Digital tools for positive impact: the playbook

How to develop your digital product to create lasting change

interfacewerk

Table of contents

Introduction

1 Introduction

Saving the planet is a decision, not a research goal. For decades, this statement would have been false. But today, it is true, and it will forever be true. We already know everything we need to know to save the planet – we're just not putting it all into action. So, why aren't we just doing it?

It's a complex issue that by its very nature defies simple answers. But on a metalevel, there are still only two major reasons holding humanity back: 1) lack of political will and 2) information deficit.

Nobody reading this will need to be told how important it is to shift political realities and decisions towards a sustainable future. This will, however, not be the topic of this book.

Countless other books, podcasts and essays have been created by way more knowledgeable people on that topic.

What I can, however, give a new perspective on, is the information deficit problem.

Even for the most diligent of us, who want to make all decisions considering their sustainability impact, be it in their private or in their work life, it is often hard in practice to do so.

Even seemingly simple, every day decisions – What fruit do I buy today? – quickly become an information nightmare, once you want to really dig deep to find out the real answer, not just an apparent, superficial answer that might be on the packaging.

And those are only the simplest decisions. Other questions are way more complex:

What kind of transportation should I use for my vacation?

How should I package my wares?

Where should I buy my supplies?

Where should I donate money to maximise impact?

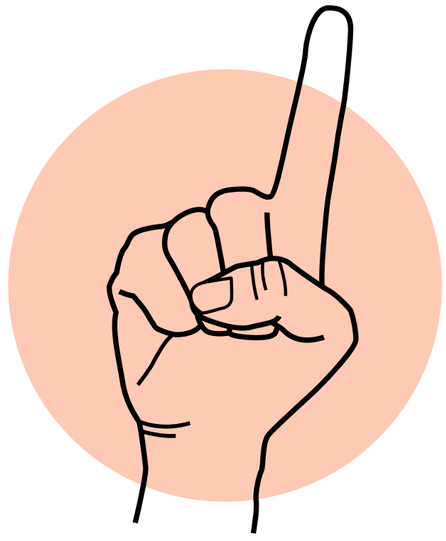
Where can I invest my savings so they also have an impact?

How should I adjust my company's strategy to make it future-proof?

All of us struggle with these questions, because all of them don't have simple answers. Why don't they? It's merely an information deficit, nothing else. It is completely possible to figure out which fruit is the „greenest" or which organisation has the highest sustainability impact. And many smart people are working on that, even coming up with useful results. Lifecycle assessments of products are becoming more common – the information is being generated. But is it also distributed? In general, it isn't. We lack the tools to take these, often complex, answers to simple questions and make them accessible. Currently, this information is accessible only to the scientists and researchers themselves, and a selected few, who, like me, spend enough time and brainpower to try to make sense of the research results.

What we are lacking, however, is a digital tool "environment" (for lack of a better word) that provides simple and easy access to all this information. Despite numerous startups, nobody has succeeded in creating even the superficially simple app that can scan a barcode and show the climate footprint of the respective product. Numerous CO2 calculators have been created to assess the climate impact of our private lives – often with 100+ questions that less than 1% of those who start, ever finish.

In this book, I want to explore the reasons why we don't have those digital tools or apps, and what we all can do to get them.



This book is written especially for people who are interested in building their own impactful digital tools – whatever their size and scope might be.

Based on my own experience building digital tools for impact (which I have done as a startup founder as well as as a service provider), I want to share mistakes that I and others have made and provide clear steps and recommendations that lead to an impactful digital tool. You should read this book if you:

- Are wondering how to start the journey towards an impactful digital tool.
- Are already in the process of creating impactful software, but are not yet doing user research.
- Want to create something with real, lasting impact rather than just a few minutes of fame.
- Are struggling to secure funding to grow and scale your business.
- Encounter problems demonstrating your product's value to stakeholders, investors and customers.
- Are finding it challenging to measure and communicate the impact of sustainable products.
- Have difficulties building and maintaining a strong team with the necessary skills and expertise.

We start from the birds-eye view, by first assessing the goals of any such project, and zooming in step by step, so you never lose touch to the humans the tool should serve, and maximize the probability of having real impact.

Figuring out what tool to build

2 Figuring out what tool to build

Obviously the most important step for any digital tool is the beginning. In the first few months after a new project is started, many decisions are made that have the potential to steer the whole project either in the right direction or on a dangerous path.

To get a great start, it's important to utilize methods and product development processes that have proven themselves useful in the past, then you can derive the next steps from there. Let me introduce to the concepts that have been the most successful for me, my startups and my current company, interfacewerk, so far.

2.1 Covering your base

Building a digital tool for positive impact, is, in essence, the same as building any other type of digital product. Your goals might be different (impact instead of or alongside profit), but the core mechanisms of how a product is conceptualised, built, distributed and adopted remain the same. For example, like any other product, your digital tool needs a value proposition and a product market fit.

This should be great news, because it means that all the methods and materials that were developed to create more valuable digital products can also be used to create more impactful digital tools.

We recommend using a structured approach to understand the base on which you're building a new tool. The model we've had our biggest successes with is the Product Field (https://productfield.com). With the Product Field, you can get a good overview about what your idea and team already have available, and where there are gaps in your knowledge or skills.

Within a day or two, you will be able to grasp this bigger picture. In figure 1, you can see a quick overview over the Product Field. Using it, you will not only evaluate your idea and the resources you have to bring it to life, but also the key value that it brings to your market.

And even if that value is primarily for the planet, you're still gonna need to sell it to your customers and users, otherwise nobody will use your tool.

“Positive impact can mean many things.
For me, the most important unifying aspect in this wide spectrum is that the things you work on have a positive impact on the future of humanity.
That’s still a very broad term, but so are the UN sustainable development goals.
There’s a plethora of impactful things one can work on.”

The first things you will notice once you've filled out the product field, are the areas where you're lacking information. Maybe you have no idea yet how you could reach your customers (distribution), or you don't have the skills to build the solution yourself (production).
Very likely you will only guess the motivation of your users at this point. And that is ok. Because you now know what you need to research. We'll cover how to resolve these in the next chapter.

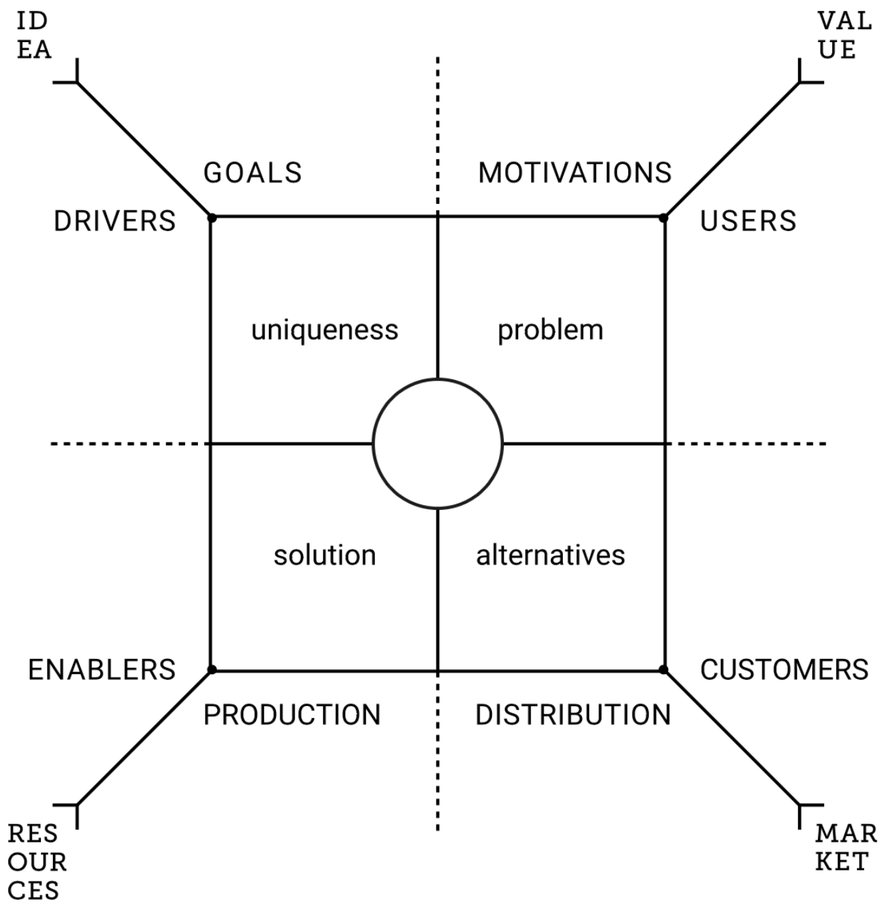


FIGURE 1: The product field provides a great canvas to see in which areas your idea is strong, and in which it is lacking.

Customer vs. user

A user is someone who uses a product or service, while a customer is someone who pays for a product or service.
In some cases, a user and a customer may be the same person, but in other cases they may be different (for example, in the case of a free app or service, the users may not be paying customers). Some digital tools have no customers, but only users, since they are free to use or at least provided for free.

2 Figuring out what tool to build

The other thing the product field will help you to evaluate, is the actual impact of your tool. It is touched on by many areas of the Product Field. The most important elements that influence your impact are:

1. What goals do you have? More ambitious goals lead to higher potential impact. It's good practice to put goals in terms of a concrete unit, e. g. reduce plastic usage of SMEs in the transportation industry by 50%.
2. How much impact does the problem have? Solving bigger problems or problems at scale leads to more impactful solutions.
3. How unique is your idea compared to the alternatives? Something unique can have a higher impact than something that already exists in some form.
4. What motivations do your users have? Impact only happens through actual, real world usage.
5. How are you going to distribute it to your customers? More or better customers lead to a higher impact.

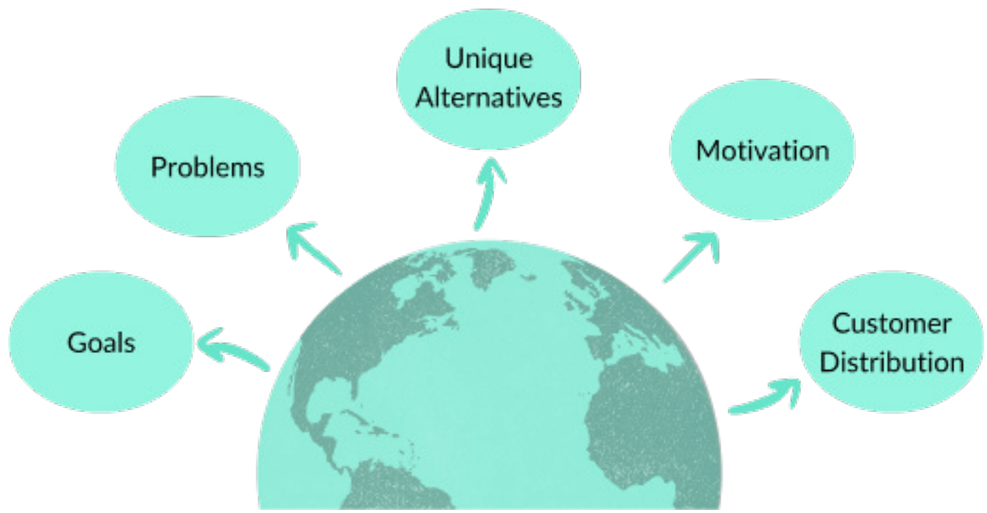


FIGURE 2: The five elements to measure your impact are goals, problems, unique alternatives, motivation, and customer distribution.

Maximising impact is one of the ideas of the Product Field, and this structured examination will allow you to at least get a sense of the potential impact of your idea. Put as a rough formula, your estimated impact would look like this:

Product Field Formula

=

Problem Size (e. g. in tons of CO2 or any other unit you care about)

× User Motivation (in %)

× Solution Quality (in %)

× Uniqueness (in %)

× Market Size (no. of potential users)

× Distribution Percentage (% of potential users reached)

Other than that, please also consider the plentiful resources provided by the Product Field team, which will also guide you towards possible next steps.

We've now almost covered our base: we know the potential impact of our tool, we know what skills we have and what knowledge we lack. Now, let's finish the job and check our idea against the real world.

2.2 Check against the real world

It pains me to say this, but most tools and products are still being built – today – without sufficient real world checks. The thrill of building something is so exciting that we often forget to check its usefulness and impact before we build it. I call this problem premature implementation.

Having an idea and the resources to build it don't seem to be the limiting factor, so to check against the real world, we're gonna examine the customers and users, to create a product market fit.

First, make sure you understand the problem and motivations of your users really well. To get there, we recommend a human-centered design approach, which I will now outline in a bit more detail. The idea behind this approach is to stay in close connection with real users throughout the whole product development process, in order to ensure the product/tool will create real impact. To get the process off to a proper start, it has to start with people.

Books like Lean UX outline how this is done in more detail. Fortunately, again, we can rely on existing best practices and don't have to reinvent the wheel here. We need to make sure that we don't just care about the user's problems, but also to uncover the potential impact that the problem's solution could have.

Concretely, this process starts with user research, usually in the form of a context analysis, in which one tries to understand the user's daily realities and problems they face, as well as surrounding information (context) that informs the potential usage of a product.

2 Figuring out what tool to build

In addition to the standard questionnaires, we recommend taking along the following research questions in this step of the process:

- What other people do our users influence and what impact do those people's actions have?
- Are there organisational barriers (e. g. hierarchical decision-making) that limit our users impact?
- Are there technical barriers that limit our users impact?
- Are there political barriers that limit our users impact?
- If our users problem could be solved 100%, how much impact would that have?
- Can the problem our users face be expressed as a number (e. g. tons of CO2)?
- What is the background rate of the impact that would happen even without your tool?

These allow you to look into the areas where your tool might need to support them most.

Is efficiency their biggest problem? Focus on making things easier and quicker.

Are there organisational barriers? Help them present results to convince their superiors. Are they working on an impactful problem at all? If not, maybe the tool can free up time so they can work on more impactful stuff.

One question that has special relevance to our success is the last one, about the background rate of our planned impact. Let's say, for example, you want to build a tool to reduce plastic generation. Now, there are already efforts underway, without your tool, to reduce plastic production.

So if the maximum impact of your tool would be a reduction of, say, 4 million tons, but the background rate (that would happen without your tool) is already 2 million tons, the potential impact of your tool is only 2 million tons. Why? Because even though there were 4 million tons reduced with your tool, 2 million of that would have happend anyway, so you can't take credit for them.

This comparison to the background rate will make sure that you don’t fool yourself into thinking you have some impact, when you actually don’t.

Once you’ve understood your users' problems and the im potential, you’re ready to think about the next steps.

2.3 Human-centered vs. impact-focused

In our approach, we've highlighted that working with humans to do real world checks of our ideas is essential. One might ask, if impact is our goal, why do we focus on humans, not on impact itself? The reason for that is simple: humans are the ones creating impact. If we don't help our users create impact, we're not creating impact at all.

This is why, if you want digital tools that create real impact, you have to work with the people that harbour that potential for impact, and help them realise it. If we wouldn't already have developed all the technologies we need to stop climate change, that might be different.

In that case, focussing on research alone might be the most beneficial approach. But that's already been done. What we need now is to change minds, tweak culture, re-design systems — and help people to get to work!

We especially need to help people to understand the information that's already there, and derive correct, concrete, impactful actions from that information. That is what any tool that wants to make an impact, should focus on.

And that is where human-centered and impact-focused can go hand in hand. Impact is at the core of your project goals, and you've set up those goals so you can measure your impact.

The human-centered approach will then guide your feature decisions on a path towards reaching your project goals.

The human-centered approach is known by many names, and all of them have slightly different methods, focus or outputs. For example, many people talk about UX Design, Design Thinking, user-centered Design or just Design. User-centered product development is the umbrella term, which is why we use it here.

The mindset and approach are more important than concrete methods. And that is, I think, what any team that builds a digital tool for impact needs to take to heart:

> *"Put the humans that create impact at the centre of your development process, and you will maximise your impact."*

Depending of the domain of your digital tool, different methods are best applied. We'll examine two very different case studies in the following chapters. But before we do that, let's make sure we have a good team to do so!

2.4 Gather your team

Using the Product Field, look closely at drivers, enablers and production. They can show you if your team is lacking key roles or knowledge that's needed for success. In short, your team needs at least these key roles (keeping in mind that one person can fulfil multiple roles)

1. A person taking care of the business and financial side of things (CEO, CFO, or a project lead with budget responsibility)
2. A person exploring and representing the user's perspective (e. g. a Head of UX)
3. A person understanding the technological challenges (e. g. a Lead Architect or Head of Development)
4. A person structuring the development process towards the goals (e. g. Head of Product or Product Owner)
5. A person providing access to your target group/market (e. g. Head of Marketing)

If you lack any of these roles, they can usually be filled. Local startup or impact hubs might even have a team or a digital tool to help you find the right people.
If you can't expand your core team to accommodate all perspectives, I would advise you to seek consultants who will ensure you against failure in any of these topics. Most often I've seen project teams that had either no Head of UX or no Head of Product. Having those roles covered with external consultants, at least for the beginning of the project, will set you up for success, because they will, in essence, use their experience to reduce your risk of failure – or at least reduce your risk of wasting your time on one path, when a better way forward would be available.

Of course, not all consultants can fulfil that promise, so here are a few ideas on how to find the right fit for you:
They listen to you first, then listen some more, then ask good questions. Be wary of those who talk too much about themselves.
They understand and want to work towards your goals, first and foremost, and always adjust concrete methods to that end. Be wary of fixed mindsets and processes.
They bring experience, but aren't already too invested in the domain details of your area of expertise – you're trying to get a fresh perspective, after all.
Your mindsets match – you feel like they will be evangelists for your cause, and will use their network to get more support if needed.
They have proven their worth – look at testimonials, maybe even talk to previous clients.
To illustrate the above approach, let's examine two case studies that make all this theory more concrete.

2.5 Case study: a company impact dashboard

Dashboards are interesting animals. They are usually created on a whim, based on what data one already has in some database, and then creating decent-looking figures and graphs to visualize that data. This tech-driven process leads to a tech-driven result, which is usually completely useless to the humans who look at it.
A useful dashboard answers the questions that deciders (humans!) need answered in order to make informed decisions. So you have to start the process by uncovering those "data questions". This is done best with qualitative interviews, since the number of people who'll use the dashboard later is very limited. Interviews will help you figure out what they actually need – which, be careful, might not be what they say they want! You might need to interview only 3-5 people to uncover the main data questions that the dashboard will need to answer later.

Here's a list of questions you might ask during the interview:

- What decisions do you make every day?
- What impact do these decisions have?
- What information do you consider?
- How do you use this information to make those decisions?
- What information are you lacking?
- How would that information guide your decision?
- If you'd ask a colleague for the information you need, how would you ask them?

In these interviews, it's important to use plain, natural language, no tech speak. Because if the terms you are using are too technical, you'll be thinking in the solutions that you know the technology can provide, not in problems that people actually need solved. Focus on the problem in this stage!
You'll then have a list of data questions, that you can even structure and prioritize based on the information in the interviews.

Now it's time to prototype some ways a dashboard could answer those questions. Before we do that, one quick word of caution: there are lots of best practices for dashboards (google that!), which will help you not make mistakes others have already made. A skilled designer working with your team can shortcut this process for you.

The most important thing when prototyping is to keep things really quick and simple. Don't invest too much time into a single prototype, otherwise it will be emotionally hard for you to trash it when it doesn't work. And that's the key thing: build prototypes, then check if they work. As a starting point, use pen and paper. It's quick, easy to make, and easy to trash. To come up with your first prototype, write the most important data question on the top of your piece of paper. Then try to come up with a way to show the information that would answer this question and allow your users to make a decision. After you have 3-4 ideas visualised, check them with your team: can they answer the question based on what they see? If yes, check with real users: can they do it? Importantly, don't ask them if they like it.

Make it specific: show plausible information in your prototype, and ask them to make a decision. Can they make that decision? If yes, you're in fine-tuning mode for that visualisiation. If no, then you need to come up with different ideas. Iterate that process until the results work (= users can make decisions). This usually takes only 3-4 iterations. Repeat this process for all the other data questions.

Now, that's the hardest task done. Now let's build it! This is probably much closer to the expertise you as a reader already have, so I won't go into much detail. The important thing is: you might need to gather new data in order to enable the visualisations that you came up with. That is normal. Since you've started with the humans and uncovered their requirements, it would be almost weird if you already had all the data ready. So data collection needs to be adjusted first, then data processing (to create the information you want to display), and then building the actual dashboard. There are many tools and libraries to help you build a dashboard – no need to reinvent the wheel.

2 Figuring out what tool to build

After building and creating your first release (which could be only one, important, visualisation as the Minimum Viable Product, MVP, which is sent out via email every week), make sure to stay in touch with your users. Are they actually making the decisions based on the information they now have? Are they failing? Are they missing something? Are new users not understanding what they see? Take this feedback and improve. Even initially unhappy users will become happy users if they see that you listen to them. After a while, feedback tends to converge and the dashboard will likely go into a rather stable mode of operation, where everything works smoothly.

Please notice how humans, the real impact creators, have been at the center of this process from the beginning, and until after the release. Everything revolves around them, and the technology is a servant for their needs.

2.6 Case study: an app for checking the carbon footprint of products

Why would a person want to know the carbon footprint of a product? Simple: in order to decide whether to buy it – or something else. The corporations that provide the every-day products of our lives have a huge impact on the climate, and driving consumers away from wasteful products and towards carbon-efficient or carbon-neutral products would help a lot. We have to acknowledge, though, that only people who can afford these choices will be in our target group. First, this app will need to be built on a lot of knowledge, even a structured database with many products is needed.

That's why this kind of app needs the support of sustainability experts that help figure out the key scientific questions, first:

- What is the carbon footprint of a specific product?
- Can the carbon footprint for a product be estimated if it's not available from the manufacturer?
- How can this data be collected, integrated and kept up-to-date in one single database?
- What are the types of products where this decisions matters most (i. e. has the biggest impact)?
- What are the types of products people would happily switch, if they knew the most climate-friendly option?

To answer these questions, user research would not be the best choice, since they are scientific questions that have already been answered by researchers around the globe. There are, however, questions that concern our users, that we will have to figure out doing user research:

- What do people already know about the carbon footprint of products?
- How much time are people willing to invest to make purchase decisions?
- Are people willing to take action, or are they mostly just curious?
- How can an app encourage action?

2 Figuring out what tool to build

So the first step for our project would be to basically do a scientific meta-analysis with our research questions, and then add our own user research (mostly via interviews) to answer our starter questions. Since we aim to provide an app for many people, we have to methodically move into more quantitative and statistical methods quickly, in order to cater to our full target group. That means quickly building a decently scalable MVP, for example a quick website that can be packaged as an app.

The important part here is that we can measure how users interact with our MVP, otherwise we won't know where to look for improvements. That means that basically all actions of our first users take must be tracked, e. g. via an anonymised tracking tool like Plausible. Then we can quickly learn what users actually do with our app and sense areas for improvement. This is only possible, of course, if we manage to find new users. Our circle of friends and family, which is a good starting point but might also be biased, will be exhausted quickly.

We can solve this through marketing (possibly a simple website with good SEO will be fine in the beginning), by buying test users via an online service or by visiting local usability test events. Alternatively, most bigger cities have usability test events or meetups, where you can gather data and feedback in person. Those can usually be found by searching for "usability test meetup [your city]". The important point here is to work up some numbers, gather data, and enter into a sense and respond cycle.

The sense and respond cycle is very simple:

1. Use our tracking tool to identify roadblocks for users on the way to our goals.
2. If necessary, do interviews to get ideas how the roadblocks could be resolved.
3. Implement the changes and release the new code.
4. Rinse and repeat.

2 Figuring out what tool to build

Once the roadblocks become smaller and less important, it's time to think about a bigger release, and maybe make the app look nicer as well, because that's not a priority in the beginning. First, it has to work, then, it can look nice. People are not going to use an app that looks nice but doesn't work – you wouldn't wear an awesome looking pair of pants if you couldn't breathe when they're zipped up.

If we haven't already done so, now's a great time to make sure our tracking and statistics reflect not only steps towards our goal, but also our actual goals. We should make sure we can ask the following questions using our statistics tools:

What actions did people take based on information given?

Are those actions temporary (one-time) or sustainable longterm (behaviour changes)?

Can the statistics of people's decisions be used to change corporation's with wasteful products?

What gets measured gets done, so we have to make sure we measure what we want done to reach our goals. But now, since we're iterating and tracking our progress, we're set up for the success of our carbon footprint app.

Building the right tool

Setting goals, discovering problems, iterating solutions – we've now been able to figure out the most important points for our own tool.
In our case studies, we've already hinted at the actual building of the tool, which is the other most important point in the creation of our tool.

Because no matter how good the idea and the concept, it can only have real world impact if we actually build it, and build it well.

As you probably know, software projects are akin to construction projects, in the sense that they almost always take much more time and money than was originally planned.

That is because most software projects are not run in a way to ensure maximum impact for the time and money utilised. Through countless successful and failed projects under my supervision, I've learned the key points that make a project stay on track.

Fortunately, we've already laid out the groundwork for a successful tool, by doing the following:

- Setting up a clear vision and clear goals for the project.
- Involving our users from the beginning to ensure a problem-solution fit.
- Iterating our solution towards our goals.

There are some tools that can be built exactly as designed, without much iteration during the development process. That can happen mostly if the scope is limited and it was possible to build and evaluate a fully functional prototype before development.

3.1 Starting to iterate

In most cases, however, iterations and continuous improvement need to be part of the development process. The main reason for that is, that, depending on the scope of your tool, it might take anywhere from 3 months to 3 years to develop it into something that will reach your goals – and maybe much longer until you get close to the product vision

During that time, there is bound to be new knowledge about the users and the market, even the underlying assumptions and the science will change in that timeframe. All of that has to be taken into account, otherwise you'll build and release a tool that's already outdated.

To avoid that, we generally don't make long-term feature plans or detailed feature roadmaps. Planning as far as we can see, and then sensing changes and adjusting for them, is the right way to go. So, to be able to start development, you need only four things:

1. A development team (in-house, freelancers, external, …)
2. A UX and product team (in-house, freelancers, external, …)
3. A product owner, i. e. the person who prioritises features and plans the development iterations with the development team
4. A goal for the first release (e. g. "solve problem X for our users")
5. Detailed designs and technical requirements for the first, most important features.

Let's go more into detail for these five.

3.2 A Development Team

Obviously, you need people to do the work you need done. Ask yourself these questions to start:

- Do you already have developers in your team?
- Can they cover all the required areas (usually backend development, frontend development & operations/delivery)?
- Do you have a very experienced developer (10+ years experience) who can set up a smart software architecture, i. e. the structure and relationship between the different parts of the software?

Depending on your answers, different paths might be best for you to proceed. For example, if you have a team that covers all areas, but lack a software architect, you can hire that role yourself, find a fitting freelancer or hire a service provider to fill that role for you.

If you don't have any development team and nobody with tech skills, it's usually best not to hire freelancers, since they need competent management, but instead look for a service provider who can offer you the whole package, to be your development team.

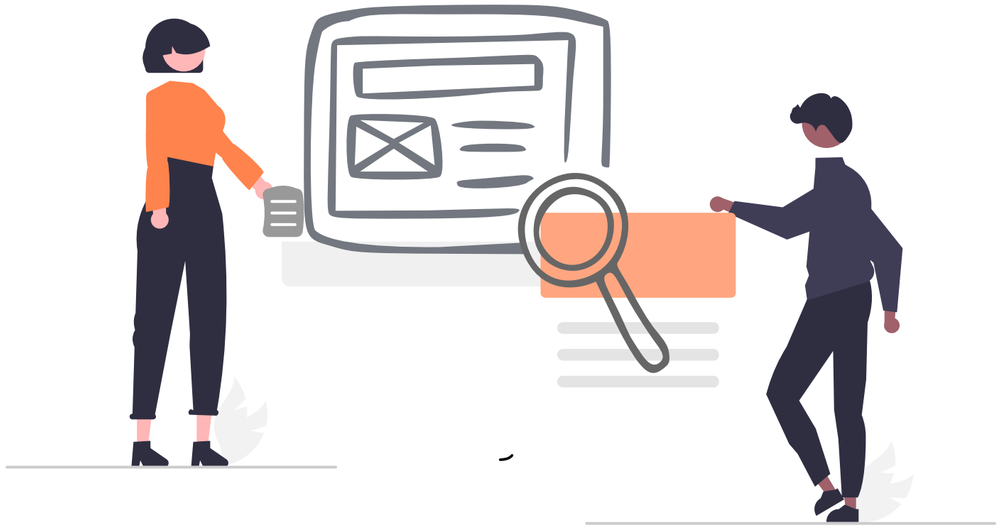
I've seen many teams try to start this without a capable architect, and I would strongly advise against doing that. The amount of technical debt (i. e. problems in your code structure that will cause severe problems later on) can become huge within the span of a couple of months. Better start from a solid foundation rather than tare the foundation down after a couple of months because it's not holding up any more.

3.3 A UX & Product Team

Without solid UX support, a development team will usually work on the wrong things, because their decisions are not guided by facts about the user needs.

A UX team will support the product owner with facts to make good feature decisions, create concepts for those features to make sure they have a good structure, and evaluate those concepts with real users to ensure they will have the intended effect in the real world. Questions to start:

- Who in our team will stay in touch (at least every couple of weeks) with real users?
- Who has the research skills to make sure we don't fool ourselves when talking to real users?
- How will we avoid coding things that we later have to throw away (or at least reduce the probability that this will happen)?



Most teams will lack UX / product knowledge and experience, and will either have find someone to join the team for these tasks, or hire someone external to support.

While it is a key role in order to create a successful product, it's also one that can be easily added from an external service provider or through hiring a freelancer, since UX people are generally very inquisitive and can adjust quite well to work with different teams.

Here are the most important questions to ask potential UX team members, from my experience:

- Do you have experience in working with real users? How do you make sure you discover their needs, not their wants? → If they don't, they might not be UX people, but UI people, who will become important only much later in your project.
- Do you enjoy sharing & discussing concepts & designs while they are work in progress? → Classically trained designers often don't want to share their work early. Avoid those in the beginning, since they will slow prototyping down.
- How do you balance user needs, impact goals & technical feasibility as a designer? → They should care not just about the user, but also about the goals that your project has and if what they design can actually be built. If they see their role too limited, they will limit what you can achieve.
- How do you ensure that your concepts & designs will help people have real impact? → They should show some scientific literacy in their answer, maybe even mention critical thinking skills or examples where they applied them. Designers need to think like scientists, otherwise they might delude themselves – and you.

One of the core tasks of the UX & Product team is to help the product owner make decisions, so let's talk about this role next.

3.4 A product owner

This can be a Chief Product Officer (CPO), Head of Product or an actual Product Owner. The key is that there is a single person who facilitates and coordinates all product decisions, in order to create a clear path towards the goals and the vision.

If this person is missing, or this person is missing key information (e. g. when there was no user research done), you risk that your team spends weeks discussing features and feature details instead of moving forward and iterating a solution. A well-supported product owner creates product clarity.

3.5 A goal for the first release

The first release can be many things, for example the implementation of one key feature in order to then test it with limited users. It can also be a first public release, maybe as a designated beta version, with limited or unlimited access. Both of these could be defined as MVPs (Minimum Viable Products).

Don’t only think about the first public release, though. Depending on your overall goals, the first public release might be more of a MMP (Minimum Marketable Product), a much more polished product than an MVP.

To start developing, developers need to know the first goal they are working towards, and it should not be a huge goal far in the future. It should be a relatively small goal that is achievable within at most 2 months.

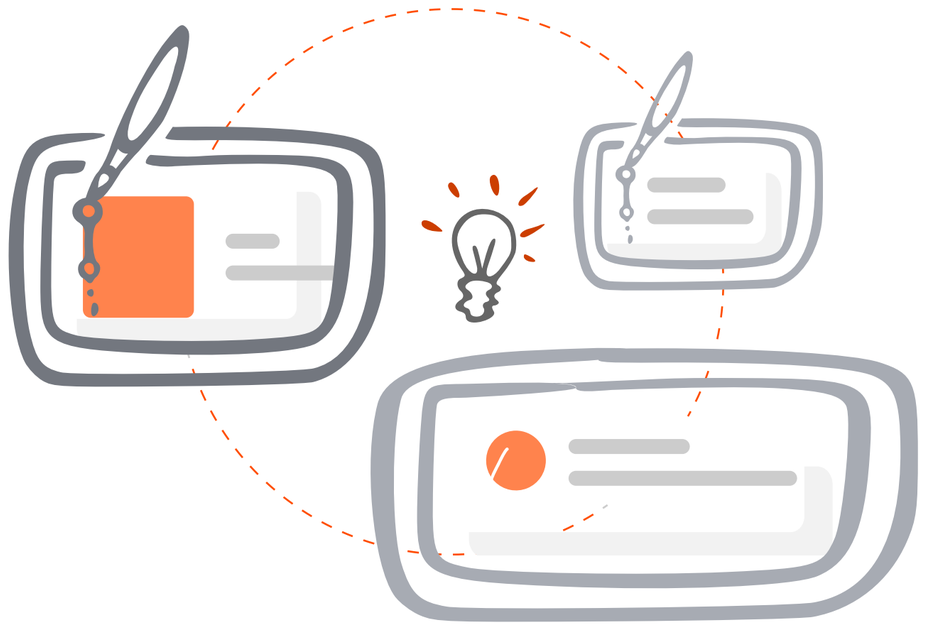
Not only does this create a regular sense of accomplishment for the team, it also allows you to adjust goals after releases, with new knowledge coming it. In order to adjust goals, you need to create knowledge, and this knowledge creation should be part of the goal for a release.

Here’s an example: For our MVP, when want to implement feature X as designed, in order to then evaluate if our test user group can solve their problem Y in the real world.

3.6 Detailed designs and technical requirements

There are books written on that topic alone, so I'll stick to the basic "must haves" here. For each feature, the development team needs the following in order to implement it, which your UX & Product team can & should provide:

1. Layout and navigation: where does the feature live within the context of the whole tool? (Can be done via the design tool)
2. Screendesigns: how shall the feature look and what elements will comprise it? (Can be done in a design tool like Figma)
3. Interaction: what will happen to the user interface when the user interacts with the elements? (Can be done in a design tool as a click prototype)
4. Behaviour: what will happen in the background (invisible to the user) when the user interacts with the elements? What data needs to be processed and sent where?



3.7 Continue iterating

Once you've gotten into your grove by finishing the first "release", the same process basically continues, with two big caveats.

The first change is that, to continue iterating, you need to find a way for new knowledge to find its way into the development process. Otherwise, you risk to follow a static plan instead of reacting to change.

That will, inevitably, result in an outdated result. Your goals for the previous release should already contain questions for that previous release to answer, as we discussed earlier. The step that most teams forget, is that these questions don't answer themselves, and you should not postpone answering them any minute longer.

You have your release, now answer those questions. Everyone except the development team can work on that, but usually it's the UX designers who are methodically best prepared to do so. Because these answers cannot come from within your team.

The new knowledge must come from external sources, probably users (via user tracking, usability tests, …), otherwise your team will live in an echo chamber and simply reinforce views that it already held.

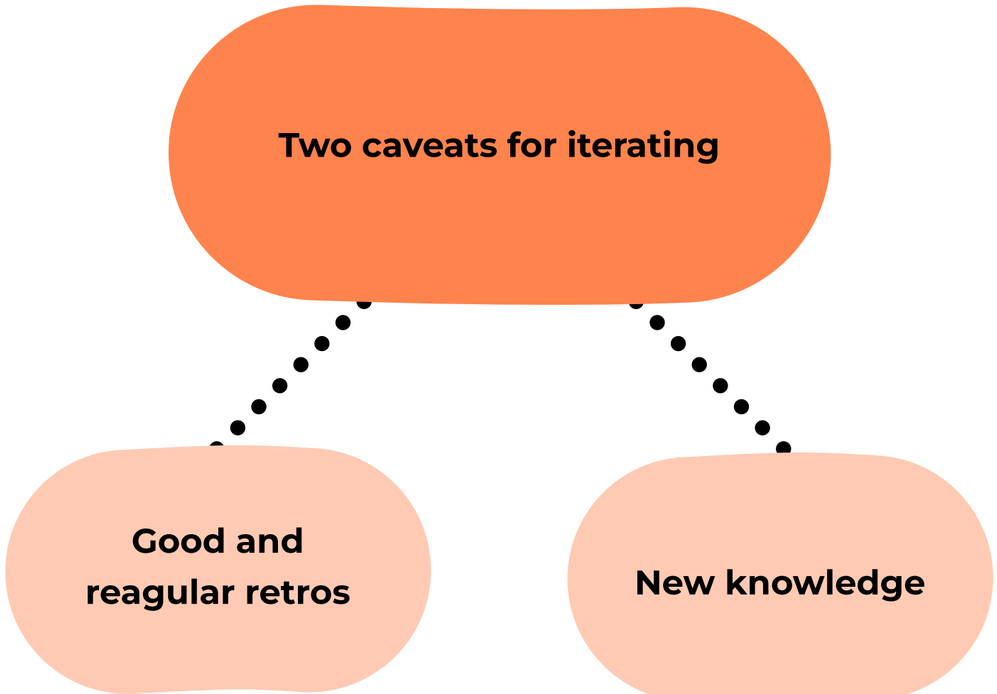
For some reason, the resistance against leaving the echo chamber is huge. Involving externals is often postponed, discouraged by leadership or simply forgotten. My hypothesis is, that this is because by checking for external validation, you allow for the possibility that you are wrong. And people don't like that. Except for those who want to succeed more than they want to be right. Please be one of those!

The second change means implementing everything that you learned by doing a retrospective. Again, there is much literature available on doing good retros.

But most teams just don't run them often enough or implement changes decided during the retro. So whoever is responsible for doing the retro (in a SCRUM team, that’s usually the SCRUM master), needs to make absolutely sure that the learnings actually find their way into the every day development work. In short, you need to iterate your process as quickly as your product.

If you don’t, the team will quickly feel out of control and passive, as if locked into a hamster wheel of more and more features, and will lose their intrinsic motivation. Keep them interested, keep them inspired by sharing everything with them and listening to their ideas how to improve product and process.

These two seem simple, but once the wheels of development have started turning, and new habits are being formed, they are exceedingly difficult to continuously implement. Yet they are critical to success.



Ensuring impact in the real world

4 Ensuring impact in the real world

Most teams reach a point, after a couple of releases, where they somewhat achieve their initial vision for their digital tool.

It's finished – or so it seems. If we were designing a sustainable cast-iron cooking pan, we might really reach that point. Digital tools and products, however, are never finished.

Never.

Once you stop caring for them, they quickly die. There are multiple reasons why digital tools need care and maintenance over time. Let's examine the most important and what you can do about them.

4.1 Evolution of technology & security fixes

You didn’t build your tool from scratch. You likely used programming languages and frameworks, that themselves undergo development.

They change over time, because the technical environment also changes. As computers become faster and can do different things, communication protocols change and become more secure over time.

Framework developers adapt their frameworks to that changing environment. Also, over time, security problems in frameworks or protocols will be found and fixed. To benefit from those changes and fixes, your code needs to be updated regularly.

Otherwise, you might lose all your users because your system is hacked or simply stops working because of outdated technology.

4.2 Changes in the user base

Over time, new and different people will start using your product. They bring different backgrounds and might have slightly different problems to solve.

Also, once the original problems were solved, your initial users might discover new problems to solve, and ask you to solve them. Keeping in touch with your users through feedback forms, surveys and user tracking will enable you to not fall behind your user's problems.

4.3 Ensuring impact

Having covered the two key issues that necessitate product care and maintenance, let's now get back to the core of our vision and goals. You want to make a positive impact in the world, that's why you started the project in the first place. But are you actually having an impact? Now, since a version of your tool is out in the world, it's time to measure and assess its impact, so it can be maximised.

Through previous steps, you should already know what the core steps are that your users need to got through in order to manifest impact. These steps need to now be examined, through user tracking, user shadowing ("looking over their shoulder"), usability tests and other methods. If you're familiar with sales or marketing, you know the term "funnel" already.

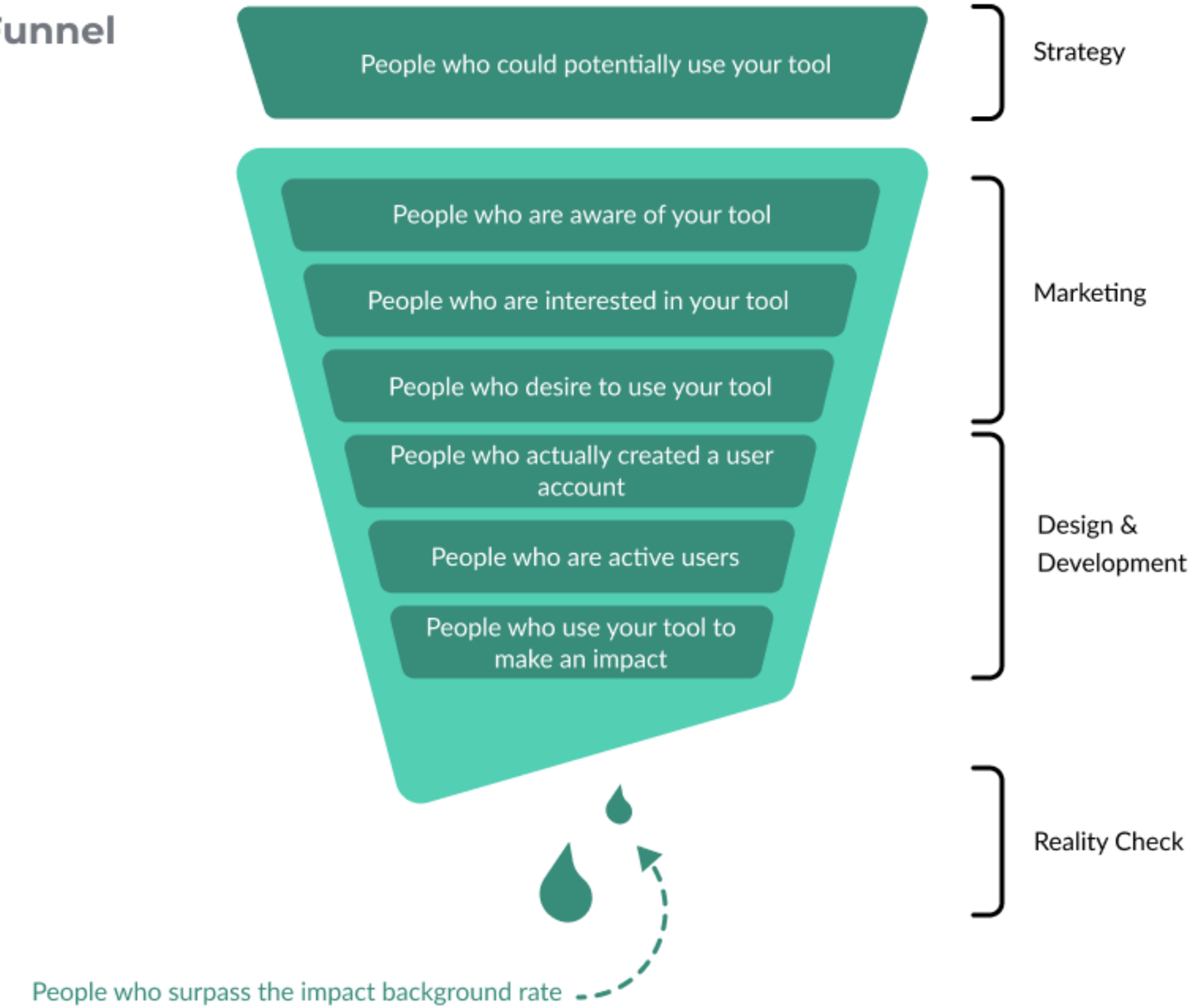
It's a progressive funnel or ladder, where your marketing prospects get closer to buying, going one step at a time. In each step, there is a reduction in prospects – there are more people interested in a product than those who will actually buy it.

One goal of sales and marketing is to maximise the ratio of people who finish the funnel based on the people who started the funnel.

Instead of a marketing funnel, we recommend doing your impact analysis with an impact funnel. Where companies measure turnover or profits, you measure impact. From each step to the next, you have a loss of impact. Here are the stages or our impact funnel.

1. People who could potentially use your tool
2. People who are aware of your tool
3. People who are interested in your tool
4. People who desire to use your tool
5. People who actually created a user account
6. People who are active users
7. People who use your tool to make an impact
8. People who surpass the impact background rate

8 - Step Impact Funnel



Now, these are a few more steps than the classic marketing example. But more steps allow for a more detailed analysis and more concrete actions. Because actions matter, not the analysis itself. The steps need to be separated in a way so you can derive actions from them. Usually, core features or major steps in your digital tool can be separated in funnel steps. Because once you've set up the funnel, and adjusted your tracking to represent that funnel, you'll be able to put numbers on the respective steps. How many people who look at your tool on its website actually start using it? There's a key percentage right there.

A guideline is, that everything below 20% from one step to another signifies an issue that needs to be resolved. Numbers over 50% are usually fine. Now, the statistics (quantitative methods) only give you an indication that you need to do something. What exactly, that's to be figured out with more in-person (qualitative) methods.

You can only answer the question "Why are people not …?" by talking to people – using the same approach as during our initial prototyping: first figure out the problem, then iterate solutions (with feedback from real users), until the numbers are good enough. Do that for each funnel step, and you'll have maximised the real-world impact of your tool.

Wrap-up

5 Wrap-up

In this playbook, we've outlined the approach of having real users, their problems and the impact of the problem's solution at the core of your concept and development process. This can ensure the success and impact of your tool in the real world, if you stay diligent.

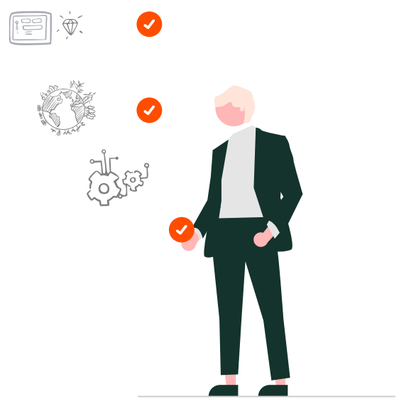
But it can also help you discover that the product goals you have are not realistic or hard to manifest in the real world – which is fine as well. Most ideas don't survive contact with the real world. Some can be pivoted and adapted to solve some real world problems, and some simply cannot.

The key is to have a process that allows you discover that knowledge, and to act on it, as early as possible. Pivoting before development starts is a huge time-saver. Adapting your vision to solve fewer or different problems will ensure more real world impact.

So don't be discouraged by setbacks or data that conflicts with your initial idea and goals – it's completely normal. As long as you're ready to learn and adapt, to sense and respond, you can create something awesome!

If you would like know what the next steps from here could be, check out the next pages!

If you need to process the information and figure out the perfect way for yourself, we wish you good success and keep in mind that we are there in case you have any questions or difficulties!



5.1 What can be a next step?

As you might have noticed, Sebastian wrote this ebook to give an outstanding and thorough overview of what it needs to create digital products with a positive impact. With this book, we aim to help NGOs, startups and SMEs that struggle with their product.

And what you also might have noticed is that this knowledge comes from real projects we did. Of course with real clients such as Systemiq, The Recycling Partnership, Moving GmbH, Project Climate GmbH...

So this is what we do as a company: Help other companies to succeed with their products. We follow our method, **the Product Impact Process**. However, we see each client, each product, each team as an individual situation.

And that's why a good, efficient communication is so important. So: Let's talk if you feel like you have any questions.

This would be the first step if you don't know where to start and how to develop or improve your digital product.

If you consider working with us, the next step would be our strategy workshop. As the name states, this is where we dive deep into your product strategy, vision, your users' view, your team structure, and the stakeholders.

Schedule a first call on our Website

5.2 Our method – The Product Impact Process

After you have learned and read about the process of digital product creation for impactful tools, your next questions might be: How do you solve it? Is there a process that you use to ensure all the mentioned points and how do you come to a good end result?

The simple answer is: Yes, we do! We call it the Product Impact Process. A method of 6 steps (usually; depending on the status quo) that we use fo all our projects.

At its core the Product Impact Process is:

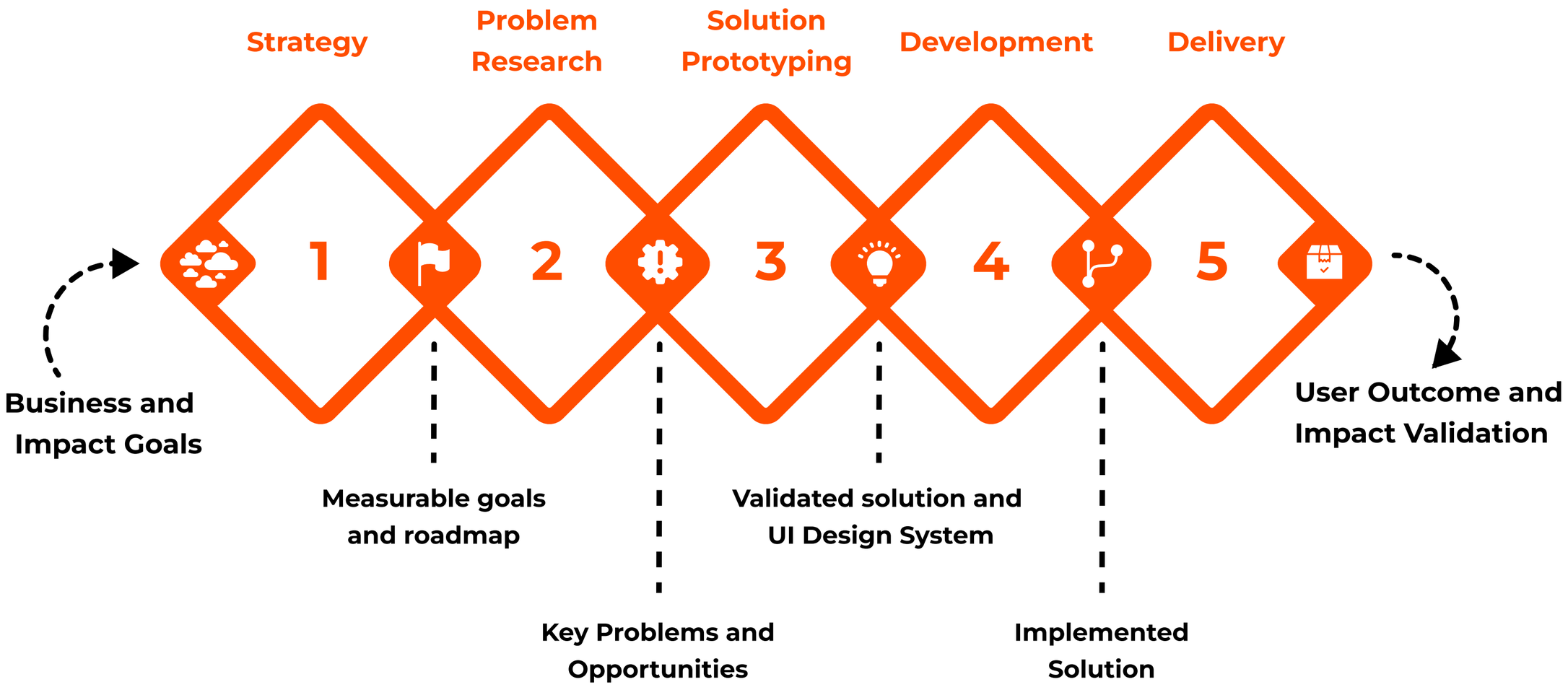
holistic - scientific - pragmatic.

We combine the main disciplines of product strategy, human-centered design, software engineering and managed operations.

And that is it – we follow the steps and ensure our clients have a digital product that has a great UX, creates impact, meets the business goals, such as more customers, the next funding round etc. and that can be iterated fast and maintained to state-of-the-art standards.

Check out our method on our website

The Product Impact Process



5.3 The author

Sebastian has a passion for reality, and uses the scientific method to know what's really real in a world (increasingly) full of fake. He wants to live the future and pull the present up to him, especially regarding working environments, teamwork, trust & collaboration. Meditation, mindfulness, music, and reading help him to focus and grow. Sebastian is all for taking over responsibility and acting responsibly, especially in regards to his family, society and humanity as a whole.

As the host of the podcast "menschzentriert", Sebastian loves to spread knowledge and talk with interesting guests. In his career, he has founded three startups and interfacewerk. His goal is to build a focused & human-centered working culture in addition to doing great work (as a team) on digital products.

Franconia is Sebastian's homeland and he lives at the edge of a forest with his family of four and works on making his family's old house future proof (net zero!).

Sebastian is fighting against the (proverbial, not literal, unfortunately) windmills of the old economy. He does not see the point in working on projects without being able to prove their worth/impact. Bad leadership (in all areas and on all levels) is the most frequent barrier he sees because of its immense stopping power.

5.4 About interfacewerk

It started off in 2014 as a Munich-based UX agency, helping their clients design, develop and improve the user experience of their digital products and services.

However, over time, the founders, Sebastian and Moritz, recognized that their work had more potential.

Namely, to not only improve the user experience but also create positive impact for society and the environment.

The company began to shift its focus towards using human-centered design and sustainable UX to drive system change.

interfacewerk is a full-service digital product agency that helps companies create impactful digital products.

They focus on using design and innovation to create sustainable and socially responsible solutions for their clients.

Among other long-standing customers, interfacewerk works with startups and NGOs (Non-Governmental Organizations), and mentors their projects – from the initial vision to the iteration of the developed digital product.

Here the Product Impact Process, developed and successfully trialed by Sebastian, ensures that positive impact becomes a calculable factor and part of the product vision.

We are committed to helping companies create digital products that are meaningful and make a real difference in the lives of users.