeed a clear lack of focus on the problem in the Operations Tech team. Requests are often already solution-oriented and higher-level strategy strongly influences what to build.

Moreover, the ability to adapt and find strategic fit is ever so important; dynamically as strategic fit is not a state that can be achieved, but rather a process to manage constant change. Looking at strategic planning on a functional and initiative level, there is an opportunity for Design Thinking to help teams prior to and after commitment to building a certain backlog item. In the former case, Design Thinking can help teams to dive into the core of the problem, gain a better understanding of the user and stakeholder needs and other elements that are part of creating strategic fit such as technological possibilities, in order to question and validate a stakeholder request prior to commitment to building it. In the latter case, in which commitment to building is already established, there is an opportunity for Design Thinking to support (dynamic) optimization of user-centric solution-fit, and to build and communicate accordingly to improve user-experience and stakeholder satisfaction.

The Ansoff matrix was used to create a framework that enables classification of current and future projects. However there is one quadrant (user group penetration) in which Design Thinking is not relevant, it is difficult to generalize needs and forms of Design Thinking within the other quadrants. Therefore, project goals, (user-) assumptions and needs regarding further problem exploration have to be considered per project. The matrix has been useful in supporting a shared understanding of project goals and supporting the conversation around corresponding needs concerning problem exploration and Design Thinking.

To understand why there currently is a lack of focus on the problem, and how Design Thinking might help in this context of the Operations Tech team, the next chapter will focus on pain points and challenges the team faces in terms of problem exploration and 'building the right thing'.

Chapter 4

Boundaries to Problem Exploration

In this chapter, main insights and conclusions into the current challenges and boundaries of executing problem exploration and a focus on building the right thing prior to development, coming from a focus group and interviews conducted with the Operations Tech team, will be presented. Next to that, literature will be explored to make sense of the findings. The boundaries and challenges within the defined context that come out of this chapter, serve as input for the practical conditions and implications fundamental to support the implementation of problem exploration initiatives.

4.1 Approach

The goal of this research phase is to better understand the reasons behind the lack of focus on the problem in the context of the Operations Tech team to learn about how Design Thinking might be able to support current practices. Therefore, pain points and challenges the team faces related to problem exploration and building the right thing are uncovered.

4.1.1 Collecting data

Different methods were used to collect data, which will be elaborated in this section.

Semi-structured interviews

To explore the context and current situation concerning problem exploration and related boundaries, seven semi-structured interviews were conducted with Operations Tech team-members; three interviews with product managers, three interviews with product owners, and one with a developer. Throughout the project, these people were also involved in validation sessions and/or discussions when possible.

The semi-structured approach was chosen, as it allows for both focused and explorative questions (Rowley, 2012), and encourages a natural dialogue (Fielding & Thomas, 2001). The interviews were held in either Dutch or English. The interviews were between 30 and 60 minutes long and are recorded and immediately transcribed afterwards. In order to respect anonymity, the participants will be referred to as P1-7 in random order. The interview guide and transcripts can be found in appendix 5 and 12 respectively.

Presentation/discussion session
Next to the interviews, I attended a presentation including discussion session held by a technology product manager outside the Operations Tech unit, with the topic of solving the right problem. Gathering notes and quotes, insights have been turned into statement cards as well. The presenter is referred to as P8 on the statement cards. Notes can be found in appendix 12.

Innovation experts
Next to that, the perspective of two internal innovation/design experts were taken into account as well. These conversations have been less formal and were not recorded. Notes can be found in appendix 12.

Workshops
A workshop (split into two sessions, together 1.5h) was done with a group of product managers, product owners and a product analysts (7 participants) to reflect on the innovation design sprint and other Design Thinking experiences, to reflect on opportunities for Design Thinking in day to day ways of working. This workshop allowed for insights with regards to the attitude towards Design Thinking and boundaries in turning plans into action. Statement cards are indicated with a 'workshop insight' note. Notes can be found in appendix 14.

Documents
Internal documents related to the specific software tools, the user journey, the operations functional area and technology contexts and documents related to projects such as project proposals were studied during the project.

Furthermore, observations, informal conversations and analysis of internal documents supported a general understanding of the context required to interpret the data.

4.1.2 Analysis procedure

The analysis of the interviews, workshops and observations was done by performing initial coding (Birks & Mills, 2015), and clustering using an 'on the wall' approach (Sanders and Stappers, 2012) (see fig.21).

Affinity mapping was used to cluster the codes, which allows for organizing the codes into distinct clusters. Affinity mapping is an inductive approach used to 'externalize and meaningfully cluster observations and insights from research' (Martin & Hanington, 2012) and helps to stay grounded in the data throughout the process. Insights, requirements, observations etc. are captured on individual statement cards, after which the cards are individually interpreted and underlying significance is considered. In this way, a story emerges about the people, their context and the nature of the problem. The book 'Universal Methods of Design' refers to the affinity diagram as 'the voice of the customer [user], and a partner in design' (Martin & Hanington, 2012).
The statement cards all have a colored band referring to the corresponding participant. Using the method described by Martin & Hanington (2012), the statement cards represent single observations, insights, concerns etc. rooted in research data. The blue post-its describe aspects of the issue 'boundaries to problem exploration' that the statement cards reveal, the pink post-its describe specific issues clustering the separate aspects, and finally the green post-its describe an overarching area of concern. An example of a cluster can be found in fig. 22, the full overview of clusters can be found in appendix 15.

The process started with 249 statement cards, 18 were discarded during the analysis mainly because of their focus on context-description instead of the relation to the issues involved.

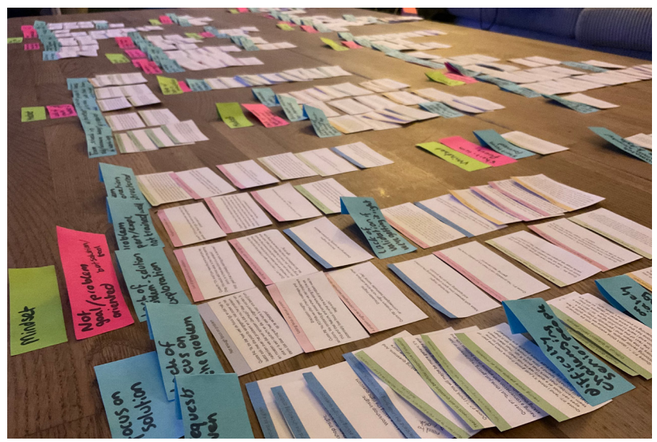


Fig. 21: Picture during the clustering of statement cards

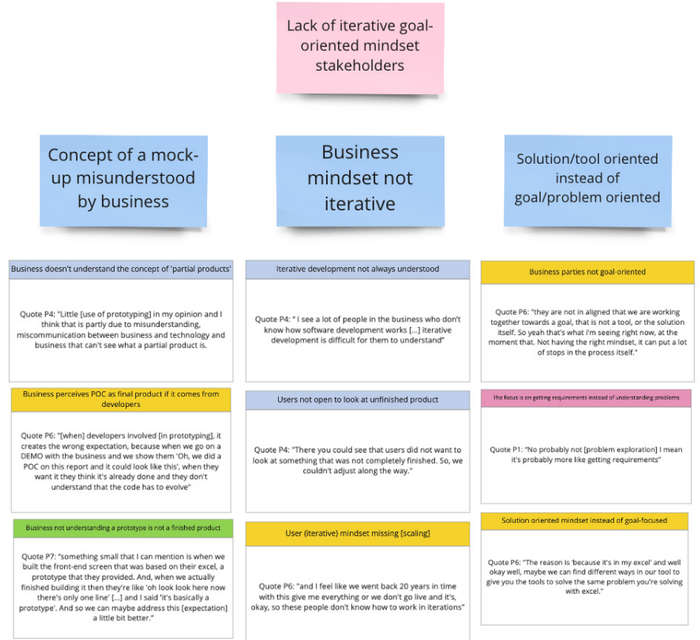


Fig. 22: example of a cluster of the affinity diagram

4.2 Results

In this section the issues found to influence and limit problem exploration will be discussed. These are grouped into overarching areas of concern which will be discussed in the discussion section (§4.3), but will already be used for the sake of clarity. An overview of the areas of concern, specific issues and main corresponding quotes can be found in appendix 13.

4.2.1 Area of concern: Mindset

The first area of concern includes issues that can be related to the mindset. An overview of the related issues can be found in fig 23.

Issue 1: Lack of problem-/goal-oriented mindset

A main challenge observed and reported is that the start of the process (regardless of whether a request is coming from a business stakeholder, user or management, or if the team sees opportunities themselves) is usually tool- and solution-oriented, skipping the problem exploration phase. In most cases the business comes to the technology team with a request to build a certain solution, e.g.:

> *"[...] it was like here is the excel which we want to have in the form of a tool, and that's the way it generally goes [...] we're past that phase in which we would look at what the real problem is and what we're trying to solve." (P5);*
>
> *"a lot of times the question is to build an app, it would help to challenge that and to get a good vision around it to create a solution to the actual problem" (P4).*

The focus on solutioning is reinforced by the analytical and solutioning skills from the developers, who are *"keen on building software" (P1)* and have a hard time postponing their 'solutioning mode' (focus group). Also, on a product owner and product manager level the focus is often on tools and features:

> *"We [PO/PMs] constantly get asked if we can implement this and that feature and somewhere along the line we potentially lose focus on what the actual problem was to begin with" (P8)*

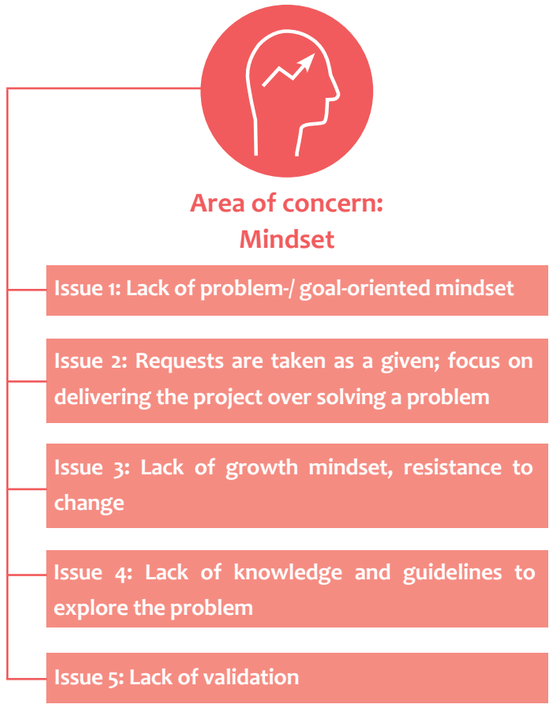


Fig. 23:. Overview of the issues grouped into the 'mindset' area of concern.

Next to the mindset of the team, the solution-focused mindset of other stakeholders collaborating in the development process limits the space for problem exploration and user orientation prior to solutioning and developing as well. The business stakeholders *"are not aligned that we are working together towards a goal, that is not a tool or the solution itself. That is what I am seeing at the moment, not having the right mindset, which can put a lot of stops in the process itself." (P6).*

> *"I tend to say trust the process for almost everything that I see that sometimes we miss the pre-work which is educating the people in the mindset they need to have" (P6)*

Strengthening the solution-focused mindset might be the lack of an iterative mindset needed to iteratively learn about the problem and solution space in the problem exploration phase (and thereafter in agile development):

> *"[business stakeholders don't understand] the incremental way of working. It is [normal for] business [people] to have ambitions, but have smaller milestones and check how things are going instead of going for the end-result" (P6);*
>
> *"I see a lot of business people who don't understand how software development works [..] it's iterative development that is hard for them to understand" (P4).*

Issue 2: Requests are taken as a given; focus on delivering the project over solving a problem

Next to being solution-focused, there is also a high focus (and pressure, see issue 7) on delivering a project. Hence, requests are taken as a given and are not questioned.

> *"I see that there is a lot of focus, and I indicated that last time with the post-its as well, not on thinking about the problem and if that is the problem we should actually solve in the first place, but more like 'okay there is an assignment that has to be executed" (P4)*

The lack of questioning and the importance of addressing this was also concluded in the focus group:

> *"We need to challenge them [stakeholders]. Get behind why they want something and understand the underlying problem."*

Towards stakeholders, the focus immediately goes to getting requirements (P1) with a technical lens. Though, the lack of questioning problems also happens on a team level:

> *"We have quite a few seniors in the team and generally if they make a statement the rest agrees." (P7).*

As requests might not always address the right problem, consequences of not questioning and validating a request were experienced in practice:

> *"[..] then in the end users are not happy or you're not solving any problem, you just check a box that the project is delivered." (P6)*

Issue 3: Lack of growth mindset, resistance to change

Limiting change within the company is a lack of a growth mindset, which is observed in the ongoing agile transformation and required for Design Thinking, as the mindset of a design thinker includes characteristics such as openness to risk-taking, embracing failure and

a comfort with ambiguity (see §2.1.4).

"[People] don't have a growth mindset yet, more a fixed mindset […] that people dare to experiment and to make mistakes, and to learn from that, that you can say to your manager 'we spend a week on this and it did not work, shit happens, let's continue to the next thing'." (P5);

"We're often solving within existing frames/ boundaries [..] people don't dare to look outside the box […] that is what I miss a bit" (P5).

This resistance to change might also be linked to a lack of awareness, or routine rigidity:

"We have been doing these things for so long, and we think we are right [...] but we should be asking why are you doing it like that. [...] Having done it for a while, I will just use whatever I've learned in the past, but that might not necessarily be the best way to do this." (P7)

Issue 4: Lack of knowledge and guidelines to explore the problem

Moreover, focusing on the problem is currently not present in the structure of projects and there is no time or capacity dedicated to the problem deepdive (focus group). It is up to the teams and collaborators if they want to take the step of problem exploration and validation prior to development. Therefore, it is dependent on the skills and motivation within a team if the problem validation phase will take place after a request:

"if a certain engineer is good at it then [problem validation] will happen and if people have less experience with it then it will not happen" (P5);

"we probably like I said don't have that [problem-]validation loop in place, it's more like 'okay, this is what we're going to do'" (P1).

This is recognized to open up opportunities for Design Thinking:

"I think it's in the structure. Design Thinking could give guidelines we can hold on to in the process. So if we would get a request from business […] we could say 'okay looking at the Design Thinking process, this is the phase in which we need to understand the problem very well', going consciously to that step, consciously investigating the problem before going into a solution direction." (P5)

Issue 5: Lack of validation

A small but interesting insight that relates to reinforcement of the mindset is the lack of validation. In terms of 'building the right thing' hypotheses need to be tested to see if and how things change and to learn from that (ideally already early on in the process, see the case study about Spotify §2.4.2), e.g. to find out if the tool or feature actually makes the forecast more accurate as was intended or if the intended revenue uplift was reached. However, measuring and assigning certain gains or losses to specific causes (e.g. tools) might be difficult and ambiguous, currently it is often not measured at all (focus group). Not validating if the right thing is built does not encourage focus on this goal from the start of the process.

"I see us adding value so that makes me think that we're getting it right, but. Again, like I don't have any points of validation to say like 'Yes, we are', or like 'no we're just getting lucky' or 'actually you know we're delivering stuff but it's not solving the problem'" (P1)

4.2.2 Area of concern: Organizational Structure and Strategy

There are also factors related to the organizational structure and processes that influence the room for exploration activities. An overview of the related issues can be found in fig. 24.

Issue 6: Lack of space for problem exploration because of the organizational structure

However the teams are following agile ways of working, they are still part of a large company that is bound to its existing structures that concern more heavyweight practices (see §2.2.2) that arguably do not work well with Agile Software Development.

"It [change] is very difficult with such a large organization in which there is a large middle-management layer that is sticking to old structures." (P5), and

"we're a big company and the way we're structured, also with budgeting on specific initiatives and all lines in which reporting about it is necessary. […] in terms of company culture [structure] that doesn't work that will within software development" (P4)

Existing structures around budgeting for submitted project proposals conflict with cases in which value is coming from exploration activities, as this value might not be known or determinable up front and a part of the experiments might even provide no return on investment at all. Resources are generally provided to solution-based initiatives, meaning that teams have to come up with solutions in order to get the capacity to spend time on that proposal. If a proposal gets a budget assigned, it ends up on the backlog of the teams and *"If it's on the backlog, it might be too late to start problem exploration, because at that point it's already prioritized" (focus group)* Thus, structurally no resources are assigned to problem exploration and the step of researching the problem and validating if that solution is indeed the right thing to build is skipped.

Fig. 24 Overview of the issues grouped into the 'organizational structures' area of concern.

Issue 7: Lack of space problem exploration because of time-pressure

Even if there is the intention to implement a problem exploration phase or to start using more Design Thinking, high workloads, time pressure and strict deadlines make it difficult to prioritize exploration of problems:

> *"You take little time to think, are running from meeting to meeting, but nobody plans some time to calmly think about how to best approach something." (P5)*
>
> *"People often think that there could be more room to think about the problems themselves. But there is often no space to think about problems because of the task to be completed that lies in front of you" (P4).*

This results in a focus on just checking boxes to finish tasks as soon as possible instead of taking a step back to consider the underlying problem. (see issue 2)

Highly related to and influencing this focus on efficiently checking the boxes is the way employees are (consciously and subconsciously) rewarded; there is a focus on Key Performance Indicators (KPIs) related to velocity, efficiency, eliminating all 'overhead', stimulating to skip any seemingly unnecessary tasks.

> *"It's difficult to get it prioritized because it is all about short-term focus on what we need to deliver this quarter or next sprint. And if it is not directly contributing to this, it will be difficult to get priority for it." (P5)*

Issue 8: Role of technology; technology not included on problem level

Technology teams are generally mainly or solely involved in the solution phase, showing a sequential relationship between the business side and technology side:

> *"In most cases they [business stakeholders] are already pretty far with a solution when they come to us. […] like 'this is what we need' and then we're past the stadium in which we look at what the actual problem is that we are trying to solve. So there should be a balance and I think that would be pretty difficult to find." (P5)*
>
> *"Business needs to include us at a point when their ideas are less mature, so we can help in the process of understanding the problem and solutioning" (focus group).*

Not involving technology early on in the process significantly limits the room for problem exploration. Ideally "understanding the problem is a conversation, a partnership between tech and the business. Exploration is something that needs to happen hand in hand." (focus group)

4.2.3 Area of concern: Clear shared vision

A third main concern that was found is not having a clear and aligned goal or vision as a boundary to exploring what the right thing is to build. As the problem deep dive extends beyond a team-activity, the vision is an important element of it (workshop insight). An overview of the related issues can be found in figure 25.

Issue 9: Need to understand higher level vision on team level; vision translation gap

There is a need to have an understanding of the higher-level vision on a lower-level for the team and users to understand why a certain problem needs to be solved, why a certain solution is proposed, and ultimately what the right thing is to build from a business perspective.

> *"If that [vision] is clear you can decide how to support business in the best way as a tech organization [..] and empower people to move towards that vision. And that vision or how to deal with that is what I miss" (P5);*
>
> *"[no overview] about what are the large problems that we want to solve and then looking at what we could do instead of 'oh we could implement this, maybe it helps'. I would rather do that based on a vision" (P4).*

The problem does not seem to be a lack of a higher-level vision, but there seems to be a gap in translating that vision to the tech-teams and users, to make sense of what is on the backlog.

> *"[…] there is a higher level vision from [the operations functional area], where we want to go. But it's such a high level that it is not tangible for an average planner or tech-team. So a translation has to happen somewhere." (P5);*
>
> *"You have the higher level vision and strategy […] and you have what eventually lands on our backlog. And how those match is not a structured and transparent process." (P5).*

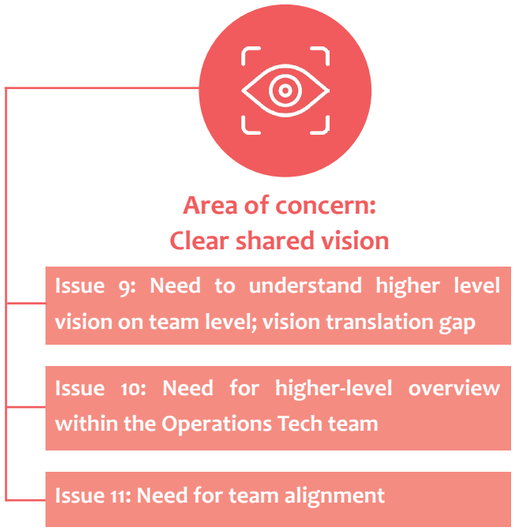


Fig. 25: Overview of the issues grouped into the 'organizational structures' area of concern.

Issue 10: Need for higher-level overview within the Operations Tech team

A lack of investigation of certain business areas results in a lack of overview needed to find out what would make the most impact to the business and users, i.e. the right thing to build.

> *"Apparently there are gaps, how can we make sure that we have an overview of [the operations functional area] as a whole instead of mainly focusing on the old demand planning. […] the different [planning] business functions and if we support them or if there are gaps; such analyses are not there or too little" (P5);*
>
> *"There are so many more processes within the business that could be better supported with technology, but they don't get visibility or attention to be investigated in the way we are organized at the moment." (P5) .*

Issue 11: Need for team alignment

In order to align squads within the Operations Tech team, there is a need to look more broadly at a shared vision and overarching priorities "*instead of team priorities, because*

they don't make sense." (P4). Because of high dependencies, this alignment is important:

> *"Because we have some dependencies between the teams, we need to be really clear on what it is that would add the most value to the business" (P1);*
>
> *"we are reliant on another team and there's not always visibility on it" (P7);*
>
> *"we need the other team, which is why we have to align with that team so that they can reserve capacity at the right point in time." (P5)*

At the moment team-priorities are not aligned at times and squads work rather siloed instead of goal- and human-centered:

> *"[…] so what I said now is 'let's stop looking at it as a front- or a back-end, let's look at who is our client and how can we solve this together'" (P4)*
>
> *"that information [about the user] kind of sits within each of the teams and each of the applications that we're building, you know so is that something that we need to bring to a higher level." (P1)*

Alignment is difficult because *"teams have different cultures and you know, we have one team working in Scrumban the others working in scrum" (P1).*

4.3 Discussion

The issues found can be grouped into three main areas of concern: mindset, organizational structure and a clear shared vision. In this section, these three areas will be discussed, compared to literature and previous insights of chapter 2 and 3.

4.3.1 Mindset

Findings indicate that the solution-oriented mindset of technology teams, users and other stakeholders form a boundary to problem exploration. Both tech-teams and stakeholders could use guidance in getting to the goal-oriented perspective, as this way of thinking does not come natural to them. Adding this mindset and potentially design skills to the team could support the collaboration and problem exploration phase. Adding these skills to the collaboration through a UX researcher or designer does not guarantee a shared mindset within a development team or collaboration, Design Thinking can support the right mindset meta-disciplinary (Lindberg et al., 2011) (see §2.3.1).

The lack of mindset is a common bottleneck (Dunne, 2018), but argued to be the most crucial aspect of the design thinking approach (see side box in §2.1.4).

In §2.1.4 Design Thinking was presented from three perspectives: mindset, process and toolbox, and a framework was presented combining these three perspectives. The area of concern 'mindset' directly links to the core of this framework. A fixed mindset of stakeholders and routine rigidity of the engineers - being stuck in existing behaviors - seem to be limiting factors concerning mindset. The lack of a growth mindset can be linked to the design thinker characteristics: openness to risk-taking, comfortability with ambiguity, embracing failure and optimism to change. The lack of an iterative mindset can be linked to the dynamic characteristic and reflectiveness.

Next to that, the solution-oriented mindset at the core of current process and collaboration within the Operations Tech team influences the perspective of Design Thinking as a process. Instead of starting with a problem, as is assumed in Design Thinking models (see §2.1.3 and §2.1.4), the starting point is a solution. In order for the Design Thinking process to make sense, there is a need to reframe the proposed solution to the core of the problem. This will allow for evaluation of the solution (and alternatives) to conclude 'the right thing to build' and to optimize for example problem-solution fit and user-fit (see §3.4.1 for elements of strategic fit). Without this initial reframing, questioning the proposed solution, the Design Thinking process does not make sense.

As there is currently no structure in place, Design Thinking could provide guidelines to create awareness and allow for conscious consideration of the problem before committing to a solution direction.

4.3.2 Organizational structures

Established processes predominantly focus on efficiency and short-term gains (exploitation), which conflicts with the focus on exploration and experimentation needed for problem exploration, which are central to Design Thinking (Carlgren, Rauth & Elmquist, 2016). Lack of room for problem exploration related to the organizational structure is in line with findings in literature. Encouraging deviations from the initial problem brief (or in this case solution proposal) and the iterative way of working, conflict with the logic and efficiency of linear mainstream processes. In the culture and structure of large organizations, such a focus on experimentation and iterations, is based on a different logic (Lester & Piore, 2004) and corporate culture (Dunne & Martin, 2006) than the traditional one, and easily conflicts with the non-adaptive established organizational processes (Walters, 2011).

The boundary of a focus on over-exploitation in current structures and KPI's, meaning predominantly focusing on efficiency and short-term gains, is a well-known challenge in innovation literature (e.g. March, 1991; Tushman & O'Reilly, 1996). Balancing exploitation and exploration is called 'organizational ambidexterity' and is found to be necessary to adapt to changing environments (O'Reilly & Tushman, 2008). However an emphasis on exploitation is effective in the short term, it is argued and proven to be destructive to a company in the long run (March, 1991). Having organizational structures and routines in place that allow for a focus on exploration makes a firm more likely to be innovative (Miller and Friesen, 1982).

In order to support exploration, there is a need to consider different KPIs and possibly budgeting streams in order to release the pressure on efficiency and short-term return on investment, e.g. possibly through structural ambidexterity (O'Reilly & Tushman , 2011).

Next to that, findings indicate a need to reconsider the role of technology teams in the front-end of the software development process. Currently, technology teams are included in a sequential manner instead of as a team or collaboration with the business side; they get a request to build a solution and start acquiring requirements skipping the step of problem exploration (i.e. understanding the user, the purpose behind the request, the fit with the bigger picture etc.). In this way skipping the question if the solution proposed should be built at all and/or if there are better solutions to address the core of the problem.
However the current way of working might be the most efficient way to 'tick the box' of delivering a project, in order to thoroughly understand what should be built it is essential to dive into the problem space and understand the goal. And thus, it is suggested to involve technology teams earlier on in the process to explore the problem thoroughly together.
In this way, technology teams can support the business regarding knowledge about technological possibilities, the user-perspective and a focus on aligning the goal of the project with these elements. Moreover, the technology teams are able to build prototypes to – in collaboration with the business - learn iteratively and (in case of rapid prototypes) cheaply about the problem at hand prior to starting development. In order for technology teams to work goal-oriented, and to leverage the technology-perspective in problem- and solution-exploration, it is key that they are involved earlier in the process.

4.3.3 Clear shared vision

Looking at the Design Thinking philosophy (chapter 2), building the right thing means integrating and aligning the business, technology and user perspective. In chapter 3 it was found that, in the context of the Operations Tech team, the business perspective is highly influential in deciding what to build. In this chapter, findings indicate the importance of a clear and shared business vision, which is recognized by researchers as well, e.g. Briner et al. (1996) state that "the most significant success factors for project teams is that they have a common and shared idea of what difference they are trying to make as a result of the project" (Briner et al., 1996, p. 89). In exploring the boundaries to problem exploration, it is found that a clear (shared) understanding of the vision is however lacking. It is suggested that this is not due to a lack of vision but rather due to a gap in higher-level vision translation.

To be able to set up a preferred project outcome, or goal, with various project stakeholders, the reasons-for-being need to be defined by leadership so that it is well understood by those who are of influence to the successful execution of the project (Norrie & Walker, 2004). As the current focus on technology and economic value in translation of the higher-level strategy might result in losing the sense of purpose initially intended. Therefore, it is recommended to consciously link the phases of strategy translation to the human-centered purpose intended, to allow for alignment around bringing value to people. The role organizational leaders play in setting up the vision is recognized by researchers (Norrie & Walker, 2004).
Recognizing the dilemma of a lack of a clear vision, Baccarini (1999) and Davis (1995) set up the Logical Framework Method (LFM) to define project success. The framework clearly distinguishes two components: project management success, which focuses on the process in terms of costs, time and quality, and product success, which focuses on 'the effects of the project's final product' (Baccarini, 1999). Therefore, regardless of whether a clear vision is communicated top-down, it is suggested that a clear understanding of the effects of the final products are established. This fits with the need to reframe the solution to a goal in order to be able to evaluate and diverge from a proposed solution (see §4.2.1)

Chapter 5

Conceptual Model

From insights to a conceptual model

In this chapter, insights from the literature review towards problem exploration (i.e. Design Thinking, user-centered design) in Agile Software Development, and insights into the context, boundaries and opportunities come together in a conceptual model. The model represents and highlights the needs and goals for problem exploration (based on the Design Thinking framework) in the context of focus. The framework will be translated into a usable artifact afterwards; its structure will be explained in this chapter. First, building blocks of the framework will be explained to create an understanding of the elements. The framework is built in an iterative manner, different representations used in conversations.

5.1 Purpose of the model

The aim of this conceptual model is to capture and represent the key opportunity areas for Design Thinking to support agile software development in the context of this research in a comprehensive way. By conceptualizing the vision, the model can be used to (contextually) guide the implementation of elements of Design Thinking in a way that focuses on main bottlenecks and absences.

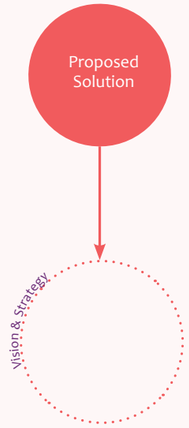
The conceptual model will be further used in this research to derive the design goal, product principles and product requirements, through which it will be translated into a usable artifact to support users to put the conceptual model into practice. Thereby allowing users to directly work on the identified opportunity areas.

5.2 Building blocks

In this section, an overview of the building blocks used to develop a model representing the role of Design Thinking the explored context is presented. The building blocks come from conclusions of chapter 2, 3 and 4. Formation of the building blocks and subsequent model (which is presented in the next section) has been an iterative process of analysis, ideation and discussion (see §5.5 and appendix 7).

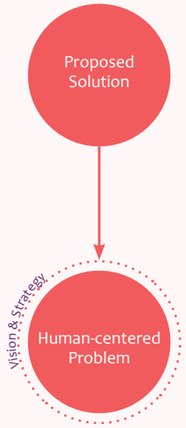
Proposed solution
The starting point is a proposed solution

Requests are generally solution-oriented (§4.3.1). The solution-oriented mindset forms a boundary to problem exploration (§4.3.1).



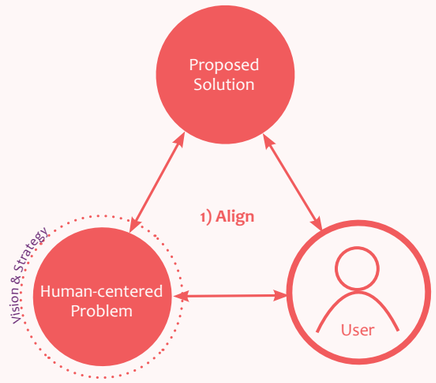
Vision & Strategy
Understand the vision and strategy behind the proposal

A clear vision is key to align around the (human-centered) reason-for-being of the project (§4.3.3). Project goals, user assumptions and needs regarding further problem exploration need to be considered per project (chapter 3)



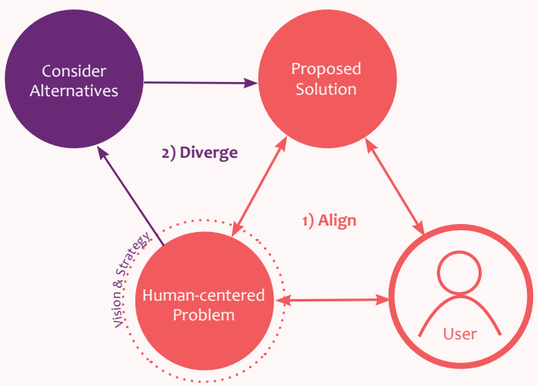
Reframe solution to core problem
Explore the root of the problem behind the solution proposal

Reframing is essential to be able to evaluate the solution (and alternatives), to conclude 'the right thing to build' and to optimize strategic fit (§4.3.1). This frame needs to be user-centered (§2.1; case studies §2.4). There is need for guidance and awareness in reframing towards a problem-/goal-oriented perspective.



Evaluate strategic fit
Evaluate strategic fit by considering:
1) problem-user fit
2) problem-solution fit
3) solution-user fit

§3.4.1 shows the importance of strategic fit. There is an opportunity for Design Thinking to support establishment and (dynamic) optimization of strategic alignment (chapter 3; Lindberg et al., 2011). See also 'design decision' §5.3.1)



Consider alternatives
Diverge to consider alternatives to the initially proposed solution.
Ideate, prototype and test to learn about the strategic fit.
Consider reframing the initial solution proposal.

Divergent thinking is at the core of Design Thinking (chapter 2). By exploring and considering alternatives, the solution (and strategic fit) can be optimized.

5.3 The conceptual model

The conceptual model (fig. 26) is a representation of the proposed problem exploration phase in between getting a request and starting software development. It is a simplified model of reality that helps to understand and consider elements to dive into after receiving a request/idea and prior to software development.
The model aims to support teams (product managers, owners, engineers and their stakeholders) in working problem oriented, and considering alignment and alternatives – facilitating the step of questioning initial requests and ideas.

Next to that, it aids in creating awareness of the need and opportunity to explore the problem space prior to starting development. This foundation can be used to examine and advocate for problem exploration steps in discussion with stakeholders.

5.3.1 Design decisions

Taking the business and strategic fit models (§3.4.1) and Design Thinking (chapter 2 and §3.4.3) together, as well as the required focus on problem, user, and strategy and vision (see §3.4, §3.5 and §4.2), the strategic fit represented

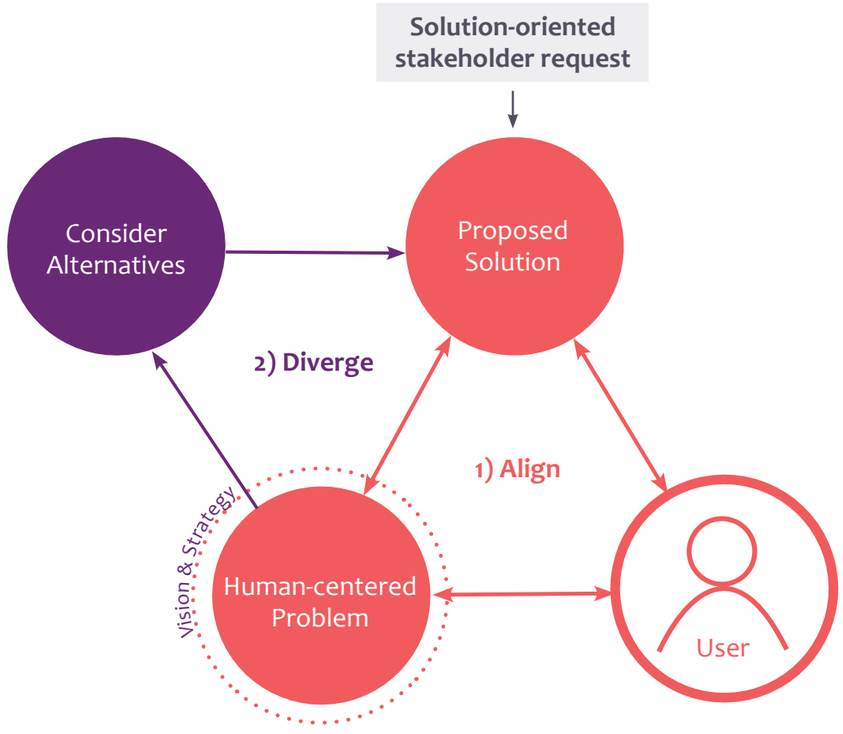


Fig 26. Conceptual model

in this conceptual model includes the problem, solution, user, and strategy and vision. In theory the strategic fit model can be extended to further detail for example technology and organizational resource focus, however as those areas get sufficient attention, the choice is made to focus on key elements that require attention in the problem exploration phase in the context of this research.

'Considering alternatives' is presented in a different color as the other three elements can be seen as a separate model representing strategic alignment.

The 'vision & strategy' element is presented as a circle surrounding the problem, as the problem to be tackled should contribute to the set vision and strategy. As written in chapter 3.4 the problem solved needs to create significant business value and meet current business priorities in order to be considered.

5.4 Fit with Agile

First of all, the model focuses on gaps present in agile methodologies, specifically concerning explicit focus on the core of the problem behind the solution developed. Next to that, the model can be followed to run through the elements prior to and/or throughout agile software development projects. The former is in line with the design up-from approach suggested by Lindberg et al. (2011) to limit risk of conflict with existing processes and structures. In this case, users are encouraged to run through the model (and i.e. reframe towards the core of the problem, build and validate alignment, and diverge to consider alternatives) to consider the underlying problem and consequently to evaluate the proposed solution prior to continuing to actual development.
In the latter option (visualized in fig. 27), considering the model throughout agile software development, the model is viewed in an agile way as one can use it to reflect on the

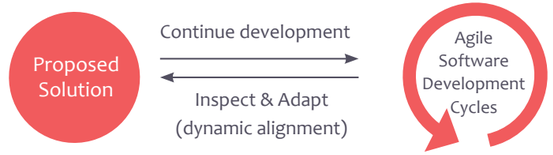


Fig 27. Fit of the conceptual model throughout agile software development cycles

elements - 'inspect & adapt' - any time. This can be done systematically during retrospectives, to stay grounded towards the goal of solving a user-centered problem and to keep optimizing alignment dynamically with more insights coming from agile learning cycles. It can also be done when changes happen in one or more elements of the model, e.g. organizational changes impacting user goals and journeys or impacting vision and strategy, to reflect on what the changes mean for the proposed solution and alignment between the different elements.

5.5 Iterations and validation

The process of getting to this conceptual model has been a journey of simplification. An overview of the two main iterations preceding the final version can be found in appendix 7. An elaboration of a version that captured the same core principles, but in a less comprehensive way, that highly influenced the final model can be found in appendix 16. Feedback was acquired during coaching sessions by design professionals, an internal UX design expert (see notes in appendix 12) and during conversations with members of the Operations Tech team.

Notes of the key insights are presented in appendix 7.

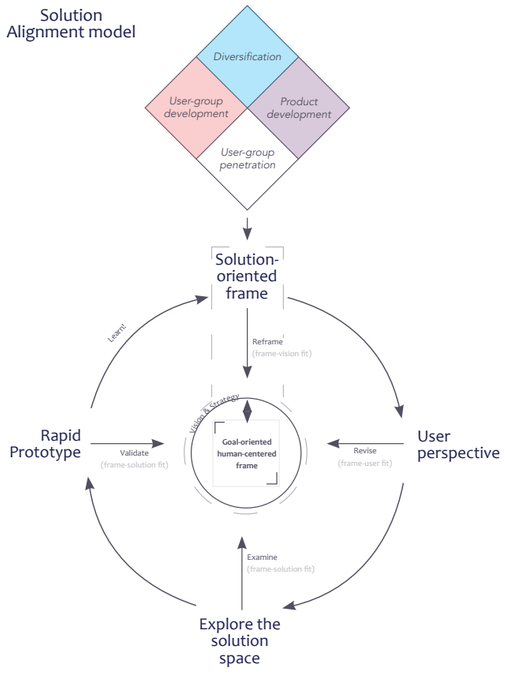


Fig. 28: Main iteration preceding the final conceptual model

Main points of feedback:
- Try to reduce complexity
- *"You are noticing all the problems that there are today"*
- It is confusing how Agile is treated
- It is unclear where to start; consider adding numbers
- Mentioning the goal of alignment makes it more clear

Key choices made comparing the iteration presented in fig. 28 to the final model:
- The diamond including the different scenarios (product maturity levels), see §3.2.3, was removed as the different scenarios did not seem to impact the structure of the elements in a consistent manner (i.e. a project in which a new feature is developed has to align the same elements as a project in which a product is scaled to new users). Further research into the diamond and product maturity levels is recommended to find out if there are generalizable needs related to specific quadrants that could guide the user at the start of a project.
- Wording (e.g. of 'solution oriented frame') was changed after different conversations to optimize understanding.
- Core principles were derived, letting go of the circle representing Design Thinking more completely to focus on the key elements fitting the context of research.

Chapter 6

Concept Design

Turning the Conceptual Model into a 'Problem Deep Dive Canvas' prototype

In this chapter, the conceptual model will be translated into a usable artifact: a 'Problem Deep Dive Canvas'. First, the reframed problem statement and subsequent design goal will be stated, followed by design requirements and product principles based on the conceptual model. Next, a prototype of the canvas will be presented and validated through six validation sessions in which the canvas was applied within the Operations Tech team to actual current requests. Concluding, a final iteration of the product will be presented, taking the gathered feedback into account.

6.1 Design focus

In this section, findings of the theoretical analysis and analysis of the internal context and boundaries to problem exploration are integrated towards a design focus; a problem statement, design goal and design requirements will be presented.

6.1.1 Problem statement

Bases on previous conclusions and the conceptual model, the reframed problem statement is formulated as follows:

> *"Agile practitioners within the Operations Tech team lack the focus and structure to approach solution-oriented requests in a problem-oriented and human-centered way, blocking effective evaluatuation of strategic fit and consideration of alternatives that might better align."*

6.1.2 Design goal

Following the main building-blocks (see §5.2), the focus of the design will be on 1) reframing and 2) user-centered exploration of the core of the problem, to subsequently allow for 3) divergent thinking and evaluation, validation and/or optimization of strategic alignment between the problem, user and solution.

The design goal is formulated as follows:

> ***"Design an easy-to-use, low-threshold tool, supporting product managers and owners, and their collaborators (development team and business stakeholders) in reframing solution-oriented stakeholder requests, (dynamic) strategic alignment between the problem, user, vision & strategy, and solution, and divergent thinking, to put the conceptual model into practice and raise awareness for its key elements."***

6.1.3 Target group

The main target group envisioned to use and initiate using the canvas consists of product owners and product managers in collaboration with business stakeholders and the development teams (and potentially other parties with whom they share dependencies).
The role of product managers and product owners include understanding users needs and the problem (though, as mentioned before, often requirement focused), leading sessions with stakeholders and the team to understand and align needs and possibilities, and translating insights into development requirements and project decisions (see also personas §3.1.7), a responsibility to 'build the right thing', making them a good fit to iniate this change. The openness to experiment with and improve the problem part of the development process and the interest in this topic can be concluded from the interviews and other conversations throughout this research project.

6.1.4 Product principles

The principles directly follow the conceptual model (see §5.2 for more information about the building blocks of the model) and are stated as follows:

Principle 1: Think human-centered and problem-oriented at the core

Principle 2: Dynamically work towards (optimizing) strategic fit

Principle 3: Continuously learn through ideation and experimentation

Principles are meant as guidelines for the user to hold the right mindset during the process, and as the aim of the product is to support this mindset, the solution will include those principles.

The key focus will be on the first principle 'Think human-centered and problem-oriented at the core', as this principle is regarded to be a critical requirement to be able to effectively align (i.e. an understanding of the core problem and users are required in order to evaluate solution fit) and diverge (i.e. an understanding of the core problem and users are required in order to ideate effectively in the right direction to find solutions that better fit for purpose), respectively the second and third principle.

6.1.5 Design requirements

Based on the research, insights and conceptual model, the following design requirements are set-up following the three elements presented in the Venn-diagram in fig. 1 (see § 2.1.2): desirability, viability and feasibility.

Feasibility

DR 1.1: The product can be used by a product manager or product owner in collaboration with stakeholders and/or software developers, without external facilitation (e.g. by a designer)
DR 1.2: It is clear when, why and how to use the product
DR 1.3: The product does not significantly conflict with current structures, to encourage engagement
DR 1.4: The product can be used online (as global stake-holders might be involved and/or a global pandemic requires working online)

Desirability

DR 2.1: The product encourages engagement; people are motivated to use the product and spend time on problem exploration
DR 2.2: The threshold to use the product is low enough (as time-pressure is usually high and the Design Thinking experience of the agile practitioners usually low)
DR 2.3: The product supports shifting from a solution-oriented to a problem-oriented mindset
DR 2.4: The product is easy to understand and use

Viability

DR 3.1: The product supports in creating a better under-standing of a project situation and next steps for further problem exploration
DR 3.2: The product provokes new thinking (within a team and/or stakeholder collaboration)
DR 3.3: The product supports evaluation and optimization of strategic alignment (i.e. better fit-for-purpose solutions are build)

Note that some requirements can potentially be placed in another category as well and are placed in the most fitting category to avoid overlap, e.g. ease of use and understandability might influence all categories, and the threshold to use has a broad influence as well.

6.2 Initial prototype

6.2.1 The problem deep dive canvas for solution-focused requests

The product presented as initial prototype is a canvas that guides the user through the conceptual model, see §6.2.3. for an explanation of how the model and canvas link to each other. This initial prototype will be tested (see §6.3), after which an iteration is made.

Form

A canvas is chosen as form, as it is assumed to provide a low-threshold way to touch upon the product principles, and externalizing knowledge on a canvas assumably allows for (iterative) alignment around the elements on the canvas, e.g. problem, user, solution, opportunities.
The canvas comes with a tool guide explaining why, when and how to use the canvas, see §6.2.2.

Workflow

Similar to the conceptual model, the starting point of the canvas is a solution-focused request. The (core of) the problem behind the request will be analyzed (causes and symptoms), and reframing will take place towards a human-centered problem in the form of a HowMightWe..?' statement. The input comes from questioning the client requesting the solution about pain points and the vision and strategy behind the request, and by talking to users to gain a better understanding of related goals and pain points. The HMW output is used to inspire consideration of alternatives.

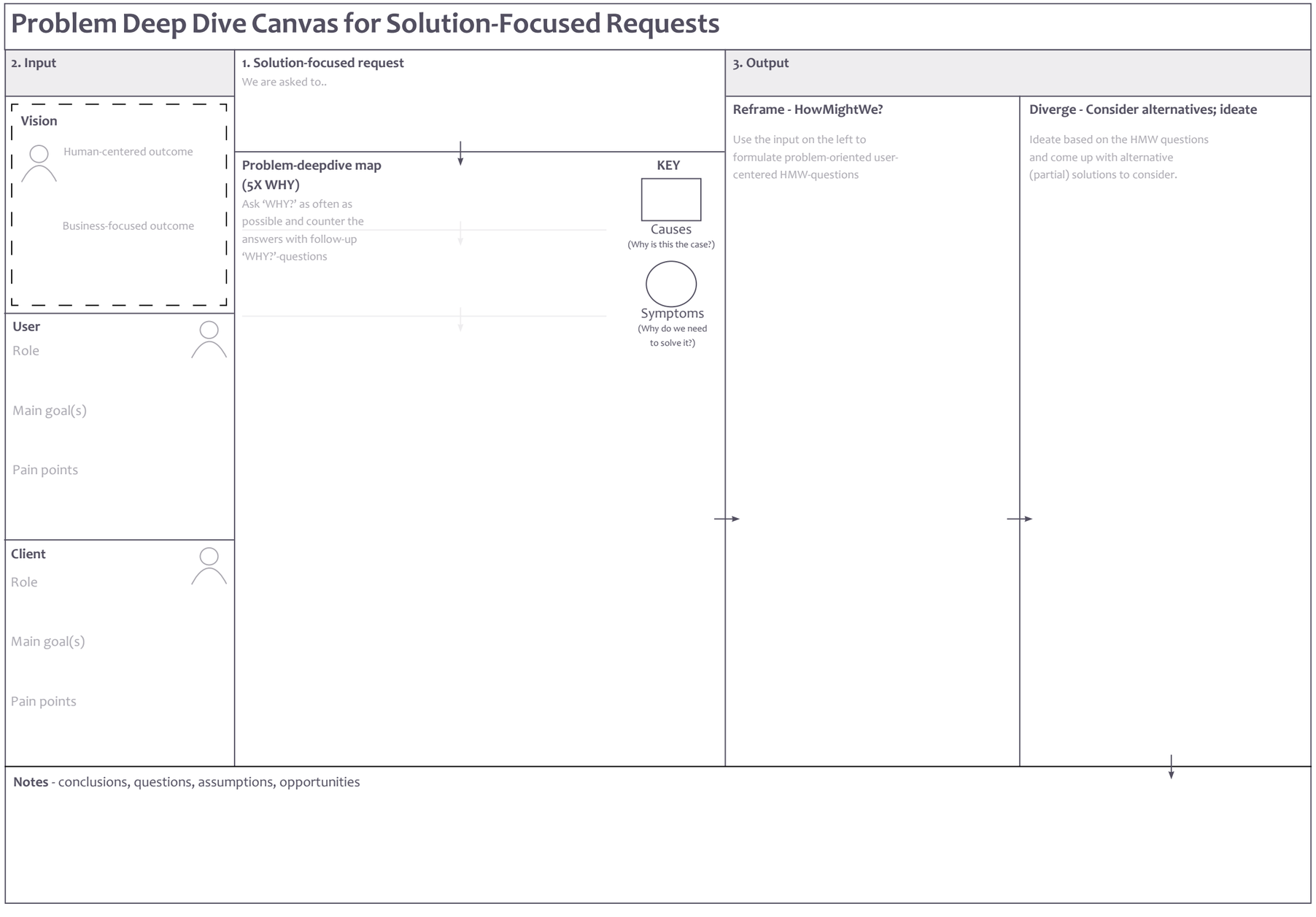


Fig. 29: Problem Deep Dive Canvas prototype. The final version of the canvas can be found in §7.1 and appendix 10.

The output of the canvas
- Eventually, the outcome of the canvas is an alternative assumably better fitting solution based on the core of the problem uncovered. The alternative solution can be tested (e.g. through rapid prototyping) as a next step.
- In the process, partial problems might arise, which can be ideated on as well; solutions to partial problems can possibly be integrated in the (new) solution proposed.
- Especially in initial iterations, the outcome of the canvas might not yet be a potential better fitting solution. In those cases, the canvas provides an overview of knowledge gaps to fill in and assumptions to test; information that needs to be gathered to iterate on the canvas.
- Note that it is also possible to conclude after filling in the canvas and considering alternatives, that the initially proposed solution is indeed the way to go. In this case, the canvas provides an overview of the data grounding this decision.

Low-threshold use of the canvas
The canvas is envisioned to be used in collaboration with developers and stakeholders, to critically reflect on the elements, fill in knowledge gaps and to (dynamically) align the elements as a team, making sure that everyone is on the same page.

To allow for low threshold experimentation and familiarization of the canvas, product managers and product owners are encouraged to fill in the canvas alone or with someone else involved when a requirement comes in to kick-start problem-oriented thinking and to critically reflect on the problem behind the request. As a next step, the canvas can be used to communicate insights, integrate different perspectives and facilitate discussion and alignment within a broader audience.

6.2.2 Problem Deep Dive Tool Guide

The tool guide answers the why, when, how and what of problem deep dive canvas and accompanies the canvas. The tool guide can be found in fig. 30 and enlarged in appendix 11. An overview of the elements present in the toolguide is presented below:

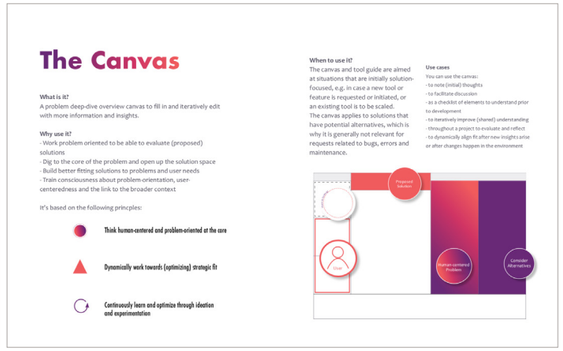
What is it?
A problem deep-dive overview canvas to fill in and iteratively edit with more information and insights.

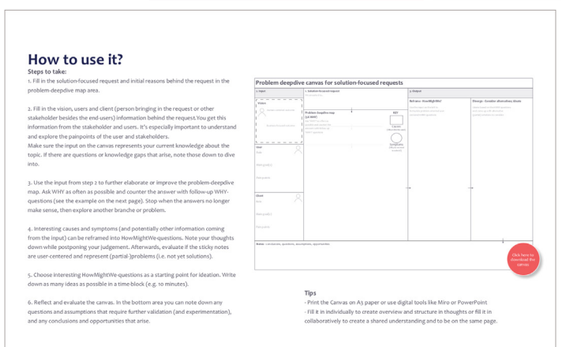
Why use it?
- Work problem oriented to be able to evaluate (proposed) solutions
- Dig to the core of the problem and open up the solution space - Build better fitting solutions to problems and user needs
- Train consciousness about problem-orientation, user- centeredness and the link to the broader context

When to use it?
The canvas and tool guide are aimed at situations that are initially solution- focused, e.g. in case a new tool or feature is requested or initiated, or an existing tool is to be scaled.
The canvas applies to solutions that have potential alternatives, which is why it is generally not relevant for requests related to bugs, errors and maintenance.

Use cases
You can use the canvas:
- to note (initial) thoughts
- to facilitate discussion
- as a checklist of elements to understand prior to development
- to iteratively improve (shared) understanding
- throughout a project to evaluate and reflect
- to dynamically align fit after new insights arise or after changes happen in the environment

How to use it?
Steps to take:
1. Fill in the solution-focused request and initial reasons behind the request in the problem-deepdive map area.
2. Fill in the vision, users and client (person bringing in the request or other stakeholder besides the end-users) information behind the request.You get this information from the stakeholder and users. It's especially important to understand and explore the painpoints of the user and stakeholders.
   Make sure the input on the canvas represents your current knowledge about the topic. If there are questions or knowledge gaps that arise, note those down to dive into.
3. Use the input from step 2 to further elaborate or improve the problem-deepdive map. Ask WHY as often as possible and counter the answer with follow-up WHY- questions (see the example on the next page). Stop when





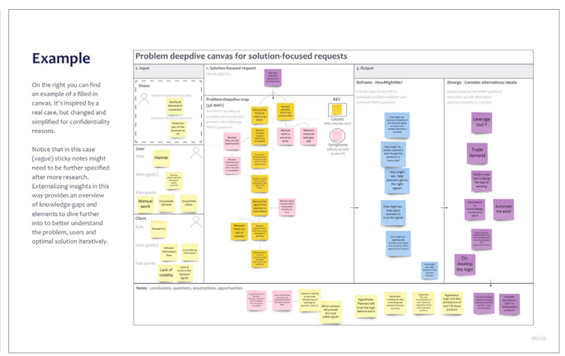


Fig 30. Tool guide booklet

the answers no longer make sense, then explore another branche or problem.

4. Interesting causes and symptoms (and potentially other information coming from the input) can be reframed into HowMightWe-questions. Note your thoughts down while postponing your judgement. Afterwards, evaluate if the sticky notes are user-centered and represent (partial-)problems (i.e. not yet solutions).
5. Choose interesting HowMightWe-questions as a starting point for ideation. Write down as many ideas as possible in a time-block (e.g. 10 minutes).
6. Reflect and evaluate the canvas. In the bottom area you can note down any questions and assumptions that require further validation (and experimentation), and any conclusions and opportunities that arise.

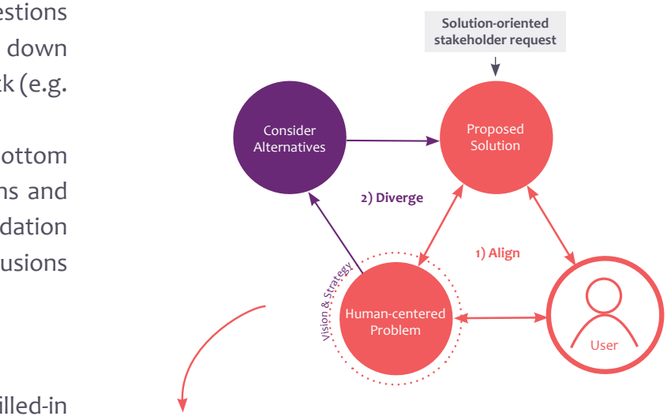
Example

Next to that, the tool guide includes a filled-in example. The example is inspired by an actual project, but is changed and simplified to take confidentiality into account.

6.2.3 Fit with the conceptual model

As visualized in fig. 31, there is a direct relation between the conceptual model and the problem-deepdive canvas.

The canvas allows for dynamic alignment, as information can be reconsidered, added, changed etc. Also when development is started, the canvas can be used to not lose sight of the vision, user-perspectives and the problem that is to be solved. Next to that, by evaluating the elements to see if there are any new insights or changes (e.g. in the vision, the environment of the user, or problems that arise). The solution and related communication can potentially be adapted.



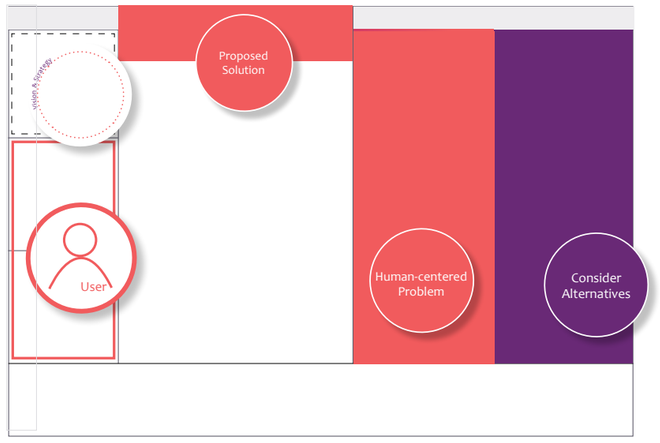


Fig. 31: Direct link of the canvas and the conceptual model

6.3 Concept validation

To validate the canvas, six sessions were held with either one or two participants (product managers, product owners and developers) to go through a current solution-focused request case. Validation insights come from observing the participants during the sessions, and from feedback from the participants during the sessions and through a survey sent out after the session. Five sessions were with team-members of the operations technology area (the scope of this graduation project), one session (case 6) with someone from the marketing technology area (unprimed to the subject) to get insights into generalizability.

6.3.1 Working sessions set-up

The goal of the sessions in which the canvas was tested, was to find out if and how the design requirements (see §6.1.5) are met, if it effectively supports the set-out principles (see §6.1.4) , and how the canvas is perceived and intuitively used.

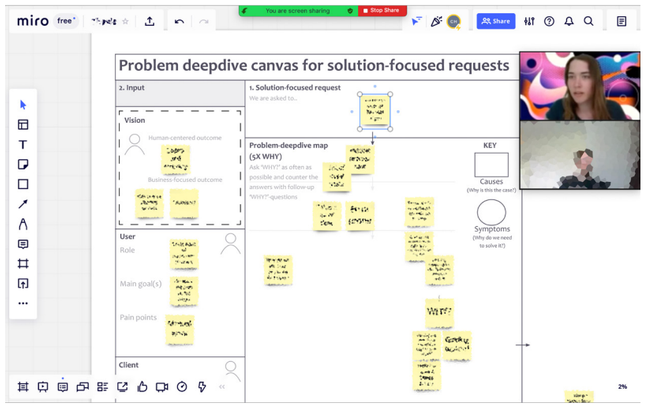


Fig. 32: Screenshot of a working session

For the participants the goal was to fill in a first version of the canvas with their current knowledge about the project, conclude knowledge gaps and assumptions, and consequently determine next steps in the project.

The canvas and the booklet were sent in advance and participants came in with a certain solution focused request as a starting point.

The sessions were all done online via Zoom (online video communication platform) and in Miro (online whiteboard and collaboration platform) with the exception of one session in which the canvas was tested in Adobe Acrobat (PDF viewer and editor). Each session took 45-60 minutes.

To keep the threshold to use and get used to the canvas as low as possible, the canvas was tested with either one or two participants in one session. Ideally, someone gets comfortable with the canvas alone or in collaboration with someone else prior to facilitating it within a larger group.

An overview of the sessions, main session insights related to the cases and main session insights related to the use of the canvas are presented below. To respect confidentiality of the cases and participants, the cases are simplified and anonymized. Three canvases filled in can be found in the appendix 16.

6.3.2 Case 1: add feature to automate the input of targets (simplified)

PO Dev

Main case insight by using the canvas:

The targets need to be entered in time by the team lead to unlock the next step for regular users (bottleneck pain point users), however why the users are blocked and if they can be 'unblocked' prior to entering the targets is an open question. If possible, the solution space changes significantly.

Session insights about the canvas:

- The day-to-day team of the participants builds and maintains one product with the same clients and users. Participants assume that the left side is therefore known and that provides the benefit of efficiency in meetings in which initiatives are proposed – the focus can be on the 5xWHY.

- There is an interest to use it in business refinement meetings, initially possibly scoped down further to the 5xWHY exercise: *"Our focus can be on the 5 whys and we can do it quite quickly in business refinement meetings […] and ask more questions about the underlying problem instead of what they think they need." (Participant A)*

- There is uncertainty about when to move on, e.g. when is a HMW question framed correctly?

- In this session, it became especially clear that the canvas could support in raising awareness to ask questions about current assumptions and limitations.

- The outcome of this session is surprisingly impactful and usable; it impacts the current trajectory of the project, and the relevance of the initial request.

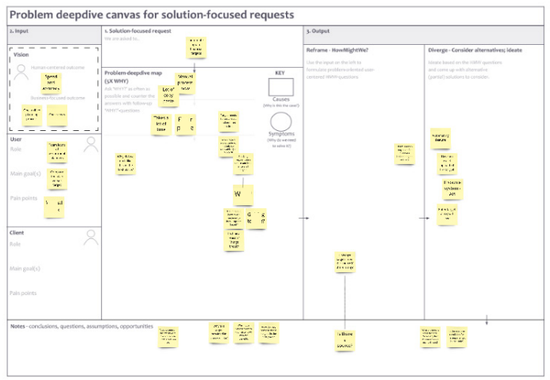


Fig. 33: Filled in canvas case 1 (blurred concerning confidentiality; see appendix 16)

Participant A: "It's definitely useful, because it - especially the 5 why's - forces you to kind of dig deeper than you initially did and look at underlying issues."

Participant B: "I have the same feeling. It is helpful for refinement as a team and it opens up questions we don't question ourselves for example, because we see them as part of the design [of the software product]. This exercise made it apparent that we can question those things."

6.3.3 Case 2: get manual reports from the users to have the required data (simplified)

PO

Main case insight by using the canvas:

Manual reports are not updated anymore because of changes in the organization; we might be able to get the currently unavailable manual reports we need from users (to enable other users to work with the output) in a different way (e.g. generating data, sending it through a different channel).

Session insights about the canvas:

- The initial broadly formulated solution-oriented request to scale the product to the digital channel was already 'validated'; the project passed that stage. In this case, the solution focused request quadrant was further scoped down after an initial problem deep dive: *"If we frame it to the solution-focused request at this moment is to get manual reports from the users to have the required data" (Participant C)*. Scoping down iteratively in this way worked well in this case.

- In this case there are two types of users: the end-user of the tool (output) and the user providing data into the system (input). The pain point we're addressing in this case is related to the latter. There is currently no space to consider and address multiple users on the canvas.

- The participant missed a space on the canvas to note dependencies. In this case that would be the vendor and different parallel teams sharing resources and systems. If there are changes in the backlog they need to be notified. *"I would like to add a box for dependencies, upstream and downstream, in terms of systems, technology, data and resources." (Participant C)*

- The outcome of the session provided clarity on current limitations and assumptions that can be tested today.

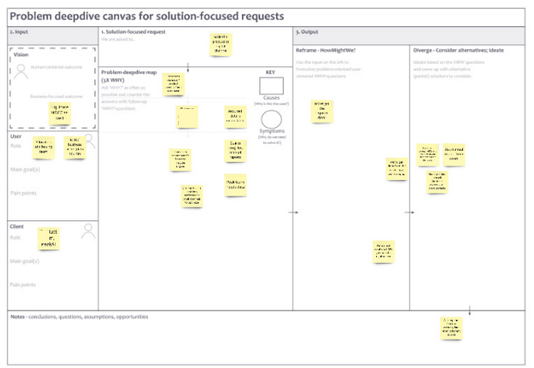


Fig. 34: Filled in canvas case 2 (blurred concerning confidentiality; see appendix 16)

Participant C: "It is very insightful in understanding what the current problem is that we have to focus on now, the assumption we can start testing today. That makes it very agile, also acknowledging that things can change."

Participant C: "The end-user is what we tend to forget, we focus on technical debt and forget the fact that it is in the end going to help end-users. The canvas gives a different perspective focused on delivering value to the end user and taking that into account in the decisions we make further down the line."

6.3.4 Case 3: turn excel solution into tech-supported solution (simplified)

PM

Main case insight by using the canvas:

Excel is often the starting point of the technology work, a kind of prototype coming from the business users. However, it limits business users at a certain point and it is prone to error. We might be able to advance their 'prototype' in the first place (e.g. by making a template in or light version of another tool, or partnering with analytics)

Session insights about the canvas:

- A critical insight is that the formulation of the initial request and the goal/intention with which one fills in the 5xWHY, significantly impacts the direction of the reframing. Not specifying it can lead to interesting insights, however when there is a certain underlying goal related to the project, it is important to get clear about it. E.g. one can dive into:
  - why we need a certain solution on a problem and benefit level (e.g. a lack of visibility or accuracy, saving time and money),
  - why we need a certain solution on a functionality level (e.g. having a weekly forecast and further details), or
  - why we are having a certain problem in the first place (e.g. knowledge about a more advanced tool lacking).

  In this case, the initial request was too broadly and vaguely formulated and we broadly dove into why this problem exists in the first place, however on hindsight the expectation or wish was to find out more about the functionality level and how it might be achieved in a better way.

- A main question this participant mentioned throughout the session is when we consider that we have an answer and can move on to the next area step.

- By diving into the core of the problem, the outcome of this session provided insight and awareness concerning possible prevention of getting a similar request (i.e. preventing business users to have to work in Excel with its limited functionalities in the first place and thereby making their initial prototype more sustainable).
  However it did not provide insight into the details of the current request (short-term goal), insights can be used to potentially solve this problem on a higher level (long-term goal).

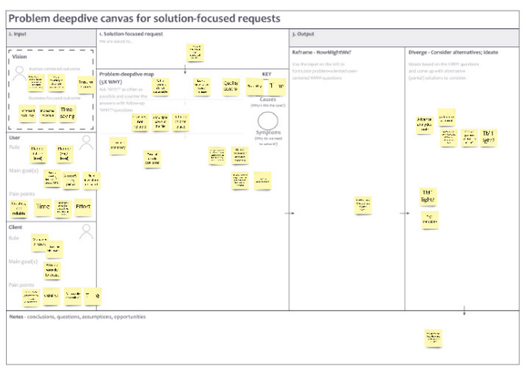


Fig. 35: Filled in canvas case 3 (blurred concerning confidentiality; see appendix 16)

6.3.5 Case 4: source data from a central location (simplified)

PO Dev

Main case insight by using the canvas:

The participants found out that the canvas is not relevant in either/or technical cases. To dive deeper into the problem, iterations of using the canvas will have to include business counterparts in the discussion to cover knowledge gaps.

Session insights about the canvas:

- Participants tried to map a situation in which two options were already known ('should we take solution A or solution B?), which made the solution-focused mindset and intuitive solution-focused use clear. Interestingly, by 'failing fast' they found out that the canvas doesn't work for such technical well-defined problems and concluded that they "*could use the canvas for any business problem that tech could get a solution for*" (Participant E).

- Participants chose to use Acrobat Reader, which was not ideal as comments are not visible at a glance. For those who don't work with Miro, PowerPoint could be an alternative.

- At one point the participants got stuck and found out that for the business side of this problem they need the business counterparts to fill it in, "*it's one step earlier than technical problems we are mostly solving in our day-to-day.*" (Participant E).

- The participants are enthusiastic to introduce it to their business counterparts at the stage in which new initiatives are proposed, because: "*We see enough challenges, but there is no structured way how it comes to us [e.g. the loudest voice]. This canvas could really help.*" *(Participant E);* "*[…] during the session with senior leaders, to get the initiatives in this way. I would like to try that out!*" *(Participant F);* "*We could test it out right after planning, in the first few weeks of the quarter in which first ideas pop up, and see if it helps.*" *(Participant E to Participant F).*

- The assumption is made by the participants that introducing this canvas to business counterparts as the starting point of initiatives could start the conversation earlier, adding the tech team earlier on in the discussion. "Once they come to us there is normally months of discussion prior without us and the product is half finished already."

> *Participant E: "I see it as an initiating platform. This is the stage to combine all tech teams. You can consider how broad to take it, you could even go further and let go of the solution that is already in place."*
>
> *Participant F: "I like that it is Tech initiating the search for problems in the business"*

6.3.6 Case 5: automate the process of order alignment (simplified)

PM

Main case insight by using the canvas:

We're not 'optimizing' the alignment (vision), but matching the quantities to align more efficiently in the system. This is a short-term pain reliever. Diving into the underlying problem is useful to consider solving the actual and long-term problem (as well).

Session insights about the canvas:

- There is uncertainty about when to proceed to the next step. *"When to go from the problem deep dive to the next step? What conditions must be met? Could there be a 'control question' like 'do you have enough input to ideate to more solution options than the initial request?'" (Participant G)*

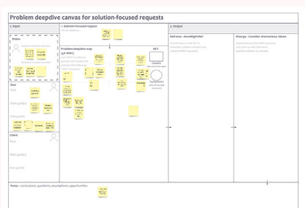
- Filling in the canvas and getting thoughts onto paper supports the participant in getting thoughts clear, *"the canvas steers you in a methodical way to get the problem clear", "it visualizes thoughts in a clear insightful way and it's easy to do it together." (Participant G)*

- The human-centered focus was noticed, understood and consciously focused on throughout the session. *"I like that the sheet is business focused AND human centered. Because we are usually pretty much focusing on the business case in terms of time and money saved." (Participant G)*

- By diving into the core of the problem, the participant became aware of the difference between the short-term and the long-term problem behind the request, which is an important insight to be able to start solving the core of the problem.

- The participant mentions how the canvas provides a different user-centered perspective to business case initiations: *"It's a different way to initiate a business case I think […] it can be a stepping stone towards a business case in which you focus on more than quantifying output. Another perspective to the same story focusing on the main goals and user pain points." (Participant G)*

- The participant mentions concerns related to people who don't see the need or don't have the intention to dive one step deeper into the problem, *"if they have to fill it in they can just turn it to their hand and go to the next steps without really filling in the problem details." (Participant G)*



Participant G: "We tend to immediately go to the technical details, this [canvas] forces you to consider the bigger picture and the details on user-level. I think that also helps in communicating benefits to the business, e.g. senior management, if we have an initiative."

Fig 36. Filled in canvas case 5 (blurred concerning confidentiality; see appendix 16)

6.3.7 Case 6: automate contract feature (simplified) – marketing technology area

PO

Main case insight by using the canvas:

There might be possibilities to reach the same goal in different ways, e.g. through a plug-in instead of building a whole new architecture, we are currently not considering at all.

Session insights about the canvas:

- The participant mentions that the essential points discussed in proposal meetings are in the canvas (opportunities, questions, assumptions) and that the canvas could help to structure such a session: *"for me leading such a proposal sessions, I can direct the attention in a structured way. I think we lack that structure at the moment and there is an attention span involved of course. Now I could say 'okay, we spent enough time on vision, let's go to the problem deep dive' for example." (Participant H)*

- The structure provides an overview at a glance, useful to browse through quickly and get everyone on the same page. *"We have a stakeholder meeting, twice a quarter, in which all different stakeholders come with their ask. It would help if we could discuss canvases, as it helps to understand different perspectives of the ask that a certain stakeholder brings to the table. […] Free styling is often counterproductive than following structure. It also forces to become concrete." (Participant H)*

- The participant mentions that people might shy away from the problem deep dive when teams don't have much time, but that it is the responsibility of the PO to make sure the problem is well understood and validated. *"When teams are stacked with work and there is a lot on the backlog, a proposal meeting might be focused on just getting the requirements. Developers don't care. I see it as my job to make sure it happens in a presentable and structured way." (Participant H)*

- Considering which key stakeholders to involve in a session is something to think about in advance to make the session more productive.

Participant H: "I'm enthusiastic. It really nails down a lot of things!"

Participant H: "Communication gaps are a large pain point for us and bridging it is key. The canvas is a way to bring transparency, normal and simple, helps to put concrete thoughts, questions and ideas onto paper."

Participant H: "It's very agile: now that we have this information we make these adaptations e.g. to the problem statement."

6.3.8 Survey results

Six out of the eight participants filled in the survey afterwards (three product owners and three product managers). The set-up and results can be found in appendix 9. One provided answers to the open questions verbally, which are taken into account in case 5.

Key insights:

- It is key to position the canvas well in order to manage expectations; i.e. it needs to be clear that its intended use is mainly diving into business problems that technology can provide a solution for, not well-defined technology problems in which diverging from the solution proposed is not an option.

- It should be clear what the outcome of the canvas is; currently the last section is labeled 'notes' while it actually holds main takeaways from the document.

- One participant mentions that "the value of the canvas becomes very evident by using it", which fits my observation as well - value becomes apparent in use, when actual insights are provided, which means people have to get motivated to try it out on a real case to be able to assess its value.

"The value of the canvas becomes very evident by using it. So I'd recommend any team to try it out a few times to assess the value and fit for purpose."

- All six participants who filled in the survey mention that they would use the canvas again. User-centricity, agility, better fit-for-purpose solutions and the notion to challenge decisions are main reasons mentioned:

  "Canvas encourages a problem oriented thought process and stresses on looking at the problem from a user centric approach. Additionally it reflects fundamental Agile principles to build user centric potentially shippable incremental solutions."

  "It will enable us to think outside of the box and provide a more fit-for-purpose solution"

  "It challenges the development of design decisions that were not really or strictly user centered."

6.4 Discussion

Insights gathered in the validation stage can be used to evaluate whether the proposed solution aligns with the design goal as stated in §6.1.2: *"Design an easy-to-use, low-threshold tool, supporting product managers and owners, and their collaborators (development team and business stakeholders) in reframing solution-oriented stakeholder requests, (dynamic) strategic alignment between the problem, user, vision & strategy, and solution, and divergent thinking, to put the conceptual model into practice and raise awareness for its key elements."*

Key insights will be used to improve the tool and will be taken into account in the final discussion chapter covering recommendations, limitations, and implications.

The proposed tool will be discussed in relation to the design requirements (see §), covering the viability, feasibility, and desirability aspects. Moreover, the design goal includes the three main product principles following the theoretical model (see §6.1.4), which will be evaluated.

6.4.1 Evaluation design requirements - Feasibility

DR 1.1: The product can be used by a product manager or product owner in collaboration with stakeholders and/or software developers, without external facilitation (e.g. by a designer)
Five out of six participants who filled in the survey agree that the canvas is easy to understand and logical to follow without an external facilitator; one participant remains neutral about this. It might be hard for participants to judge this factor at this stage as they are just introduced to the new tool. Assumably, it might take some practice to get familiar with the steps and the process - learning by doing to become better at it.

In the sessions, participants did have questions about when to move to the next stage, e.g. *"When to go from the problem deep dive to the next step? What conditions must be met?"* (participant G), see also DR 1.2. To ease facilitation by product managers and owners, adding more guidelines about when to move to the next step would be helpful.

Considering the collaboration with software developers, in two sessions, a developer joined a product owner to fill in the canvas together. Both developers were positive about using it in their projects and felt comfortable about it. The product was understood and verbally applied into context (e.g. indicating when it would be useful), implying general understanding.
In case 4, the intuitive start was technical solution-focused. However, this allowed for 'learning by doing' about the fit of the canvas, and understanding was observed through the comments made about the fit of the canvas. Digging deeper and critically asking further why-questions worked immediately well in case 1.

A product owner or manager's use of the canvas in collaboration with stakeholders was not tested. However, willingness to use it in proposal meetings has been indicated, which suggests confidence to use it without external facilitation.

Participants' confidence in their ability to use the canvas evaluated after the session is not fully representative of first-time users who will experiment with it independently, i.e. in the validation sessions questions could be asked to the researcher when the process blocked. Questions participants asked during the session already cover some of the initial questions future users might have that can be considered in the canvas iteration.

Insights suggest that this requirement is met; however, feedback after further independent use is needed to validate this.

DR 1.2: It is clear when, why, and how to use the product

Multiple participants raised their concerns about when to proceed to the following steps on the canvas, e.g. *"When to go from the problem deep dive to the next step? What conditions must be met?"*

In some cases (e.g. cases 1 and 2), particular key insights made it immediately clear what path to explore further as those insights sparked new opportunities. In other cases (e.g. cases 3 and 4), it has been trial and error, iteratively moving towards a logical story. As it is an iterative process, one can continue to come back to earlier stages when the process blocks (e.g. when no useful or promising alternatives are found); participants might need to experience the intuitive feeling of when to proceed to fully understand it.

Participants came in open-minded and curious about what the canvas would bring to a specific request case they brought in, without specifying (or being asked) about their intentions or goals to use the canvas. For exploratory purposes and learning about the value and fit of the canvas, this approach works for both the participant and the researcher/observer. However, to use the canvas most efficiently, it is recommended to specify the intention before using it. This intention can be to 'freestyle explore a request', but it can also be specifically about understanding product characteristics to find out how they might be achieved in a better way (case 3). Not specifying the intention might result in exploration of other tracks, which might be interesting in terms of innovation opportunities but less efficient for the project at hand and expectations of the users.

The intention impacts the formulation of the why-questions in the 5xWHY exercise. To improve this, adding a section about the intention of using the canvas to the tool guide, as well as directional WHY-questions, can guide the users.

This requirement can continuously be improved with more text and more information. However, it is encouraged to learn by doing; it is a tool, not a strict procedure, so one can use the tool to one's advantage.

After using the canvas in the validation session, all six participants who filled in the survey indicated that it is clear when they can use the canvas. Note that the initial questions of participants were answered during the validation sessions. The missing information can be taken into account in the information provided to new users.

DR 1.3: The product does not significantly conflict with current structures to encourage engagement

As the product can initially be used in little time and up-front, there are no significant conflicts with current structures or processes. Participant E even mentions that the canvas could even support the current structure of initiatives coming to the technology teams: *"We see enough challenges, but there is no structured way how it comes to us [e.g. the loudest voice]. This canvas could really help." (Participant E).*

DR 1.4: The product can be used online (as global stakeholders might be involved and/or a global pandemic requires working online)

All validation sessions were conducted online through zoom, which confirms the ability to go through it online. Miro is a great platform to set up, share, edit and update the canvas; the digital sticky notes allow for a clear and visual overview. Digital tools that do not allow for an overview at a glance (e.g. Adobe Acrobat in which comments are hidden, tested in case 4) should be discouraged. PowerPoint could be an alternative to Miro.

The canvas was not tested offline, as it was required to work from home due to the pandemic (Covid-19). As Design Thinking canvases are usually mainly used offline, no significant problems are expected when using the canvas in person.

6.4.2 Evaluation design requirements - Desirability

DR 2.1: The product encourages engagement; people are motivated to use the product and spend time on problem exploration

By providing a low threshold, simple, structured process, the tool encourages people to dedicate time to problem exploration. The feedback received suggests that this indeed works, e.g. the survey indicates that all participants are motivated to use the canvas again and would recommend it to a colleague. However, for objective validation, further evaluation can be done to conclude if this requirement is quantitatively achieved by spending more time on problem exploration.

DR 2.2: The threshold to use the product is low enough (as time-pressure is usually high and the Design Thinking experience of the agile practitioners usually low)

The threshold to test the canvas is relatively low as no large commitments or sacrifices have to be made in terms of time or changes to the current process when starting out (while there are also possibilities to expand the impact by using it more extensively). As participants were enthusiastic to test it out in their own projects, no resistance to the threshold to use the canvas is concluded. However, this judgment is made after an initial run-through in which insights are already gathered and the value of the product

is experienced. However, the threshold concerning the use of the product is low as it takes little time to get started and test it out, the perceived threshold to use it independently for the first time has not been tested (i.e., users were invited to a session to test the canvas; the threshold to use it without being asked to use it might be higher).

Testing was done regarding the set up of an initial version of the problem deep dive canvas. It would be interesting to test and reflect on the threshold to use the canvas for a longer period of time, in which users come back to the canvas to iterate with new knowledge.
In the feedback survey, five out of six participants indicated that the threshold to use the canvas is low enough. One participant remains neutral about this statement.

DR 2.3: The product supports shifting from a solution-oriented to a problem-oriented mindset
The canvas supports problem-oriented and human-centered thinking. This can be concluded from the reframes made, questions asked and remarks made by participants during the session, and the feedback given in hindsight.

> *"Canvas encourages a problem-oriented thought process and stresses looking at the problem from a user-centric approach. Additionally, it reflects fundamental Agile principles to build user-centric potentially shippable incremental solutions." (survey)*
>
> *"It challenges the development of design decisions that were not really or strictly user-centered." (survey)*
>
> *"I like that the sheet is business-focused AND human-centered. Because we are usually pretty much focusing on the business case in terms of time and money saved." (Participant G)*

Survey results show that all six participants who filled it in feel that the canvas helps shift their thinking from solution-oriented to problem-oriented.

DR 2.4: The product is easy to understand and use; the example provided is representative and easy to understand
Feedback concerning the information provided and the example given has been positive; the example clarifies the use of the canvas, is representative of cases (e.g. initially targeting manual work). The example given answers some questions participants have and provides inspiration in terms of the type of questions that can be asked in the deep dive and the simplicity with which one can fill it in. Participants turn to the example if they block in the process and for inspiration.

As the example is used as a resource when people get stuck in the process and for inspiration purposes, it can be considered to add more examples of filled-in canvases, including the main insights derived from using it, to build an inspirational library of use-cases and added value. In this way, patterns between canvases can be sought (e.g. in terms of the type of questions) to learn more holistically about the type of problems solved.

See DR 1.3 for more information about unclarities in using the canvas.

As some participants missed certain elements (e.g. dependencies and having multiple users in case 2 and considering stakeholders in case 4), allowing individual adaptations by making the canvas modular could be an option. As other participants did not miss the same elements, keeping the main canvas as simple as possible is recommended to avoid confusion and keep the threshold as low as possible. Therefore, allowing for modularity and not integrating those elements into the main canvas might be best.

6.4.3 Evaluation design requirements - Viability

DR 3.1: The product supports in creating a better understanding of a project situation and next steps for further problem exploration
As the canvas supports reflecting on knowledge gaps and assumptions, the project situation and next steps become clearer.

> *"It is very insightful in understanding what the current problem is that we have to focus on now, the assumption we can start testing today. That makes it very agile, also acknowledging that things can change." (Participant C)*

In most cases, outcomes provided action points towards knowledge to acquire to further dive into the problem (e.g. cases 3 and 4) and/or assumptions to test that potentially significantly impact the trajectory of the project (e.g. case 1, 2, and 6).

All six participants who filled in the survey indicate that the canvas is useful to point out knowledge gaps.

DR 3.2: The product provokes new thinking (within a team and/or stakeholder collaboration)
Feedback provided by the participants indicate that the product brings a different - specifically user-centered - perspective and allows for out-of-the-box thinking, e.g.:

> *"It's a different way to initiate a business case I think […] it can be a stepping stone towards a business case in which you focus on more than quantifying output. Another perspective to the same story focusing on the main goals and user pain points." (Participant G)*
>
> *"It will enable us to think outside of the box and provide a more fit-for-purpose solution" (survey)*
>
> *"The end-user is what we tend to forget, we focus on technical debt and forget the fact that it is, in the end, going to help end-users. The canvas gives a different perspective focused on delivering value to the end-user and taking that into account in the decisions we make further down the line." (Participant C)*

DR 3.3: The product supports evaluation and optimization of strategic alignment (i.e., better fit-for-purpose solutions are built)
The product does allow for evaluation of strategic fit, in an agile/dynamic manner, by encouraging to revisit and reflect on the canvas when more information is gathered. This was not extensively tested due to the time and

scope of the graduation project.
Iterations made and suggested throughout and at the end of the sessions do suggest that awareness and potential concerning 'working dynamically towards strategic fit' when new or more information is acquired or when changes happen is present. E.g. in case 2, the problem was scoped down after an initial run-through; in case 4, an iteration was proposed after evaluating the initial outcome of the canvas; and participants in cases 2 and 6 explicitly mentioned the agile nature of the canvas.
However, as the canvas has not been used throughout a project, further use and evaluation are required to validate if this requirement is met to determine how users can potentially be further supported in this process.

Next to that, it is hard to judge at this point if the canvas actually supports improving strategic fit. Though, looking at the outcomes of the sessions in which the canvas was tested on actual running requests, outcomes could significantly impact the further trajectory of the project (especially in cases 1, 2 and 6) or potentially start new (longer-term) projects directed at the core of the problem (case 3 and 5). The assumed better fit for purpose solutions and their significant impact on the projects suggest potential value towards improving strategic fit.

6.4.4 Evaluation product principles

The first principle, concerning problem-oriented and human-centered thinking, is covered in DR 2.3 and 3.2, and it can be concluded from the actual reframing done in the sessions and the feedback provided that this principle is well supported by the canvas.

The second principle, concerning dynamic alignment, is covered in DR 3.3. The product allows for evaluation in a dynamic manner, and initial iterations suggest that awareness concerning this principle was raised. However, further evaluation over an extended period of time is required to validate if the principle is effectively achieved.

The third principle, concerning ideation and experimentation, is not covered in the design requirements. The focus has been mainly on the other two principles, especially the first one, as an understanding of the core problem and the users is required to ideate and experiment effectively in the right direction to find solutions that better fit for purpose. However, the product does touch upon this principle to create initial awareness for it.
The product does support ideation as a specific step in the process. During the sessions, participants were aware of the ideation principle, e.g. participant G mentioned if there can be a control question to go from the 5xWHY to the next step based on this element, "Could there be a 'control question' like 'do you have enough input to ideate to more solution options than the initial request?'". In the first version of the canvas, set up by participants during the individual sessions, multiple alternatives were considered in cases 1, 2, 3 and 6 (the other cases did not reach this stage yet). However, the participants did not push themselves further than the initial initiatives they had in mind. So, a pitfall could be that the users immediately fall back into their solution-oriented mindset after reframing the problem and that limited ideation happens.
On the one hand, the solution might already be better fitting and solving the core of the problem in a more effective and/or efficient way, achieving the purpose of the canvas to a certain extent. On the other hand, it would be a missed opportunity if solution exploration is limited, especially regarding (more radical) innovation opportunities. Giving guidelines concerning time-blocking and challenging users to come up with a certain amount of ideas per 'HowMightWe..?' can potentially address this pitfall.

Ideally, in a later stage of working with the canvas, multiple options are potentially explored and assumptions are tested, e.g. through rapid prototypes, to validate (partial) solutions and to learn more about the problem at hand. The 'experimentation' part of this requirement has not been part of the scope of this project but can be added to extend the tool and guidance in this area when the canvas is established. Therefore, the presence and support of this part of the third principle can not be validated.

6.5 Chapter conclusions

In this chapter, an initial prototype of the 'Problem Deep Dive Canvas' was presented following the design goal and requirements specified. This prototype was tested through six validation sessions, and evaluated regarding its feasibility, desirability and viability afterwards.

Concerning its **feasibility**, with the current information available, design requirements concerning independent use (no designer or external facilitation required), understanding of the information, canvas and example (also by highly technical developers), and online usage are met, suggesting feasibility. To increase clarity concerning how and when to use the canvas, especially for first time users, guidelines can be added concerning the intention of use, the WHY-questions in the problem deep dive section and continuation to next steps on the canvas to help prevent particular bottlenecks (coming from the validation sessions) in the process. How and when to use the canvas will become clearer by using the canvas and experimenting with it - learning by doing.
Moreover, feasibility is increased by the notion that someone is not dependent on external support, resources, or other information to be able to try and experiment with the product. As there are different levels concerning the intensiveness of use (see §6.2.1), the threshold to get familiar with the canvas and to test it out is considerably low, increasing the chance of adoption as no significant commitments or sacrifices have to be made in terms of time or changes to current process when starting out (while there are also possibilities to expand the impact by using it more extensively).
The product is already ready for use and can result in impactful and valuable insights; outcomes of the validation sessions have already been impactful regarding the further trajectory of projects (case 1, 2, 6), which increase feasibility.
Further testing over a more extended period of time is required to evaluate the threshold and fit for independent use and use throughout the course of a project.

In terms of **desirability**, the threshold to test the canvas is low as no significant sacrifices have to be made in terms of time or changes to current processes. The product fits the current level of effort product managers and owners are willing to spend on an initial problem deep dive. The low initial threshold to test is the first step towards awareness and further adoption of the canvas.

The product supports shifting the solution-oriented mindset to a problem-oriented one in a simple yet effective manner, increasing desirability. The product fits the current level of effort product managers and owners are willing to spend on an initial problem deep dive. Therefore, the participants indicated motivation to use and recommend the canvas.

Further testing needs to be done regarding the threshold to use the canvas dynamically for a more extended period of time and regarding individual use without the ability to ask questions to an external facilitator when the process blocks.

Regarding **viability**, the idea is that the canvas supports the development of products that better align with the core of the problem one intends to solve and the related user pain points, which would save time and rework, and would increase strategic fit and customer satisfaction (see also §3.4.1 about strategic fit). Problem exploration does take some time in the short term, but more optimal fit-for-purpose solutions make up for that time in the long term.
The initial validation phase of the canvas does not include validation concerning the improved effectiveness of solutions, as the focus has been on an initial run-through of the canvas. As the insights gained about the initial solutions proposed in the validation sessions potentially change the trajectory of those projects significantly (see DR 3.3), it can be suggested that improved strategic fit can potentially already be experienced in the first session. Testing the canvas for an extended period of time, considering significant changes in the trajectory following insights developed in sessions in which the canvas was used, would allow for evaluation of the effectiveness of the canvas.

The key take-aways for the next iteration are:

- To ease facilitation by product managers and owners, adding more guidelines about when to move to the next step in the canvas would be helpful.

- As it is recommended to specify the intention to use the canvas before using it, which impacts the WHY-questions asked in the problem deep dive, a section about this can be added to the tool guide to support users.

- To allow for individual adaptation and personalization, adding elements to the canvas in a modular way will be helpful to users.

Insights will be used to build an iteration of the product (proposed changes are presented in §7.1). The final iteration of the canvas developed in this graduation project will be presented in the next chapter.

Chapter 7

Final Product

Iterating towards the final Problem Deep Dive Canvas and Tool Guide

In this chapter, proposed changes to the initial prototype will be introduced based on the observations, feedback, and evaluation of the design requirements presented in the previous chapter. Subsequently, the final iteration of the canvas and tool guide made during this graduation project will be presented.

7.1 Final Problem Deep Dive Canvas

Changes to the canvas prototype

- To clarify the step between the HowMightWe problem reframing and the ideation, a possibility to conclude the problem reframing step in a visual manner will be added.
- The 'notes' section will be rewritten towards 'conclusions' to indicate the importance of this final step, including the outcomes. A line will be added to explain this step.

The final version of the canvas can be found in fig. 37 and enlarged in appendix 10.

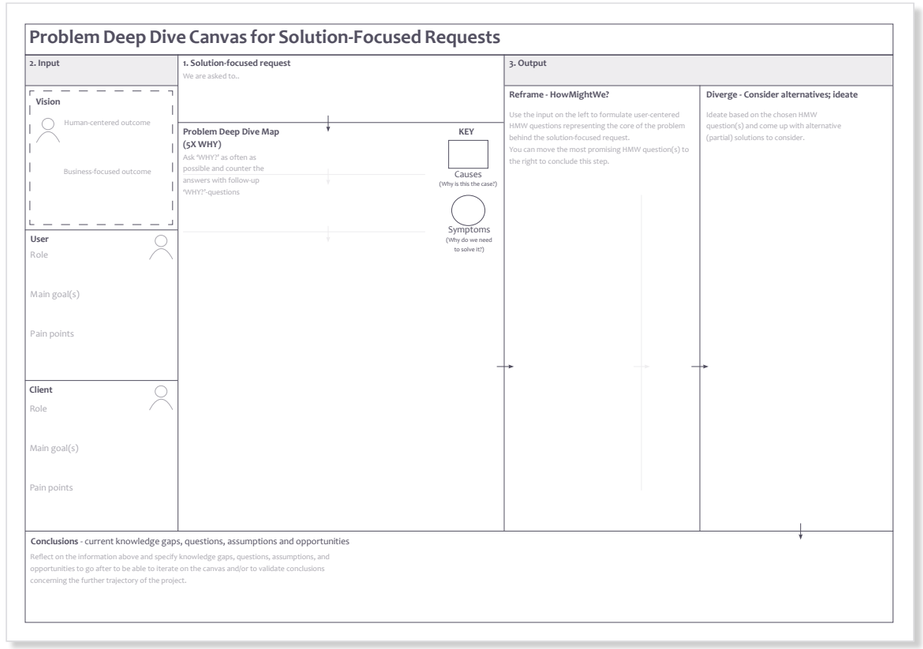


Fig. 37: Final version of the Problem Deep Dive Canvas

7.2 Final Problem Deep Dive Tool Guide

Changes to the tool guide prototype

- A section will be added to guide the user in setting an intention prior to using the canvas
- Inspiration for the formulation of WHY-questions will be provided, taking the different intentions into account
- A section will be added to support the step between the 5xWHY step and the HMW step in the form of a simple checklist
- A section will be added to support the step between the HMW and the ideation by providing a way to reflect on the HMW question(s)
- To allow for modular adaptation of the canvas, initially, a module will be added to add a quadrant on the left of the canvas to include dependencies, multiple users and/or stakeholders involved. Moreover, an empty module will be added to allow for other adaptations.
- A possibility will be provided to extend the deep dive toward the users by the extended persona. In a possible future iteration of the tool guide, more tools can be added to entend the deep dive in specific elements.
- To aid facilitation, more information about brainstorming (including time blocking and postponing judgment) will be added

The sections will be added to the appendix of the tool guide, as adding more pages to read prior to using the canvas will result in a higher threshold to use the product. The idea is that the user only has to read the short ‘what’, ‘why’, ‘when’ and ‘how’ to be able to start experimenting with the canvas. When questions arise, or a different reason to look for more information (e.g. after a first trial session), the user can consult the appendix. In this way, the threshold to get started with the tool will be as low as possible. A mock-up of the tool guide can be found in fig. 38, the final pages are extracted in fig. 39 and enlarged in appendix 11.

Fig. 38: Mock-ups of the final tool guide

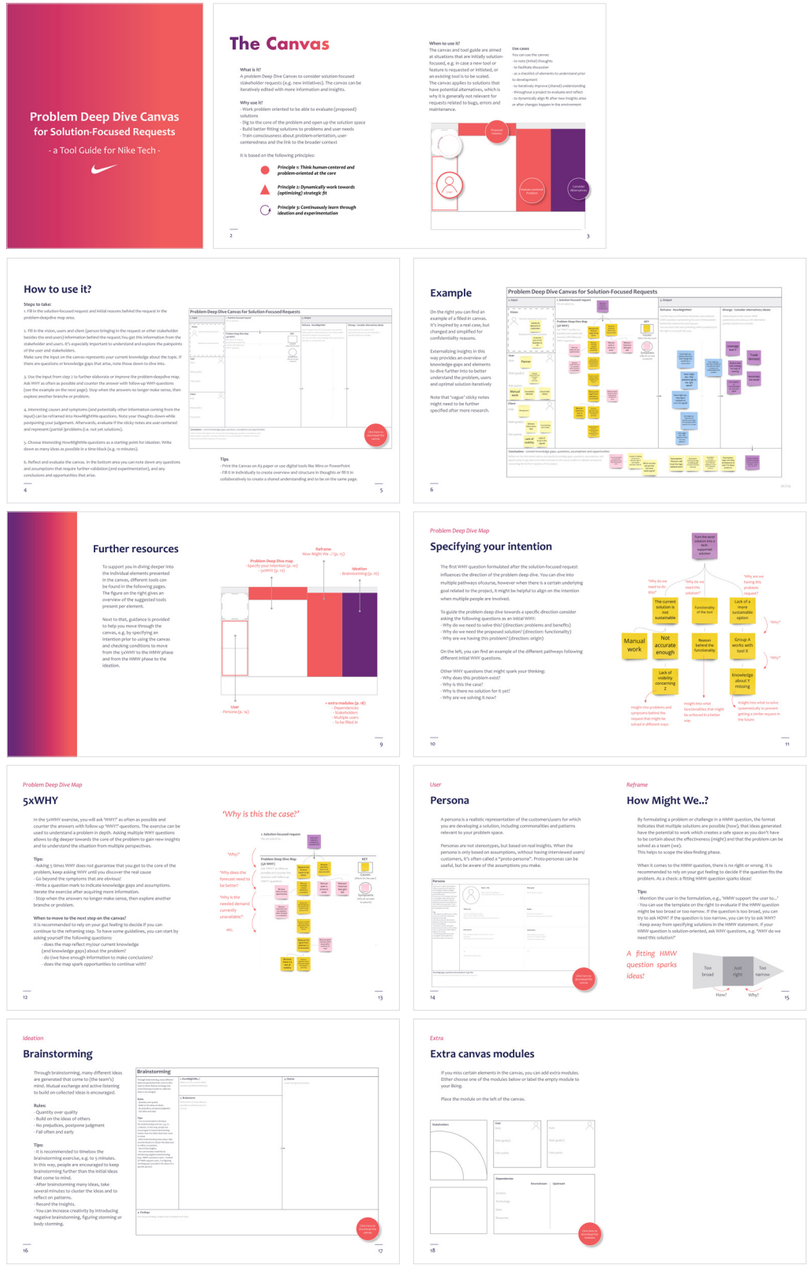


Fig. 39: Extracted version of the tool guide, see appendix 11 for a larger version

Chapter 8

Discussion & Conclusion

In this chapter, a general discussion including recommendations, theoretical implications and limitations will be provided, as well as a final conclusion. The chapter will end with a personal reflection on the process of writhing this graduation project report.

8.1 Discussion

8.1.1 Recommendations

First of all, implementation of a problem exploration phase is recommended in an agile manner. The canvas provides structure to this phase and the possibility to dig deeper into a problem in a low-threshold way without significantly impacting current processes and structures. In this way, the product can be viewed as a prototype that can already deliver insights and learnings. Following the stages of the curve of commitment (Conner & Patterson, 1982), a graph that helps to understand the stages towards the option of a new mindset (see fig. 40), it is concluded that experimentation precedes adoption and institutionalization (including the adaption of organizational structures to accommodate new ways of operating, new mindsets and behaviors) of a change in mindset and behavior. Approaching the product in this way as management allows for learning about the problems and opportunities in an agile way and can provide input for further changes within organizational structures supporting the development of more effective solutions.

Top-down support of management is crucial to the adoption of change and can actively increase the problem-oriented mindset and approach to the problems by introducing the canvas when considering new initiatives. An initial version of the canvas can be set up in less than 30 minutes prior to acquiring more information to fill in knowledge gaps and iterating on the canvas, and it is recommended to initiate and support this to kick-start problem-oriented thinking.

To canvas does not conflict with existing structures and processes and can be used by product managers and owners to work towards a better understanding of underlying problems behind a certain request and can also be added to existing proposal or intake meetings with business stakeholders. As the canvas can be used independently, without the need for external facilitation, approval, or resources, the threshold to start change is low.

The product focuses on key opportunities present at this moment in time, taking into account the little current experience with

problem exploration of the team and their current interest in the topic and potential value. The principles mentioned are regarded as requirements that need to be established prior to further extending problem exploration activities in an effective way, i.e., one needs to think problem-oriented to be able to align and diverge, and consequently to test promising assumptions. Therefore, it is recommended to focus on the initial principles first (more specifically, prioritizing the first principle) and to work towards unlocking and supporting further problem exploration capabilities in an agile manner.
When the product is used more often and users advance in the initial principles set out, different needs might arise, e.g. towards rapid prototyping and advanced problem deep dive. It is recommended to consider elements of Design Thinking, as the approach might be able to (partially) support further needs as well.

After further testing and experimenting with the canvas and tool guide, optimization is possible. By giving users a simple way to adapt the canvas to their needs, further experimentation is enabled and encouraged. Evaluating the use of the canvas and fulfillment of needs in different scenarios is recommended to enable process optimization within teams. Further exploration about the role of stakeholders, i.e. when and how to involve them in the process of using the canvas, might be benefitical.

In order to raise awareness for the product, it needs to be introduced to product owners and product managers. In the Operation Tech unit, this has been done already. By using the canvas as a communication and documentation resource in conversations, a snowball effect might arise. Next to that, the tool can be introduced in technology-wide meetings to encourage further experimentation. As value becomes apparent by using it, use cases and insights that might significantly impact the further trajectory of the projects involved (e.g. in the validation session cases presented in this report) can be used to increase credibility.

To allow for easy access, it is recommended to place the canvas, tool guide and separate templates in the cloud-based content

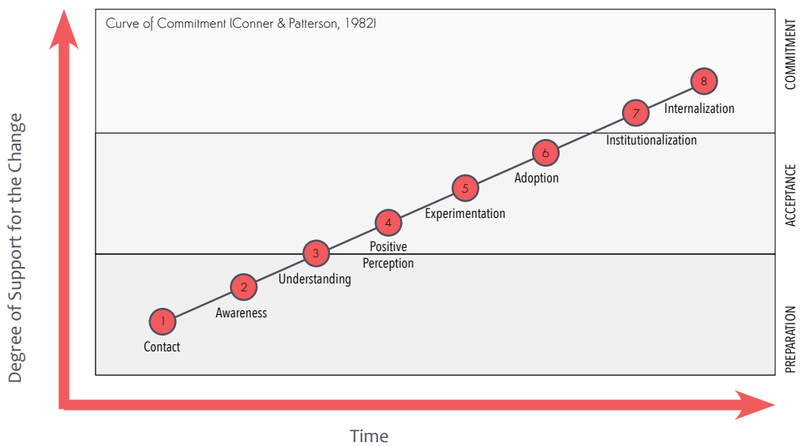


Fig. 40: Curve of Commitment (Conner & Patterson, 1982)

management tool used within the organization. In this case, the links have to be integrated in the tool guide referring to the right canvases.

8.1.2 Limitations

First of all, this graduation project has focused on one unit within the operation technology area, which limits generalizability. As the agile software development teams considered are different in terms of their ways of working (e.g. in terms of the number of tools and main stakeholders, levels of software maturity, and agile practices applied) a certain degree of generalizability can be assumed. However, further testing is required to validate this.
Moreover, as change is the only constant, it is important to be aware of limitations regarding the timing of this research and the impact on the results, especially regarding the current mindset, organizational structures and clarity of vision. Interviews reflect a snapshot of a time in which changes in the environment (e.g. due to the pandemic) and organization most likely influenced the results. In line with this, future changes (e.g. in organizational structures) also impact the fit of the outcome of this research.

In line with this, the proposed solution fits the willingness and threshold (considering time, processes and structures) towards problem exploration of the interviewees coming from one organizational unit. Testing fit regarding those aspects when scaling the product beyond this team is recommended to ensure user-fit.

Furthermore, the validation phase in this research has been limited due to the scope and timing of the project. One version of the canvas has been extensively tested through six validation sessions with a focus on filling in an initial version of the canvas. However, this allowed for testing assumptions related to the understanding of the use and the threshold to get started (key requirements for adoption), it does not provide validation regarding the actual effectiveness of using it through multiple iterations over the course of the (initial phase of a) project.

The product focuses on three principles and provides a structured way to approach initiatives in a problem-oriented way. However, it does not remove structural boundaries in place in the organization (e.g. organizational structures and process, full backlogs and time pressure, efficiency-focused KPIs, and late involvement of technology teams in the process skipping problem exploration). Next to encouraging low-threshold experimentation and adoption of problem exploration through the product presented, it is recommended to evaluate the structural boundaries to support and encourage this change.

8.1.3 Implications

Research on Design Thinking often focuses on entire organizations and practitioners are introduced to a generic form of Design Thinking. Recognizing that Design Thinking is contextual, there are a great deal of opportunities for future research to dive into how and where capabilities associated with Design Thinking exist or are needed within an organization and the use cases they are needed for.

In this way, answers to how Design Thinking can be leveraged become more nuanced and design approaches can further help engineers, managers and other practitioners to become more effective in their work. E.g. in which cases are specific approaches most helpful and how can design approaches support other approaches to frame and solve problems?
Further research would help companies unpack their needs in more specific areas that could also benefit from different approaches to problem-solving; viewing Design Thinking as a means to an end that could support their goals when appropriate to a given situation.

Generally, using Design Thinking is associated with coming up with new concepts to offer the market and, as innovation research advocates for, seen as a process to come up with radical ideas. However, it is also recognized as a more general way of thinking and solving indeterminate problems (e.g. Dunne & Martin, 2006, see §2.1.2); more research is required towards the value and application of Design Thinking in this way.

This graduation project suggests the significant value of applying elements of Design Thinking in the dynamic pursuit of strategic fit in the context of internal agile software development and in tackling the bottleneck of having a solution-oriented mindset that withholds people from exploring the actual problem to develop better fit-for-purpose solutions.
This also signals implications in terms of research towards who is using Design Thinking. Researchers advocate for different perspectives, e.g. as a management approach (Liedtka & Ogilvie, 2011; Martin, 2009), for multidisciplinary teams (e.g. Beckman & Barry, 2007), or for everyone (Brown, 2008). Recognizing the meta-disciplinary application of Design Thinking will open up research opportunities to further specify specific user groups (e.g. product managers, product owners, and software developers), to investigate how Design Thinking can support them in their goals and how to introduce the subject to them considering their (e.g. analytical) mindset, skills, and experience, in order to meet their needs and to provide them with more tools in their toolbox to solve problems.

While recognizing the value of Design Thinking in the context of coming up with radical innovations, which are important to the survival of organizations, a great deal of the time (or even all of the time, which is a different discussion) in teams is spent on managing and exploiting day-to-day activities and stakeholder requests. In this research, needs concerning the alignment of the problem, solution, user, and vision, and considering better alternatives do suggest relevance and value of elements of Design Thinking within this use case.

Finally, this research suggests the impact of the solution-oriented mindset as a bottleneck towards other elements of Design Thinking that can be useful in the context of software development. Further research is recommended to provide insights into and recommendations for the introduction of Design Thinking within teams and organizations.

8.2 Final conclusion

The goal of this graduation project was to identify opportunity areas to leverage the Design Thinking methodology in the process of agile software development and to find out how processes and relevant tools can be tailored to fit the needs of the target group (Nike's Operations Tech unit).

Research question: How might we use Design Thinking to our advantage (according to theory & practice) in this specific context?

Both literature about agile software development and findings show that there is a lack of focus on the problem exploration phase preceding the solution phase. Therefore, an opportunity arises to better understand users and the problem the solution is trying to solve to evaluate if the proposed solution is indeed the right thing to build, and to potentially pivot to alternatives to optimize solution fit.
This graduation project expands on literature regarding the contextual use of Design Thinking by codifying Design Thinking elements in the context of internal agile software development in a large organization.

The research aim has been approached through a literature review, case studies of similar companies, a context analysis and explorative interviews regarding current boundaries to problem exploration in this context.

Main areas of concern include a lack of the right mindset, specifically considering the solution-oriented mindset instead of a problem-oriented mindset when considering initiatives; organizational structures limiting space for problem exploration in terms of time, processes and the role of technology in the problem exploration phase; and the need and impact of having a clear vision and alignment around it.

Integrating literature and exploratory research findings, the answer to the research question was translated into a conceptual model covering three key principles: problem-oriented and human-centered thinking, dynamic alignment towards strategic fit, and divergent thinking to consider more fit-for-purpose alternatives.

Finally, to put the principles into practice, a 'Problem Deep Dive Canvas' was developed, a tangible artifact for product managers and product owners to use in collaboration with their software development teams and stakeholders. Initial validation results are promising considering feasibility, desirability and viability of the product.

8.3 Personal reflection

In the week of my green-light meeting, I found a graph visualizing the Dunning-Kruger effect (see fig. 41), in which I immediately recognized my thought process during this graduation project. While observing the agile software development context during my internship, opportunities for Design Thinking looked very clear, which resulted in a great deal of confidence and optimism. With little actual knowledge about the context, the Dunning-Kruger graph notes this initial peak of optimism 'Mount Stupid'. By diving further into the topic, complexity arises and you find out that it is actually way more complex than you initially thought. Then, by unraveling complexity, making sense of all the elements and bringing complexity back to core elements, you can start to climb out of the valley of despair.
It shows me two main things about being a designer. First of all, that initial optimism gets you somewhere, as it helps to dare to dive into topics and face complex problems. Secondly, it shows me to trust the process, which is not always easy when you're in the middle of complexity.

In line with this, if I had to describe my process in one word, it would have to be simplifying. I like the challenge to find patterns and logic and to integrate many different elements, in this case into a conceptual model. However, unnecessary complexity should be avoided as it complicates communication and application of the model, which I experienced during this process. By challenging myself to continue clarifying my thoughts, understanding the elements better, the outcome of this thesis became a lot stronger. At this moment, I can't imagine introducing people to an initial version of the model I set up and the process of having to translate that model into an effective and usable artifact. Through trial and error - 'failing fast' - it became clear to me that simplification is crucial for the adoption of a process or product

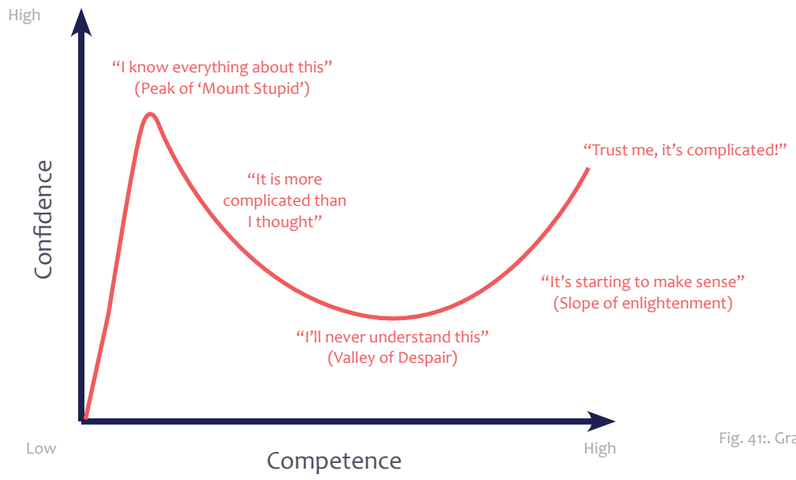


Fig. 41:. Graph visualizing the The Dunning-Kruger Effect and thought process

"That's been one of my mantras - focus and simplicity. Simple can be harder than complex. You have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains."

- Steve Jobs

as it significantly lowers the threshold to understand and use it. In my initial project brief, I mentioned that "actually getting people along and increasing the chances of implementation within a company is a completely new challenge I want to learn as much as possible about". The impact of simplification is a huge insight towards this challenge.

Looking back at the goals formulated in my initial project brief, I can conclude that I have definitely been practicing my research skills, integrative thinking and reframing. Specifically, in the context of a large multinational company, this has been a new experience for me.
Working iteratively towards problem-solution fit has been an almost natural part of the process because of the complexity of the context. The intention to not be afraid to fail helped me to put information out there, e.g. a presentation and discussion about 'not solving the problem you're asked to solve', different ways of reflecting within a team and making my thoughts concrete on paper to test understanding. I learned a lot more about the problem at hand from making these ideas concrete quickly and putting them out there to discuss with others.

In the project brief, I also mentioned the awareness that Design Thinking is a buzzword, which I considered. Literature about the contextual nature of the concept helped me in forming my opinion about it. I agree that Design Thinking is overused as a general concept that can be applied by anyone to everything. In my opinion, implementing Design Thinking should not be seen as an objective itself but as a means to an end. The 'end' is to support the users of Design Thinking in their needs, which have to be explored in the specific context of focus. In this way, (elements of) Design Thinking can be leveraged in an effective and focused way.

Finally, I am proud of the personal development achieved throughout this graduation project, the process I went through, the mindset I had during the project, and the final outcome. All six validation sessions were interesting and insightful towards the request at hand and actually exceeded my expectations of the impact the canvas could have on the projects discussed in the sessions. This makes me excited about the potential and value of the product.

This journey has been an interesting one; I have learned a great deal and I can't wait to implement my learning in my future career and life.

References

A

Abrahamsson, P., Warsta, J., Siponen, M., & Ronkainen, J. (2003). New directions on Agile methods: A comparative analysis. *25th International Conference on Software Engineering, 2003.* Proceedings. https://doi.org/10.1109/icse.2003.1201204

Adikari, S., McDonald, C., & Campbell, J. (2013, July). Reframed contexts: design thinking for agile user experience design. In *International Conference of Design, User Experience, and Usability* (pp. 3-12). Springer, Berlin, Heidelberg.

Anderson, D. J. (2010). *Kanban: Successful evolutionary change for your technology business.* Blue Hole Press.

Ansoff, H. I. (1957). Strategies for diversification. *Harvard business review*, 35(5), 113-124.

Azis, M. (2016, February 9). *The making of IBM design thinking.* LinkedIn. https://www.linkedin.com/pulse/making-ibm-design-thinking-miroslav-azis

B

Baccarini, D. (1999). The logical framework method for defining project success. *Project Management Journal*, 30(4), 25-32. https://doi.org/10.1177/875697289903000405

Baseer, K. K., Reddy A, R. M., & Bindu C, S. (2015). A systematic survey on waterfall vs. Agile vs. Lean process paradigms. *i-manager's Journal on Software Engineering*, 9(3), 34-59. https://doi.org/10.26634/jse.9.3.3471

Bass, J. M. (2014). How product owner teams scale Agile methods to large distributed enterprises. *Empirical Software Engineering*, 20(6), 1525-1557. https://doi.org/10.1007/s10664-014-9322-z

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development.* agilemanifesto.org. https://agilemanifesto.org

Beck, K. (2000). *Extreme programming explained: embrace change.* addison-wesley professional.

Benson, J., & Dresdow, S. (2015). Design for thinking: Engagement in an innovation project. *Decision Sciences Journal of Innovative Education*, 13(3), 377-410. https://doi.org/10.1111/dsji.12069

Bicen, P., & Johnson, W. H. (2015). Radical innovation with limited resources in high-turbulent markets: The role of lean innovation capability. *Creativity and Innovation Management*, 24(2), 278-299. https://doi.org/10.1111/caim.12120

Birks, M. Mills, J. (2015). *Essentials of grounded theory. Grounded theory: a practical guide*. (2nd ed., pp. 10-14). London: SAGE.

Bloomberg. (2021). *Bloomberg - Are you a robot?*. https://www.bloomberg.com/profile/company/WORK:US

Boehm, B., & Turner, R., (2003). Using risk to balance agile and plan-driven methods, *Computer (Long. Beach. Calif).*, vol. 36, no. 6, pp. 57–66, 2003.

Boland, R. J., & Collopy, F. (2004). *Design matters for management*. Stanford, CA: Stanford University Press.

Brenner, W., Uebernickel, F., & Abrell, T. (2016). Design thinking as mindset, process, and toolbox. *In Design thinking for innovation* (pp. 3-21). Springer, Cham.

Briner, W., Hastings, C., & Geddes, M. (1996). *Project leadership*. Aldershot, UK: Gower.

Brown, T. (2008). Design thinking. *Harvard business review*, 86(6), 84.

Brown, T. (2009) *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation.* HarperCollins, New York.

Buchanan, R. (1992). Wicked problems in design thinking. *Design Issues*, 8(2), 5–21. https://doi.org/10.2307/1511637

Burton, R. M., Lauridsen, J., & Obel, B. (2004). The impact of organizational climate and strategic fit on firm performance. *Human Resource Management*: Published in Cooperation with the School of Business Administration, The University of Michigan and in alliance with the Society of Human Resources Management, 43(1), 67-82.

Burton, R. M., Lauridsen, J., & Obel, B. (2002). Return on assets loss from situational and contingency misfits. *Management science*, 48(11), 1461-1485.

Buxton, B. (2007). *Sketching user experiences.* San Francisco: Morgan Kaufmann.

C

Carlgren, L., Rauth, I., & Elmquist, M. (2016). Framing Design Thinking: The Concept in Idea and Enactment: Creativity and Innovation Management. *Creativity and Innovation Management* 25, 1 (March 2016), 38–57. https://doi.org/10.1111/caim.12153

Chandler A. (1962) *Strategy and Structure.* MIT Press.

Chang, Y ., Kim, J., & Joo, J. (2013). An exploratory study on the evolution of design thinking: Comparison of Apple and Samsung. *Design Management Journal*, 8(1), 22–34. https//doi.org/doi 10.1111/dmj.12001

Chesson, D. (2017). Design thinker profile: Creating and validating a scale for measuring design thinking capabilities.

Chicoria, A. (2018). *IBM enterprise design thinking?* SlideShare. https://www.slideshare.net/AdilsonJardim/ibm-enterprise-design-thinking

Cho, J, (2010). An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum, *All Grad. Theses Diss.*, p. 599, 2010.

Conner, D. R., & Patterson, R. W. (1982). Building commitment to organizational change. Training & Development Journal.

Corral, L., Sillitti, A., & Succi, G. (2015). Software assurance practices for mobile applications. *Computing* 97, 10 (2015), 1001–1022.

Correa, H. J. (2008). *Introduction to Scrum.* CODE Online. https://www.codemag.com/Article/0805051/Introduction-to-Scrum

Cross, N. (1982). Designerly ways of knowing. *Design Studies*, 3(4), 221–227. https://doi.org/10.1016/0142-694X(82)90040-0

D

Davis, K. H. (1995). Logical framework analysis: A methodology to turn vision into reality. *Proceedings of the AIPM National Conference*, Adelaide, Australia, 393-397.

Davis, S.M., & Lawrence, P.R. (1977). *Matrix.* AddisonWesley: Reading, MA

Design Council. (2018, September 10). *What is the framework for innovation?* Design council's evolved double diamond. Retrieved July 1, 2021, from https://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond

Dignan, L. (2019, October 27). *Nike bets on tech CEO Donahoe to accelerate digital transformation: Will it work?* ZDNet. https://www.zdnet.com/article/nike-bets-on-tech-ceo-donahoe-to-accelerate-digital-transformation-will-it-work/

Dingsøyr, T., & Moe, N.B., (2014). Towards Principles of Large-Scale Agile Development - A Summary of the Workshop at XP2014 and a Revised Research Agenda, in Agile Methods. *Large-Scale Development, Refactoring, Testing, and Estimation*, vol. Volume 199, 2014, pp. 1–8.

DIS. (2010). Ergonomics of Human-system Interaction: Part 210: Human-centred Design for Interactive Systems. *ISO.*

Dix, A., Finlay, J., Abowd, G.D., & Beale, R. (2003). *Human-Computer Interaction.* Harlow, Prentice Hall.

Dunne, D. (2018). Design Thinking at Work: How Innovative Organizations are Embracing Design. (n.p.): *University of Toronto Press*.

Dunne, D., & Martin, R. (2006). Design thinking and how it will change management education: An interview and discussion. *Academy of Management Learning & Education*, 5(4), 512–523. https://doi.org/10.5465/AMLE.2006.23473212

F

Fielding, N & Thomas, H. (2001). Qualitative interviewing. *Researching social life.* 123-144.

Fjeldstad, Ø.D., Snow, C.C., Miles, R.E. and Lettl, C. (2012). The architecture of collaboration., *Strategic Management Journal*, Vol. 33 No. 6, pp. 734-750.

Fraser, H. M. (2007). The practice of breakthrough strategies by design. *Journal of Business Strategy*, 28(4), 66–74. https://doi.org/10.1108/02756660710760962

G

Ginsberg, A., & Venkatraman, N. (1985). Contingency perspectives of organizational strategy: A critical review of the empirical research. *Academy of management review*, 10(3), 421-434.

Girma, M., Garcia, N. M., & Kifle, M. (2019). Agile scrum scaling practices for large scale software development. In 2019 4th *International Conference on Information Systems Engineering* (ICISE) (pp. 34-38). IEEE.

Google. (2008). *Design sprint kit.* Share and engage with the Design Sprint Community. Retrieved July 1, 2021, from https://designsprintkit.withgoogle.com/methods/

Gothe, M. (2016). *Adopting IBM design thinking for solution development.* DESIGN.GOTHE.SE – Mats Göthe Design Portfolio. Retrieved July 1, 2021, from https://design.gothe.se/1605/

Gurusamy, K., Srinivasaraghavan, N., & Adikari, S. (2016, July). An integrated framework for design thinking and agile methods for digital transformation. *In International Conference of Design, User Experience, and Usability* (pp. 34-42). Springer, Cham.

H

Hamel, G., & Breen, B. (2007). *The Future of Management* Harvard Business School Press: Boston.

Hassi, L. and Laakso, M. (2011). Design Thinking in the Management discourse; defining the elements of the concept, 18th International Product Development Management Conference, IPDMC.

Hehn, J., Mendez, D., Uebernickel, F., Brenner, W., & Broy, M. (2020). On integrating design thinking for a human-centered requirements engineering. *IEEE Software* pp. 25–31

Highsmith, J. A., & Highsmith, J. (2002). *Agile software development ecosystems.* Addison-Wesley Professional.

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer,* 34(9), 120-127.

Hildenbrand, T., Geisser, M., Kude, T., Bruch, D., & Acker, T. (2008). Agile methodologies for distributed collaborative development of enterprise applications. International conference on complex, intelligent and software intensive systems. *CISIS 2008* (pp. 540–545), Barcelona.

Highsmith, J. (2009). *Agile project management: creating innovative products.* Pearson education.

Hildenbrand, T., & Meyer, J. (2012). Intertwining lean and design thinking: software product development from empathy to shipment. *In Software for people* (pp. 217-237). Springer, Berlin, Heidelberg.

Hobday, M., Boddington, A., & Grantham, A. (2012). An innovation perspective on design: Part 2. *Design Issues*, 28(1), 18–29. https://doi.org/10.1162/DESI_a_00137

I

IDEO. (2021). About IDEO: Our story, who we are, how we work. IDEO is a global design and innovation company | ideo.com. Retrieved September 24, 2021, from https://www.ideo.com/about/

Itami, H. (1987) Mobilizing Invisible Assets. *Harvard University Press.*

J

Jiang, X. (2009). Strategic management for main functional areas in an organization. *International Journal of Business and management*, 4(2), 153-157.

Johansson-Sköldberg, U., Woodilla, J., & Çetinkaya, M. (2013). Design thinking: Past, present and possible futures. *Creativity & Innovation Management*, 22(2), 121–146. https://doi.org/10.1111/caim.12023

Junginger, S. (2007). Learning to design: Giving purpose to heart, hand, and mind. *Journal of Business Strategy,* 28(4), 59–65. https://doi.org/10.1108/02756660710760953

K

Kalenda, M., Hyna, P., &Rossi, B. (2018). Scaling agile in large organizations: Practices, challenges, and success factors. *J. Softw. Evol. Process*, no. September 2017, 2018.

Kimbell, L. (2011). Rethinking design thinking: Part I. *Design & Culture*, 3(3), 285–306. https://doi.org/10.2752/175470811X13071166525216

Kimbell, L. (2012). Rethinking design thinking: Part II. *Design and Culture*, 4(2), 129-148.

Klamann, K., Shah, V. (2020). Sprinting Toward a Failed Product: How to avoid the pitfalls of product delivery through product discovery. *Frog Designplat.* https://info2.frogdesign.com/hubfs/Sprinting Toward a Failed Product_frog.pdf ^^ H2 quote deze??

Kniberg, H. (2013, January 18). *How Spotify builds products.* Crisp's Blog - from the Crisp Consultants. https://blog.crisp.se/wp-content/uploads/2013/01/HowSpotifyBuildsProducts.pdf
Kniberg, H., & Skarin, M. (2010). *Kanban and Scrum-making the most of both.* Lulu.com.

Koh, J., Chai C., Wong B., & Hong H. (2015). Design thinking for education: Conceptions and applications in teaching and learning. *Singapore: Springer.*

Kumar, V. (2012). *101 design methods: A structured approach for driving innovation in your organization.* John Wiley & Sons.

Kupiainen, E., Mäntylä, M. V., & Itkonen, J. (2015). Using metrics in Agile and Lean Software Development–A systematic literature review of industrial studies. *Information and software technology*, 62, 143-163.

L

Lafley, A. G., & Charan, R. (2010). *The game changer: How every leader can drive everyday innovation.* Profile Books.

Lawson, B. R. (1979). Cognitive strategies in architectural design. *Ergonomics,* 22(1), 59–68. https://doi.org/10.1080/00140137908924589

Lawson, B. (2006) *How Designers Think: The Design Process Demystified*, 4th edn. Elsevier, New York.

Lee, M. Y., & Edmondson, A. C. (2017). Self-managing organizations: Exploring the limits of less-hierarchical organizing. *Research in organizational behavior*, 37, 35-58.

Leffingwell, D. (2007) *Scaling software agility : best practices for large enterprises.* Pearson Education, 2007.

Lester, R. K., & Piore, M. J. (2004). *Innovation—The missing dimension.* Harvard University Press.

Liedtka, J. (2013, November 6). *Design thinking.* Jeanne Liedtka. Retrieved July 1, 2021, from https://jeanneliedtka.wordpress.com/design-thinking/

Liedtka, J., & Ogilvie, T. (2011). *Designing for growth: A design thinking tool kit for managers.* West Sussex, NY: Columbia University Press.

Lindberg, T., Meinel, C., Wagner, R.: Design Thinking: A Fruitful Concept for IT Development?, pp. 3–18. *Springer,* Berlin, Heidelberg

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., & Kahkonen, T. (2004). Agile software development in large organizations. *Computer*, 37(12), 26-34.

Lockwood, T. (2010). *Design thinking: Integrating innovation, customer experience, and brand value.* New York, NY: Skyhorse.

Lucena, P., Braz, A., Chicoria, A., & Tizzei, L. (2016, November). IBM design thinking software development framework. *In Brazilian Workshop on Agile Methods* (pp. 98-109). Springer, Cham

M

Mandel, T. (2009). *Elements of User Interface Design.* Hoboken.

March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization science*, 2(1), 71-87.

March, J. G., & Herbert, A. (1958). Simon1958 Organizations. New York: WileyMarchOrganizations1958.

Martin, R. L. (2009). *The design of business: Why design thinking is the next competitive advantage.* Boston, MA: Harvard Business Press.

Hanington, B., & Martin, B. (2019). *Universal methods of design expanded and revised: 125 Ways to research complex problems, develop innovative ideas, and design effective solutions.* Rockport publishers.

Mee, J. F. (1964). Matrix organization. *Business Horizons,* 7(2), 70-72.

Meinel, C., & Leifer, L. (2011). Understanding Innovation (pp. 17-18). *Springer.*

Miller, D., & Friesen, P. H. (1983). Strategy-making and environment: the third link. *Strategic management journal,* 4(3), 221-235.

Mishra, D., & Mishra, A. (2011). Complex software project development: agile methods adoption. *Journal of Software Maintenance and Evolution: Research and Practice*, 23(8), 549-564.
Moran, A. (2007). A study of the influence of goal alignment on multi-organizational projects: A system dynamics approach. Oklahoma State University.

Mullins, J.W. & Walker, O.C. (2009). *Marketing Management: A Strategic Decision-Making Approach* (6th ed.) Europe: Mcgraw-Hill Education.

N

Nigam, C., & Gupta, S. (2017). Agile Methodology for Software Development. *Inst Innov Technol Manag - J Inf Technol 2017;3:54–64.*

Nike. (n.d.). *Global technology.* Nike Careers Home. Retrieved July 1, 2021, from https://jobs.nike.com/technology

NIKE, Inc. (2019). *NIKE, Inc. - Investor Relations.* https://investors.nike.com/default.aspx?SectionId=1699651a-428e-4d77-9df4-973d389ab618&LanguageId=1

Norrie, J., & Walker, D. H. (2004). A balanced scorecard approach to project management leadership. *Project management journal*, 35(4), 47-56.

O

O'Reilly III, C. A., & Tushman, M. L. (2008). Ambidexterity as a dynamic capability: Resolving the innovator's dilemma. *Research in organizational behavior*, 28, 185-206.

O'Reilly III, C. A., & Tushman, M. L. (2011). Organizational ambidexterity in action: How managers explore and exploit. *California management review*, 53(4), 5-22.

Owen, C. L. (2005). *Design thinking. What it is. Why it is different. Where it has new value.* Paper presented at the International Conference on Design Education and Research, Gwangju City, Korea.

P

Pereira, J. C., & de FSM Russo, R. (2018). Design thinking integrated in agile software development: A systematic literature review. *Procedia computer science*, 138, 775-782.

Pfeffer, J. & Salancik, G.R. (1978). *The External Control of Organizations: A Resource Dependence Perspective*. Harper & Row.

Pichler, R. (2010). *Agile product management with scrum.* Addison-Wesley Pearson Education, Boston, MA, USA.

Plattner, H. (2009). An introduction to design thinking: Process guide. *Palo Alto*, CA: Stanford Institute of Design. doi:10.1007/978-1-4302-6182-7_1.

Poppendieck, M., & Poppendieck, T. (2003). *Lean software development: an agile toolkit.* Addison-Wesley.

R

Radović-Marković, M., & Omolaja, M. A. (2011). Management dynamics in the new economy. *International journal of engineering education*, 498-508.

Ries, E. (2011). *The Lean Start-up.* Crown Publishing, New York

Rowley, J. (2012). Conducting research interviews. *Management research review*, 35(3/4), 260-271.

S

Sanders, E. B. N., & Stappers, P. J. (2012). *Convivial toolbox: Generative research for the front end of design.* Bis.

Scaled Agile, Inc. (2021, May 12). *Program increment.* Scaled Agile Framework. Retrieved September 1, 2021, from https://www.scaledagileframework.com/program-increment/

Schach, S. & Schach, S. (2004). *Intro to Object-Oriented Analysis and Design with UML CD*. Boston: McGraw-Hill.

Schwaber, K., & Beedle, M. (2002). Agile software development with Scrum (Vol. 1). Upper Saddle River: Prentice Hall.

Schweitzer, J., Groeger, L. & Sobel, L. (2016). The design thinking mindset: An assessment of what we know and what we see in practice. *Journal of Design, Business & Society*, 2(1), 71-94. https://doi.org/10.1386/dbs.2.1.71_1

Simon, H. A. (1969). *The sciences of the artificial.* Cambridge, MA: MIT Press.

Solinski, A., & Petersen, K. (2016). Prioritizing agile benefits and limitations in relation to practice usage. *Software quality journal,* 24(2), 447-482.

Sommerville, I (2011). *Software Engineering*, 9th ed. Boston: Addison-Wesley, 2011.

Statista. (2021a). *Topic: Nike*. Retrieved September 1, 2021, from https://www.statista.com/topics/1243/nike/

Statista. (2021b). *Nike's revenue worldwide from 2005 to 2021.* Retrieved September 1, 2021, from https://www.statista.com/statistics/241683/nikes-sales-worldwide-since-2004/

Statista. (2021c). *Number of employees of Nike worldwide from 2009 to 2021.* Retrieved September 1, 2021, from https://www.statista.com/statistics/243199/number-of-employees-of-nike-worldwide/

Statista. (2021d). *Total Nike retail stores worldwide from 2009 to 2021.* Retrieved September 1, 2021, from https://www.statista.com/statistics/250287/total-number-of-nike-retail-stores-worldwide/

Statista. (2021, April 12). *Spotify - statistics & facts.* Retrieved September 1, 2021, from https://www.statista.com/topics/2075/spotify/

Stickdorn, M., Hormess, M. E., Lawrence, A., & Schneider, J. (2018). *This is service design doing: applying service*

design thinking in the real world. " O'Reilly Media, Inc.".

Sutherland, J., & Schwaber, K. (2007). The scrum papers. Nuts, Bolts and Origins of an Agile Process.

T

Thompson, K. (2013). *Recipes for agile governance in the enterprise*. RAGE.

Tushman, M. L., & O'Reilly III, C. A. (1996). Ambidextrous organizations: Managing evolutionary and revolutionary change. *California management review*, 38(4), 8-29.

V

Vallon, R., da Silva Estacio, B. J., Prikladnicki, R., & Grechenig, T. (2018). Systematic literature review on agile practices in global software development. *Information and Software Technology*, 96, 161-180.

Venkatraman, N., & Prescott, J.E. (1990). Environment-strategy coalignment: an empirical test of its performance implications. *Strategic Management Journal*, 11, pp. 1- 23.

W

Walters, H. (2011, March 24). *"Design thinking" isn't a miracle cure, but here's how it helps.* Fast Company. https://www.fastcompany.com/90186356/design-thinking-isnt-a-miracle-cure-but-heres-how-it-helps

Williams, L., & Cockburn, A. (2003). Guest Editors' Introduction: Agile Software Development: It's about Feedback and Change. Computer, 36(06), 39-43.

Witte, R. (2019, May 9). *With new fit technology, Nike calls itself a tech company.* TechCrunch. Retrieved July 1, 2021, from https://techcrunch.com/2019/05/09/with-new-fit-technology-nike-calls-itself-a-tech-company/?guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_s