10CLOUDS

Digital Product Design

FOR STUNNING WEB AND MOBILE APPS

Index

CHAPTER 1

Basics

4 Introduction to the World of Design

Product Design Process

8 The Urban Legend of a Universal Project Process

14 Focus and Vision: the Keys to Cooperation

19 Design Workshops: Where Magic Happens

CHAPTER 2

Cooperation

25 Interdisciplinary Teams Create Better Products

28 Presenting UX Like a Pro

34 Brothers in Arms: UX and Backend

CHAPTER 3

Product Development

Design Goes Agile

40 Rocking Design Sprints at 10Clouds

46 UX Flow for Easier Team Cooperation

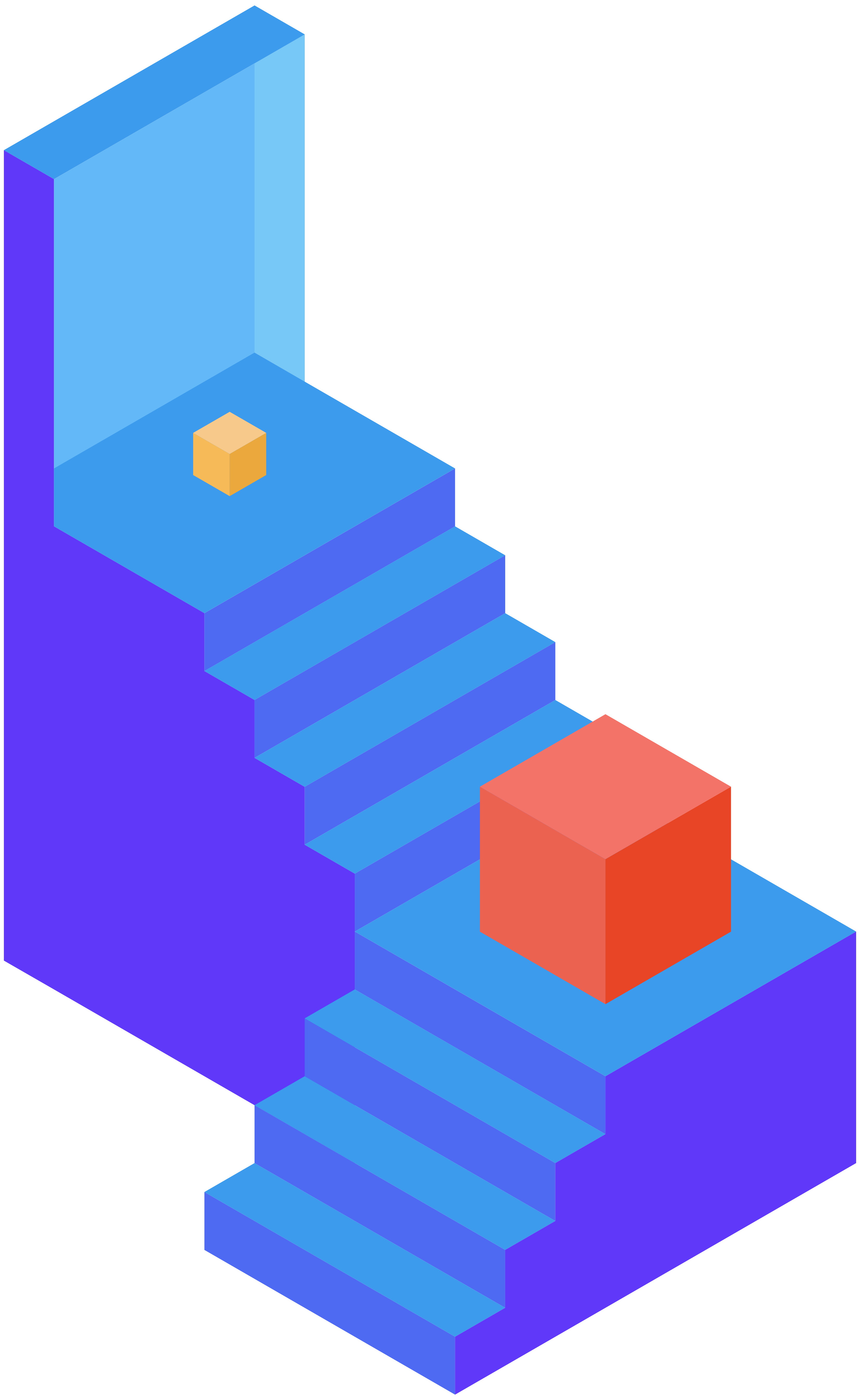
50 Trust and Empathy Make a Great Project Team

Iterations in Progress

56 The Awkward Secret to Valid User Interviews

61 Making Things Work with Usability Testing

68 Make KPIs Your Friends, Not Enemies



Basics

CHAPTER 1

Introduction to the World of Design

If you were to explain to your grandma what design is, how would you start? Explaining it to yourself can be hard, let alone someone not tech-savvy! Design is a very intangible discipline. Most people have problems with understanding the variety of outcomes it delivers. We all declare we get the value of design yet everyone expects something different from it.

The trick with design is that it meant different things during the software industry evolution. I would like to guide you through this story to help you better understand the plethora of meanings and dimensions applicable to design. Ready, steady, go!

Design as a Visual Outcome

Not that long ago, building a commercial product revolved mainly around technology and marketing. Most products were physical objects and the Internet only began to grow. Marketing was how well the product looked and how many features it had. The role of the designer was to create a beautiful packaging to sell the product.

For many business owners, this is still the way they see design. Just as an obligatory phase in a bigger process. The role of designers is to execute the brief and deliver work that meets specified business goals, and - let's be honest - the taste of the stakeholders.

On the other hand, designers have always loved to focus on the visual aspect of the product. Oftentimes, there was not enough communication between the business and designers. It lacked a shared goal and greater understanding of the context. And above all that, users are usually excluded from the equation.

So what can be done better? Get users involve, test, validate and create the process of constant improvement!

Design as a Process of Improvement

For many IT business owners, focusing on the user started with ergonomics and usability. Everyone was talking about creating products so that they were easy to use. The business started to be interested not only in the overall user satisfaction but more specifically in ways to improve their product. Design began to mean more than a pixel and the general "feel".

Now business understands that there is a need for users' advocate in the team. Designers started to be the part of the product creation process. They were evolving and specialising in different fields. The overall goal of designers changed to translating business requirements combined with user needs into tasks.

Sounds perfect, doesn't it? Great, but we can still get more. Invite designers to the table and get involve in design alone.

Design as a Collaborative Process of Improvement

Design Thinking for many is nothing more than a buzzword. For us at 10Clouds, it's an eye opener that makes you think about design as a constant collaboration and validation process. Design teams have now more power to lead innovation and suggest changes. Concepts like MVP (Minimum Viable Product) and Lean Startup are much more popular. The idea of testing assumptions as fast as possible is implemented in big companies, such as IBM or Facebook. Frameworks like Google Ventures Design Sprint are getting more traction.

There are businesses that understand the need to have some constant touch point with users and to use this loop for building a better product market fit. Analytics, together with user research, are important data sources used in the decision making process.

This type of design is a collaborative game that requires involvement from all sides. Business brings the requirements and opportunity. Development brings technical boundaries. Designers provide a framework to tie it all together.

It's the UX designer who should be a hub connecting all team members and helping them set a proper goal. Design begins to be a process where there is no single source of all decisions regarding the structure and visual aspects of the product. In UX, the team is responsible for forcing the change by design. There is a drastic need for design leaders within an organization.

Wrapping Up

The design is no silver bullet, but it provides a variety of tools that can help you improve your product. It's important to understand that it requires great collaboration, open-mindedness and, most of all, trust. Real design is iterative, and - against the popular opinion - it should be measurable. And the thing that is often missed: good design is both business- and user-centered.

I hope you'll find many answers to your questions about design in our ebook. It covers all stages of product design and shows you how strongly they are connected with product development and project management. Enjoy your read!

Jacek Suwalski
Senior UX Designer

The Urban Legend of a Universal Project Process

We’ve already mentioned the fact that Design balances on the verge of technical constraints, user needs, and business goals.

Now we can move onto how design works in organisations and products. I’ll try to cover the most popular approaches within an App Development community and Startups.

Blossoming in Agile

The Agile Manifesto brought a lot of changes into Product development. The shift from a get-it-all-done-at-once approach to a more let’s-ship-it-and-improve-it approach has added a certain leeway into design. We have now switched from keeping our fingers crossed that everything goes right to knowledge-based decision making.

Even though you can put an enormous amount of effort into preparations - market and user research - being right is not always a certain. Learning is a step-by-step process, where at each step you have an opportunity to discover something new or cross out something that no longer applies.

And that’s what release cycles give you. Time. Time to retest your assumptions, make sure that you haven’t missed anything or can’t do something even a little bit better. Agile requires a set of iterations to make sure that your Customer has a competitive advantage.

But how much design can you pack into a two week sprint?
Well, a lot.

At its base, every Agile approach is a contract between the development team, the Project (Product) Owner (Manager) and the business. If you need more time for research, you can negotiate putting it a sprint prior to the development. If you need separate time for improvements, you can negotiate starting a dual-track Scrum.

> ” Agile isn’t moving tickets in Jira.
> It’s about building great things from
> smaller pieces.

Even though there are plenty of approaches in Agile (with Scrum and Kanban being the most prominent ones), the method itself doesn’t mean much without an effort made by the entire team. A push for releasing better increments - with clear requirements and necessary research to back it up, is something that you and your team can strive for together.

Design blossoms in environments with great communications and constant contact between the business and developers. And it’s also what Agile has been built on. The constant need for building and releasing increments, forces you to create great things that can be developed in a specific timeframe.

When to use it: if you are already working in Agile. Try to negotiate including more user feedback, push for more time for a better design. Talk more with developers about what they need and how you can work together better. Talk to your stakeholders and try to include their thought into what you are designing. Great design isn't a standalone process, it's a team's effort.

When not to use it: when your company and your team aren't ready for it. Agile requires a lot of trust between the business and team members. It sometimes may take time.

Experimenting in Lean

The Lean Startup book by Eric Ries has been out for a couple of years now. Lean heavily relies on three principles:

- Build
- Measure
- Learn

Lean gives designers a lot of freedom when it comes to delivering the final design. You can spend time researching in Learn, designing in Build and using either remote or in-lab testing in Measure.
The principle of Lean is a constant need of validating. You create hypotheses, build and test them and make sure that you got them right. Driving a business by learning is an approach that allows you to avoid guessing whether what you got was correct - it gives you actual data to back up your assumptions.

In Lean, there are plenty of ways that the Designer can use to try to shape the product. They can use User Research to test hypothesis before they make it into the product or to test whether a solution is working. They can run A/B tests or design alternative versions to check if something can be improved. They can also run Generative or Market research to get a further understanding of the target audience. The possibilities are endless and mostly rely on the ability to form a working hypothesis.

> ” A business that heavily relies on testing and measuring actually shortens the gap between your designs and the market.

In case the early adopters don’t engage enough with the current version of a feature - you can always reshape it, optimise it and release it within a short cycle. In case the new engagement ever drops, you can always repeat the process.

You can also use the Lean principles to reduce waste - hours you would otherwise spend building something that has little to no market value. It doesn’t only increase the chances of you not doing pointless work, it also increases the overall chance of success of the entire product.

When to use it: when you are a startup that heavily relies on data and is willing to try going Lean. When you want to build something small and then improve it as you go.
When not to use it: the techniques contained in the book are fairly new. There is no optimal way of running a Lean startup. If you are not willing to optimise your own process, Lean might not be for you.

Prioritising in a Fixed Scope MVP

The Fixed Scope is not really as tough as people make it out to be. Almost always it’s a set of features that have to make it onto the market to test how the product will perform. And usually the product has a very solid stance in the reality of said market.

In this case, the focus of design is to make the product as usable as possible. The efforts of the designer should be focused around delivering high Usability prototypes that can be implemented within a timeframe of the product. Of course, the effort can include a lot of usual techniques: Workshops, User Research, Interviews.

> ” In case of a Fixed Scope MVP most of the optimisation comes after the release.

When the Client starts getting some traffic, you can verify the business assumptions and cooperate on improving the overall experience of the product. You can plan out the roadmap for the next releases and set up

What makes Fixed Scope MVP fun is the fact that once it hits the market, it’s usually a well-catered set of features that can be used to improve a very specific business need. While giving the impression of a complete product, it’s usually only a core of the product that can be later improved by implementing a new set of features.

Designing a Fixed Scope MVP is a challenging process. It usually requires you to prioritise functionalities, not visual aesthetics. But, at the same time, it’s very rewarding to see how the product will fare with its competition.

When to use it: a Fixed Scope MVP is usually a business decision. You can help with prioritisation of certain features and have insight into what should be delivered first. You can also perform market research and benchmarking to figure out the scope of the project.
When not to use it: in case you have time and resources to pick another approach, you can try using Agile or Lean to have more time for improvements and User Research.

Delivering the fullest value

While the process itself is quite important to the development of Product, what's more important is the reasoning behind the choice and application. Even the biggest and best organisations can struggle with proper application of Agile, small Startups may have problems with measuring in Lean and Fixed Scope may include some wasted effort.

But it can change easily. Developing Software is mostly a learning process that allows you to draw insights and apply them in the development of next features. You can easily apply knowledge that you learn from your team members, improve communication channels and iterate some more on core features.

While the process of getting an app on the market is getting lower yearly, the process of gaining knowledge is a lifetime experience.

Jacek Suwalski
Senior UX Designer

Focus and Vision: the Keys to Cooperation

The very basis of Software Development is synergy. Software Development is a cooperation between multiple teams (Dev/Design/Sales/Marketing), accountable for different KPIs and goals. These teams can work separately, without much co-operation, even if housed in the same building. Yet, they can't coexist without one another, and they achieve better results when they work together.

And this article is about the cooperation between all those teams.

Focus and Vision

The easiest way to describe Vision is that Vision is the set of company's goals – it's what the company wants to provide to its users. The Vision doesn't really have to be complicated: it can be a monetary goal (we want to be worth ten million dollars) or an achievement (we want to be the best video streaming platform in the world). It just has to portray the service well enough that everyone working at the company will understand it.

Focus is the practical implementation of Vision. It can be applied to every part of the business. For example, you might either want to focus on maintaining the current user base to decrease churn or focus on acquiring new users. Push out new features every sprint or improve the experience of using current ones.

The Vision makes it very easy for the entire company to align their Focus. It doesn't matter whether it's six months or six years down the line, it will always be important to keep the company on the same page.

Prioritise

Let me preface this by saying: acquiring development debt is an inherent part of growth. **During our lifetime, we acquire thousands of developmental debts.** We do it in every aspect of our lives. It sounds counterintuitive, it does. But it's true nonetheless.

If you don't apply enough focus to learning calculus, you'll have a harder time learning geometry (mathematical debt). If you overwork yourself, you'll get very tired (sleep debt).

> ” Acquiring development debt is
> an inherent part of growth.

The notion of debt (technological, design and any other) in Product Development isn't really that much different. Complex computer systems are powered by **abandonware**, that is software developed ten to fifteen years ago by developers who went MIA soon after pushing their last commit.

Because of the nature of development – starting from scratch and building software from the ground up – we're in a constant cycle of learning. When we implement new features, other parts of the application become dated or obsolete.

It is fairly easy to prioritise the backlog when you apply Vision.
If we want to push out new features, let's switch our Focus here and maintain a manageable debt.
If we want to improve existing features, we can Focus on clearing debt and making small changes in the Product.

In case we want to Focus on a marketing campaign, we can create landing pages and special features for new users so that their experience of using the Product is better.

Exceed All KPIs!

Most KPIs in every company are based on the assumption that everyone is doing their best. If there's not enough inbound marketing, the Sales team won't get enough leads. If there's a lot of churn, the Product team won't be able to show an increase of usage of specific features.

It's fairly easy to overlook that there are plenty of outside factors when setting up metrics. And it shouldn't prevent you from setting them up. Goals are made to be met.

The important thing about KPIs is that they set out a course for the entire year.

- Where do we want our company to grow in the next quarter?
- What do we want to have less of?
- What do our investors want?

> ” Goals are made to be met.

When you sit down and discuss a specific feature, you’ll be able to talk about its effect on a specific KPI. When you start measuring its impact, you’ll be able to see how it aligns with your Vision.

Products are living organisms. A change in one part of the service impacts how another performs.

An often requested feature might improve your Sales quota. An increase in inbound signups might help with the usage of the application. When setting up Product Roadmaps, it’s vital that you spread Focus among all of your KPIs. Cooperation between specific departments in achieving a certain metric will increase the overall performance of your service.

The ASAP of ASAP?

There are ASAPs that are not really ASAPs. Marketing campaigns generally take at least a couple of days to prepare. Specific, user-requested features are usually gathered in a series of meetings.

There is an inherent trust built into Vision. **We, the company, want to achieve the same end goal. We just have different means of achieving it.** There are probably a couple different factors that we need to take into account when doing something new. We need to have some meetings that will help us prepare. But we'll do it because we need to co-operate.

I have a couple of friends who work in a very Agile environment of ASAP management. Their entire process is based on requests that need to be accomplished within days. There is no long term commitment and no cooperation.

It's very easy to overlook the necessity of cooperation, especially in big companies. People are busy. There's a lot of stress, a lack of Focus. Vision doesn't solve all of that. But it helps. Even if it's just to ask a question of whether thing A is more important than things B, C, D that were prepared specifically to accomplish a specific goal.

Can You Work Without a Vision?

Yes. I know a lot of companies, some of them successful, that never established a common goal. They are a patchwork of individual cells that rarely interact. And it's fine.

But it's always better to have a common goal. Thing that rallies all employees. Thing that priorities efforts of multiple departments just so you can get something done. And pretty much every company can achieve much better results if they co-operate.

There isn't one way to establish Vision. And there are multiple ways of spreading Focus. But there are usually factors that you can take into account when talking about achieving a common goal. And it's not an ASAP process.

Kamil Kołodziejczyk,
UX Designer

Design Workshops: Where Magic Happens

A design workshop is a set of meetings held at the beginning of a project. It can be seen as a UX-focused project kick-off. Design workshops are designed to help the team understand what they have to build and discover potential problems they might run into. But they do a lot more than that.

Benefits

Workshops are a central part of the UX process at 10Clouds. We invested a lot of time in perfecting our techniques that allow us to get as much infrmation about the project as possible.

What are the benefits of design workshops?

- Your team will learn to work together

  Workshops are perfect for sparking a conversation among people. You gather insights and generate ideas together – it lays a foundation for efficient team communication during the upcoming project. The team members will meet the stakeholders, understand their mindset and expectations and learn how to work with them.

- **You will build the right thing**

  Without a proper workshop, you are bound to make mistakes once the project starts. You are likely to operate under wrong assumptions, and at some point, you'll have to backtrack and review your approach.

- **You won't miss anything important**

  Stakeholders are experts in their domains, and your first task is to learn from them. If you don't do it properly, then you cannot be sure you will be making informed decisions later on. Workshop techniques are designed with the aim of bringing up various topics related to the product. Going through them will help you cover all the bases.

Planning a workshop

Every product is different, which means you need a tailored process to learn enough to start working efficiently. So, even before you begin planning the workshop, you need to understand the project first.

- **Pick the right techniques**

  You have to go through every document you have and start figuring out what's missing. User needs, frustrations, their current habits, business goals, existing solutions, market research, what worked and what didn't – this is but the beginning. Once you have a list of things you need to learn, you need to choose the techniques that will help you obtain that information.

> ” There is too much to cover before you can start designing. You will miss things if you don’t have a plan.

Once you have a list of things you need to learn, you need to choose the techniques that will help you obtain that information.

- Group techniques into themes

If you have more than one day for a workshop, select a theme for each day. This should be a topic you will keep at the back of your head during the meetings, something to focus on. Then, assign techniques to it that will help you explore the theme. This approach prevents people from jumping back and forth between different topics – you just stick to the chosen theme throughout the day.

To give you an example, you could start your first day of workshops with analysing users. You’d work with techniques that help you get into users’ mindset and develop empathy for them. On day two, you could get down to the nitty-gritty, talk about business goals, obstacles and competition.

On the third day, once you have enough information to start talking about your app, you could delve into details, with navigation diagramming, scope definition, and information architecture.

This approach makes the workshop structure incredibly easy to follow. People will know what they will roughly be talking about, and they will be able to prepare for it.

- Write down an agenda

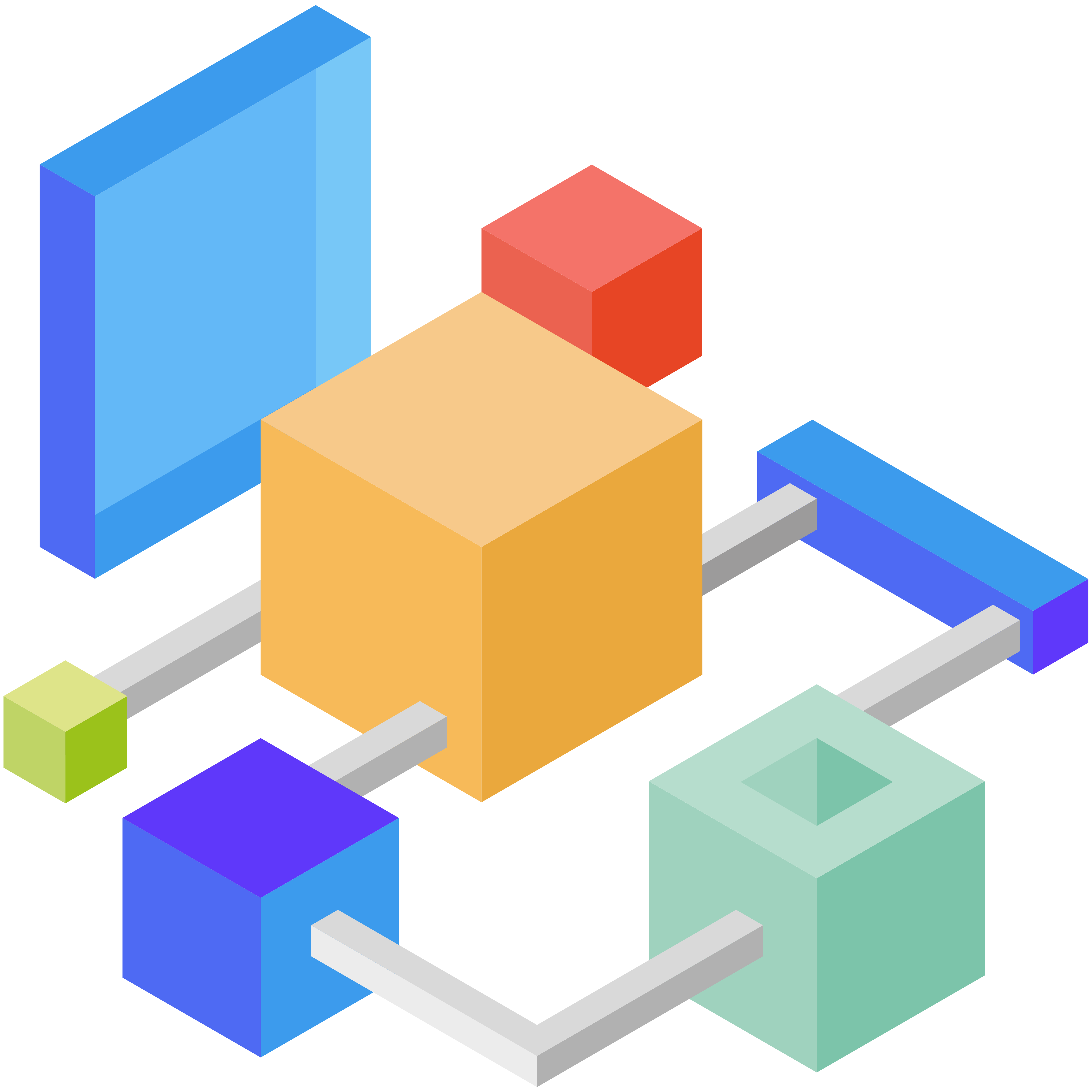
  Create an agenda and share it with everyone ahead of each meeting. An agenda is a document that will help everyone focus and stay on track. It also tells stakeholders what to expect and how to prepare for the workshops.

- Use a parking lot for ideas

  Prolonged discussions and going into too much detail have the potential to derail the whole meeting. But, on the other hand, every piece of information can prove to be important. Keeping a “parking lot” will help you deal with that dilemma. Have a dedicated space on a whiteboard where you will ‘park’ things that require further discussion. Just remember to get back to them at the end of the workshop.

All products are complex. Even if they seem simple at first, there is always something along the way that will surprise you. Workshops take care of that. They help you get a bird’s eye view of the project you’re about to start.

For most products, you have only one chance to get it right. Put everyone in the same room and figure out the strategy for a product and use it to your advantage.



Cooperation

CHAPTER 2

Kamil Kołodziejczyk
UX Designer

Interdisciplinary Teams Create Better Products

Software is no longer created by separate departments and in a sequential fashion. It's an ongoing process ran by people with various skills and areas of expertise. If you just put together a group of people and give them narrow sets of responsibilities you won't end up with a good product. They have to be able to discuss problems and learn from each other. This is what makes the team interdisciplinary.

Why is that important? Because UX Design is not limited to designers anymore. In interdisciplinary teams, people are expected to contribute ideas to the entire project, not just the part they specialise in. It's crucial because when it comes to design, a lot of depends on a particular perspective.

Common goal

For companies which run according to the Agile development process, it's natural to set up dedicated product teams. Such teams are combinations of developers, UX and UI designers, QA testers, and a project manager. You need all of them, because only when you have all of them, you will have gathered all the possible unique perspectives on the project. If any skill is missing, it will slow down the development process.

In this type of team, everyone has a different field of expertise, but their goal is the same – to deliver the best possible solution. As a result, the team don't only design or develop, but actually solve problems. This is possible in a user-oriented software house, as opposed to a typical agency workflow, where design is just a deliverable prepared for a different team. People have to work closely together.

Designers who work alone might be coming up with new ideas, but they're often unaware of technological limitations. Their ideas might favour the overall experience at the cost of much longer development time, while developers will try to cut down the number of features to make the code cleaner and more manageable. But when you put them together and encourage discussion, those issues will be recognised instantly. And better ideas will emerge.

Benefits

Improved team collaboration looks like a solid benefit of interdisciplinary teams. The way such teams improve the development process goes far beyond that, though. When people with different specialisations work together, they grow. They learn by experiencing different perspectives, ways of communicating, and approaches to problem-solving. With time, they become the so-called T-Shaped people, with deep expertise in one subject and basic knowledge of all neighbouring disciplines.
Shared experience reduces misunderstandings and fosters empathy.

How?

- People speak the same language and help each other with their responsibilities.
- Mutual support creates a better atmosphere in the team and boosts creativity.
- People who work closely together are not afraid of sharing ideas, which results in more innovative concepts and solutions of higher quality than those developed by individuals who work separately on their small bits and pieces.

New approach

Building products is a big challenge that requires a lot of hard work and a lot of time. For a project to succeed, you need people that can walk the extra mile. When problems arise, you need to be sure people will solve them by teamwork.

UX designers are no longer in charge of User Experience – they don't own it. Their role is starting to be all about facilitating collaboration within the team. Wireframes are no longer a key deliverable. They rather serve as a conversation starter to help the team find the best solution. Projects become more and more complex, so everyone has to take part in the UX design process – everyone can think like a designer to some extent.

Kamil Kołodziejczyk
UX Designer

Presenting UX Like a Pro

This article was originally intended to provide designers with useful tips on presenting their work to clients and stakeholders so that they understand the value of the presented design. You may ask: if this ebook is aimed at non-designers, why does it include tips for designers? The reason is simple. It lets you take a glance at the designer's perspective, learn what to expect from a professional and grab a few tips on simplifying and improving your cooperation with a designer.

Everything we suggest as designers has to meet specific business goals and be technically feasible. While clients hold a certain vision of their product, potential users have needs that may not be defined yet, and we have to make sure UX solutions strike a balance between the two points of view.

There is a set of skills, often overlooked by designers, that has an enormous impact on our work, namely presentation skills. In this article, I listed a set of basic guidelines on how to create a presentation valuable to both the client and the designer.

Why Presenting Is Important

Most designers work with an assumption that good work will stand on its own. But the reality often gets in the way of the perfect process. Even the best interface could be turned down by a client if you are unable to clearly explain its value. Stakeholders might not understand how the interface will help meet their goals or why the approach presented to them is the best for their business.

> ” A design is the conclusion to everything we’ve discovered, and it should be presented as such.

It’s the designer’s job to communicate all of the above. A design is the conclusion to everything we’ve discovered, and it should be presented as such. I have realised that it’s best to approach a presentation as if it were yet another design project.

Research

When trying to figure out which way to go with a project, starting with research is the safest bet.

The most common problem in design presentations is... designers forgetting to take their audience into consideration. Taking a moment to think about the client and their expectations can come a long way. All it takes is a few minutes to go through the following questions:

- Who will be listening to the presentation?
- What is their role in this project and what do they need from you?

- What output should the designer get from the meeting?

Pro tip for non-designers: Don't be afraid to ask basic questions from the very beginning. The sooner you clarify any misconceptions, the better you will communicate in the next stages of the meeting. Your designer may not realise what's clear or unclear to you as a non-designer, and it's perfectly fine to address it immediately.

Defining Goals

Finding the answers to the questions listed above helps ensure that the meeting will be productive and that nobody will be disappointed by the end of it. Writing down your findings would be the natural next step in this process. When doing this, be sure to:

- **Define the needs of the participants**

  When you present your work, the only sensible context are the client's needs. To help focus on the client's needs and understand them better —write down the goals of each participant, also considering their expectations and knowledge. They might be interested in multiple things, and they might want to know how your proposal will affect users, costs or sales.

- **Write down the type of feedback you expect.**

  This will help you set the tone of the presentation and steer the discussion that will take place afterwards.

Pro tip for non-designers: Listen carefully and answer the designer's questions as accurately as possible. Make sure you can explain your points of view clearly and try to avoid general phrases such as "I don't like it", "It won't work", "It should look different".

Designing the Presentation

It's much easier to plan a presentation with clear goals in mind. I've learned that at this stage, there are a few important things to be aware of.

Setting the stage is crucial. That's why it has to be carefully prepared:

- Explain what you'll be presenting.

  Every presentation needs a proper introduction. Explain what you are about to show and what type of feedback you are looking for.

- Anticipate questions.

  People often get anxious at meetings, so it's best to address their worries upfront. Tell attendees how long the meeting will take, which phase of the project you are addressing, how they will contribute, and what the expected outcome of the meeting is.

Of course, the presentation has to be properly structured as well. Instead of showing everything right away, it's best to build up a bit of anticipation. This ensures that your idea will be understood and properly received:

- Start with a pitch.

  Begin with an elevator pitch of your idea to get people's attention. Make it a summary of what it is and what makes it special.

- Don’t show your design upfront.

  Make it a logical conclusion of the presentation. Summarise the problems you are attempting to solve and the outcomes you expect.

- Stress the benefits

  Summarise project goals, the problems you discovered and the constraints you faced. Use this part to explain how your solution meets all requirements and is a logical conclusion of all you had agreed upon.

While presenting the actual design, guide everyone through it carefully. Choosing the right narrative is of the essence:

- Acknowledge their vision

  Clients come to us with a vision for a product. It’s up to us to capture it (usually through workshops) and carry it through the development process. That’s why we need to demonstrate that our design fits within this bigger picture.

- Tell a story

  Storytelling is the best tool to prove our solution aligns with the product vision. By creating a narrative told from the user’s perspective, we make it easier to imagine the app out in the world. Get in a character and present your design by following one of the typical scenarios of use.

Iterating

It wouldn't be a design process without iterations. Initial presentations help us get to know our client, so we have to use this opportunity to adjust our approach:

- Understand priorities

  Every client is different. We can usually learn enough to get a feel of their priorities during the first meeting. With that in mind, we can adjust our approach and make further improvements to our presentations.

- Verify methods

  The fidelity of our work can sometimes be a bit off. Aim too high, and it'll encourage people to get too much into details. Aim too low, and they'll have problems trying to imagine how the product will work.

This sums up the process that I've developed having worked on a few design projects.

When Presentation Skills are Crucial

Some clients are more involved than others only because they want to make sure that everything you propose them is right for their business. This is essentially what happened to me when I faced my first corporate client.

Everyone was more concerned with what would be good for the business, often at the expense of the actual users. I realised that we reached a moment when the project was about to pivot in the wrong direction. I was losing control. Something was wrong with my approach, and it needed to be fixed fast.

Before the next meeting, I took more time to prepare. I worked out my first take using the process mentioned above and applied it step by step. Shortly after the presentation, it became clear that my new approach managed to put the project back on track.

The dynamics had changed, and everyone seemed to have more trust in what I was saying. There were a few reasonable concerns, but other than that everyone agreed to move forward with my idea. From that moment on, I was sure that taking the additional steps outlined above can change the outcome of a meeting.

During every project, there are a few critical points like this. At those times you need a designer who can ensure the project takes the right course. Someone who can explain their reasoning clearly and convince everyone that this is the way to go. That's how you recognise a professional.

Jacek Ziobroń
Account Executive and UX Designer

Brothers in Arms: UX and Backend

The main goal of the design process is to solve problems. You must identify the key issues and find the perfect balance between users' needs and the client's business goals, as well as between design and technical aspects. This is the role of a designer: to take all these things into consideration and deliver the best possible solution. It's in designers' best interest to join forces with back-end developers on a mission to deliver.

> ” Design is an ongoing conversation among the whole team during the development process, and with the users after the release.

As you delve deeper into the process, things always start to get complicated and you face more and more challenges and questions to ask (and answer). But designers are not alone on the battlefield, because they have back-end developers to watch their backs (pun intended). Let's find out what the most efficient way for them to co-operate is.

Cooperation Will Improve Overall User Experience

Everyone needs to remember that they don't work in a vacuum but as part of a larger team. This is especially important when working in Scrum methodologies, because things change quickly in this type of environment, and a lot of people influence how the final product is created (especially developers).

To deliver better products, we – the designers – need to spend more time talking about problems and solutions with developers. That said, there is often one major issue that stands in the way: finding common ground.

The Key to a Truly Immersive Experience

Designers tend to think that learning to code in HTML/CSS/Javascript will be the answer. Most of the discussions about understanding technology that occur among designers often focus solely on the front-end side of the process.

But there are many more aspects to consider if you want to create great design. We should include back-end developers and their point of view of the entire process in order to create a genuinely immersive experience.

"We Can't Do That"

I think every designer has heard that phrase from developers at some point. This could be about anything, even dividing the first and last name fields in a contact form.

Why is this even a problem? With VR, AR, bots and Artificial Intelligence being all the rage? We can do anything, so let's get down to the business.

> ” Designers don't work in a vacuum, but as part of a larger team.

The truth is that sometimes we aren't even aware of the development constraints and what is feasible to build within the scope of your product. When was the last time you asked yourself the following questions before starting to draw your perfect solution:

- Is it enough to create your own database from scratch or will you need to fetch data from a third party?
- Should we store all data if users don't finish their process? What if users want to continue on different devices?
- At which point do we send a query with a user request to the database?
- How will you structure all information in this database so that it is usable later on?
- How will the app behave offline? Should users be able to access basic functionalities?
- What about user-generated content? Should we cache and send it to the server when back online? What information should users get in all cases?
- What options will be available in the CMS for the person responsible for managing the app later on?
- How can we use this back-end magic in the user's favour?

The last one is my favourite.

We can see the results of the co-operation between designers and back-end developers almost every day in most of our favourite apps. Let me give you a few examples.

Example 1: Undo Taken Actions

Have you ever accidentally removed something important from Google Drive? No worries, you can just click “Undo” to restore the file, and with one simple trick, because no file has actually been removed yet. Even if you receive notification that it has been removed, no action has been taken on the back end.

It is only when this message disappears after a couple of seconds that the file actually gets deleted. So “Undo” doesn’t really undo anything, but prevents the server from taking any action.

Example 2: Speed Up Performance

In order to increase the performance of user actions inside Instagram, a lot of them are treated optimistically. Click the heart icon and it always lights up, even if your connection is broken. Of course, no data will be uploaded to a server. The app doesn’t wait for the server to confirm the user’s action, instead of that it reacts as if it had. The result of your action shows immediately, even if the server hasn’t confirmed that the action has been approved.

Another example: Instagram starts uploading a photo to the server the moment you choose it. In most cases when you click “publish a photo”, it will have already been uploaded to Instagram’s servers and ready to be displayed in the feed.

Example 3: Help Filling Out Forms

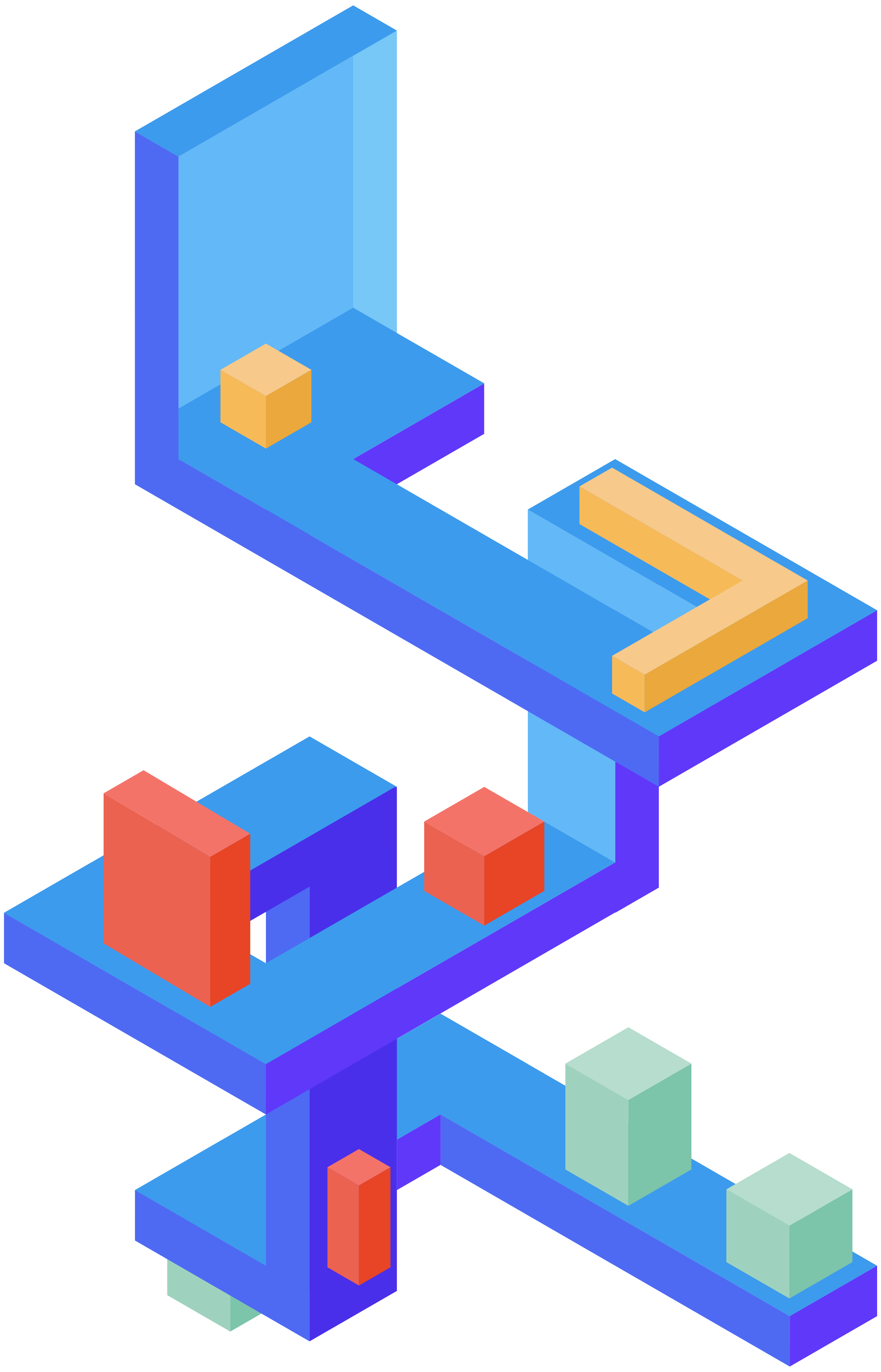
Do you want to reduce the chance of making an error when your users fill out forms? Try this solution if you have the option of fetching postal codes from a third-party database. Just change the order of the fields and let users type in their postal code first.

The database will automatically fill the town, street and county/state fields once the postal code is provided (the amount of information you can glean out of a postcode depends on the country you are in).
These are only three basic examples of a successful co-operation between designers and back-end developers. The more time you spend on analysing and thinking about these little things, the more of them you will see.

Wrapping Up

The back-end world is a terra incognita for many designers. It may look scary at first, but understanding it and including in the design process certainly won’t hurt. Instead, your team will be able to create more immersive experiences in your products.

A thing to remember is that design is a conversation – among the whole team during the development process, and with the users after the release. If we want to create great products, we need to listen to what other team members have to say. Then magic will happen!



Product development

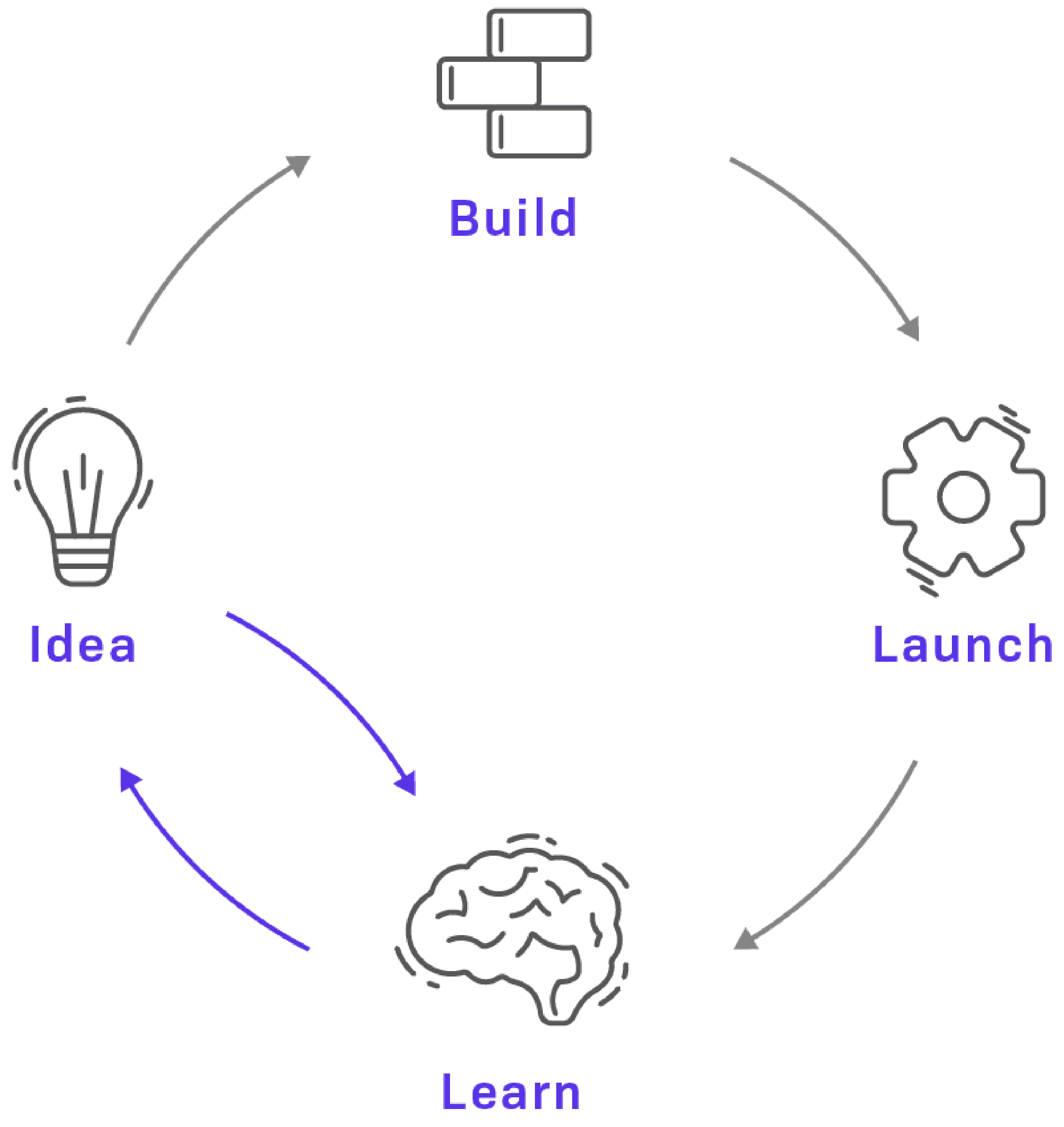
CHAPTER 3

Jacek Ziobroń
Account Executive and UX Designer

Rocking Design Sprints at 10Clouds

When working with startups, we often look for the easiest ways to validate our ideas quickly. In recent years, we've seen the rapid growth of empirical-based agile methods in software development process.

On the other hand, fans of Lean Startup methodology use the build-measure-learn loop and try to constantly listen what their customers say about their product. The next emerging topic is the design sprint. A brilliant concept or just another buzzword? Let's find out.



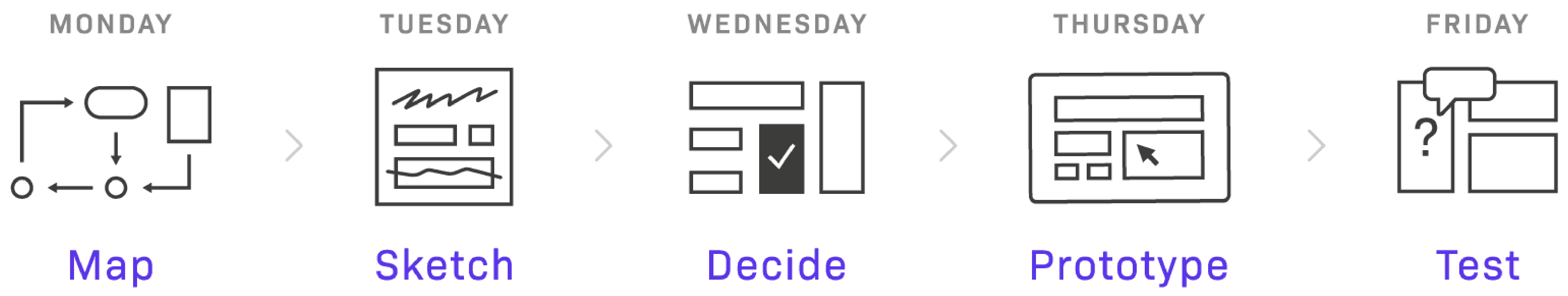
Basically, the design sprint is a 5-day framework for the process of validating your ideas quickly and cost-effectively. It was introduced by GV (formerly known as a Google Ventures) and developed to help companies make the decision-making process shorter. The authors of the concept used it with a lot of businesses (i.e. Nest, Flatiron Health, Medium) to create better products.

But how exactly does it work for us?

How to Use a Design Sprint

At 10Clouds we do not work for clients – we work with clients to build great digital products. To do that we need to understand the whole concept quickly and communicate about it with our partners clearly and frequently. What helps a lot in achieving that are product workshops.

Last year, we started using design sprints during the workshops with some clients. It was thought out as one of the techniques that would help us achieve a better understanding of important product goals.



The complete sprint cycle takes five working days and lets the team go through the process of problem-idea-learning without the need for building the whole solution. The team just validates a hypothesis so that they know whether the concept at hand meets the business goals and provides value for end customers.

The most important thing is to choose the right hypothesis to validate. And this happens on day 1. This day helps us understand the product issues by sharing knowledge, discussing problems, and choosing a target for the week's efforts.

Sprint is a perfect fit to conduct it both on starting as well as ongoing projects. During this day, the client shares all the information about our target users and their needs, competition, market conditions, and business goals.

We don't delve directly into countless ideas without a solid understanding of what the biggest strategic or tactical issues are that we need to solve next week. Is it the low retention rate? Or maybe people are lost within one of the scenarios? The first day's purpose is understanding the problem.

> ” The most important thing is to choose the right hypothesis to validate.

The next three days are mostly about the solution itself. There is a lot of sketching, making notes and discussing which of the solutions could be best for our defined hypothesis.

What we always learn about this part, again and again, is that we are all designers – everyone in the team can come up with an excellent idea. Lateral thinking not a domain-internal capability of the so-called creative minds. We work together to find the best possible solution to the problem and the sprint works because people want to take part. **The real value of a Design Sprint is that this framework is all about teamwork.**

The other thing is a Design Sprint is a very democratic process. Everyone has an opportunity to say whether they liked and didn't like the particular idea. We gather solutions that could be the best fit and let everyone vote for their personal favorites after a short explanation.

This also incorporates a critical, but too often underestimated, aspect of a design critique. I have personally seen too many times how hard it is for many companies to establish a culture of healthy feedback.

But because within Sprint every prototype is tied to our business goals, this framework forces everyone to be very candid and specific. We never talk about 'likes' and 'dislikes' but about meeting our goals defined during day 1 and affirming what's working about each idea.

In the beginning, we were a little bit afraid that maybe there wouldn't be so many designs, and we would have not much to discuss, but we found out very early that the team can generate a lot of good concepts within a short span of time.

During a Sprint, we have a chance not only to learn how to solve the problem, but we can also get to know each other a little bit better – both within the team and with the client. Did you know that one of your developers is learning about machine learning after hours? Or maybe one of your marketing guys is really into sports and has information that could actually be crucial for solving this problem?

> ” During a Design Sprint the team can generate a lot of great ideas in a short span of time.

The last day of a Design Sprint allows you to validate whether the idea you chose to test solves the problem your team defined during the first day of the sprint. We do that by showing our prototype to other people. Alternatively, if target users are not available, we just validate it with the client.

This is a really valuable day, because it can, for instance, happen that the initial idea client came up with doesn’t really solve the problem. Or it turns out that the solution that the team figured out is not that good, which is also a really valuable lesson for us. The team learns quickly what works and what doesn’t in a presented prototype without spending money on weeks of development.

Is a Design Sprint Worth It?

If you work for a startup, or software development company working for startups, you know that this environment changes quickly and unpredictably. At 10Clouds we keep in mind what Eric Ries, the creator of the lean startup concept, said about winning in this context:

> ” The only way to win is to learn faster than anyone else.

Taking that into account, we believe that incorporating design sprints into your product development routine will be a big step towards building better solutions. Design sprints are a really valuable tool – they successfully combine various techniques coming from the fields of user research, customer development, agile, and lean startup. It is also a great tool to get to know your teammates and clients a little bit better. All in five days.

Kamil Kołodziejczyk
UX Designer

UX Flow for Easier Team Cooperation

The start of development has always been a challenge for me, because explaining UX work to developers is hard.

Traditionally, I would walk everyone through a prototype, summarize a roadmap, explain features, and more. And once I was done going through all of that, I'd still get a lot of questions about the app and a ton of follow-ups.

Our flow from the designer—me—to our developers was inefficient. We needed a new approach. I needed to figure out how to effectively communicate with the team without extensive documentation on my end.

Keeping It Lean

Some techniques exist, such as Use Cases, Sitemaps, Task Flows. They are designed to help, but it would be best to use a couple of them for proper documentation. And, honestly, it's hard to update them in an Agile environment.

Lean UX is against extensive documentation because team collaboration should be at the center. The problem happens if you rely only on mockups—you'll end up explaining the same things over and over again.

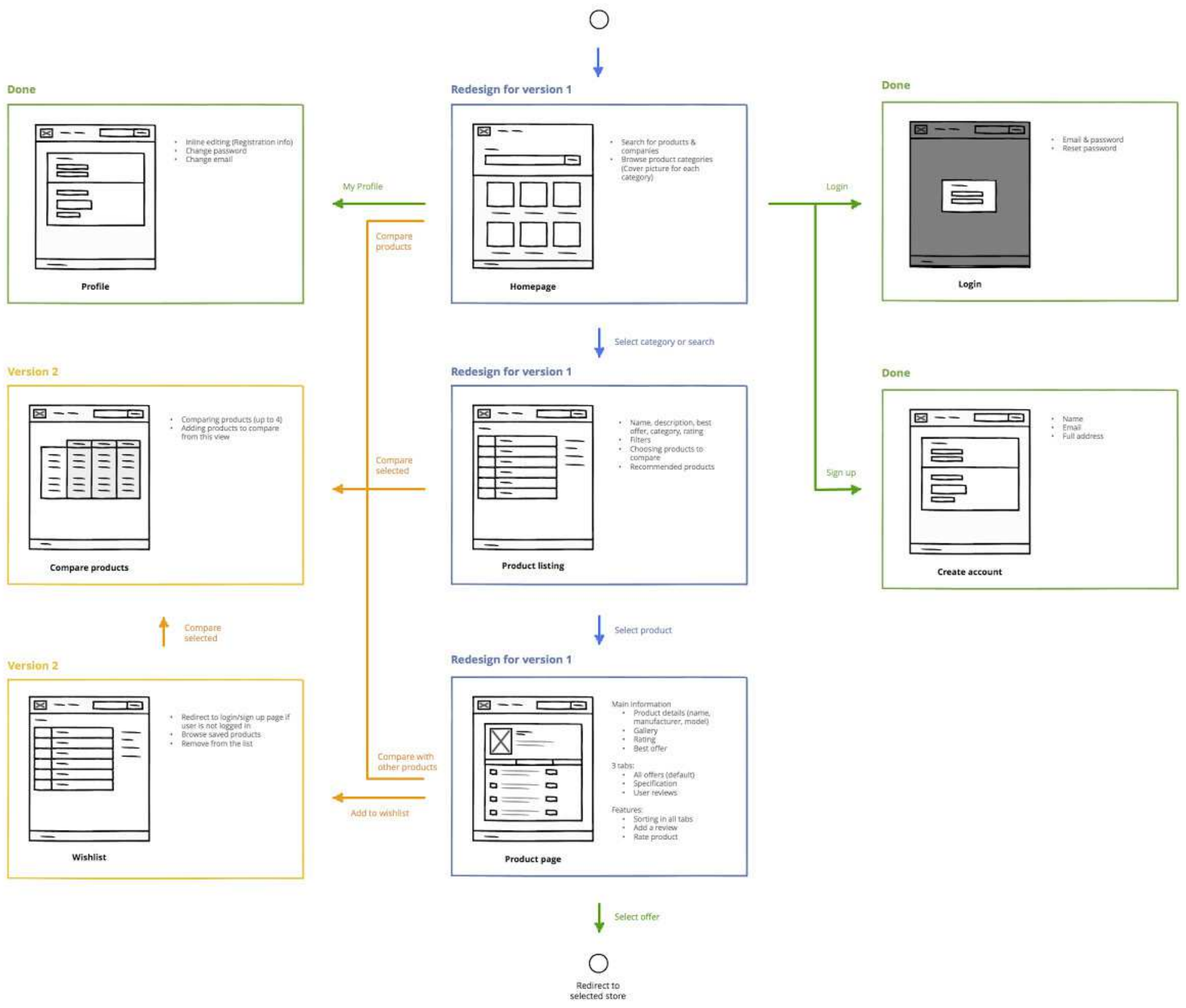
If you want to invest time into an artifact, it should engage the whole team by improving communication and keeping everyone on the same page.

Flow Redesigned

I needed a new tool to help our team flow stay lean, agile, and effective. I started by writing down the things I needed to communicate, looking to outline the project without getting into too many details.

Then I created a flow diagram as a base, and started mapping the key views. Only then did I move on to adding details: drawing mockups, testing different fidelities, writing down features and listing the data to show.

Here's the finished flow:



Full resolution available here

This version just felt right. I shared the flow with the team developers and braced for an extensive Q-and-A. Instead, they got it right away—we all felt awesome about it, and it's worked wonders for us.

Designing in Flows

So what makes this flow work so well?

Designers focus on users, but in a team environment, the system perspective is just as important.

Developers usually have to imagine the whole system before they can start planning their work. They need to know how big the app is, how complex the views are, and what can be reused.

Typically, all that has to happen before you start prototyping.

There are a few things that can help a team deal with it:

- A low-fidelity representation of each view shows its complexity.

  Will there be a grid view? A table? Are there any tabs? You can present all that with a simple sketch. At this point, it will be enough to decide if the design is going to take the whole sprint to implement, or just a small part of it.

- List all information that will be visible on the screen.

  The data you show affects the database structure, filters and search criteria you can use. It's much easier for the team to imagine the desired outcome of their work if you list everything at the outset.

- Annotate arrows to explain the navigation.

  Links between views are another crucial element of the flow. Arrows help you show the structure of the app. Annotating them also describes which features will be available.

Even though I focused on developers, this diagram supports the whole team. Project management can see where the team is in the process, and UI designers know how much work they have left.

Oh, and did I mention that UX work is much easier too? No more missing links or forgotten views (it’s pretty cool).

Wrapping Up

The flow diagram has become a standard approach for me when communicating with the team.

And it’s not only about communication. It also helps to maintain consistency in the design and speeds up development. It’s an essential part of the entire design process at my company—and maybe it could also help yours!

Agnieszka Siedlczyńska
UX Designer

Trust and Empathy Make a Great Product Team

Cooperation and trust are the foundation of every strong team.
To build both, all team members have to understand and accept goals, trust the process, have shared vocabulary and take responsibility for their role. It's not easy, but when you achieve it, the sky's the limit.

What had the biggest impact on my decision to switch my career from architecture to IT? The overwhelming need to work in an interdisciplinary team. I've always believed that a well coordinated group is stronger than one person, even if this person is a genius. The opportunity to work with people who have different perspectives, yet still share the spirit of building a common solution, is priceless.

Two Heads Are Better Than One

People are different. It's natural that with a different type of personality comes a different work style. I like to see teams through the lens of Belbin's nine roles:

- Leaders

  who have the ability to see the big picture, can define priorities, locate the resources, and motivate others.

- Plants

  who are constantly thinking up lots of innovative ideas.

- Implementers

  who need to move forward by nature, and turn all their ideas into life as quickly as possible. Chaos is their ally, but they tend to be impatient. They're doers.

- Monitor evaluators

  who calmly think over every aspect of the concept, finding all the pros and cons.

- Resource investigators

  who can build bridges with their outgoing and straightforward manner.

- Team workers

  who nurture the team, maintain a good atmosphere and help to stave off conflicts by paying attention to every team member.

- Coordinators

  who need to control chaos – they're able to organise meetings and manage the process.

- Completers
  who are perfectionists caring about every single detail, however, they tend to focus only on current tasks.

- Specialists
  who have in-depth knowledge about specific areas critical to a project.

There is never a personality that entirely corresponds to one given type. Usually, one personality is dominant in a person, but one or two others are somewhat active too.

As you can see, some personality types are conflicting, such as the Implementer and the Coordinator. However, in the process of building a product team, contradicting characters become complementary and prove that two heads are better than one.

Awareness and Empathy

Cooperation cannot exist if team members fail to understand what the business goals are and for whom they are developing the product. When it comes to user research, I find that my responsibility is not only to collect feedback from our users and monitor their actions, but also to build the awareness and empathy in my team members. There are a few ways to do this:

Inform

No one actually has the time to look into the documentation written by designers. The only way to update the team is to organise a meeting. Be prepared that the idea of holding meetings might be met without much enthusiasm. No one likes meetings, especially developers.

The first meeting with my team concluded with a long discussion. Everyone realised that there are real people “on the other end”, who have to deal with all the bugs and confusion. The more people reported the same request, the stronger the message was. Letting go of current tasks and a feature-centered approach isn’t easy, but it’s amazing to watch what happens when people actually do that.

My team invented the so-called Happy Points based on the feedback they received. Happy Points are derived from a number of repeated users’ requests. The more repetitions, the more points the request gets. Next to the regular Story Points in SCRUM they play a key role in defining priorities and planning sprints.

Show

Highlighting quotes can never be as effective as showing what real users do and who they are. This is why I like to use screen recording tools. A visual example with basic info about the user strengthens the feedback and makes it more plausible.

This is also the reason why Personas are popular. They are evidence-based descriptions of a typical user or target group. Archetypes with goals, concerns and needs. Personas provide solid support in building awareness in your team members. Placed somewhere where everyone can see them, they act as a reference point.

Another excellent method is to invite your team member to be an observer during usability tests. Watching a real person deal with the interface can be an eye-opener.

Put Them in Your Users' Shoes

Informing your team members might sometimes not be enough. People are emotionally egocentric. Their emotional state can distort their capacity for empathy, especially if other people's situation is entirely different to their own. This is why informing and showing should be combined with putting your team's members in users' shoes.

It would be perfect if your team members used your product on a daily basis. When they experience the same difficulties as other users it's more likely they will empathise. If you don't have that luxury, at least ask one of your team members to take part in a pilot usability test. It will not only change his perspective but also help you prepare for the research.

Shared Responsibility

- To cut a long story short (you will find a detailed version in another post): we support each design phase by tech research.
- Developers put their insights on the table and assume the user's perspective at the same time.
- They judge how much time the designed concepts would take to implement.

- We brainstorm and work out a general idea.
- We slice it into smaller chunks so that each chunk can be completed during one sprint.
- Each part meets the business goals and makes sense from the user's perspective.

As a result, design, implementation and process go hand in hand. Each team member, as a co-author of the idea, is prepared for the planning. Everyone knows what's going on in the backlog and feels comfortable with priorities. Estimation is easy to conduct.

One for All, All for One

Maybe this heading sounds a little lofty but working in a well-coordinated team really gives lots of satisfaction. Team spirit, synergy and fun on the one hand. Trust, professionalism and honest communication on the other.

Of course, let's not forget why we're here. Every team exists for the sake of developing a product and the job has to be done. Yet, you decide how it will be done. Cooperation and trust pay off with high performance, which will make your team achieve goals that no specialist working alone could ever achieve.

Agnieszka Siedlczyńska
UX Designer

The Awkward Secret to Valid User Interviews

Look at what people do, not at what they say – this is common knowledge in the product development industry. That said, listening to what people say might also be beneficial – properly conducted user interviews can be a valuable tool for every researcher. Answering the question “Who are the users?” can be game-changing, especially when it comes to the design of value propositions.

I conduct user interviews to supplement testing tools. Over numerous interviews, I have learnt many surprising facts that later became crucial in product development. Asking the right questions and showing a participant that you are willing to listen helps get the right flow going. Interviews are the foundation of every method of user research. (Un) surprisingly, the most difficult part of interviews is keeping your mouth shut...

Chit-chat vs. Interview

Small talk is pretty much what keeps a casual conversation going. When people interact with each other, they often fail to listen to what the other person has to say. They tend to focus on how they want to present themselves and try to find ways to entertain each other in order to avoid awkward silences. With such expectations about everyday conversations, the users interviewed might think that interviews don't differ much, and they're just casual chit-chat.

Although this is a valid point, an interview has its own specific formula, and very few rules from real life can be applied here. Transferring the habits and thinking patterns from everyday conversations to user interviews is a terrible mistake, especially if you are the researcher.

> ” A user interview has its own specific formula, and very few rules from real life can be applied there.

Researchers that are new to this trade can easily make the fundamental mistake of not shutting up, because they are too afraid of letting the interview go silent and descend into awkwardness. In such case, wouldn't you get the impression that the person interviewed ends up talking only about themselves?

The situation becomes even worse if your interviewees are also your colleagues, as it happened with 10Clouds' own product: timeION.

Two Scenarios

Imagine someone asks you: "What does your morning look like?" Even such a simple question requires some thinking before you can answer. To break the silence, the interviewer will more often than not end up adding guidelines such as: "You know, what you eat for breakfast and when you go out to work." This way the interviewee loses focus and provides answers based on the new suggestions, assuming that this is what matters to the researcher. There are two situations a researcher might ruin the interview by speaking too much.

Scenario 1.

The interviewer doesn't know how to engage in the conversational rhythm with the person interviewed because they aren't familiar with each other. As a result, they try to recognize each other's reactions.

As I described above, it is also tempting to suppress silence while the interviewee is processing a question, because the sheer thought of staying silent while waiting for an answer can be nerve-wracking.

Scenario 2.

This one happens when the interviewee finishes answering the question and looks at the interviewer. S/he stares back and a few long seconds pass in silence. This is when the researcher probably thinks: "Bingo! Now I will ask my next question!"

These moments taught me the tremendous impact of the signals my body sends, especially when my interviewee starts to look at me with desperate eyes. As a matter of fact, this is still the most challenging part for me.

> ” Accept the awkwardness, because it is a natural element of a user interview.

If it hadn't been for a book by Steve Portigal that once fell into my hands, I would have been unaware for ages. The author wrote something extremely simple, yet it changed my point of view entirely. "Accept the awkwardness", because it is a natural element of a user interview. There is no need to fight it. With that knowledge in mind, and a few trials, you can finally make your interviews more effective.

Silence Is Good, Awkwardness Is Even Better

The first scenario shows that people need time to process a question when they are asked. The trick is to know that processing information manifests differently in different people. For example, a person may look intensely at the wall, frown or stare at you with no expression.

> ” Be sure the participant knows it is experiences that are under scrutiny,
> not the people.

You need to remain concise, ask your question and give the interviewee ample time to answer it. Just do not speak. If the interviewee is lost, s/he will ask you to repeat or explain the question. Just be sure the participant knows it is experiences that are under scrutiny, not the people. Otherwise, s/he might feel insecure and give dishonest answers.

The second scenario shows another important factor for any interview: people talk in paragraphs. After concluding one part, they need to pause, take a deep breath and put their thoughts together. Some people may take longer to do this and if so, just let them do it their way. If you rush with your questions, you risk losing crucial information. Even when you think that a full answer was given, bite your tongue and wait for a few more seconds. It is very often the moment when you can gain some very important insights.

Let the Chips Fall Where They May

Interviews are challenging for both parties. Our natural conversational behaviours are raring to take over. However, an interview has its own rules: silence is crucial and awkwardness is inevitable. Hold your horses and let the interview be about the interview. It is you, the researcher, who conducts the meeting, and you need to reassure the other person that everything is under control and will go as planned.

Agnieszka Siedlczyńska
UX Designer

Making Things Work with Usability Testing

We get attached to our ideas. The more nights we sleep with them, the stronger the bond. Admitting that your idea sucks is a massive challenge and coming up with a new one isn't any easier. How to do both and remain an expert in the eyes of a client? Usability testing brings you the precious insights about your upcoming product without wasting a single hour.

Before I get down to the business with usability testing, I'd like you to take a minute and think about the goal of this strategy. You do usability testing to gain knowledge about your users and make better decisions in the process of product development. Does it influence your status of an expert in the product team? It doesn't, or at least shouldn't.

Before I discuss the practical side of usability testing, let me give you some food for thought about the process of acquiring knowledge. What are the boundaries between knowing and not knowing? And how knowing this difference can help you conduct better usability tests?

Does an Expert Need to Know Everything?

I don't know what I know	I know what I know
I don't know what I don't know	I know what I don't know

An expert defined at the beginning of this section lives only on one level: „I know what I know". They work in the One-Direction-Process, defining the problem by themselves, coming up with a solution and implementing it. Easy peasy. The only thing is that this expert uses only 25 percent of the knowledge needed. What's the result? Many of the products developed this way die. Have a look at this article if you want to see which ones.

A true expert works on all four levels. They see gaps in their knowledge and know what to do to fill those gaps. They are aware of the power of the iterative process, where the ideas are validated over and over again.

> ” The true experts are those who know what they don't know, and know what to do to learn what they don't know.

So What Do You Do to Know?

Producing a service is based on hypotheses. Before we go into the implementation stage, we need to validate all the assumptions. Every project and its users are different, and there is no single good solution or process you can take advantage of. There is one golden rule, though: the more tests you do, the better chance your solution will meet users' true needs.

On the other hand, the business world requires operating in boundaries defined by the budget and the market. You can't afford to go on discovering 10 000 ways that don't work before you make a breakthrough, as Thomas Edison did. So how do you strike a happy medium?

Instead of a big, expensive study, which you can afford to conduct only once, choose many quick, small tests, way less expensive and far more effective. Actually this is the only way to a high-quality user experience.

> ” Fail often so you can succeed sooner.
>
> Tom Kelley, General Manager, IDEO

The true experts are those who know what they don't know, and know what to do to learn what they don't know.

There are several benefits of this approach:

1. You identify and fix usability flaws sooner.

2. There is no room for an unproductive discussion about what who thinks and feels. You get straight down to business outside the company's walls.

3. You don't take test results too personally because there is no time to get attached to your ideas. Saying goodbye is easier to handle.

4. You don't waste your time on polishing designs. The iterative agile approach helps to focus on moving forward – your goal is to get to the final result.

5. You use your team's resources better. Most defects are identified before even a visual designer and a developer get down to their work.

6. You and your product's stakeholders save a lot of money and reduce the risk of business failure.

What Are the Tools?

My favourite tool for this approach is usability testing. There are many other tools which are great, yet usability testing is the most effective and efficient when it comes to rapid prototyping. Unfortunately, it brings some challenges as well. I want to present the three things you should especially remember about.

Prepare materials of an adequate quality

When preparing a test, the researcher has to define the scope, scenarios, test duration, the number of participants and the means of recruiting them. The location dictates whether it's going to be in-person, remote or online testing. However, the biggest challenge is to prepare the materials for testing. The performance depends on the development stage, and the methods of testing depend on the case.

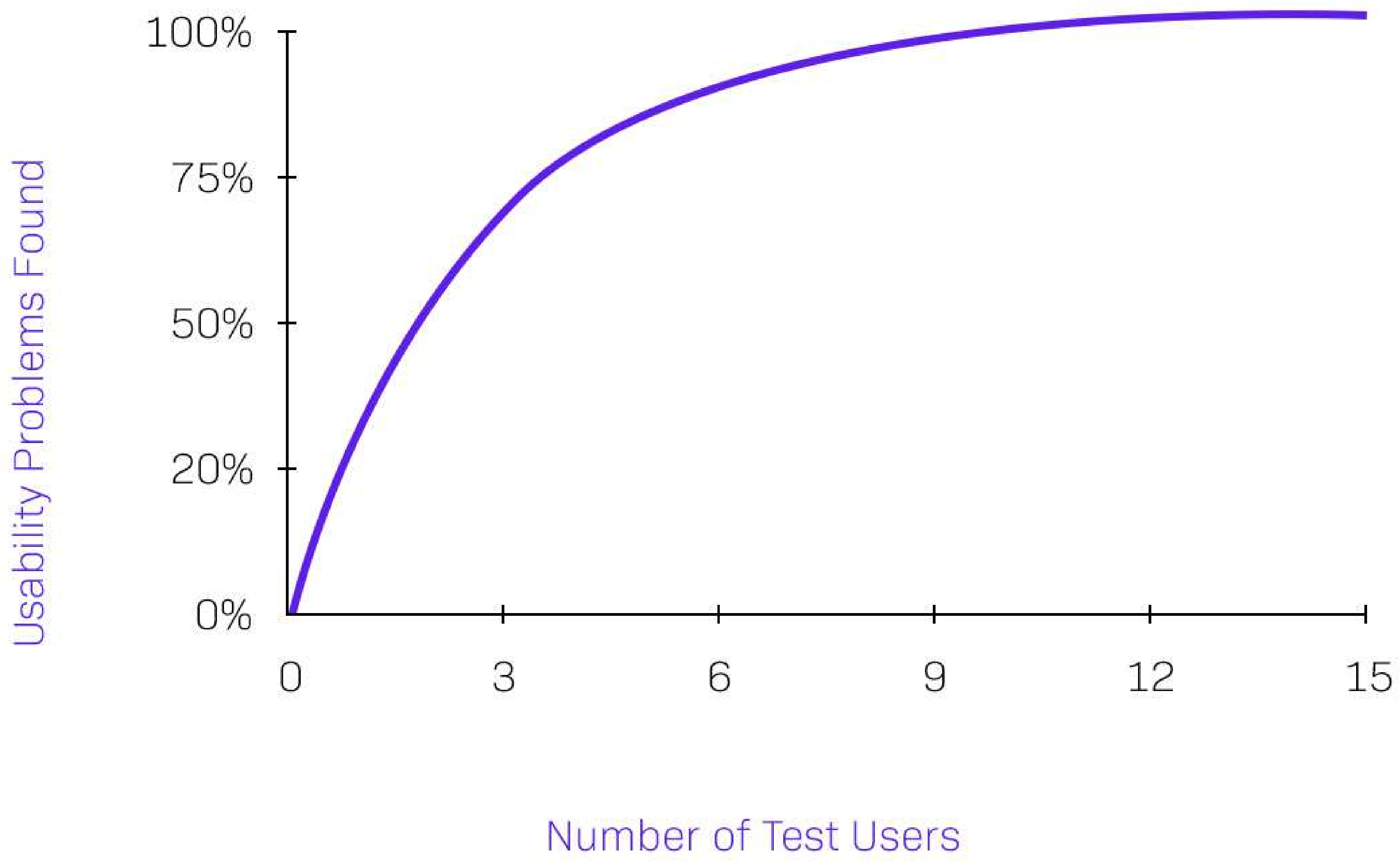
For example, one of my projects required three different methods of testing at the stage of maintenance. When it was about new mechanisms, simple interactive mockups worked fine. But when in came to specific visual aspects and the way certain multiple elements were displayed, static high-fidelity UI designs were the best option. Sometimes the fastest and the most efficient way was to implement the feature into the test environment.

My friend Kamil faced a different challenge. He was responsible for the designs of a mobile application created for a client in the USA. The product was still at the concept stage. What Kamil had to do was to produce fully interactive high-fidelity prototypes so that the client could conduct all the tests on their own. It took a lot of work to protect the prototypes from all edge cases. Kamil spent a good few days on preparing the first iteration.

Recruit the right participants

This is probably the second most challenging part. The right participants are the key to valid test results. The perfect scenario is that you get a representative sample for every target population. It's always best if the client has access to his/her own users, or if the designer has direct access to the users. If you can't get the access, you'll need to spend additional time before you find participants, or you may even need hire a recruiting agency in the end. The whole process can take up to a few weeks.

The number of participants is also important. The good news is that about five to six representative individuals from every target group are absolutely enough for the study to remain robust enough. Especially when you work in iterations. This is why the assumption some people make that usability testing needs a huge budget is invalid.



Source: Nielsen Norman Group

If – for any reason – you just can't get the right participants, you should still conduct the tests with people outside of the target group. You will at least discover and fix the most fundamental mistakes, which is always better than nothing. That said, I will emphasise once more: finding the right people for usability testing is the best thing you can do for your product to succeed.

Enjoy the silence

Finally, it's time to actually run the test. You turn on the recordings, give your participant the scenarios, and watch. Everything seems to be going smoothly... But for one tiny detail. It appears that refraining from interacting with the user and keeping your mouth shut is the biggest challenge. There are always dozens of moments when you desperately want to tell the participant where to click or what your point was when you made the designs.

Don’t! You will just have to bite your tongue and take all the mistakes on your chin. If it helps, treat it as an investigation and play a detective. Believe me, I know it’s hard. This is one of the many reasons why UX research professionals are in such a demand at the moment. Since they don’t make the actual designs, they detach themselves from the situation and don’t take the mistakes personally. There’s another piece I wrote about user testing in case you want to learn about the role of silence in the life of a UX designer.

Fun fact. One episode of the ever-funny Silicon Valley (season 3, episode 9) was dedicated to facing feedback from usability tests and focus groups. Highly recommended!

Wrapping Up

Usability testing is like a crossroads sign. It shows the right direction or which way definitely not to go. Both are extremely useful. Both take us closer to the final solution.

Now you know what to do when you meet a User Experience Designer who says that s/he has so much experience and has encountered so many cases that s/he’s 100-percent sure about his/her design decision. You can do two things: run or say „before we go further let’s find out how our users are getting along with that”.

Jacek Suwalski
Senior UX Designer

Make KPIs Your Friends, Not Enemies

UX Design relies on one underlying principle – it's crafting hypotheses and testing them with real users. UX Research can only get you so far, though. You can make use of it before you implement a change or a new feature. You can even continuously test your product to check whether your company's goals align with the product's performance.

Yet, it's even more important to see how your design performs over time. It gives you the ability to react to changes in user patterns or stakeholders' decisions. It also allows you to base your decisions on actual data, instead of your experience or hunches. You have to bear in mind, though, that while user testing gives you answers to specific questions, it doesn't say anything about your product as a whole.

More importantly, the data you obtain will increase the impact you will have on your target group. It will allow you to craft your app in a way that responds to their precise needs. You can track users from the moment they visit your website to when they convert, even if it's many months later. You can learn about what makes them tick and what discourages further use. It's a goldmine of data for decision making.

Pushing Too Hard

Big companies usually drive up their KPIs to the limit. They want people to achieve goals that are very difficult to achieve. They expect the best results. Their departments have to deliver results every quarter. To help you realise the pitfalls of such attitude, let me tell you a story about the outer space, a story that brings you... down to earth.

Apollo 1 didn't make it to the Moon. Neither did Apollo 2, 3, and 4. It took nine trials of hard iterative work to achieve the formidable result. Getting your product right is like rocket science, the literal rocket science. The basic principle is the same – achieve goals. It is usually a lot easier to achieve goals in product development, in comparison to rocket science, but it still takes some time.

> ” Getting your product right is like rocket science.

Creating a goal-driven working culture is the key to getting there faster. The knowledge of what is going right and what can be improved speeds up the process significantly. It allows you to try out new things in an iterative fashion and then either pursue them further or scrap them altogether.

Rules for Creating KPIs People Care About

Less is more.

How many Key Performance Indicators can a company have? 1,000?

10,000? There are usually much fewer of them. The inclusion of the word key should trim down the list by a lot.

It's pointless to have an endless list of metrics – you will not be able to pay attention to all of them. Some will slip your attention, even though they might require some fine tuning. Some of them will not even be reported correctly.

> ” The inclusion of the word "key" should trim down the list of KPIs by a lot.

It's usually better to create a small list of, let's say, five KPIs for each department. Then build it up from there. It's best to remove things that are no longer used or don't bring much value. Good KPIs reflect the company's direction. You are going to focus more on Facebook and less on LinkedIn? Maybe it's time to strike LinkedIn off the list and let some fresh air into your performance spreadsheet.

No one will read a thousand-row-long document during a break. Your employees won't access it on a day off to check how the company is doing. Nonetheless, a decent-sized document with things people care about is a great way to share your company's vision.

Precise Naming

When I was a marketer, I used to go to a lot of meetings where people had trouble understanding what a particular KPI meant. Let's say that you have a KPI called "number of leads". Are these qualified leads? How far are they in the pipeline? Did you inflate the KPI by sending an outreach campaign to tens of thousands of people to have a hundred cold calls set up?

A KPI should help you improve. Everyone, even with only the basic knowledge of what you do, should be able to understand it. You should be able to turn it into a PowerPoint file and present it with slick charts.

A good KPI is very precise. Let's say you send out an outreach campaign to ten thousand people:

- How many emails did you send?
  Number of emails sent

- What was your response rate?
  Response rate to initial e-mail

- How many of these people did you follow up?
  Number of follow-ups to initial email sent

- How many people responded to the follow-up?
  Response rate to follow-up email

- How many people told you to leave them alone?
  Churn rate

As you can see, these are fairly simple questions. They make great KPIs. When you go to a meeting with stakeholders, they will easily be able to tell what is going on with your campaign. This is pretty much the way you should create KPIs. Ask questions, give engaging names to your indicators and provide data. You shouldn't have a KPI called "'buy' button clicks", try "number of purchases from X page" instead.

Teams also appreciate good names. It's hard to get behind a KPI called landing page button clicks.

Growth Direction

A rule of thumb for any good KPI is: **is it actionable?**

You should be able to easily pinpoint how the KPI is doing and why. And then be able to improve it.

I used to consult for a lot of companies that celebrated every minor success such as a temporary influx of sign-ups. On the other hand, they would complain whenever their KPIs went down, even by the tiniest of fractions. This is not the way how it is supposed to work. It's normal for KPIs to fluctuate.

That said, you should know why they fluctuate and act on it. You've hit the jackpot with your Facebook campaign? Invest in it! Your blog doesn't increase newsletter sign ups? Maybe you should run an A/B test? Your e-mails don't convert well enough? Change copywriting and test again.

> ” A rule of thumb for any good KPI is:
> is it actionable?

There are also KPIs that get reported for consistency's sake. Week in and week out you get to hear almost the same exact number. A stagnant KPI that doesn't really change should not exist at all. Let's say that your pages take 3 seconds to load every time. Is it good enough for your users? Maybe. But can't you knock it down to 2.5? Maybe preload some content to make sure that users don't feel like they have to wait.

> ” It’s normal for KPIs to fluctuate. You should know why they fluctuate and act on it.

If you want to measure something for the sake of consistency, don’t report it unless something happens. When something happens, you will know whether it’s good or bad. If nothing happens, so be it.

Time and Goals

Metrics have an integral time component. They are measured live and can be turned into a report at the end of each period. If metrics are reported, they are usually reported on a weekly, monthly or quarterly basis.

KPI goals, on the other hand, don’t have a time component unless you add it yourself. Let’s use a real-world example.

John wants to get fit. He doesn’t really do anything about it, lifts the same amount of weights every week. Sometimes, he just skips training altogether. He hires a personal trainer, Kate, who sets him a goal. In three months he should be able to run ten miles. He should be able to lift two times as much weight. Now his training is truly productive.

> ” KPI goals don’t have a time component unless you add it yourself.

It’s the same with goals. Let’s say you want to have 3000 sign ups per month. It’s a ten-fold increase, and you have six months to deliver it. You can break it into two parts of three months. Now it’s 500% in the first quarter (300 to 1500) and a 100-percent increase in the second quarter

(1500-3000). You now have 12 weeks in each period to deliver value, and you can track the progress more easily. You can also reestimate the goal each quarter to assess your real capacity.

Q&A Session

Q: What tool should I use?

A: There are plenty of options for great tools, and there are plenty of top 10 lists for each and every thing you might want to measure.

The painful truth is: tools are just tools.

It doesn't matter which tools you use as long as you are getting value from them. Some companies run on SQL searches on the back end, then run functions in Excel or SPSS. It works for them. Some companies have 60-70 tools and don't really have time to check them all.

You should find a tool that supports your goals best. It might be an A/B testing tool (like Optimizely or Visual Website Optimizer), it might be a combination of analytics tools (like Kissmetrics or Google Analytics). If you have a product, it can be Woopra, Amplitude or Mixpanel.
In the long run, you should always reserve some time to check out new tools. You might not be getting the results you want from your current one, or you might learn that you are using the industry's best.

Q: I work for a big company, and we have to use thousands of KPIs to show the value of our work.

A: Really, do you? In most cases, internal reporting (for your department) is much different than reporting for the entire company.

Usually, internal departments track a lot of data. It's the data related to how servers operate, how many front-end things can go wrong, how many users click on buttons. The boring stuff.

These are not key metrics. For example, a key metric for the backend is uptime. If your service works half the time, you are doing something wrong. For UX, it's a derivative of conversion, not button clicks. For Sales, it's the number of closed deals, not emails with a particular copy.

You can usually reduce the number of indicators by at least 30 percent and show the truly key things that you measure.

Q: I don't think I measure enough things or don't measure them at all.

A: You should break down your service and department work into parts. What are the key things that you should do well to achieve long-term success? What steps should you take in order to do those things them? What makes you money and what makes you lose it?

You shouldn't start with hundreds of items. You should start with five KPIs for each department and see how you do. Are these the things that you really want to measure? Are you able to improve them? Will your service do better if you included another KPI?

Please remember that quality goes a long way. Quantity makes your KPI table bloated.

The Perks of Data-based Culture

A data-based culture is the driving force behind every successful business. A chaotic mess of user movement becomes much clearer if you can shape it into a funnel. A feature that doesn't perform well might perform better if you just increase its visibility. Your website might convert more if you A/B-test it. Finally, you might sell more if you improve your e-mails.

Every big organisation collects data across a wealth platforms and websites. They use this data as a driving force behind their decision making. They rethink, optimise and even rebuild products in ways that they can justify. They cut obsolete features and turn successful ones into new products. They earn money by getting their users to do what they actually want.

Wrapping Up

Let's recap

1. Pick your key metrics.
2. Give these metrics appropriate names.
3. Make them actionable.
4. Act on them.

If you get these basics right, you should be able to achieve everything you want. You will be able to track your progress and, at each step, consider your next moves. You can then iterate on them to make your growth easier.

10CLOUDS

About Us

10Clouds is a leading app development and design company in Central and Eastern Europe. Since 2009, we have built and designed software for more than 90 clients, mostly in the United States, Western Europe, and Australia.

10Clouds deliver projects in such industries as Big Data, EdTech, FinTech, crowdsourcing, e-health, transport, and more. Currently we employ more than 100 development and design experts located in Warsaw, Poznań, and Wrocław.

Design by 10Clouds

We translate business goals and objectives into remarkable design:

UI

UX

Product Design

10Clouds' Design Team strives to deliver a balanced combination of unique designs and usability. We are a group of high-achievers, passionate about our job and keeping a strong focus on individual features of every project.

Get in Touch