

Appendix - Part 1

Leveraging Design Thinking to Support Internal Agile Software Development

- An Opportunity for Nike Technology



Appendix - Part 1

Index

Appendix 1: Initial Project Brief	P. 4
Appendix 2: More information about Design Thinking phases	P. 10
Appendix 3: Principles behind the Agile Manifesto	P. 13
Appendix 4: Case studies IBM, Spotify & Slack	P. 14
Appendix 5: Interview guide	P. 28
Appendix 6: Agile development process example	P. 28
Appendix 7: Main iterations of the conceptual model	P. 29
Appendix 8: Complex version of the solution alignment model	P. 32
Appendix 9: Survey results	P. 38
Appendix 10: Final Problem Deep Dive Canvas	P. 40
Appendix 11: Final Tool Guide	P. 42

4

Page 1 of 75

An opportunity for innovation:
Exploring the role of Design Thinking in Agile Software Development project title

Please state the title of your graduation project (above) and the start date and end date (below). Keep the title compact and simple. Do not use abbreviations. The remainder of this document allows you to define and clarify your graduation project.

start date 31 - 04 - 2021 22 - 09 - 2021 end date

INTRODUCTION **
Please describe, the context of your project, and address the main stakeholders (interests) within this context in a concise yet complete manner. Who are involved, what do they value and how do they currently operate within the given context? What are the main opportunities and limitations you are currently aware of (cultural- and social norms, resources (time, money,...), technology, ...).

Within Nike Technology, the DSM Tech department develops tools for internal employees (demand and supply planners and buyers) to enable demand management pre-season, and optimization in-season. The team consists of product managers, product owners, and squads of software and data engineers.

Currently the Agile way of working is deployed, to innovate and develop the internal software. In an agile way of working, a team works with fast iteration cycles to build the product right. Agile methodologies are found to deliver successful products faster than the traditional waterfall method (source). They recognize that the project cannot be fully controlled up-front and allow to react flexibly to changes (Dingsoyr et al., 2012).

With this agile way of working, the goal of software engineering 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).

Agile development works well in cases in which the project goals are known, however challenges more often exceed technical expertise and ask for a more holistic understanding of problems/ opportunities and user needs – requiring Nike DSM Tech to re-assess the current way of working. A relevant domain in which this applies is the domain of digital transformation: there are endless opportunities with new technologies and data possibilities, however what the valuable solution should look like is unknown (Andersson et al., 2018). In such highly uncertain contexts, the fundamental question is what the right thing is to build, e.g. what features will satisfy the (latent) needs of the customer and user, before starting the agile development cycles in pursuit of building it right.

Research and practice show a possibility to align the goal of development projects by the Design Thinking Methodology (e.g. Adikari, 2013). Design Thinking is a human-centered problem-solving approach, focused on developing innovation that balances the following three aspects: the needs and wants of the user and customer, technical feasibility, and economic viability (Brhel et al., 2008). The methodology includes numerous individual methods and tools, coming from the field of design and now widely applied to managerial problems (Brown, 2008). The idea behind applying design thinking in engineering contexts is the extension of problem-solving abilities of the development teams in order to better meet (latent) user needs and improve the quality and experience of products.

space available for images / figures on next page

MOTIVATION AND PERSONAL AMBITIONS
Explain why you set up this project, what competences you want to prove and learn. For example: acquired competences from your MSc programme, the elective semester, extra-curricular activities (etc.) and point out the competences you have yet developed. Optionally, describe which personal learning ambitions you explicitly want to address in this project, on top of the learning objectives of the Graduation Project, such as: in depth knowledge a on specific subject, broadening your competences or experimenting with a specific tool and/or methodology, Stick to no more than five ambitions.

As a strategic designer finding out what the right thing is to build is a key focus (especially developed during PO3, DSP, BYS). As written in the introduction, in the era of digital transformation, data, technology, and endless possibilities, finding out what to build seems more relevant than ever. Extending the role of design (thinking) to other areas therefore interests me; my internship in the technology department makes me eager to look into the potential of leveraging aspects/mindset/tools in order to innovate and improve current ways of working and problem solving capabilities for non-designers. Next to bringing opportunities, added complexity to problem solving in these contexts brings a great deal of questions that feed my curiosity. Finding answers not only through literature, but also by applying my design skills to build, test and learn in practice seems to fit the problem well.

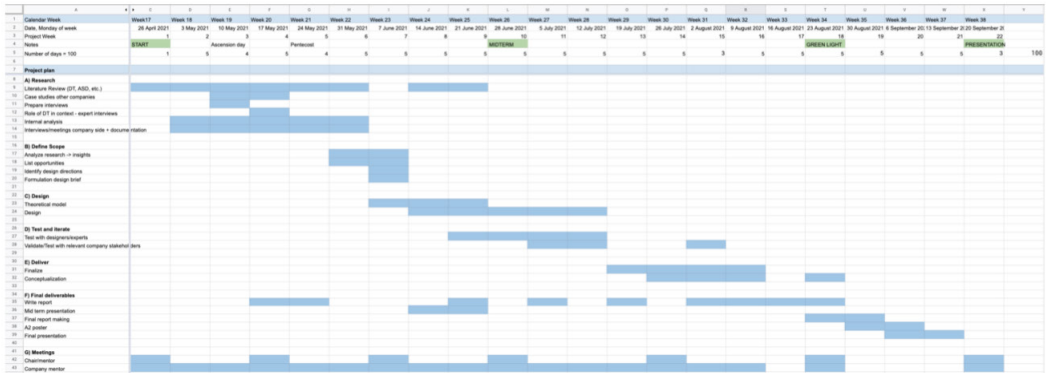
That being said, I want to prove and practice my own skills of doing research, integrating different perspectives, problem reframing, working iteratively towards the right problem-solution fit, and not being afraid to fail in order to learn. Something to point out here is that I am aware of the fact that design thinking is a buzzword being interpreted in different ways, which is something I aim to carefully consider and clarify, and form my opinion around. Working with a company, a competence to be practiced is stakeholder management and (visual) communication fitting to the audience. Designing and proposing a solution is something we practice a lot during design education, but 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.

FINAL COMMENTS
In case your project brief needs final comments, please add any information you think is relevant.

PLANNING AND APPROACH **

Include a Gantt Chart (replace the example below - more examples can be found in Manual 2) that shows the different phases of your project, deliverables you have in mind, meetings, and how you plan to spend your time. Please note that all activities should fit within the given net time of 30 EC = 20 full time weeks or 100 working days, and your planning should include a kick-off meeting, mid-term meeting, green light meeting and graduation ceremony. Illustrate your Gantt Chart by, for instance, explaining your approach, and please indicate periods of part-time activities and/or periods of not spending time on your graduation project, if any, for instance because of holidays or parallel activities.

start date 31 - 4 - 2021 end date 22 - 9 - 2021



Guideline

PROBLEM DEFINITION **

Limit and define the scope and solution space of your project to one that is manageable within one Master Graduation Project of 30 EC (= 20 full time weeks or 100 working days) and clearly indicate what issue(s) should be addressed in this project.

In the case of agile (internal-)software development, highly technical expertise is necessary to deal with technical complexity. Compared to other types of product development applying design thinking similar needs seem to be relevant, such as holistic context and user understanding, problem re-framing, ideation, and rapid prototyping to validate/reject critical assumptions and potential directions efficiently. However, contrasting to other product development processes in which design and engineering can often be treated separately, in this case “every decision about the software design unavoidably manifests at the level of architecture or code and, thus, cannot be solved without expert knowledge” (Lindberg et al., 2011), impacting the mindset and (analytical/technical) problem solving approach.

Within this context, new challenges seem to arise concerning the implementation of design thinking. Because of the technical complexity, there is a need to investigate the development process of such 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. Currently little is known about how to support the agile (internal-)software development teams with clear structure and tools in this process.

To find out how to advance Nike (DSM) Technology’s innovation/development process in highly technical 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. Thus, the main aim of this project is to identify the biggest opportunity area(s) to leverage the Design Thinking methodology in the (innovation) process of agile software development, and how we can tailor the process and relevant tools to fit the needs of this particular target group.

ASSIGNMENT **

State in 2 or 3 sentences what you are going to research, design, create and / or generate, that will solve (part of) the issue(s) pointed out in “problem definition”. Then illustrate this assignment by indicating what kind of solution you expect and / or aim to deliver, for instance: a product, a product-service combination, a strategy illustrated through product or product-service combination ideas, ... In case of a Specialisation and/or Annotation, make sure the assignment reflects this/these.

Investigate the agile software development/innovation process, opportunities and barriers, and research what/how particular aspects of Design Thinking can be leveraged to advance the current development process. Design a framework/process integrating the insights to support software development teams in developing software that captures user needs in the context of highly technical agile internal-software development.

I aim to deliver a process advancement proposal including artefacts tailored to support the context and process defined.

Appendix 2: More information about Design Thinking phases

Design Thinking model by Stanford d.school

Diving further into d.school’s vision of the Design Thinking stages; describing the what, why and how of each phase (see fig. 4)

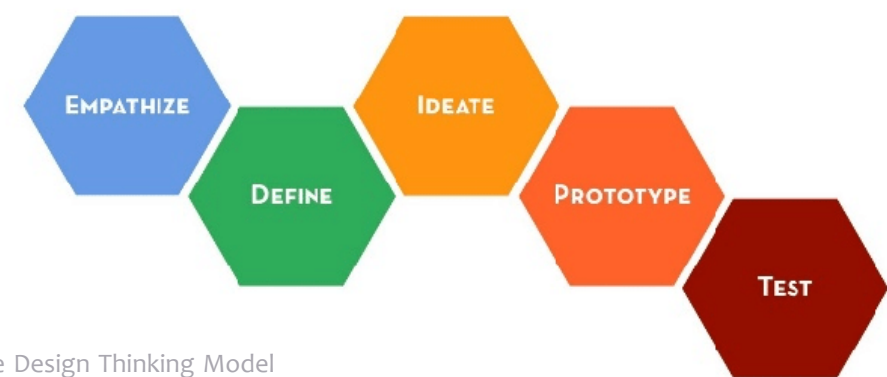


Fig. 4: The Design Thinking Model
-Image by Stanford d-school.

Empathize

What - 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?

Why - Often the problem to be solved is not the (direct) problem of the problem-solver (i.e. the developer is not the user), so in order to understand the users and their obvious and hidden needs, one must gain empathy. Uncovering information about what people feel, think, do and say by engaging with them gives insights into what they need and value. Sometimes certain thoughts and values are not immediately obvious (i.e. latent needs) to the user; ‘a good conversation can surprise both the designer and the subject by the unanticipated insights that are revealed’ (Plattner, 2009). To

recognize the insights requires an open mind, and thus empathy, as information is easily filtered out by the mind.

How - In order to empathize, one may observe, engage, watch and listen. Observation in context might bring insights about disconnects between what someone says he does and actual behavior, or about behavior normal to the user which might be surprising to the person observing. Engaging, which includes interviewing or having a conversation will bring stories from users. While engaging, the aim is to uncover deeper meanings and dive deeper towards the core of problems, values, and needs by always asking “Why?”. Through watching and listening, observation and engagement are combined: the user can go through certain tasks or steps, thinks out loud and tells why they do what they are doing. Next to the direct users, one should also consider non-users - those resisting the use of the product or service -

(Beckman and Barry, 2007), extreme users - e.g. those who are extremely excited or benefitted (Brown, 2009), and indirect users who are indirectly affected by the product or service. In broader context, Wölbling et al. (2012) mention that this initial phase of understanding might include desk research (e.g. reviewing literature), looking for analogies in other fields to reveal new insights (Rowe, 1987) and collecting unanswered questions. Furthermore it is important to postpone judgment in this phase - any ideas that emerge at this stage can be written down to refer to later (if still relevant by that time) (Wölbling et al., 2012).

Output empathize phase - To draw conclusions from the insights gained in the ‘empathize’ phase, one needs to process the information in order to understand the bigger picture and main take-aways. D.school calls this process ‘unpacking’: all the information and impressions (e.g. pictures, quotes, journey maps, experiences etc.) should get out of people’s heads onto a wall in order to start making connections.

Define

What - 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 solutioning process.

Why - This phase is a critical step towards defining and solving the RIGHT challenge. Often insights are not instantly presented to you, or

not instantly clear through all the scattered insights, which is why there is a need for this process of synthesizing information in order to discover connections and patterns.

How - By looking at insights and emerging patterns that stand out, and asking why that might be, a connection is made between the person and the broader context. In this phase, information can be visualized by clustering, through which patterns and themes emerge naturally. There are a great deal of tools to guide people through this synthesis phase, e.g. personas, user scenarios (Cooper, 1999), user journeys, Venn diagrams (Beckman and Barry, 2007) etc. Three elements are combined to create a point-of-view (POV): (1) an understanding about the user, (2) a synthesized and limited set of important needs (or even just one) to fulfill, and (3) developed insights (the ‘why’ behind the need). The POV should capture the essence of the research phase.

It is important that the POV is not too narrow, but definitely not too broad as a narrower POV tends to result in more and higher quality solutions when ideating. Next to that, when working in a team, every member should agree on and share the perspective developed, as it will guide the team through next phases (Wölbling et al., 2012).

Output define phase - There are different ways to create a POV. A common method is translating the POV into “How-Might-We..?”(HMW)-questions to naturally start brainstorming (sub-) solutions afterwards (Plattner, 2009).

Ideate

What - 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 (postpone solutioning!). Understanding of the problem space and users is combined with creativity and imagination to get to innovative solution concepts.

Why - Ideation is seen as the transition phase from identifying and understanding the problem space to creating solutions. Ideation is about exploring a wide range of ideas and going a step further than the obvious solutions to discover alternatives that might provide interesting insights; it is not (yet) about finding a single best solution.

How - In ideation, rational thoughts are combined with imagination (Plattner, 2009). For example an individual or group brainstorm can be used to build on (each others’) different ideas. Also, prototyping itself can be used to ideate; new ideas might emerge in the process of building and making decisions. Other well-known techniques include mind-mapping, sketching. An important element throughout this phase is to separate idea generation and idea evaluation, and thus to postpone judgment to allow creativity and imagination before rational examination.. Realism and feasibility are not important at this point, which allows for the solution space to grow - ‘it is easier to tone down a wild idea than to make a realistic one more radical’ (Wölbling et al., 2012). This means allowing oneself and others to voice opinions and ideas without criticism or judgment (Ambrose and Harris, 2010), which does require an open-minded attitude and atmosphere.

Output ideate phase - To avoid losing innovation potential, Plattner (2009) suggest to prototype

multiple ideas. The selection of ideas can be done by first clustering generated ideas and evaluating them by designated voting criteria. These criteria depend on the priorities and scope of the project - viability, feasibility, risk might play an important role here. Another criteria to judge ideas on is desirability: what is the most relevant solution, bringing most value to the user? Note that the discontinued ideas are not necessarily lost; documented ideas can be used at a later stage if necessary.

Prototype

What - ‘The Prototype mode is the iterative generation of artifacts intended to answer questions that get you closer to your final solution’ (Plattner, 2009). 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.

Why - There are multiple reasons to build a prototype. 1) to ideate, as new ideas emerge when making choices while building, 2) to communicate: ‘a prototype is worth a thousand pictures’ or ‘a prototype is worth a thousand meetings’, 3) to start a conversation, maybe even provoking, 4) to fail fast and cheap, as few resources are committed in quick prototyping it is a cheap way of learning up front, 5) to test possibilities, as multiple ideas can be tested quickly and cheaply without committing to a direction, and 6) to manage the solution-building process, as a problem is broken down into smaller testable parts.

How - It is important to define the goal of prototyping and to define what questions the prototype should give an answer to. While it can give interesting insights beyond the question, one should build with a particular question to be answered or assumption to be validated. The prototype is made with the user in mind. Next to that, it should be made as simple/quick/cheap as possible to get the answers.

Output prototype phase - Prototype and test go hand in hand. As said, the questions and procedure of testing have to be considered before building the prototype. How to test influence how informative the test will be - the goal is that users understand the context and can give their honest feedback.

Appendix 3: Principles behind the Agile Manifesto

On their website, Beck et al. (2001) state the following:

“We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of

conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Appendix 4: Case studies IBM, Spotify & Slack



Case study: IBM Enterprise Design Thinking

Case introduction

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. As IBM recognizes, the ‘original’ Design Thinking practice is based on the world of industrial manufacturing processes, which desirably separate design from engineering and work with well-defined specifications. However, software is built in code, and ‘because of the uncertainty of the medium, software design and software engineering are intrinsically linked, codependent activities.’ (Lucena et al., 2016). Hence, IBM adapts Design Thinking to fit their Agile Software Development process, which makes this large scale project bringing design into the company an interesting case.

First, I will analyse background information about the vision of IBM behind design and Design Thinking, and their learnings, to acquire key insights for Nike. Next, I will have a look at the ‘IBM Enterprise Design Thinking’ concept they developed and launched company wide.

What can we learn from IBM?

IBM’s (2018) expert report ‘Agile, meet Design Thinking’, and Hill (2015) and Lucena et al. (2016) provide more background information about the vision of IBM behind design, implementing Design Thinking, and their learnings. Main insights can be found below.

Why does IBM implement Design Thinking?

Agile methodologies do focus on satisfying customers, but “if agile teams become overly focused on incremental improvements, they can lose sight of the impact their iterations have on customer experiences. Design Thinking can be helpful in this situation.” (IBM, 2018). It can bring user research techniques to uncover human needs and motivations addressing the customers’ root issues, and it includes rapid prototyping to test ideas quickly. With Design Thinking, the goal is to define a solution that satisfies users’ real needs. “A customer-centric vision can improve satisfaction, reduce risk and lower the need for costly redesign cycles later.” (IBM, 2018)

- “According to a recent study conducted for IBM by Forrester Consulting [(Forrester Consulting, 2018)], adopting a Design Thinking approach can reduce time for development and testing by as much as 33 percent.” (IBM, 2018)
- Change is the only constant and the need to continuously understand and respond to changing customer needs is not going away. Design Thinking principles can be incorporated with agile methodologies to embrace user-centered innovation and to focus on the root issues to take into account and be able to test ideas rapidly.

IBM’s vision on design - change how employees think about design and redefine how teams work together

Design Thinking is for everyone, not just for designers. Diverse skills and knowledge need to be combined to design and develop great products for users. There needs to be education across the business. As it is a mindset, the approach should not be isolated in certain teams or departments, or on scattered innovation projects, but should be viewed as an essential mindset in everyday practices. Have top-down advocates to help teams adopt Design Thinking and empower teams to take more time to understand the user needs. ‘Regardless of the project phase, it’s possible to adopt a Design Thinking mindset.’ (IBM, 2018) With a flexible experimental mindset, teams gain stronger stakeholder alignment and capture higher quality user-feedback. They question themselves how to make sure the practice of Design Thinking is “part of multidisciplinary teams at the onset of projects”

User-centered goals instead of output-centric goals

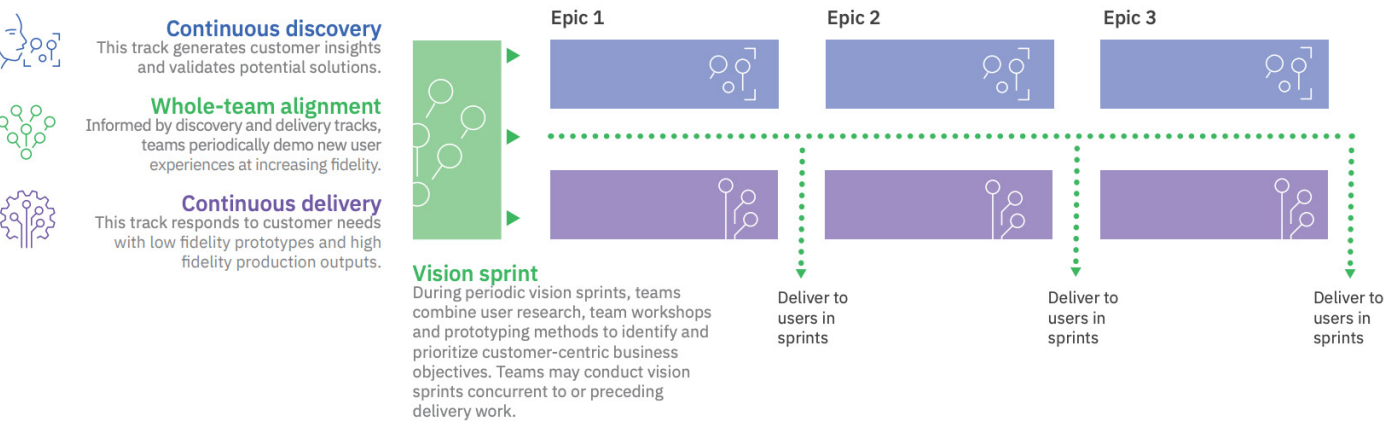
“To harmonize Design Thinking and agile, each team member’s focus must be on delivering great outcomes for users instead of on output-centric goals, such as number of products shipped or reduction in number of defects” (Hill, 2015)

The IBM (2018) expert insight report suggests to use a ‘collaboration contract’ for teams that incorporate Design Thinking for the first time, which is a quick activity to document and align leaders and team members on how the team will operate. He mentions that it is useful to have someone experienced in Design Thinking available to help teams with alignment issues.

It’s about finding a balance between discovery and delivery.

“The trick is to devise and manage a shared workflow that efficiently integrates both sets of activities.” Within IBM some teams literally set off to balance their time between those two activities (fig.) (Hill, 2015)

Integrating design thinking into an agile workflow. Teams balance efforts across discovery and delivery to maximize user outcomes



Source image (IBM, 2018)

The issue of incorporating Design Thinking in an existing sprint for the first time

The delivery schedule, and full backlog might restrict a team to try Design Thinking methods and explore ambiguous questions. IBM recognizes this difficulty to break the agile rhythm. They recommend to start small and to conduct 'hybrid sprints' in which Design Thinking activities and objectives are embedded into the sprint plan, e.g. user research is assigned within the familiar sprint structure. Starting like this helps team members to see how Design Thinking helps in connecting user needs to business goals. Continuing this practice might result in realizations in later phases that certain user stories have not been validated or even considered from a user perspective.

Key Insights

- A focus on user-centered goals is required instead of output-centric goals
- Balance team efforts between discovery and delivery
- Start small integrating Design Thinking activities (e.g. user research) in the current sprint structure ('hybrid sprint')
- Employees need to view Design Thinking as a mindset that can be adopted by everyone, and in every stage of the process (and need education with regards to this)
- It might be useful to set up a collaboration contract to align on how the team will operate

On the next pages, the 'IBM Enterprise Design Thinking' concept will be further explored.

“IBM Design Thinking Software Development Framework goal is to extend DT principles so they be applied to develop software that captures user needs at the speed and scale required for fast pace incremental software development such as on Cloud based software.”

(Lecuna et al., 2016)

Design Thinking as a CONCEPT

Let's think together.

Smarter teams, better ideas, and happier users.



The Loop drives us

Understand the present and envision the future in a continuous cycle of observing, reflecting, and making.

Observe >

Immerse yourself in the real world.

Reflect >

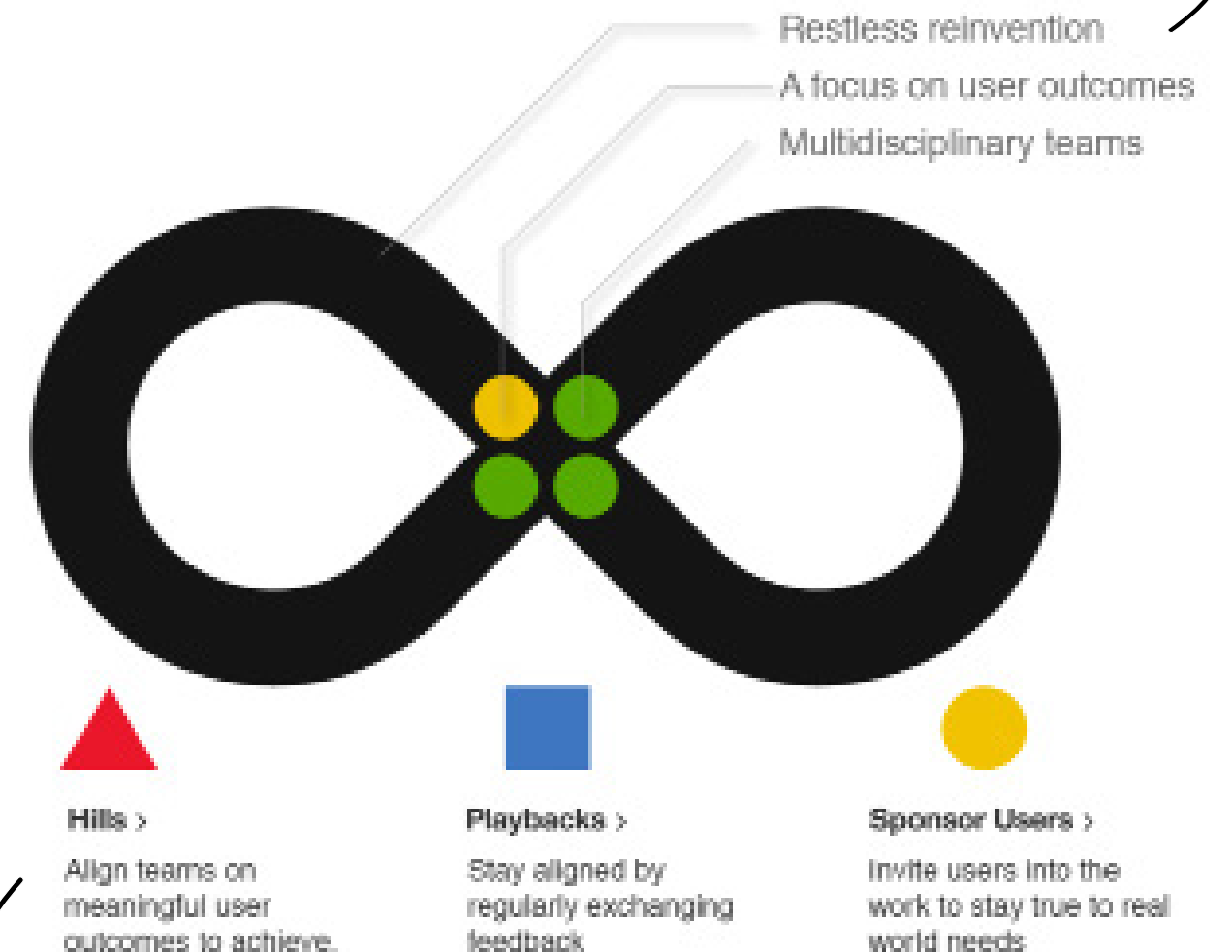
Come together and look within.

Make >

Give concrete form to abstract ideas.

The Loop is the process, based on **three main steps**: Observe, Reflect and Make, and stimulates continuous reflection.

Three principles from the model:



The model proposes **three keys** (techniques) to achieve scalability:

- 1) Hills - aim to clearly define the users' needs as project goals
- 2) Sponsor Users - real users who provide insights to improve user experience
- 3) Playbacks - different types of meetings with stakeholders to define goals, solutions, get feedback, measure progress



Hills - express user needs into project requirements

Each hill expresses a clear goal to be achieved in one or more set releases. It expresses a specific user need/problem into project requirements, informed by user research, and consists of three elements (Azis, 2016):

- ‘Who’ describes the user or specific group of users
- ‘What’ describes a problem that needs to be solved
- ‘Wow’ describes a measurable target for the Hill completion

A Complex Hill can be divided into Sub-Hills. Next to that several hills can be grouped together into a ‘Foundation Hill’ representing the product backlog for one Iteration. To allow for fast iterations, one foundation hill should not consist of more than three hills (Gothe, 2016).

Example	Who	A sales leader
	What	Can get insights from a specific market region
	Wow	By receiving consolidated data information from all available sources



Sponsor Users

Next to personas, which can only represent a part of understanding user needs (Chamberlain et al., 2006), participation of ‘Sponsor Users’ - real users - provide insights to improve user experience. Sponsor Users are existing or potential users for a certain product, and can share their individual real experiences, needs and perspectives.

A significant commitment of time is asked of them, as they will be interviewed by Product Management and the design team, and review hills, prototypes and deliverables throughout all product development phases.



Playbacks

Playbacks are ‘checkpoints’, meetings in which the project team and Sponsor Users discuss the state of the project and next steps. There are different types of playback meetings linked to different project development phases (Lucena, 2016):

Business Goals Playback (start of the project, only involves product team)

- establish initial business case
- understand users through user research
- define main users and identify Sponsor Users

Hills Playback

- product team and Sponsor Users align on three primary hills of a release
- define release strategy, major hills and their relationship in the product roadmap
- make a rough estimate of amount of work, and time and resource constraints

Playback Zero (just before delivery work starts)

- product team and Sponsor Users commit to user experience of the product
- align on final version of the hills and user experience to achieve them
- uses a Customer Journey Map (Richardson, 2010) providing a diagram of Hills from an individuals perspective of his/her experience with the service/product being developed
- product manager defines hours that could be invested on each hill
- ends with the team and stakeholders in agreement on the commitment to deliver each Hill

Delivery Playbacks

- among product development team and Sponsor Users to demonstrate a solution for a Hill
- ends with a decision on whether to release the project to real users

Three roles within IBM Design Thinking

1) Product Manager

- Responsible for:
- Understanding the [market] opportunity
 - Defining the product release
 - Project kick-off
 - Defining and recruiting Sponsor Users
 - Defining the playback strategy

2) Engineering team

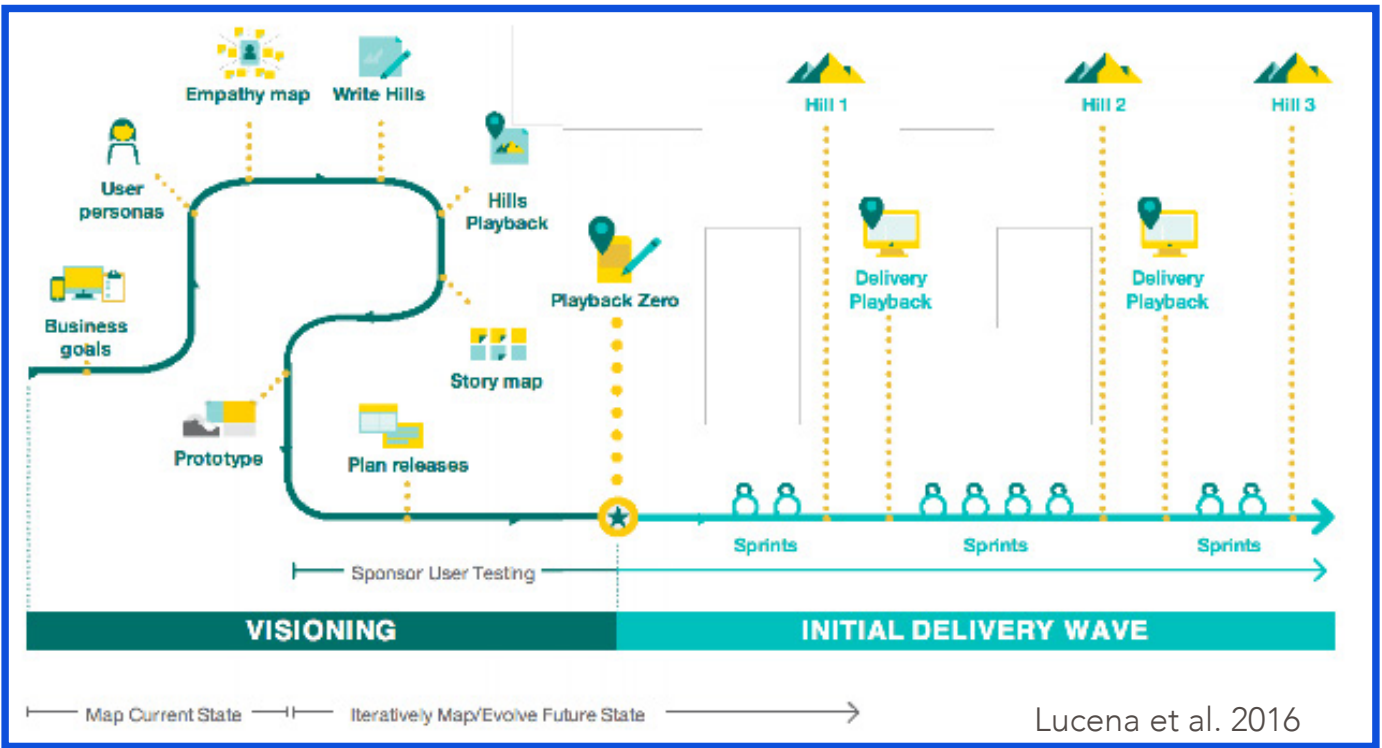
- Responsible for:
- Technical design
 - Implementation of the release
 - In control of project architecture, executable code, and prototype
 - Technical sprint-plan

3) Designer(s)

- Responsible for:
- User Research
 - User Experience
 - Functional Design
 - Engaged in developing design artifacts, mock-ups
 - Design sprint-plan

Design Thinking project work-flow

IBM Design Thinking Software Development Framework activities are divided into two main phases: the visioning phase & the delivery wave.



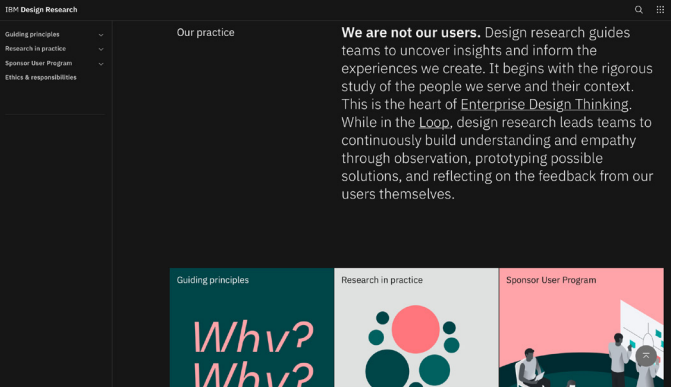
The Visioning Phase

Developing software requirements through the use of several Design Thinking practices that combines user personas, empathy maps, hills and story maps.

The Delivery Wave

Software development sprints by multidisciplinary teams that incl. sponsor users who contribute with constant feedback about delivered artifacts.

User Research Guidelines
<https://www.ibm.com/design/research/>



Field guide: comprehensive guide pocket size for Design Thinking, Tangible artifact for employees



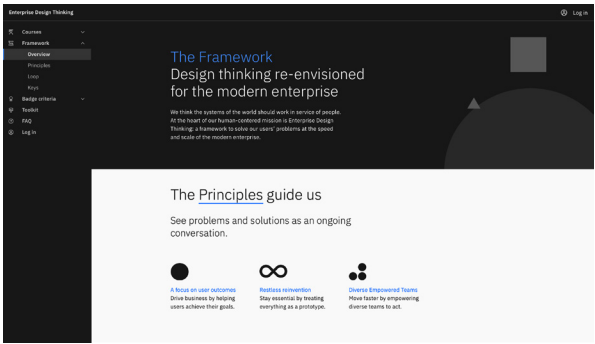
Link to an online version:
<https://www.slideshare.net/NewfluxUXBlog/ibm-design-thinking-field-guide-v34-73702608>

Take-aways from the concept

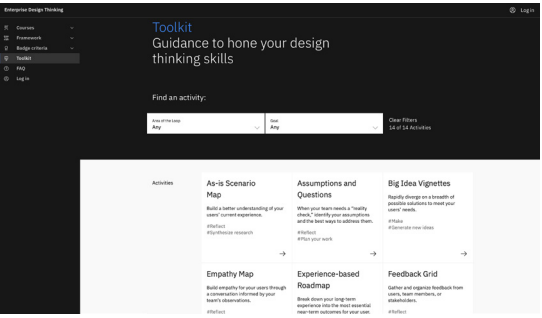
- Bringing in Design Thinking is a large scale project and requires a change in the culture of the company. It took IBM over 3 years, involving over 750 designers.
- IBM set up a tailored design process to bring the concept to employees, with visualizations capturing the core focus areas concerning Design Thinking in IBM’s context.
- Designers are present in the project teams (they went from designer:developer ratio 1:72 in 2012 to 1:8 in 2017).
- Their vision is that design and engineering are co-dependent activities in software engineering, which is why they need to be close-knit collaborators in a software development project.
- User needs are set as project goals [human-centricity at the core], a format is given
- Split the visioning phase and the delivery wave (split problem-solution) with a meeting splitting the two phases ‘playback zero’.
- Meetings with set expectations throughout the initial visioning phase and the sprints
- Next to a website, there is a field guide bringing something tangible to the change. Both are accessible for everyone.

IBM Design Thinking Resources

Website ‘Enterprise Design Thinking’
<https://www.ibm.com/design/thinking/>



Includes a toolkit:



Spotify® Case study: How Spotify builds products

Case introduction

Spotify is a audio streaming service founded in Sweden. With over 356 million active users in the first quarter of 2021 (Statista, 2021f), the products are loved by a great deal of users and artists and the company has been growing fast.

The article by Kniberg (2013) elaborates on how Spotify creates products people love, their general approach to product development. Their core philosophy is to manage innovation risks by prototyping cheap and early; to not launch based on a date but based on quality; to be great at launch and become amazing iteratively afterwards.

Major initiatives go through 4 stages:

Think it - 'figure out what type of product we are building and why'

Build it - 'create a minimum viable product that is ready for real users'

Ship it - 'gradually roll out to 100% of all users, while measuring and improving'

Tweak it - 'continuously improve the product. This is really an end state; the product stays in tweak it until it is shut down or reimaged (=back to think it).'

The stages are invented recognizing that the most common reason for failure is building the wrong product - which is defined 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).



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

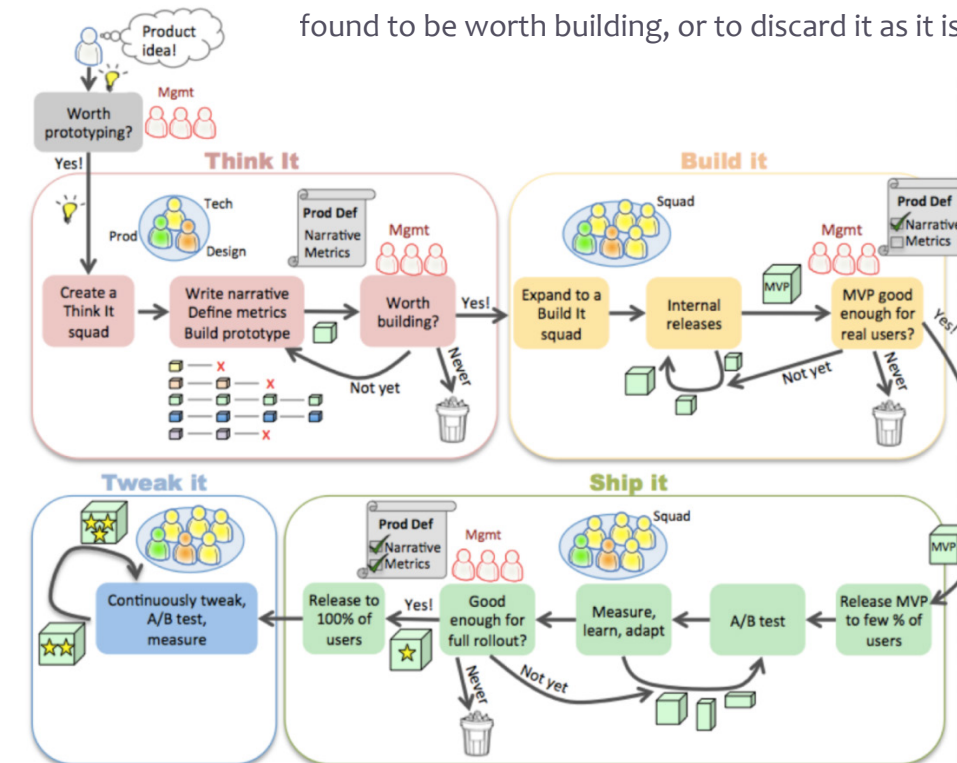
(Kniberg, 2013)

'Think it'-stage

I will dive a little bit further into the 'think it' stage as this is the phase prior to actual development in which ideas are explored. In this phase anyone can come up with ideas, new product ideas or reimagining an existing product or improvements to existing products. Management has to agree it's worth exploring and if that is the case a 'think it' squad is formed with typically a developer, a designer and a product owner aiming to write a product definition and build a prototype.

The product definition answers for example to the why of the product, who will benefit, key metrics that are expected to improve, hypotheses etc. It is not based on requirements and does not list features or resource plans, 'it is more like a data driven purpose statement' (Kniberg, 2013).

The Think it phase ends when there is consensus to either build the product as it is found to be worth building, or to discard it as it is not worth building.



I would like to point out that users are not emphasized as much as I expected; management seems to decide if it's worth prototyping and users only come in later on. Other than in the case of spotify, which is focused on external customer, the Functional Tech products in the case of this graduation project is about internal products, providing an opportunity to work closely with users from the start.

Main Take-aways

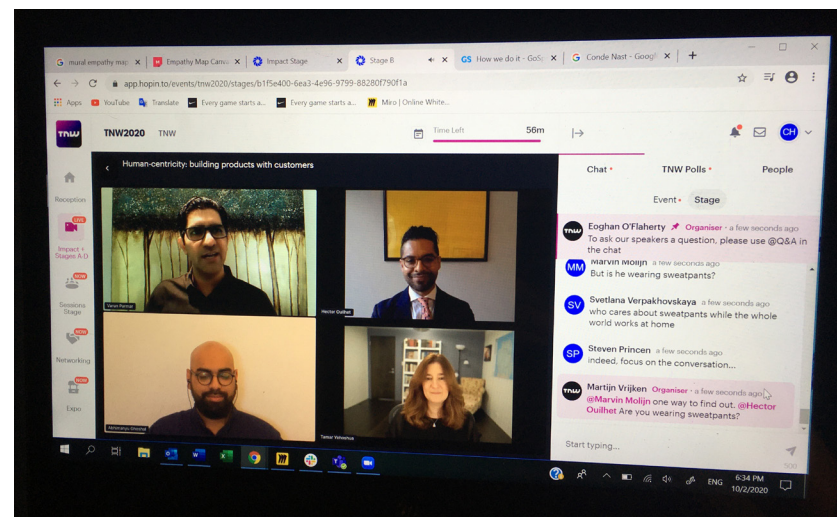
- In the 'Think it' stage, product risk can be reduced at low costs. As there currently is no stage in which quick prototypes are made prior to development in the Functional Tech team's process, there is an opportunity to make most out of this.
- Spotify defines the 'wrong product' as 'a product that 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). 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). How does/ should Nike define 'the wrong product'?

As can be seen in the graph (Fig. 10), the 'think it' stage drives down the risk at a low cost. The 'build' it stage should be as short as possible as operating costs are high but risk is reduced only little. In the 'tweak it' stage reflects that updates are needed less and less and a squad can move on.

Case study: Human Centricity at Slack

Case introduction

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 in an expert panel. 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 below.



Conference TNW on Human Centricity
1-2 October'20

Main tools used at Slack

Personas

At Slack personas are used to change the mindset of how software is developed.

“You have to know who you’re building for, don’t build for yourself”
It is encouraged to talk to a lot of people with mock-ups and prototypes.

Complaint storms

At Slack, they use ‘complaint storms’ in which the teams pretend they’re the customer and have to put yourself in the shoe of the user to get issues from their perspective.

Customer journey maps

Customer journey maps are used to get a 360 degree overview and to find the touch points, the voice of the customer. One team owns it, others tie in. The customer journey maps bring the Empathy mentality as they can help understand what customers go through. And they give people a language to what you’re focusing on e.g. consideration phase, onboarding phase - which phase are you improving?

To build empathy you need to know their language.

User research

The practice of design is about: Who are your users?
There should be a function of research hired to do:

- Usability studies and iterations
- More strategic, foundational research proactively influencing product strategy through research
- Validation and finetuning, which is cheap because of prototypes and mock-up opportunities. This should be done quick and early to improve velocity.

Prototypes and ‘ambassadors’

At Slack they say ‘prototype the path’ in which they prototype with customers whom they call ‘ambassadors’ in the same slack channel to have a close iteration loop and continuous feedback.

Represent user in the room

Have something in the room to represent the perspective of the user. Something that reminds them of the desirability and the ‘do we actually need this?’ e.g. personas

Role of top-down management

In order to improve human centricity their recommendation is to set the tone from top-down. Management needs to take accountability for some loss of velocity because you need some more time to understand.

In smaller organizations: passion drives development. In bigger corporations employees are often levels removed passion away there is a need to reinforce.

Top-down investment in resources to set best practices, professionals that understand

Training is needed to scale throughout the company

Culture that demands ‘like using’ versus ‘love using’ - create products that people love!

Measure impact

Metrics used: sentiments, NPS, user/usability metrics

Metrics shouldn’t be too focused because that incentivizes incremental change.

Move on when there is a statistical significant change.

Look at small wins for the users, to keep innovation velocity and give them the feeling that the product is continuously getting better.

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.

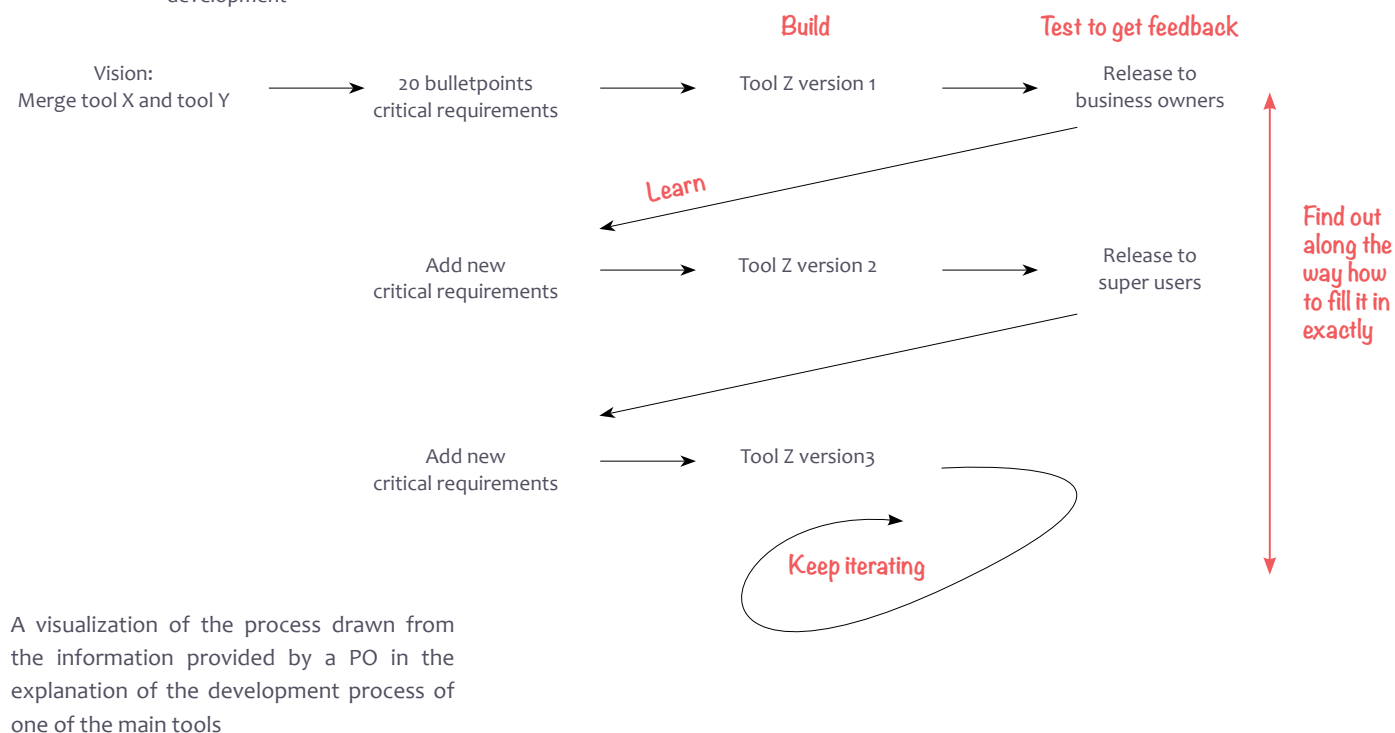
(Different) metrics apply to incentivize the right human centered change

Appendix 5: Interview Guide

- Could you describe your role?
 - Could you describe the type of tools you and your team work on?
 - Who are the users?
 - Could you describe a typical question you get to solve?
 - Could you describe steps of the process from the moment you get a certain request until development starts?
 - How and when are users involved?
 - How and when are developers involved?
 - Methods used to get to the core of the problem?
 - Pain points in the problem exploration process?
 - What is your vision concerning the future way of working?
- What has to change in your opinion?

Appendix 6: Agile development process example

Source PO
Case: Simplified MVP process tool Z
development



A visualization of the process drawn from the information provided by a PO in the explanation of the development process of one of the main tools

Appendix 7: Main iterations of the conceptual model

In this section, two main iterations that precede the final model are presented. The iteration on the right page already looks a bit like the final model. By looking at what the elements actually mean, i.e. the red elements are about alignment and the purple about divergent thinking, the model could be further simplified. This was done through an ideation process on paper, see fig. 44.

Feedback notes of the first iteration:

Expert

- “You’re noticing all the problems that are here today”
- “I put together something very similar conceptually, but different infographic.”
- “Visually it is unclear where do you start?”
- “It is confusing in this visual the way the agile development is treated.

Is it part of agile here? There are different ways of looking at it.”

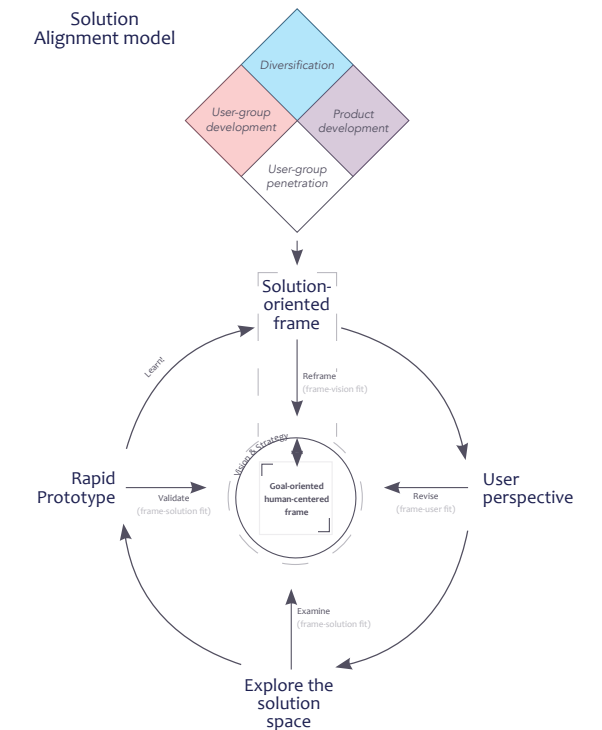


Fig. 28 First main iteration

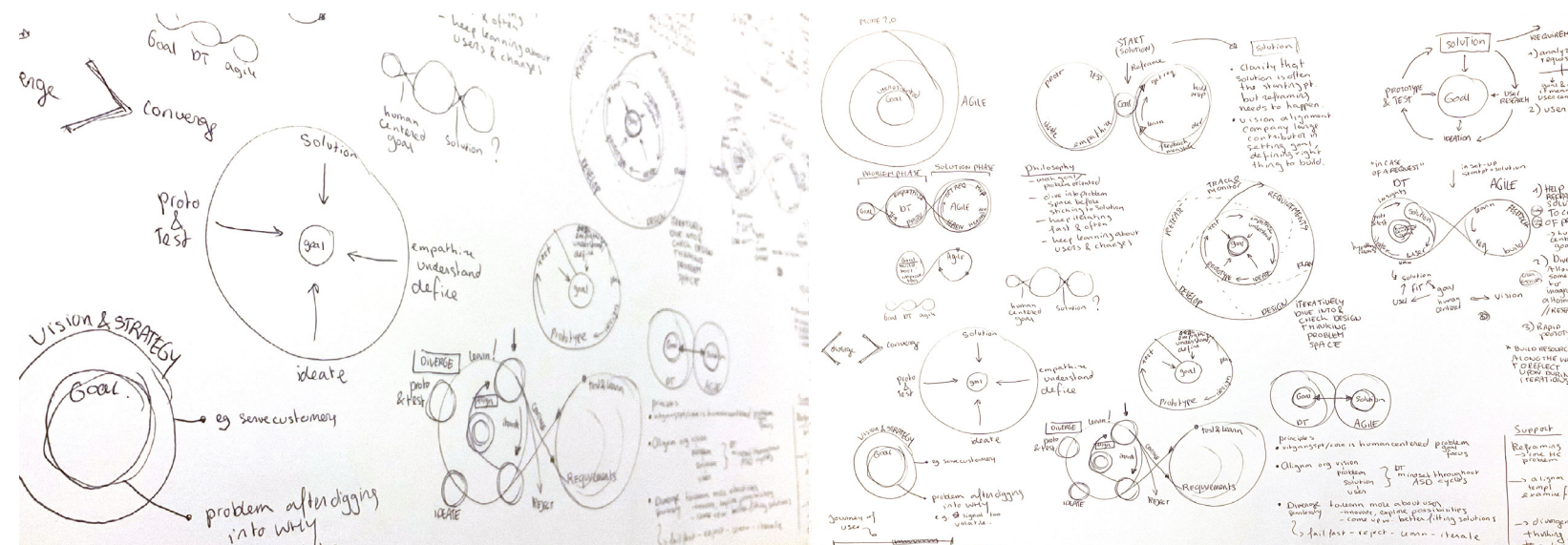


Fig. 44: Ideation towards an improved version

Team

- Where to start?
- ‘I don’t understand what I’m looking at because it doesn’t have a title’
- Going through it with an example helps
- Need clarity around the level to define the goal: ‘if my goal is to ‘improve the plan’, do I need to dig deeper?’
- ‘How is it linked to development?’
- Without explanation the arrows inside were not understood. When the goal of ‘alignment’ was clear, it made sense, so it would be good that that goal is clear prior to looking at it.
- The colored lines in the circles are all the same, which means the model could be simpler. It’s confusing as this gives the indication that there is a need to match those colors to the diamond. The simpler the better.

Coaches

- Its too complex, I have to make sure to start at common ground
 - good introduction of what the reader is going to be reading afterwards.
 - Consider adding numbers to steps
- ‘Only after reading all information it became clear to me how to use this framework. I think it’s important that you specify more clearly that you will define each of the building blocks and then provide an overview of how to use them.

In the second main iteration, the principles were already established, the model further simplified and clear. The model was based on the building blocks shown below.

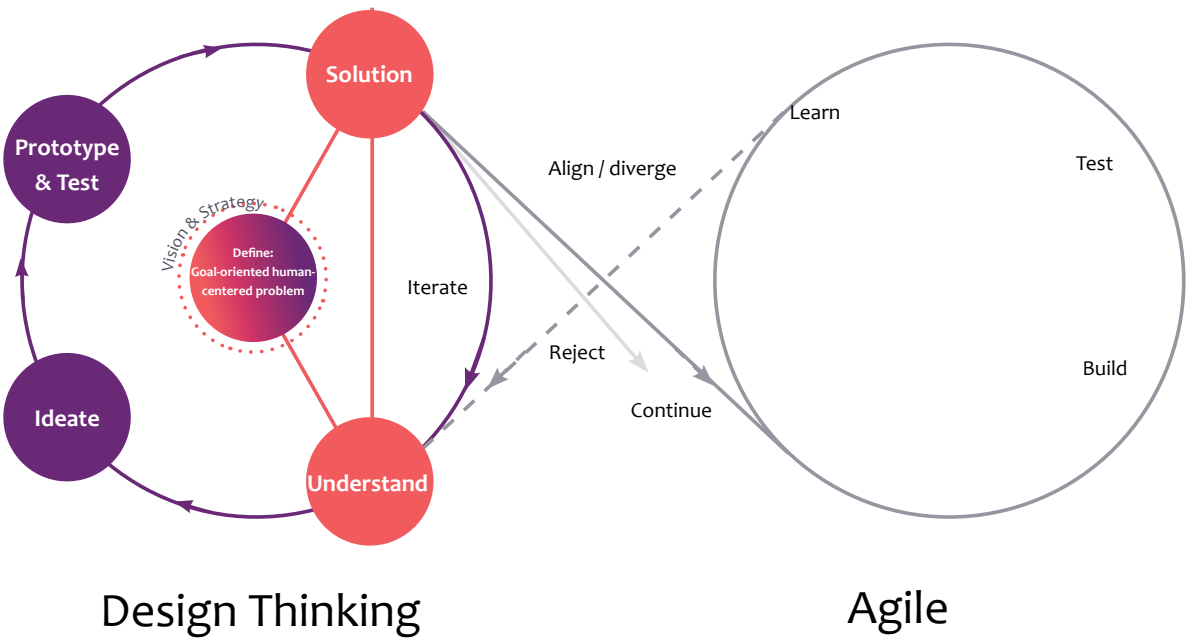
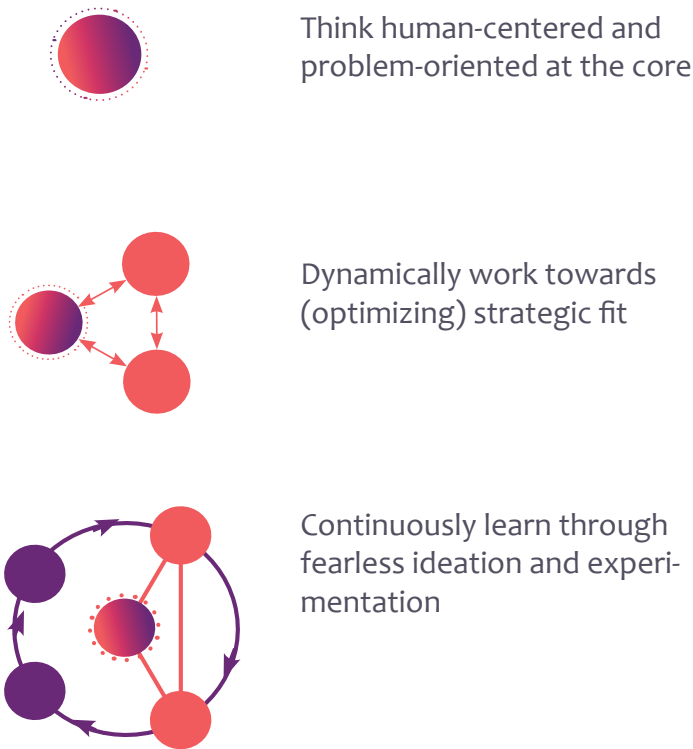


Fig. 45: Second main iteration

Appendix 8: Complex version of the Solution Alignment model

Core of the model - Alignment

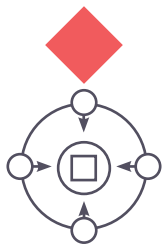
Two concepts characterizing the context of internal agile software development are the internal stakeholder requests and the solution-oriented approach to problems. In this context, the main aim behind getting clarity about a problem and user (goals, practices, user journey) is not to find innovation opportunities to tie (potential) customers to the business, as is often the case in general Design Thinking for innovation models. In this case, the users are locked-in already and the intention revolves around alignment: alignment with the leadership vision, strategy, company processes; alignment between users and management goals; and alignment between teams - finding the best fit between the vision-user-problem-solution elements. This alignment is the core focus of this model, and thereby guides for example the intention with which one would go through certain phases and use certain tools: a 'persona' can be used to find out how the user's goals and pain points fit the given frame of the request, exploring the 'user-journey' is done with the intention to examine where and how the given problem and solution fit with the practices and pain points of a user.

All in all, the phases are a means to get to and validate the necessary points of alignment. Throughout the process, teams and stakeholders need to be on the same page around these points of alignment.

Elements of the model

Like the Design Thinking models presented in

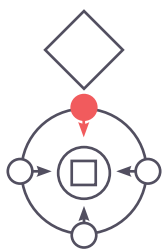
2.1.4, the phases are presented consecutively, but must be regarded as highly interconnected, iterative and non-sequential; allowing to go back and forth in the process when necessary.



1. Internal growth matrix

A project or request can be placed in this internal-growth matrix, as the strategy for internal growth of a software product can be classified to address existing/new user-groups and existing/new products/features. The first step is to analyze in which quadrant the request or idea can be placed. This will help to understand the needs within the phases of the cycle and if problem exploration based on Design Thinking will fit the problem. More about this matrix can be found in chapter 3.3.2.

In the 'user-group penetration' quadrant, Design Thinking is most likely not relevant as problems are well-defined (i.e. one logical path), e.g. concerning fixing bugs, maintenance and onboarding more users of the same group. Note that when scaling a product, there is a need to carefully consider if the growth should be classified towards existing users or new users. Unjustly placing the project in the 'user-group penetration' quadrant and skipping user orientation can have consequences related to user-misfit later in the project.



2. Solution-oriented frame

The solution-oriented frame refers to 'the (right) thing to build', e.g. a request, a proposed solution, an initial idea or a kind

Solution Alignment model

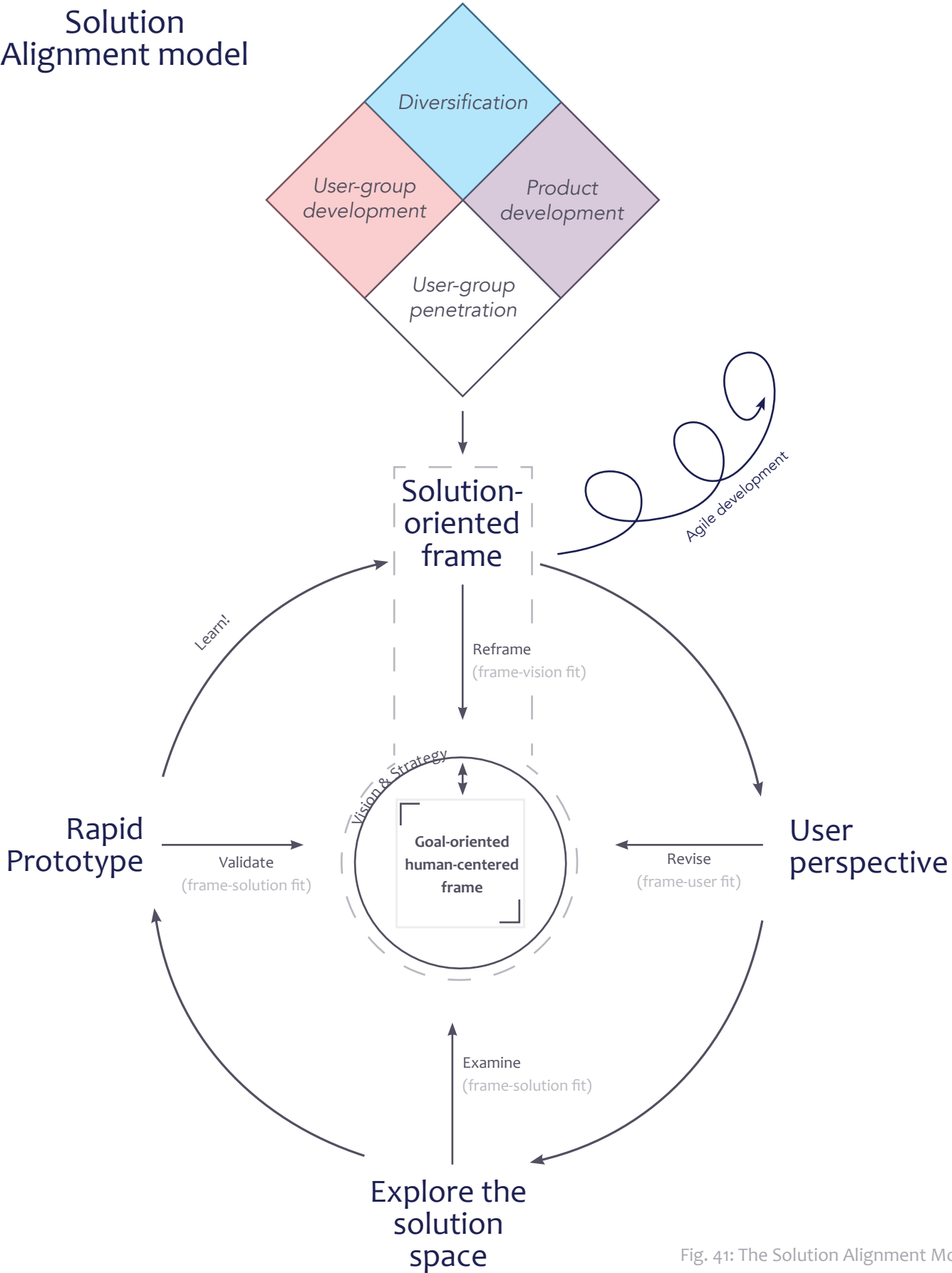
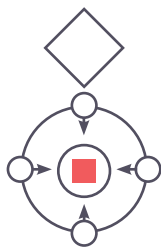


Fig. 41: The Solution Alignment Model

of prototype that a stakeholder is bringing to the table. The tendency to immediately jump to getting requirements to start building it is postponed in this model; the model suggests to first focus on problem-exploration and solution-exploration.

In this stage focus is on understanding the core reasons behind the request. Taking a step back is required in order to understand if and WHY this would be the right thing to build.

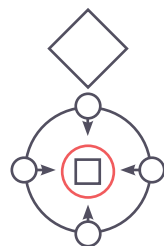


3. Goal-oriented human-centered frame

With the information gathered about the goal, vision and problem, the reframing element towards a goal-oriented human-centered frame is a key step in this model – critical to be able to consider and open up the problem space (and thereby opening up the solution space). Ideally, different perspectives (e.g. user, subject matter experts, stakeholders) are brought together to discuss the problem and the goal, not yet the solution.

This element is constantly referred back to in

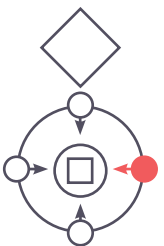
order to make sure elements align: with the vision and strategy (step 4), the user perspective (step 5), and finally with the solution (step 6, 7 and 1).



4. Vision & strategy

As we're operating in a large company with business vision and strategy on a company, department and team level, this is the phase in which the frame-vision fit has to be considered. Is the vision behind the frame clear – is it clear why we should be solving this issue/problem? Does the frame we intend to tackle, i.e. the problem we intend to solve or the goal we intend to go after, fit the higher-level vision? This stage is a key step in getting people (e.g. stakeholders and users) along in the process, and in empowering teams to proactively think about how to work towards that vision.

As mentioned before, the steps are highly interconnected. To involve the user in these discussions, one might first or in parallel investigate the user perspective (stage 5).



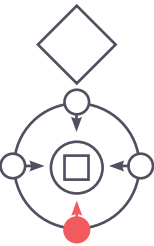
5. User perspective

In order to build the goal-oriented human-centered frame it is important to understand the user perspective, understand the users' problem, pain points, goals, practices, mindset, needs, the user journey and fit of the frame (and later of the solutions) within this journey. This will 1) help to focus on the right frame, and 2) help in understanding

the user situation and mindset to aid with the frame brought to the user towards user engagement and user adoption. As the users are the people who have to use the product in the end, the frame-user fit is essential. With knowledge about the user gained in this step, the frame (stage 3) is revised.

In the case of scaling a product ('user-group development'), this is the phase in which existing and new users can be compared in terms of their practices, mindset, goals, journey etc. Significant differences can be further explored in terms of their impact on the frame and current solution.

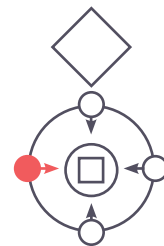
Note that this stage is part of problem-space exploration – thinking about solutions is postponed to stage 6.



6. Explore the solution space

After opening up the problem space in steps 1-5, it is time to dive into the solution space. The challenge is to diverge from the current solution proposed, come up with alternatives, allow exploration of other paths before potentially with a certain solution. Internal market research can bring insights leveraging what already exists, to not reinvent the wheel and/or to open up possibilities. External market research, e.g. trends and new technological possibilities, can bring inspiration and open up the opportunity space even further. Bringing different perspectives together in diverging and ideation will help to learn about different parts of the problem and solution.

Insights, (partial) solutions, hypotheses etc. can be concluded at the end of this phase. The aim is then to examine the fit of the solutions and the goal-oriented human-centered frame. Impact/value and effort of the solutions, and criticality of assumptions, should be examined to prioritize and make choices about what to further elaborate on prior to the start of development.



7. Rapid prototyping

In this phase, the process continues by building quick prototypes with the intention to learn more about the problem and the potential solution, and to validate problem-solution fit rooted in qualitative and quantitative user-data. Making the prototype(s) should be quick and cheap, and can be thrown away after learning about the questions and assumptions aimed to answer. The aim is to validate the frame-solution fit by testing the hypotheses and measuring the goals. Insights about the product, the attitude, mindset and knowledge of the people involved (e.g. users) etc. can help to configure the agile development stages and user engagement afterwards.

A decision about taking it into development can be made with the data gathered at this stage. The outcome can be that the solution is not validated, which could mean starting an iteration to pivot with the insights gathered or to stop the project. The outcome can also be that the solution is validated and can be taken into further development.

“The manager who comes up with the right solution to the wrong problem is more dangerous than the manager who comes up with the wrong solution to the right problem.”

- Peter F. Drucker

Fit with Agile Software Development

The vision of this model is that there is a problem exploration phase, investigating the goal, users and potential alternatives prior to accepting a certain request or solution. This is a crucial step in order to potentially break away from existing boundaries and frames.

As identified in chapter 2, the Agile methodologies focus on building a solution in the right way, but questioning if that solution is the right thing to build and thoroughly understanding the problem, context and user is not a structural part of the Agile process. Design Thinking was identified to be able to solve this gap (§2.3).

As adding features to software and scaling the software to different user-groups happens iteratively, the model can be passed through iteratively. The order cannot be generalized, e.g. a certain project might first add features prior to onboarding users of a different user-group (fig. 42A), another project might add features and scale to a different user group through several iterations (Fig. 42B), and

another project might decide to add a feature after a while (Fig. 42C). Problems classified in the user-group penetration (e.g. support, fixing bugs etc.) quadrant likely happen in parallel to the other phases.

One could view the model as continuously present throughout the growth-stages of the software (see fig. 43 for a representation):

Diversification - In the fuzzy front-end phase of the agile software development process, in which a software is yet to be developed and users are new, a project/request can be classified in the diversification phase. A new product will be built and new users will be onboarded; understanding and validation prior to building can be created through the model.

Product development - Next, new feature will be added to the software to grow the functionality of the software in an agile manner. The need of these features and priority of these features will have to be examined, which can be done through the model.

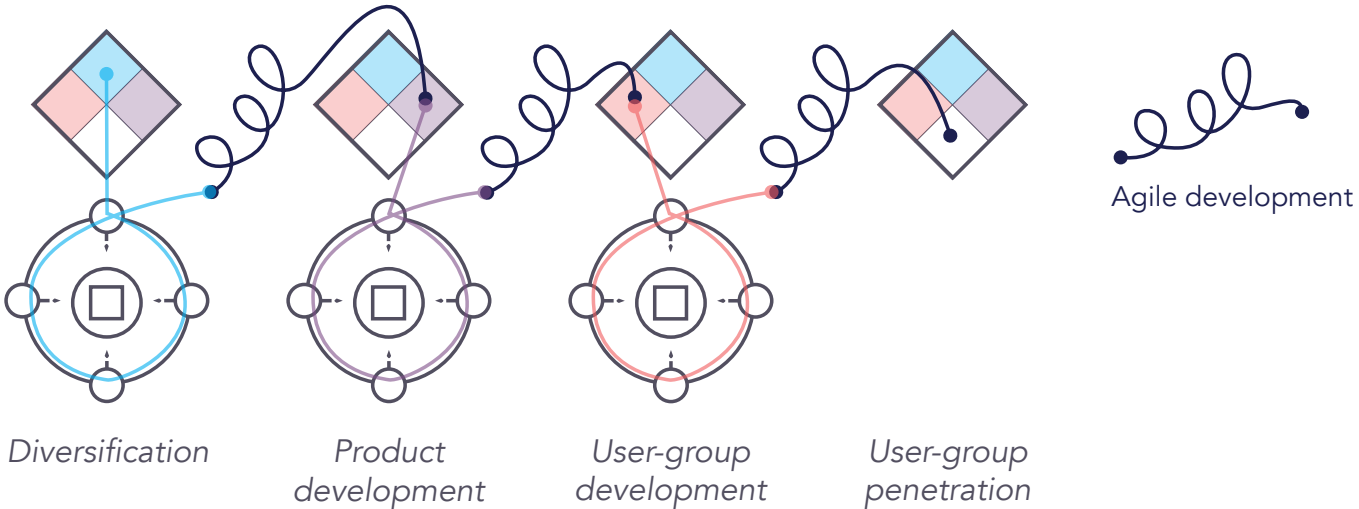


Fig. 43: Representation of a an Agile software development process with the problem exploration phase integrated

User-group development - At a certain point, a software can be scaled to new user-groups, e.g. in the Operations Tech unit’s case: planners of different areas of the business. The model can help in guiding the process of first well understanding the new user-group and assumptions concerning differences and similarities.

User-group penetration - At any stage more users of the same user group can be onboarded and support and bug-fixing will happen throughout. As mentioned before, in quadrant, Design Thinking will most likely not be relevant due to the type of (well-defined) problems. When a software is matured and no new features are added and no new user-groups are onboarded, the project will mature in the user-group penetration quadrant in which requests are mainly focused on support and maintenance.

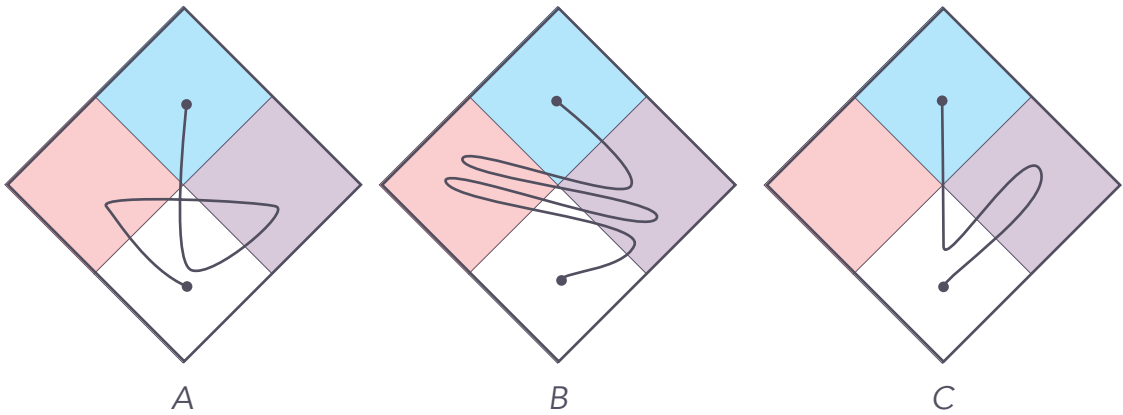


Fig 42: Agile software development project can pass through the different growth strategies in different ways during its lifecycle.

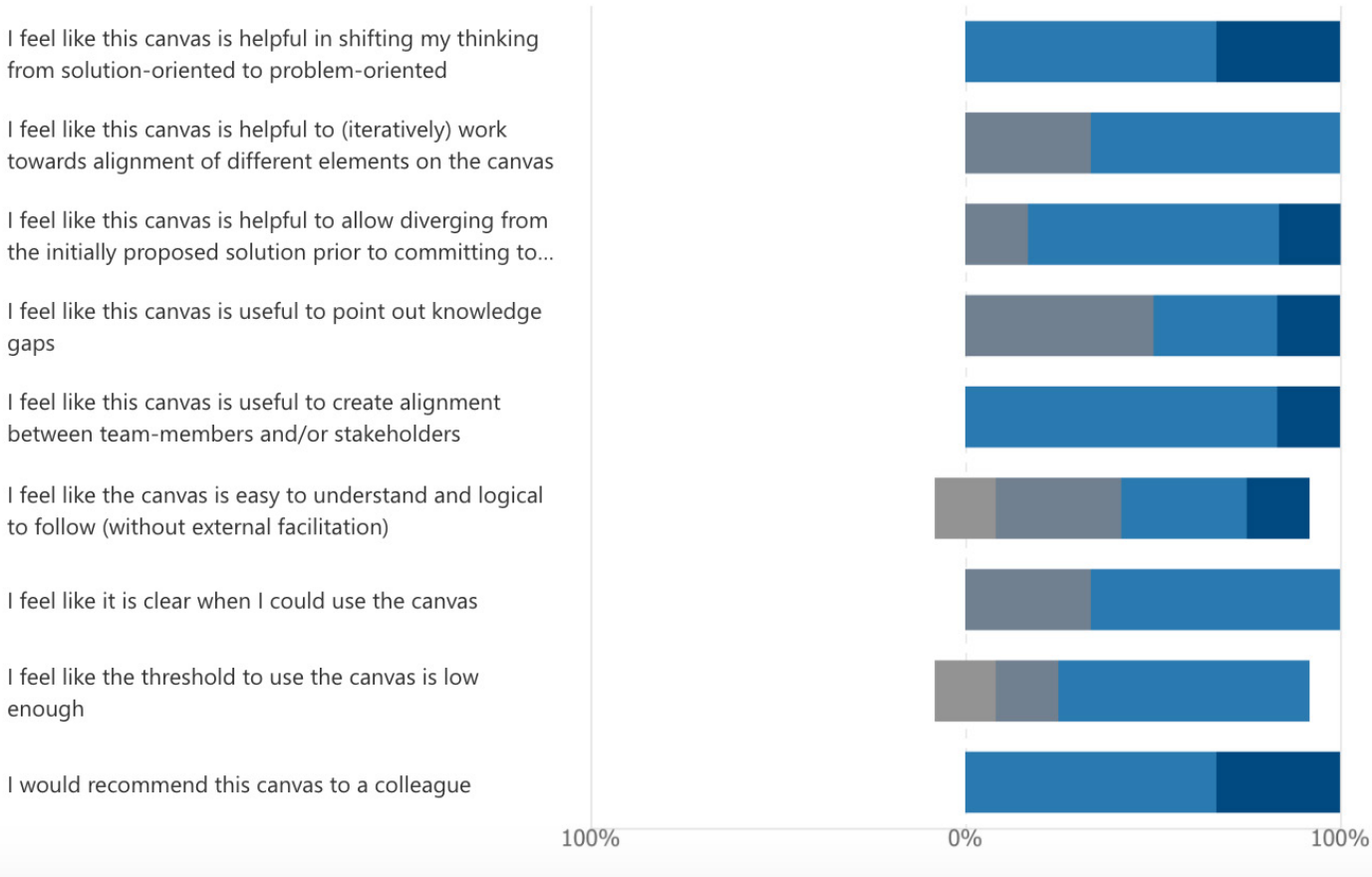
Appendix 9: Survey results

1. What is your role?

[More Details](#)



Strongly disagree Disagree Slightly disagree Neutral Slightly agree Agree Strongly agree



3. If I were to use this canvas, my biggest concern or point of improvement would be ...

5 Responses

ID ↑	Name	Responses
1	anonymous	NA
2	anonymous	Not a concern: Canvas may also have a placeholder for dependencies or dependent teams (along with end users and clients) to get a 360 degree view of the problem.
3	anonymous	to understand what exactly the outcome is - the last section is labeled "notes" which makes it feel rather unimportant, but is actually a main takeaway from the document
4	anonymous	when the users propose a solution they think they need, how to convince them that the canvas will improve user centeredness?
5	anonymous	To position it well. If not applied in the right context (like we did with only tech where business should be involved earlier in the process), you get stuck and don't get the potential value out of it.

4. Would you use this canvas again? Why? (If not, what would need to happen for you to use it?)

5 Responses

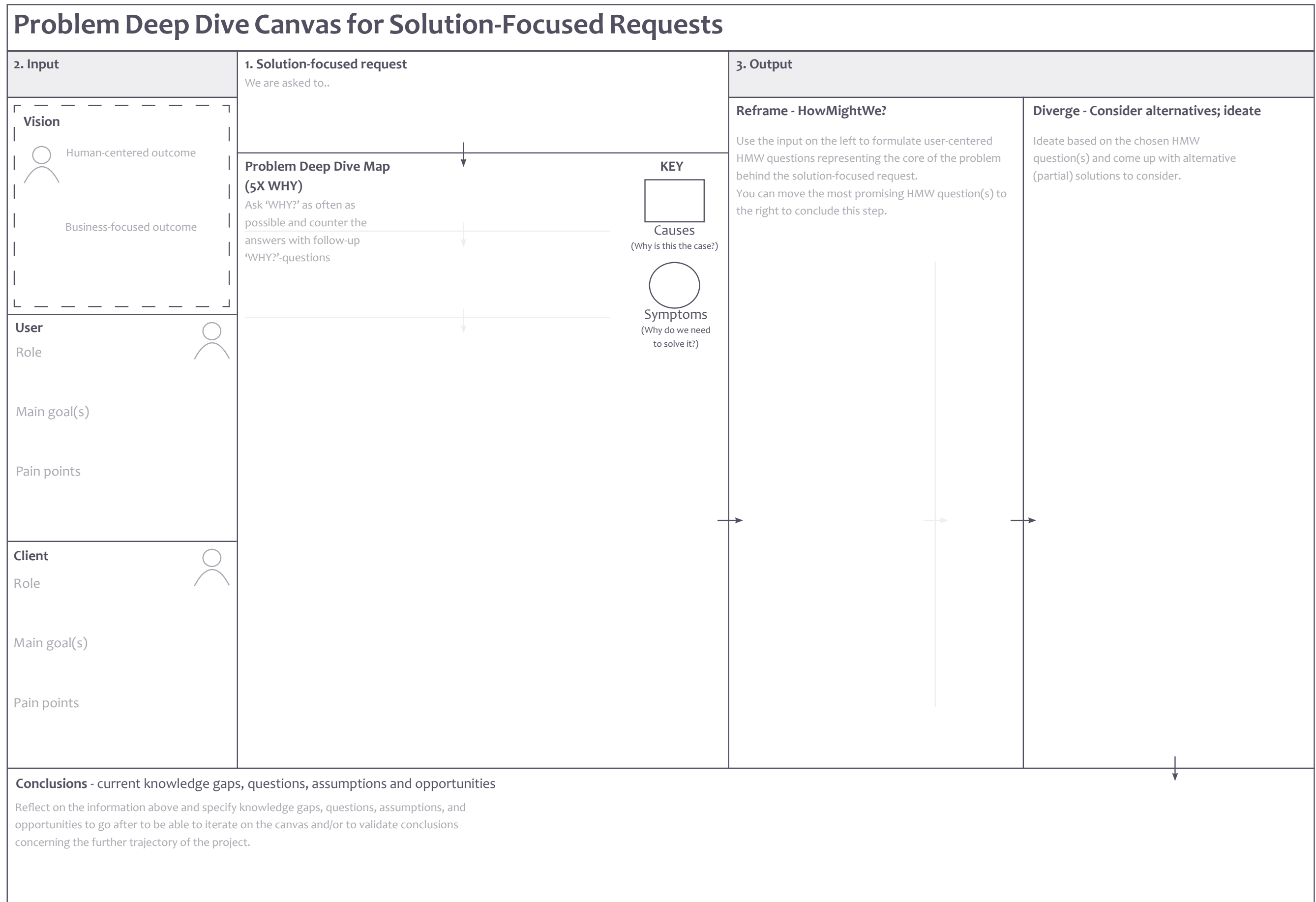
ID ↑	Name	Responses
1	anonymous	I would use this canvas again because it lists out the necessary elements to be focused while brainstorming
2	anonymous	Canvas encourages problem oriented thought process and stresses on looking at the problem from a user centric approach. Additionally it reflects fundamental Agile principles to build user centric potentially shippable incremental solutions.
3	anonymous	yes - I think it would be beneficial to use amongst our product teams when we get a "solution" request from the business. It will enable us to think outside of the box and provide a more fit-for-purpose solution
4	anonymous	Yes. Because it challenges the development on design decisions that were not really or strictly user centered.
5	anonymous	Yes, but only in collaboration with the business. From tech-side only for initiatives that span across it doesn't work well in isolation.

5. Other remarks.. notes, thoughts, feedback, etc.

3 Responses

ID ↑	Name	Responses
1	anonymous	Great work on building this Canvas.
2	anonymous	I like that the canvas incorporates several "tools" that we should/could use in product development - it touches on design thinking elements as well as user/customer personas
3	anonymous	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.

Appendix 10: Final Problem Deep Dive Canvas



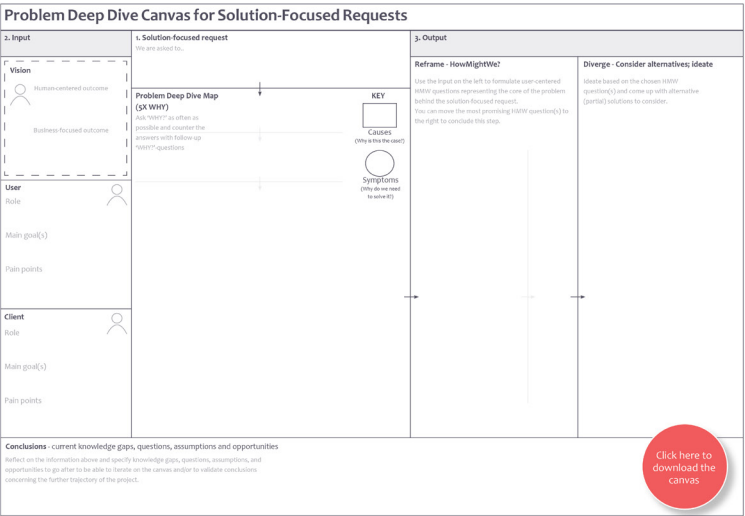
Problem Deep Dive Canvas for Solution-Focused Requests

- a Tool Guide for Nike Tech -



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 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.



- Tips
- Print the Canvas on A3 paper or use digital tools like Miro or PowerPoint
 - Fill it in individually to create overview and structure in thoughts or fill it in collaboratively to create a shared understanding and to be on the same page.

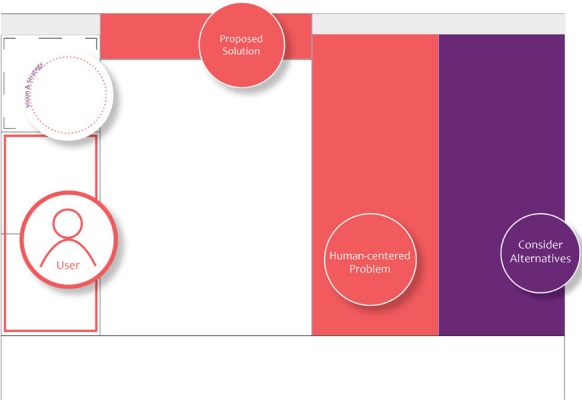
The Canvas

- What is it?
- A problem Deep Dive Canvas to consider solution-focused stakeholder requests (e.g. new initiatives). The canvas can be iteratively edited 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

It is based on the following principles:

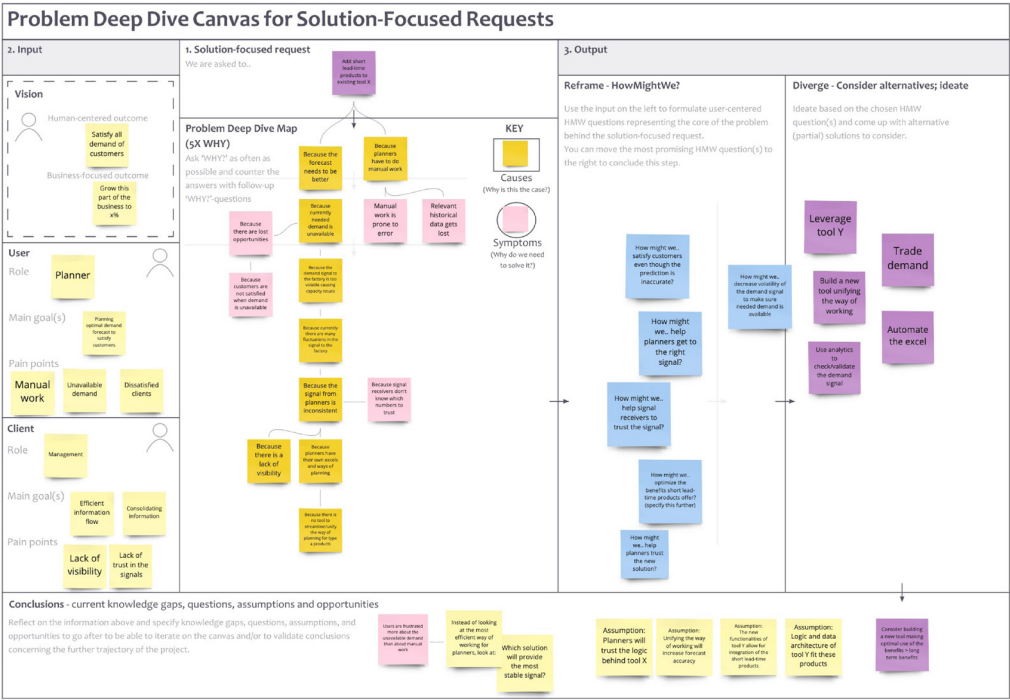
- Principle 1: Think human-centered and problem-oriented at the core**
- Principle 2: Dynamically work towards (optimizing) strategic fit**
- Principle 3: Continuously learn through ideation and experimentation**

- 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



Example

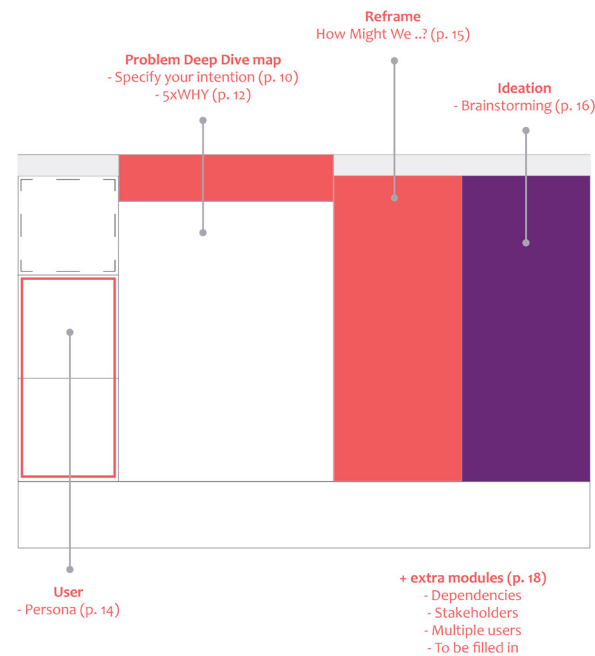
- On the right you can find an example of a filled-in canvas. It's inspired by a real case, but changed and simplified for confidentiality reasons.
- Externalizing insights in this way provides an overview of knowledge-gaps and elements to dive further into to better understand the problem, users and optimal solution iteratively
- Note that 'vague' sticky notes might need to be further specified after more research.



Further resources

To support you in diving deeper into the individual elements presented in the canvas, different tools can be found in the following pages. The figure on the right gives an overview of the suggested tools present per element.

Next to that, guidance is provided to help you move through the canvas, e.g. by specifying an intention prior to using the canvas and checking conditions to move from the 5xWHY to the HMW phase and from the HMW phase to the ideation.



Problem Deep Dive Map

5xWHY

In the 5xWHY exercise, you will ask ‘WHY?’ as often as possible and counter the answers with follow-up ‘WHY?’-questions. The exercise can be used to understand a problem in depth. Asking multiple WHY questions allows to dig deeper towards the core of the problem to gain new insights and to understand the situation from multiple perspectives.

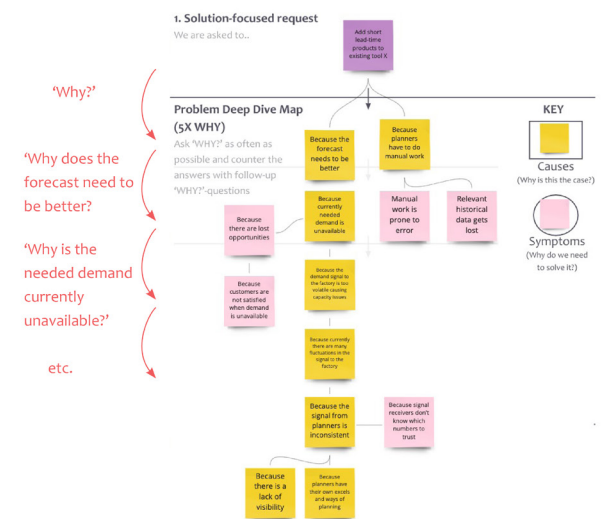
Tips:

- Asking 5 times WHY does not guarantee that you get to the core of the problem, keep asking WHY until you discover the real cause
- Go beyond the symptoms that are obvious!
- Write a question mark to indicate knowledge gaps and assumptions. Iterate the exercise after acquiring more information.
- Stop when the answers no longer make sense, then explore another branch or problem.

When to move to the next step on the canvas?

It is recommended to rely on your gut feeling to decide if you can continue to the reframing step. To have some guidelines, you can start by asking yourself the following questions:

- does the map reflect my/our current knowledge (and knowledge gaps) about the problem?
- do I/we have enough information to make conclusions?
- does the map spark opportunities to continue with?



Problem Deep Dive Map

Specifying your intention

The first WHY question formulated after the solution-focused request influences the direction of the problem deep dive. You can dive into multiple pathways of course, however when there is a certain underlying goal related to the project, it might be helpful to align on the intention when multiple people are involved.

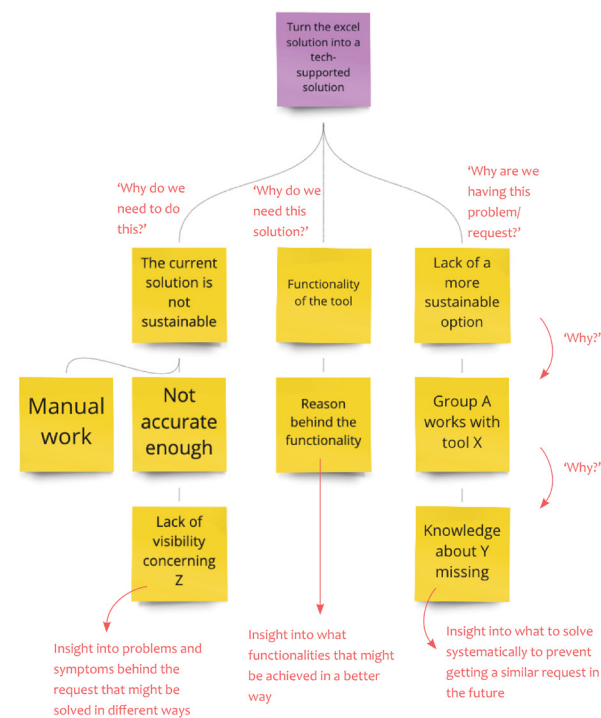
To guide the problem deep dive towards a specific direction consider asking the following questions as an initial WHY:

- Why do we need to solve this? (direction: problems and benefits)
- Why do we need the proposed solution? (direction: functionality)
- Why are we having this problem? (direction: origin)

On the left, you can find an example of the different pathways following different initial WHY questions.

Other WHY questions that might spark your thinking:

- Why does this problem exist?
- Why is this the case?
- Why is there no solution for it yet?
- Why are we solving it now?

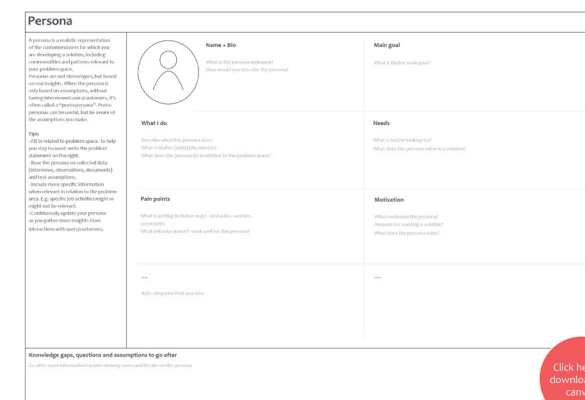


User

Persona

A persona is a realistic representation of the customers/users for which you are developing a solution, including commonalities and patterns relevant to your problem space.

Personas are not stereotypes, but based on real insights. When the persona is only based on assumptions, without having interviewed users/customers, it's often called a "proto-persona". Proto-personas can be useful, but be aware of the assumptions you make.



Reframe

How Might We..?

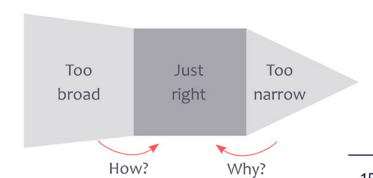
By formulating a problem or challenge in a HMW question, the format indicates that multiple solutions are possible (hovv), that ideas generated have the potential to work which creates a safe space as you don't have to be certain about the effectiveness (might) and that the problem can be solved as a team (we). This helps to scope the idea-finding phase.

When it comes to the HMW question, there is no right or wrong. It is recommended to rely on your gut feeling to decide if the question fits the problem. As a check: a fitting HMW question sparks ideas!

Tips:

- Mention the user in the formulation, e.g., ‘HMW support the user to...’
- You can use the template on the right to evaluate if the HMW question might be too broad or too narrow. If the question is too broad, you can try to ask HOW? If the question is too narrow, you can try to ask WHY?
- Keep away from specifying solutions in the HMW statement. If your HMW question is solution-oriented, ask WHY questions, e.g. ‘WHY do we need this solution?’

A fitting HMW
question sparks
ideas!



Brainstorming

Through brainstorming, many different ideas are generated that come to (the team's) mind. Mutual exchange and active listening to build on collected ideas is encouraged.

- Rules:**
- Quantity over quality
 - Build on the ideas of others
 - No prejudices, postpone judgment
 - Fail often and early

- Tips:**
- It is recommended to timebox the brainstorming exercise, e.g. to 5 minutes. In this way, people are encouraged to keep brainstorming further than the initial ideas that come to mind.
 - After brainstorming many ideas, take several minutes to cluster the ideas and to reflect on patterns.
 - Record the insights.
 - You can increase creativity by introducing negative brainstorming, figuring storming or body storming.

Brainstorming		
<p>Through brainstorming, many different ideas are generated that come to (the team's) mind. Mutual exchange and active listening to build on collected ideas is encouraged.</p> <p>Rules</p> <ul style="list-style-type: none">- Quantity over quality- Build on the ideas of others- No prejudices, postpone judgment- Fail often and early <p>Tips:</p> <ul style="list-style-type: none">- It is recommended to timebox the brainstorming exercise, e.g. to 5 minutes. In this way, people are encouraged to keep brainstorming further than the initial ideas that come to mind.- After brainstorming many ideas, take several minutes to cluster the ideas and to reflect on patterns.- Record the insights.- You can increase creativity by introducing negative brainstorming (e.g. "BMW counteract users," instead of "BMW support users.") or figuring storming (put yourself in the shoes of a specific person)	<p>1. How Might We..?</p> <p>Reflect on the current BMW question (problem/challenge)</p>	<p>3. Cluster</p> <p>Cluster the generated ideas</p>
	<p>2. Brainstorm</p> <p>Write down as many ideas as possible in a limited amount of time</p>	
<p>4. Findings</p> <p>Record your findings, insights and concluded next steps</p>		

Click here to download the canvas

Extra canvas modules

If you miss certain elements in the canvas, you can add extra modules. Either choose one of the modules below or label the empty module to your liking.

Place the module on the left of the canvas.

Stakeholders

User

Role

Main goal(s)

Pain points

Role

Main goal(s)

Pain points

Dependencies

	Downstream	Upstream
Systems		
Technology		
Data		
Resources		

Click here to download the modules

