A Minimum Viable Design Sprint and a look beyond the UX rim

Sandra Riedewald, Isaak Tsalicoglou, Chris Udell

Product Management, Proceq AG

sandra.riedewald@proceq.com, isaak.tsalicoglou@proceq.com, chris.udell@proceq.com

Summary

A design sprint is a proven way to bring fresh momentum to a project team finding itself in challenging and complex situations. However, time and resources are limited in a project's everyday routine. So, what to do in case you have only one day for something requiring five days? Our answer: just start in the best agile sense with a "minimum viable design sprint", and then build it up iteratively. In this article, we report on how it was possible to run a successful design sprint in only one day, and on how this first step triggered a chain reaction that resulted in a pilot. However, the transition from departmental waterfall thinking to an agile and comprehensive approach presents some hurdles that must be overcome. The key to the solution was interdisciplinarity: focusing on trade-offs with the Quality Function Deployment (QFD) method familiar from Lean Product Development has proven to be a valuable complement to the UX toolkit.

1 Introduction: why a design sprint?

It's a situation that is not unheard of in product development: in an important subproject, things are not progressing as expected, the stakeholders are getting impatient, and the team would stand to benefit from some fresh motivation. And, of course, there is also a tight schedule to take into account. Our company develops and produces technologically sophisticated measuring instruments. We are a medium-sized company that asserts itself through technological and business-model innovation, to provide Swiss-made quality products. Therefore, with a new development, much is at stake.
In this situation, a design sprint is a proven and well-known method to bring momentum back to project and team. Knapp, Zeratsky and Kowitz (2017) have developed the design sprint method for challenging situations exactly like the one we faced then:

"There is a lot at stake. Time is short. We are stuck."

Veröffentlicht durch die Gesellschaft für Informatik e. V. und die German UPA e. V. 2018 in
S. Hess, H. Fischer (Hrsg.):
Mensch und Computer 2018 – Usability Professionals, 02.–05. September 2018, Dresden.
Copyright (C) 2018 bei den Autoren. https://doi.org/10.18420/muc2018-up-0290

Why is a design sprint the perfect tool in such a situation? A design sprint:

a. lets you achieve best results in the shortest possible time,
b. requires only resources (people, knowledge, tools) that are already available, and
c. provides quick answers to critical questions.

The last point is, in essence, the principle of "fail fast" of entrepreneurial, iterative and incremental product development: rather learn from an early, insightful and richly educational failure with an inexpensive prototype, than have to deal with a late and therefore far more expensive true market failure with an implemented product.

2 Part 1: How can we run a design sprint in one day?

A design sprint requires a considerable amount of time and resources: a handful of experts are required full-time for five days. It is a quite high investment, with the expectation that its return is commensurate and visible. Both the company, as well as the UX designer, were new to the design sprint method – and, if you are new to something, you will tend to be cautious. On the other hand, in a situation where there has been no progress with the established methods (or lack thereof), the willingness on all sides to try something new tends to be greater. It was agility, where we found a matching methodical solution to our needs: start with the minimum, learn from it and continue iteratively – in other words: skip the five-day event and start with a minimum viable design sprint of a single day instead. This turned out to be the challenge: is it really possible to carry out a real design sprint in just one day?

2.1 The original design sprint plan

A design sprint is made for challenging, complex situations: "The greater the challenge, the better the sprint.", as Knapp et al. put it. Therefore, it is supposed to be run within five successive days, out of which an entire day is devoted solely to understanding the problem.

Day	**Content**
Monday	Establish a common understanding of the context. Identify the critical areas and choose the questions you will be able to tackle in this sprint.
Tuesday	Expand and explore the solution space for the selected question(s).
Wednes-day	Quickly decide for a solution. There may also be several alternatives. Create storyboards for the test. Specify the details according to the storyboard.
Thursday	Implement the solution(s) as a testable mockup / prototype. Different aspects might also be tested with different mockups / prototypes.
Friday	Run tests with real users. Observe and learn.

Table 1: Original design sprint plan

When trying to come up with a *minimum viable* design sprint, what is the minimum so that the design sprint is still viable? What can be scaled down, what can be accelerated, and what can be left out entirely?

It's worth looking at the essential features of a design sprint (according to Knapp 2016):

- Concentrate on one question, turn off all the disruptions of everyday working life.
- Bring together experts from different fields relevant to the question.
- In a concerted action, bring all the important information about the question to the table and make the big picture understandable and comprehensible for all by externalizing and visualizing.
- Take sufficient time to analyse, to be able to focus on the right parts of the problem (avoiding the "I had a solution, but it did not fit the problem.").
- All, not just the experts, are looking for solutions to bring in their different perspectives to the research question.
- Impose time pressure to prevent over-long disputes.
- A decision maker must attend, at the very least during the decision rounds.
- Replace guesses about solution suitability by a test with real users.
- It is not about finding the perfect solution within a few days, but getting rich answers to critical questions. This also works with the non-optimal solutions and the risky solutions with high potential.
- Alternate between individual and team work. This prevents priming effects in solution search and evaluation – and a democratic approach is not always necessary and sometimes even inefficient, or even obstructive.

2.2 How to condense a design sprint

Beyond the book theory of a design sprint, finding the time to spend an entire week represents the biggest challenge for all experts and decision makers involved. Asked how to run a design sprint in less than 5 days, Knapp et al. advice to squeeze the sprint’s first three days into two, resulting in four days. But they strongly advice against doing it all within one, two, or three days. But life is rarely like it is recommended in books. The internet is full of examples how people tried to do a design sprint within less than 4 days.

For condensing and shortening a design sprint, you basically have these options:

a. *Concentrate* – Less than five days in a row, full days, but accelerate the activities. This is what Knapp et al. suggest to do, but with a limit of 4 days. Depending on the question’s complexity, one might squeeze it even more. Hence, one day seems to be an option for more simple research questions.
b. *Split* – Full days, but have at least one day in between the building blocks.
This approach takes advantage of the fact that some of the activities are better to be done by individuals or pairs. Examples are detailing out and building prototypes. This is a good way to go when having the experts available for just two or three days in a row.

c. *Distribute* – Some days in a row, but mostly half of each day is dedicated to the design sprint. You will choose this way when you have other important tasks ongoing in parallel that you cannot interrupt. It is obvious, that in a situation like this participant's minds are not fully dedicated to the sprint's subject.

These measures may be combined. For example, Diehl (2016) proposes to concentrate and split the design sprint to be able to run its first part in just two days (figure 1).

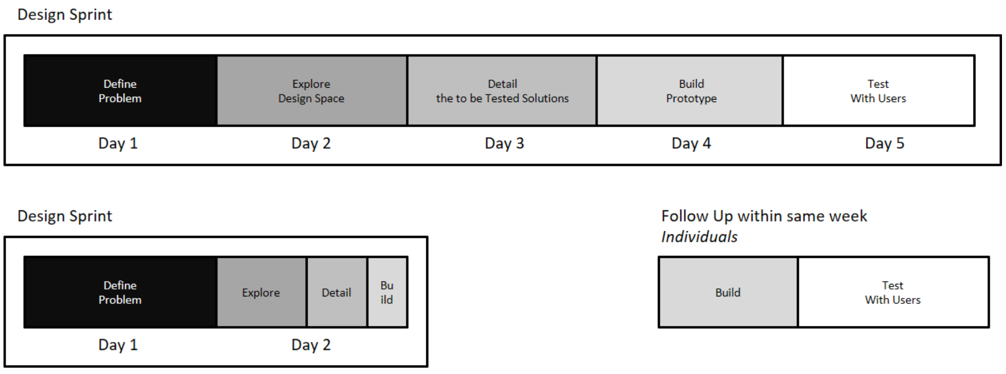

Figure 1: Condensing a design sprint.
First row: Original five-day-version after Knapp et al. (2017)
Second row: Two-day version after Diehl (2016)

It is noticeable that the reduction and condensation take place consistently from the end: the execution phases are run partially or completely after the sprint, and the solution finding and refinement are reduced by half. Only the problem definition remains in its original length. This makes sense, since a common and thorough understanding of the problem in the beginning is the foundation of tackling the right questions. And, complementary to that, the later activities (i.e., detailing and prototyping) involve a higher portion of individual work.
It is obvious that a targeted cut to just one day will also affect the problem definition phase. A reduction in the analysis, however, carries the risk of not selecting the right problem to begin with. In addition, the other three phases must be further summarized. This carries the risk of overlooking large parts of the solution space.

Our path was to make another split to shift some of the activities into the preparation of the design sprint. As we also distributed (in the sense of c, above), it was essential to this idea to closely involve the project team in the preparations and for follow-up work. (figure 2).

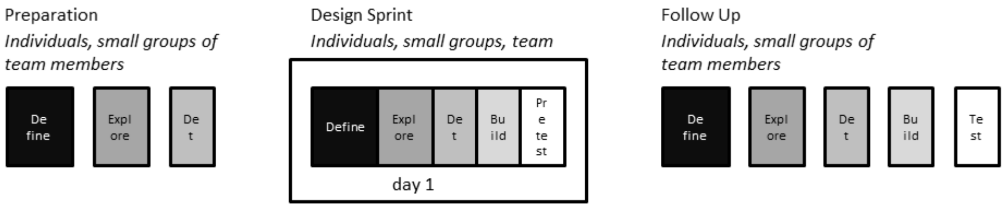

Figure 2: A design sprint in a single day

But still the idea of a design sprint is all the experts or team members coming together and doing the core activities in block. So one needs a way to make sure the relevant parts of the design space are addressed. One day is fully fine for a simple question. But because we had to tackle a complex question, we chose to increase the level of abstraction. As a result, basic branches in the design space can be explored without wasting time by getting lost in the details. What is meant by this, one can see from the example pictures given below. The downside: the resulting prototypes were too rough to test them with real users, so we ran a pre-test instead.

2.3 Sketching hardware handling and ergonomics

The sprint aimed at handling aspects of a new instrument's hardware. In definition phase, we had made sure this really is a crucial feature. Now, in exploration phase, we faced the task of how to quickly explore hardware by sketching.
Ideating includes a lot of sketching and presentation of mock-ups as an inexpensive means of conceptual prototyping. Sketching makes it easy quickly externalise one's ideas by making others able to see and share them. Especially in a condensed design-sprint workshop, efficient sketching is crucial. Hence, we looked for a way how to sketch hardware handling quickly, roughly and detailed enough for evaluation and decision.
Early on, we realised that the classic models of serious play and alternatives (figure 3) were mechanically too stiff for our purpose, preventing accurate posing of the user models to demonstrate the ergonomic benefits and disadvantages of various handling concepts.

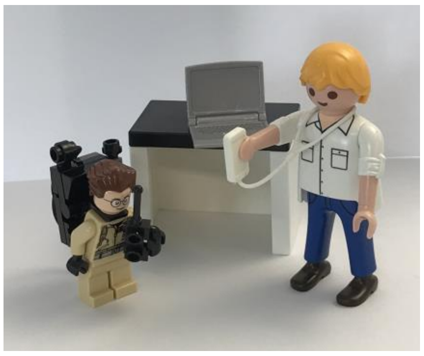

Figure 3: Classical Serious Play® and alternatives' models were too stiff for demonstrating hardware handling

We searched for alternatives and found them in artist and Barbie dolls. Due to the mechanical flexibility of the joints between their limbs, they allowed exactly for the degree of body flexibility we needed. The hardware itself was sketched via modelling clay for the parts and with lengths of wire for cables. The different functional parts of the device were colour coded. This equipment let us quickly explore the design space (see figure 4).

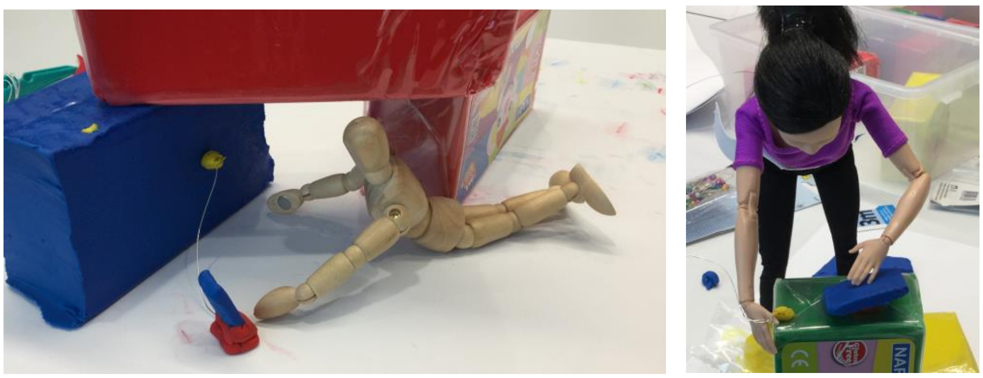

Figure 4: Sketching hardware handling and ergonomics with artist and Barbie dolls

2.4 Acceptance through transparency

When applying a new, somewhat unusual method with creative elements, one factor is crucial for success and wide acceptance: transparency. We talked with other colleagues beforehand, and shared our goals and plans with them. This resulted in them providing us with valuable ideas, thus helping us to prepare the design sprint and making sure we would explore the most important regions of the design space.

The design sprint workshop itself took place in a room with glass doors. The beneficial side-effect was that everybody passing by could glimpse what we were doing. This points to a secondary advantage of extensive externalising and visualising in a workshop: even for a hurried observer, the thorough work and dedicated enthusiasm of the workshop group was observable and intriguing.

By the way, it was not difficult at all to engage the project team itself into this kind of serious play. Having fun with solving challenging puzzles is a very motivating thing to do. It helps to create a shared mental model of the challenge, unites people into a team to address it, provides a sense of purpose, and thus also increases the likelihood of strong insights, i.e. of a successful workshop (figure 5).

Figure 5: Concentrated, hands-on design workshop atmosphere

The exploration phase finished with four basic ideas. For the detailing, each team member assigned themselves to one of them and built a quick prototype focussed on handling aspects. With these prototypes we ran a pre-test, testing each other's prototypes. The pre-test involved first to attach the device to the body or respectively holding it in hand, and then simulating some real-use movements.

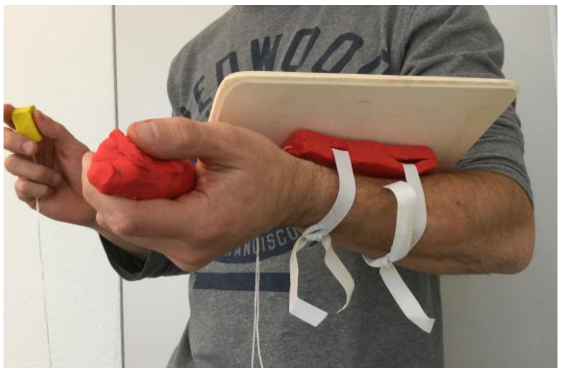

Figure 6: A prototype in the pre-test

2.5 Results

The workshop resulted in four concepts that would need to be detailed in the next steps. Furthermore, the pre-test provided us with important knowledge about the crucial properties that a handling-prototype has to fulfil for the test. And last but not least we came out with a highly motivated team and a common understanding of the subject. So, the workshop was a big success and we felt ready for the follow up: exploring the technical details and refining the prototypes to prepare the tests with real users - typical individual experts' work, one might assume.

However, quite soon we felt somehow stuck again. What happened?

3 Part 2: A look beyond UX, with Quality Function Deployment (QFD)

The design sprint fulfilled the expectations placed on it. However, after a short time we had the feeling that the follow-up steps would be somehow slow and sticky. We took the time to stop and analyse our work from a meta-perspective leading us to the following causes:

(1) Any change in how things get done within an organization has the potential to cause misunderstandings. As a team, we had overlooked that the mindset of iterative design and development was just getting introduced in our company through collaborative, workshop-led methods. Consequently, some of the stakeholders still had the misplaced expectation that our design

workshop would result in a fully-specified solution to be “frozen” for implementation. However, the four rough ideas coming out of our workshop were just the outcome of the first iteration of a longer discovery process, and still needed further investigation of details.

(2) The team found itself stuck in an outdated the-expert-will-do-it pattern. In the past, it had been standard operating procedure to hand rough concepts and precise specs off to an expert development engineer or draftsman, and expect them to elaborate the implementation details, often in isolation. With the new approach, we needed to conclude our first iteration with a learning phase as the preparation for the next iteration, so that the entire team would contribute brain power and insights towards problem-solving. And, as there were many aspects to be considered, we needed an efficient learning phase.

To address these two causes, *we needed a supporting method* that would allow us to
(1) build a communicative bridge within the team and towards the project’s stakeholders
(2) to guide us in a structured way through the knowledge to be acquired for the next iteration.

We found the answer to both of those requirements in the method of Quality Function Deployment (QFD, also known as the “House of Quality”), already well-known from Lean Product Development.

3.1 QFD, explained

Quality Function Deployment is an old but trusty method supporting product development. It enables its users to associate requirements with specifications of a system, as well as to capture knowledge about its specifications’ influence. In general, a QFD matrix helps to associate what must be developed with how it could be implemented. It also helps a team to prioritize the aspects to be taken in account during development. This is true both statically (by focusing on the requirements) and system-dynamically (by exploring the interactions and trade-offs between specifications).
Through customer and stakeholder validation of requirements, their relative importance towards realizing a desirable value proposition becomes apparent and visible on the QFD matrix. Through requirements analysis and the presentation of their relation to specifications in tabular form using the QFD, the relative importance of various specifications for fulfilling the requirements becomes clear. And, finally, through the increasing investigation of the engineering-system trade-offs between different specifications, the inherent complexity of the engineering problem(s) to be solved is made clear.
Most importantly for our project: working with a QFD matrix helps put vague impressions and strong opinions to the test against an agreed-upon scoring system.

More rarely, the QFD matrix can also serve as a first line of defence against opinionated project stakeholders or team members who might have the tendency to parachute into the problem and start shooting new concept ideas from the hip, without taking in account the complexity already addressed by and the knowledge stored within the matrix. This way of using the QFD matrix can be beneficial in terms of avoiding unnecessary confusions and chasing after moving targets. However, this is only true, provided that the team does not resort to it as in a defensive mode impervious to external input. In other words: a QFD matrix works well to deter potentially destructive or distractive stakeholders from passing their opinion as facts – and it works against the best interest of the project team if it is used as an excuse for not listening to feedback, treating stakeholder input as annoyances, or defending the work put into it as perfect. After all, application of the QFD method itself should not be a one-off occurrence in the team; and perfect knowledge of requirements and specifications is most definitely wishful thinking, especially when attempting to innovate and challenge the status quo of the target market.

As such, the QFD method is a very efficient generator of productive conflict between team members on diverging opinions, experiences, understandings, etc. By "putting the fish on the table", i.e. by laying bare all the assumed interconnections of the system to be developed, it is far more likely that a product development project team and its stakeholders will identify issues that could otherwise remain unaddressed.

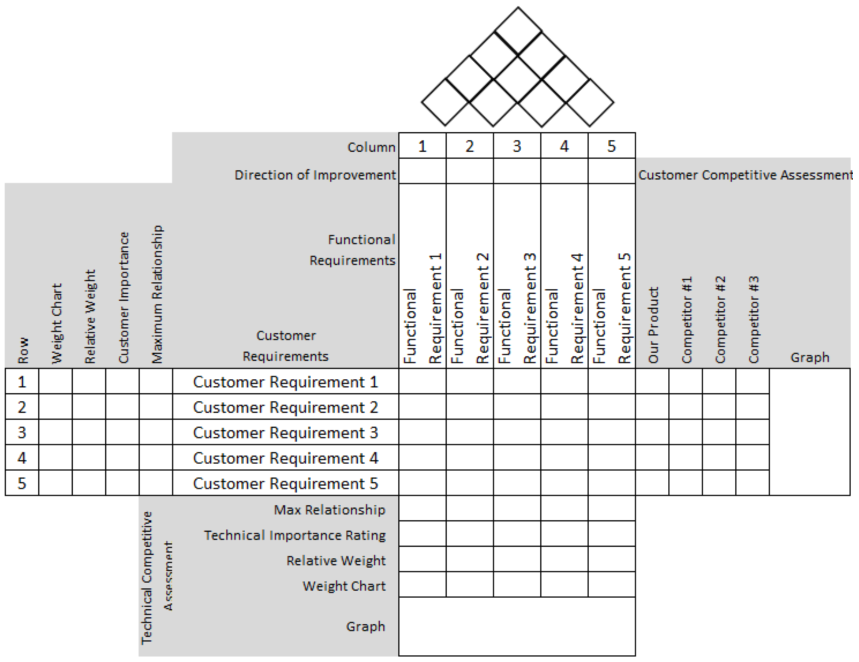

Figure 6: An example QFD matrix

3.1.1 Customer requirements

The first step in our use of the QFD method was an elicitation and documentation of the customer requirements. What features and benefits do the prospective customers of our product want it to deliver? How important is a certain requirement compared to the rest? And what exactly does it imply for the design of the system and its value proposition?
If, for example, you would like to develop a handheld measuring instrument to be used in industrial environments, then users expect it to be rugged and lightweight at the same time.

However, how important is ruggedness compared to weight? And what exactly does ruggedness imply, really? Does it mean achieving the IP67 rating? Being shock-proof or drop-proof? Surviving explosions? Or, perhaps, does it imply something more specific, such as winding up with a rope?
Discussing about customers' requirements, defining and relating them, leads directly to a detailed and shared understanding of the product-problem to be solved. With the lists with customer requirements and their importance available and understood by all, one is able to rate the competitors' products against each other and against the numerous conceptual variants the team comes up with in exploration phase of the design workshop.

3.1.2 Functional requirements (what some call "specifications")

These are the translations of the customer requirements into something one can really sense and measure objectively. For example: "a comfortable car seat" is fuzzy in its definition; comfort is subjective, and results from various other system parameters, such as the stiffness of the material used for the seat cushion. Stiffness can be described with a specific value and measuring unit, and is therefore objective. The QFD method is a practical way for teams to structure their "requirements analysis", and understand the criticality of various specifications.

3.1.3 Filling in the QFD matrix

The matrix is filled with all these relationships between its rows and columns. One customer requirement usually translates into more than one system specification. And, vice versa, one functional requirement is often influenced by more than just one customer requirement.
For example, the requirement of "ruggedness" for a device does not only influence the various choices of materials to be used, but also the different shapes the device could take, as well as its weight and volume. A delicate shape will not be as rugged as a compact one. The shape is also influenced by the requirement to be ergonomic. A lot of shapes might be in accordance with ruggedness, but not all of them are ergonomically comfortable.
Filling in the QFD matrix is a key part in the application of the QFD method. Its main benefit for the team is that it drives everyone to think about both the UX of the envisioned product and the technical parameters that will impact it.

3.1.4 Filling in roof of the House of Quality

It is not uncommon for a customer requirement to be linked to multiple specifications, which influence it in opposite direction. It's also typical, e.g. due to the laws of physics, for two specifications to correlate or be explicitly interlinked, sometimes using very precisely-defined formulas. And, most importantly: because a typical specification impacts multiple requirements, it is usually the case that 1) it is impossible to optimize the product across all of its requirements, and 2) that any great product is a result of cleverly resolving the conflicts arising between its separate requirements and specifications.

3.1.5 Good to know

The application of the QFD method benefits from the various viewpoints of participants that shed light on the complexity of the system from vary different angles, thus requiring true diversity of thought to achieve a holistic understanding. At the same time, the decomposition of a highly complex product continues down to the level where an expert will be able to establish low-level trade-offs that the rest of the team would never be able to understand or influence positively without the knowledge and experience of an expert. Therefore, some parts of the QFD method must be done in the group, whereas others are better done by few or only one. Still, on the entire-product system level, participation of the entire project team is required, possible even of the key stakeholders of the project. For example: listing and discussing the customer requirements and their prioritization is definitely a whole-group activity, as it aims to create a common mental model of the problem to be solved. Analysis of the problem, however, must in the end be followed by the synthesis of its part-concepts and part-solutions, based on a well-founded understanding of the trade-offs dominating the system. As such, the whole group together must review the completed QFD of the product system.

This systems-engineering-inspired approach works, because it:

- Triggers the right discussions within the team,
- Makes to-be-decided-upon conflicts and trade-offs clearly visible, and thereby points to where more R&D might be needed.
- Translates words into easy to be compared numbers that make a quantitative analysis for decision-making possible (if the team also analyses sensitivities).
- Is easy to understand by all, and at various levels of detail and degrees of technical understanding of the subject-matter.
- Enforces the expectation of thoroughness upon the team’s work.
- Provides a level of detail that approaches the “full specification” that some would expect, without the misconception that the outcome is set in stone and unchangeable.

3.2 Outcomes of QFD and the next design iteration

Doing a QFD takes time. And at the beginning, it had felt to us like going a step backwards. But it was exactly this halt that enabled us to go forward again. For our team, the QFD method supported our minimum viable design sprint, resulting in:

- An in-depth understanding of the requirements, and of the key questions to be answered through further team work.
- The company-wide acceptance as a model project following dynamic yet also systematic approaches for exploring the problem and its design space.
- A list of evaluation criteria for the prototypes.
- A rich morphological box with relevant design features for the next iterations.

Putting it in the scheme of a design sprint, it could be seen as a second iteration consisting solely of an extensive definition phase.

The most crucial point, however, was the bridging of the communicative gap and the new reputation as a model project. Every UX designer is familiar with the challenges that come with trying to introduce new methods and processes, with overcoming the challenges of different ways of thinking. The QFD helped to build the common ground.
The following iteration then comprised 3 days with 7 participants, who were able to explore

the design space on a deeper level with the QFD evaluation criteria as a guide and the morphological box as a source of high-variance building material, in an evolutionary approach.

4 Lessons learned

Methods are not an end in themselves, but aid in achieving certain goals in certain situations. If for a given situation and goal there is a certain method that would fit but can’t be fully applied, force-fitting the method onto the situation is unlikely to work, or do so efficiently. There are some alternatives: choosing another method, adapting the method at hand or inventing a new method. It is a good opportunity to reflect upon the essence of a method. In the end, what is needed, most of all, is the ability of the team to demonstrate both methodological flexibility and creativity.

By demonstrating both, our team concluded that yes, it is possible to conduct a design sprint in one day – and it did so in the following ways:

- By being iterative and starting with a minimum viable design sprint.
- With the project team as an integral part of the preparation and follow-up.
- With pragmatism and courage.
- With transparency and openness, including a heightened tolerance for conflict and direct feedback on the subject-matter level.
- With self-monitoring and reflecting.
- With a method from Lean Product Development, that complemented and deepened the design sprint’s definition phase.

We would like to encourage you to consciously look beyond the borders of the UX landscape, and combine design approaches with methods of e.g. Lean Product Development. We are adamant about this, because UX always takes place within a larger context and in interdisciplinary cooperation; and because, unsurprisingly, different approaches complement each other, much like different personalities and skills sets within our project teams.
The Quality Function Deployment method proved to be a powerful complement to a design sprint. It is worth the effort, especially when a subject-matter is complex with quite a lot of trade-offs between requirements. Not only did the QFD guide the team with the level of detail needed to proceed, but it also built a bridge over the gap between the iterative and the waterfall approach.

5 Literature

Diehl, J. (2016). *Der Design Sprint im Unternehmen.* In: S. Hess & H. Fischer (Hrsg.): Mensch und Computer 2016 – Usability Professionals, 4. - 7. September 2016, Aachen.

Knapp, J., Zeratsky, J. & Kowitz, B. (2017). *Sprint: Wie man in nur 5 Tagen neue Ideen testet und Probleme löst. 2. Auflage.* München: Redline.

Authors

Riedewald, Sandra

After her degree in Psychology at the TU Darmstadt Sandra Riedewald got her doctoral degree in Cognitive Science at the University of Freiburg i.Br.. Since then, she has been working 14 years as a usability engineer and UX designer in technological companies. Starting with software, she is now entering the field of hardware. Currently, she is UX designer at Proceq. Her focus is on information architecture, testing and processes.

Tsalicoglou, Isaak

Dipl. Mechanical Engineer, MBA, Pioneer of agile process development in corporate environments. He works as head of product management at Proceq. He has got ten years of experience in leading change pragmatically across functions, and in establishing Lean/Agile principles, data-analytical approaches, collaborative systems-thinking, DFSS methods, requirements engineering and workshop-led collaboration in the R&D and product development activities of industrial goods and technical service products.

Udell, Chris

Holding a MPhys in Physics and a MBA in International Management, he worked as an engineer, team leader and product manager. After his start in nuclear industry, he had been working for ten years in energy sector. Currently, he is product manager for metal at Proceq.