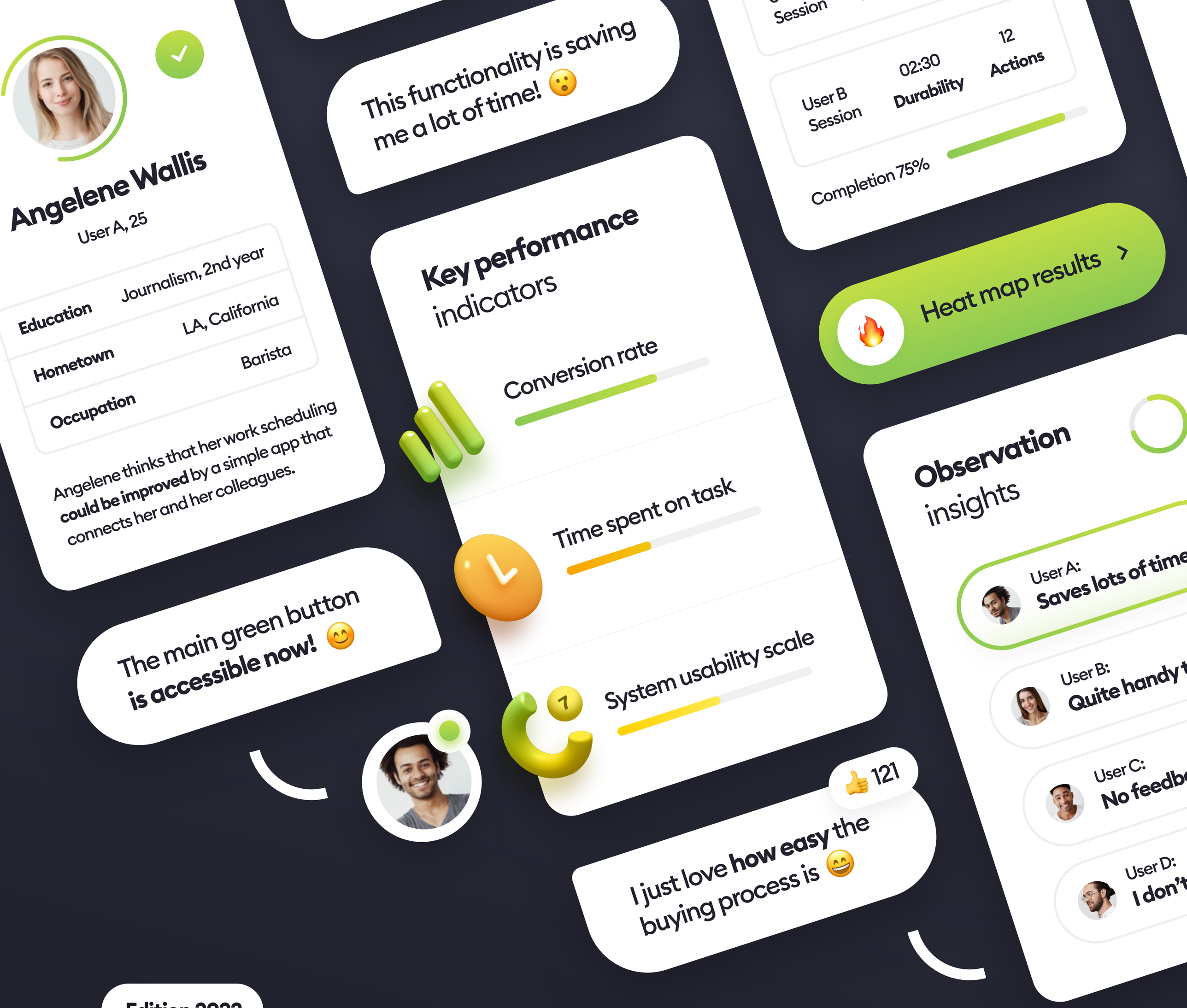


Edition 2022

Michal Malewicz

No bullshit guide to UX

Learn UX the easy way

TABLE OF CONTENTS

This is version 1 of the eBook. Based on your feedback and requests for topics to cover, I will be adding free chapters and changing existing ones to make it easier and clearer for you in the future.

The premise for this eBook comes from the fact that I'm primarily a UI designer that over the years had to learn all the research methods and techniques to keep in touch with what's needed in the industry. I am sharing my point of view with you - one that I've been successfuly using for years and that helped me complete over 500 projects in the last 20+ years.

Before we begin

Before we begin I need to say that this book is a different approach to UX than you may be used to. It's not academic by any means and it doesn't treat UX as a science. Instead I show it as **a way of solving problems,** based on my personal experiences with a large number of projects - but I'm not saying my way is the only way and that it's 100% right. I'm sharing my perspective, because I've been successfully doing this design thing for over 20 years.

Still, I encourage you to not jump in blindly. Use your own bullshit detector to get out of this read as much valuable information as you possibly can. Discard the rest.

Treat it as a guide and an overview of my personal approach - not a source of truth. You can use the knowledge within these pages to be successful, but you can also be successful using - in many cases - complete opposites of the methods I outline here.

There is no one, right way of doing these things. This book is me explaining how we did some stuff to you, as if we were in a bar. Pick one that's right for you by merging different perspectives.

With love,

Before we begin exploring the world of design together, I wanted to explain the title. Or, to be more precise the word Bullshit. Doesn't sound very UX'y right? That was exactly my idea.

Let me elaborate.

UX design education, as an industry, has gravitated towards being overly academic, scientific and serious. It desperately tries to up itself with that seriousness, so that maybe one day people will mentally place „UX Scientists" in the same row as people who cure diseases and launch rocket ships into space. After all, it's all science, right?

Not exactly. UX design is a very good job choice, as it pays surprisingly well and the rates are only going up. But that scientific image is doing the industry more harm than good.

Academic language, while may be helping some seasoned designers feel better about their perceived intelligence, makes it a lot harder to grasp for beginners.

That leads to extreme gatekeeping in UX, with the barrier of entry being set way too high for what the job actually is.

Gatekeeping

Gatekeeping means people already in the industry are making it difficult for others to join.

Let’s fight gatekeeping

I want to change that, because I used a backdoor to get to where I am. I never completed any UX Bootcamps, courses or classes, but rather simply when UX became better known, I was already years in the web design industry. I remember in 2001 I had my first web design job at a startup. Yes, a startup in 2001.

Do you know how many sticky notes were on the walls at the office? None. Do you know how often they made personas, outlined empathy maps? Not once.

Low fidelity sketches of screens were also pretty rare back then.

Did that startup fail miserably because of not using a UX process? No. It’s still around today, after over 20 years.

Now, of course the reason they are successful, is because they use the foundation of UX, a bit subconsciously. They design, test and repeat - constantly trying to improve the product.

And this is exactly my point - the actual goal of a UX designer is not to go through some mystical process every time, but rather use the method of design, test, repeat to improve your work.

At its core it is really basic.

Since 1998 I have worked on over 500 different products, from banking apps, triple A games, iPhone games all the way to pregnancy trackers. If there's a product category, chances are I have worked on at least one product in it.

That gives me a unique perspective on which parts of the UX process are used, when and how they're used and what parts are the most necessary to succeed.

In this book I will try to explain everything you need to know about UX, using simple, easy to understand language.

First, I want to focus on the essentials - because they're what pushes you and your work truly forward. But I will also cover less popular techniques, including the ones that are barely ever used.

1

Why we do UX?

Why we do UX?

I remember my first real, industry job. In October of 2001, a little after 3 PM I arrived at a 4-story marble building in the centre of Warsaw, Poland. The building was a modern office-space housing at that time offices of a couple well known companies, including the offices of IKEA. It looked modern, clean and very intimidating. Like a smaller version of a Manhattan skyscraper, with a lobby, a guy in a suit at the door and everything.

I just came back to Poland from the US, this time for good and after four years of doing cheap HTML websites I decided to find my first real job.

Lucky for me, back in 2001 the Internet wasn't really as popular yet and people who knew how to design or code were so few, that I was likely the only candidate to have replied to their job post. And yes, that job offer was posted in a regular newspaper. Printed! The dark ages!

I took the stairs and entered the office. A secretary led me to the meeting room, where the CEO and his right hand were waiting for me. It was a startup from the tourist industry - a website that allowed you to book a hotel or a holiday trip to Poland, including additional tours or attractions.

The sun was setting, leaving a beautiful orange glow on the opposite wall as I talked with the bosses about my web design experience. I got the job and were to start early next week.
This is the company I mentioned before. They still exist to this day.

How did this happen, if UX as a full industry launched around 2008-2009. Of course the term was floating around for a while, thanks to Don Norman, but the process wasn't yet fully defined, and post-it notes were only used to leave joke messages on your friends large CRT displays.

How did they manage to deliver anything of value to the customers without a UX process? There can be two answers to this question, but of course the truth lies somewhere in between.

The simple answer would be that there wasn't a lot of competition back then, so people simply *had to* use that website to book their trips and hotels.

I think it's the second option though: the company had the right approach to building the digital product. They used the simple analytics that existed back then (remember, that was before Google!) and made changes to see if it works better.

They surveyed their customers to identify some pain points, and without a single low-fidelity wireframe, they went straight to high-fidelity design, or sometimes even prototyping in code.

Do you see a pattern here?

The product cycle was a repeating series of the same steps over and over again. Design a new feature based on what the users said. Test it on said users - on production, and if it works, refine it further. If it doesn't work as expected, remove it.

1.1

The all-knowing UX Guru

The all-knowing UX Guru

Maybe I was unlucky, but most senior UX people I met were egomaniacs. A large portion of them were jerks too. They felt like gods of product, and praised the full UX process as the most important invention since sliced bread.

But was it really that innovative? In many cases it was simply taking the most logical approach to building stuff - build, test, improve.

Three steps. Simple, right? And a lot easier to comprehend at a high level - design is these three steps. And if the result is positive - i.e. we made something that's better than it was before - it means we did good design. Then the process repeats.

For years that feeling of superiority of many UX designers was annoying and confusing me.

"Why do they need to wrap a rather simple process in all that jargon and fake-difficulty?"

Now, I'm not expert enough to answer that question, but luckily I got through that when i realised what you should also consider - that it doesn't matter.

First of all, we're doing a rather easy, yet well paid job. Design is moving rectangles around, based on some simple data. It's baby research. Almost anyone can learn that.

Taking that approach many years ago, led me to understand the industry better, and yes - it also led me to improve the quality of my work. That's because having a structure and a plan of action is good. The UX process is not at all evil, stupid or useless.

It's often simply overblown a little, and especially for junior designers - it's presented in a complicated, intimidating way.

We design experiences, to solve a problem. Most often it's a business problem of not enough profit, and we want to do everything we can to maximise that profit for the company. And unhappy users aren't that keen on paying, so to achieve more profit we need to make their experience as flawless as we can.

And then test and repeat.

1.2

What is UX Design?

What is UX design?

If you’re reading this book, it’s likely that you already know that UX stands for User eXperience. I guess X is there to make it sound cool, because if we were to use the first letter of each word it’d be UE not UX.

Many people will say that UX is all about the user. After all, it’s their experience right? So we should place the user front and center and do everything to make their experience great.

It’s not 100% true.

In reality the job is about reaching that goal of more profit for the company, through making something that's easy, fun and memorable to use.

When people like interacting with our work, chances are we're doing something right.

In the real world, the goal of UX is to make more money through happy users.

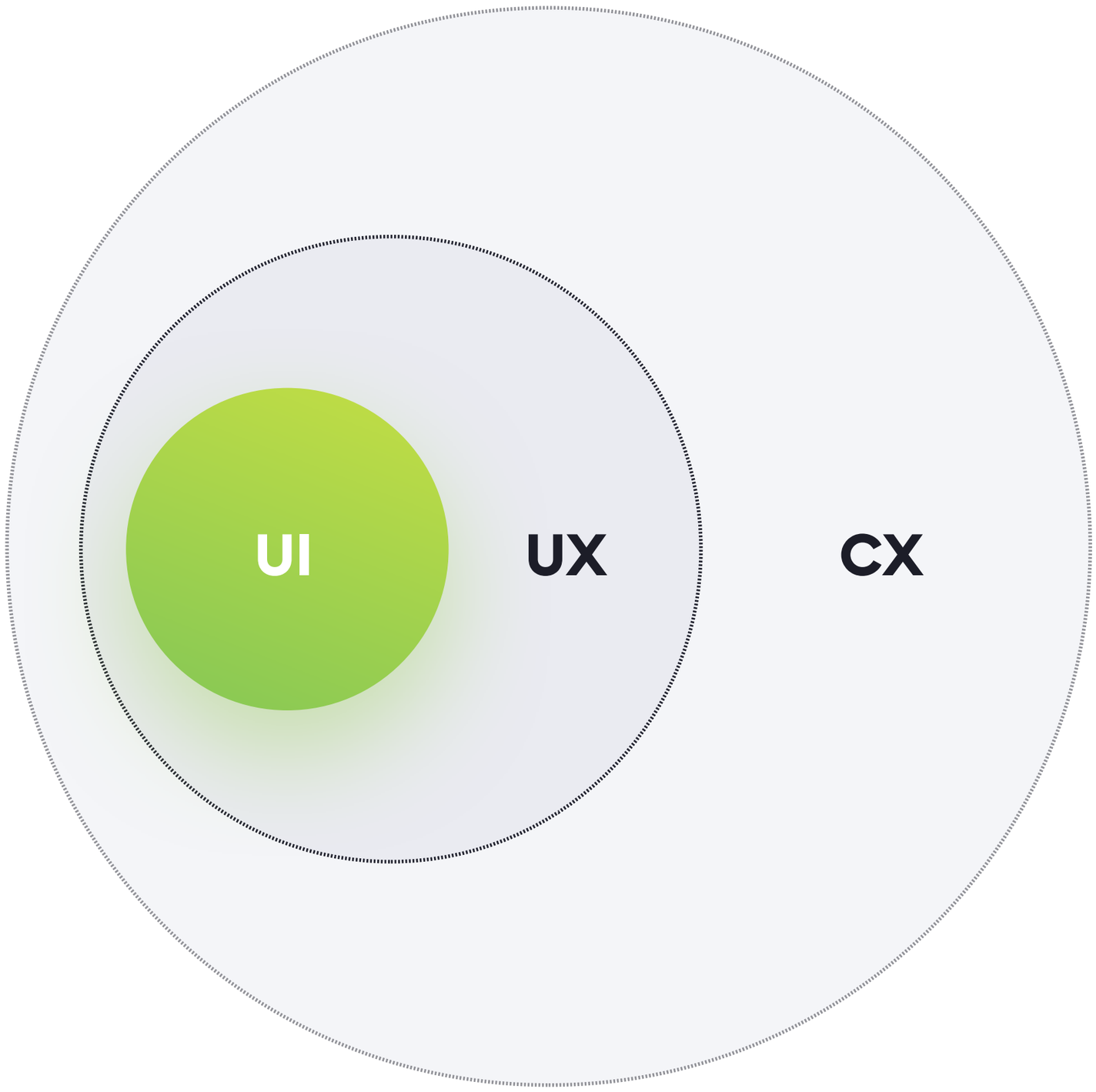
I highlighted that part for you.

To achieve that goal, we design, test, and repeat the entire process over and over again. We repeat it, because even if our design works great the first time around (and it sometimes does), we can still look into ways of improving it and innovating.

So let's start with that in mind, and go through each of the stages in more detail. *The following breakdown is also present in Designing User Interfaces book.*

Where does the humble interface fit inside the trendy industry terminology? It's often looked down upon by "real user experience specialists" as "painting with crayons." We believe this is the wrong approach. What we see is just as important as information architecture and business goals.

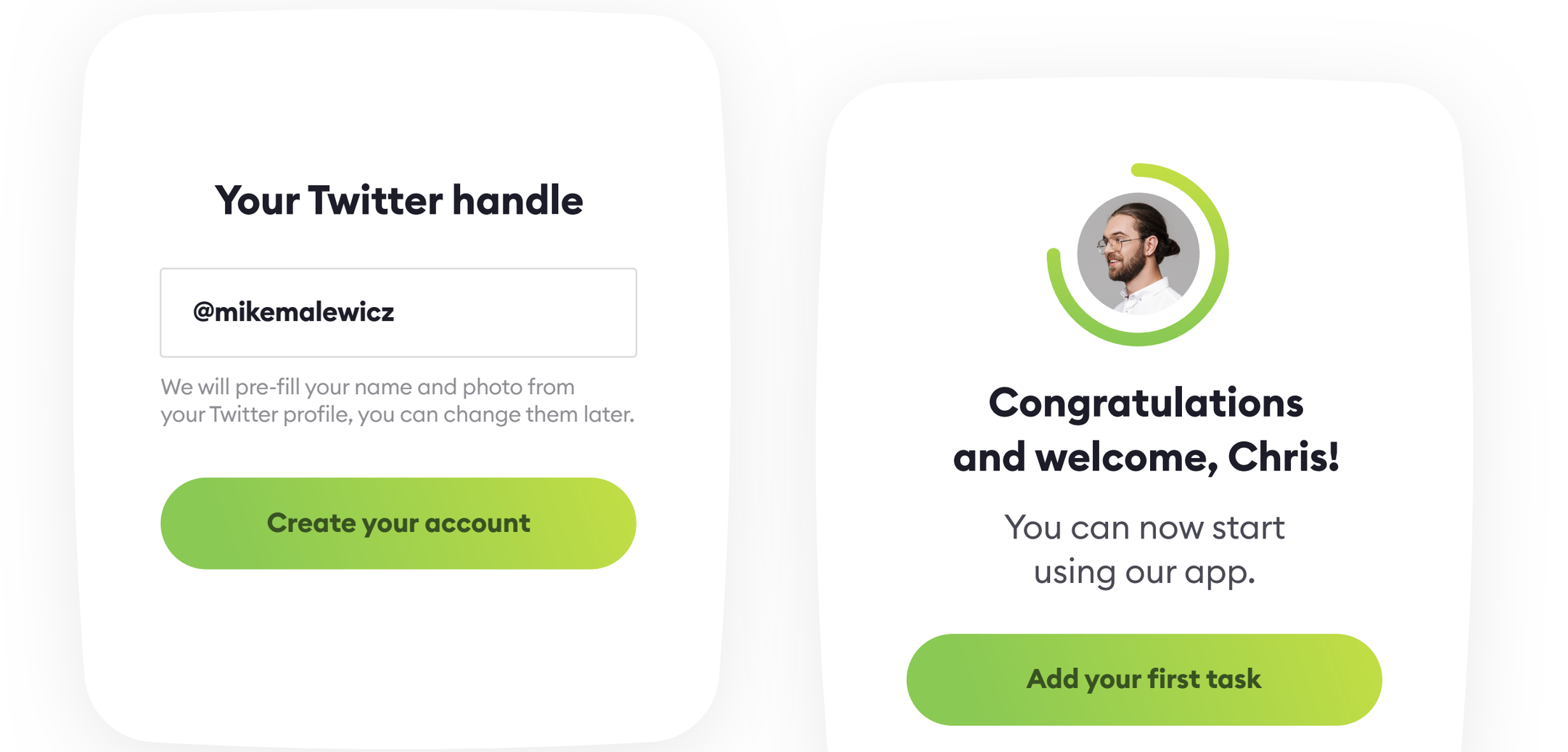
Customer experience is a high-level category with UX being a part of it, while UI being a part of the UX process. The diagram below shows the relation between these elements. To build a great product, you need to take care of every little detail with great precision.



UI or **User Interface** is the visual representation of a digital product. Its obvious use cases are apps and websites. UI is the link between the user and the functionality of a product. It helps to achieve the desired outcome through a series of human-machine interactions. It's a series of text, shapes, graphics, and photographs combined in a way that allows for a fluid, natural interaction.

An interface is a mix of grid, layout, typography, colors, animations, and microinteractions. In other words, UI is all that we experience - mostly with our eyes. UX defined information architecture is a blueprint for UI's refined final look.

The job is to define its unique style and make it fit the target market. That visual representation should be readable, usable, and devoid of any unnecessary, distracting elements.

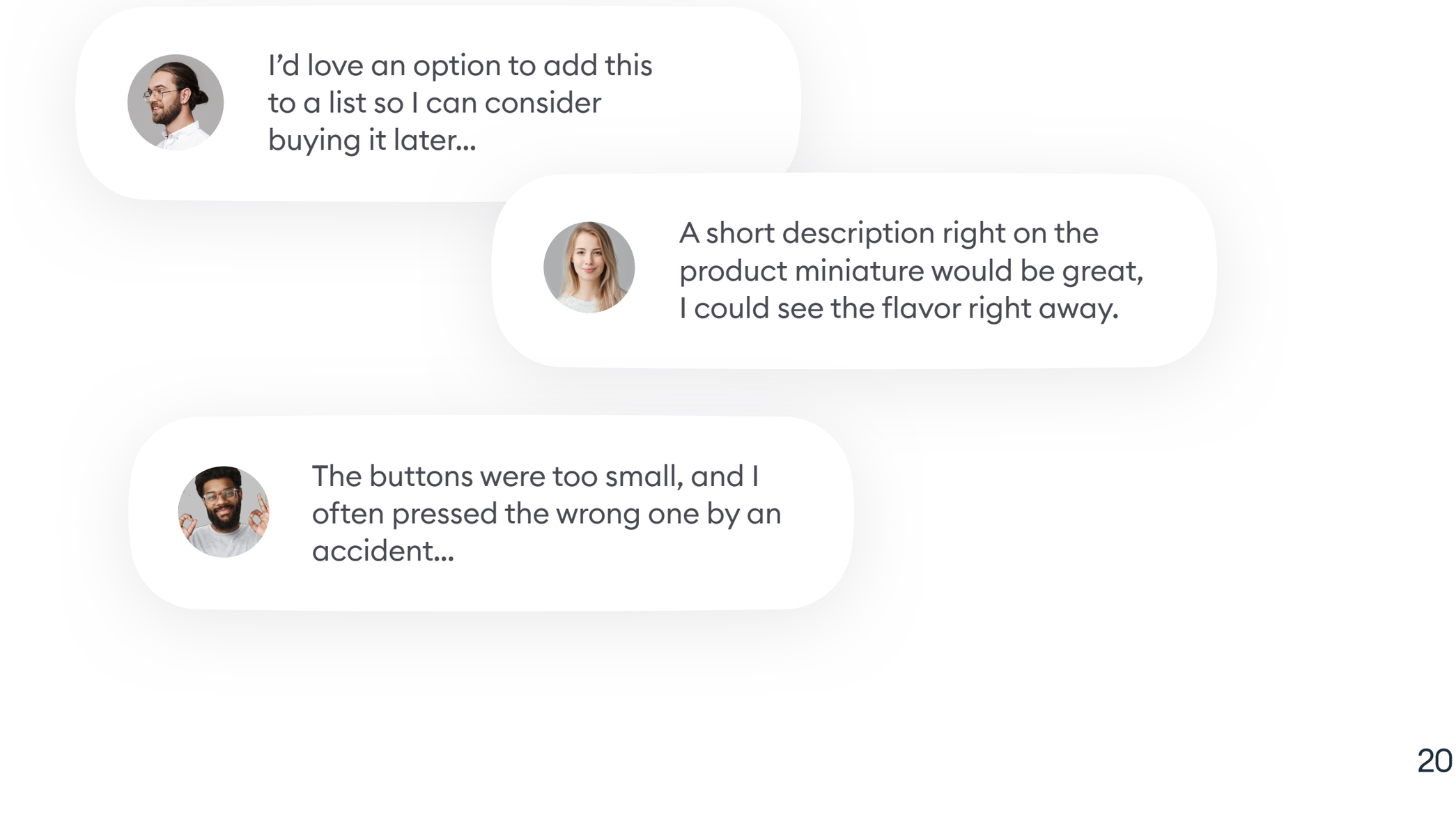


UX or **User Experience** defines and studies how easy it is to use the product. With digital products, that means the interface, navigation patterns, and communications.

The goal of UX is to allow the largest possible group of users the ability to understand and use a product. If the role of a UI designer is the look & feel, the role of a UX designer is defining how it will work.

Aside from information architecture planning, UX also covers various kinds of research (surveys, A/B tests, focus groups, interviews, workshops, and more).

UI is a part of User Experience because readability and looks are also influencing the ease of use and shape our feelings towards a product. This book covers the most basic UX principles but focuses heavily on the UI perspective of design.



CX or **Customer Experience** is often confused as being another name for UX. The truth is that CX is a top-level process that defines not only how your product works, but how your entire company operates.

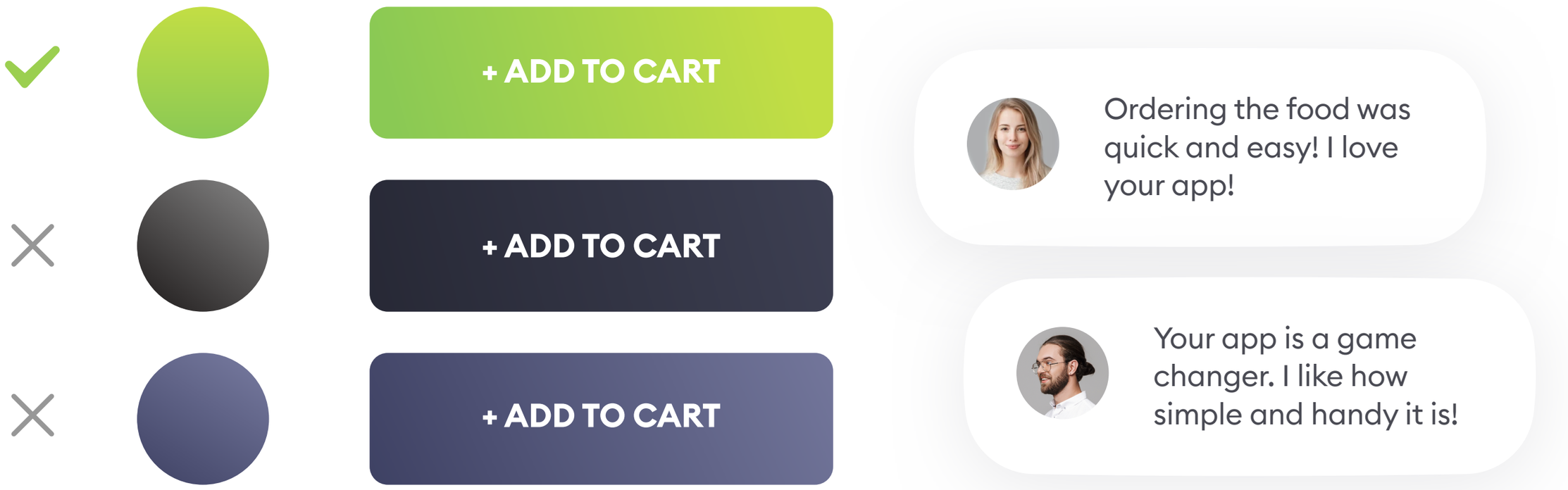
The CX process is the UX of your product, but also your branding, marketing, or friendly customer support on the hotline.

You can design Customer Experience by defining clear, consistent rules for your entire brand. Those rules apply to all the real-world experiences, as well as UX and UI. They even influence the way you write your copy.

A good example is one of the popular fitness apps. They changed the boring "save your workout" message to an "I'm awesome" button. Customer engagement went up.

All of those elements combine for a consistent, coherent vision with which our potential customer can quickly identify.

Here’s an example of what can be a part of customer experience.



Changing the brand colors to a more positive, green gradient yielded an improvement in user perception of our brand. We also tested various types of label text to see what works best.

In the last few years, many of the UX designers started calling themselves Product Designers instead. The term merges most UX competences with UI and basic research, allowing you to solve a wide range of potential problems.

This new naming convention comes from the fact that UX designer, as a title, doesn't show which skills the person has. UX designers are often product owners, researchers, wireframe makers, or all of those combined.

A product designer can help with both the business processes, choosing the right building approach, and final interface designs.

It's often the most versatile person in the company that can guide the rest of the team, including the developers.

Some say Product Design is User Experience design 2.0.

1.3

The curse of UX Agencies

The Curse of UX Agencies

2008-2012 was the golden age of UX. It is when it exploded as the only real way of doing digital products, but also in terms of financial gain. Nobody questioned any parts of the UX process because it was too new, and mostly proven to work, which led to a very serious case of dishonesty from UX consultancies.

Here’s a real-world example.

A popular UX consultancy schedules a client meeting to start the UX process on a new product the client wants to build. The meeting is in person, and four people from the UX agency are attending. Of course, one person is the lead there, and that person does practically all the talking. Everyone else is supposed to add two or three mini suggestions throughout a full three hour workshop. That’s because it’s easy to remember a couple of suggestions, so the client will know those people contributed and were necessary.

The lead starts with a role-playing game, asking the client to imagine some unrealistic scenario that often doesn’t have any connection to the product - like: you’re running late for a bus to work and your coffee is sour, so you’re frustrated, but you need to use this and that feature of the mobile app - how do you feel about it?

After many “games” like that, and general discussion, there is of course some requirement gathering, but it’s often already distilled by the agency lead during the meeting. The three hour workshop turns out the be summarised in seven bullet-points, but they never show those to the client.

The meeting is over, and because UX professionals are in short supply, the agency bills for all four people, times three hours. That often repeats many times before any actual work gets done. In the best case scenario, they go back, and start preparing their initial proposal after the first workshop. It always has it all: personas, empathy maps, customer journeys, storyboards, low-fidelity sketches, low-fidelity digital wireframes and a fifty page report that is basically a stretched out version of the previously mentioned seven bullet-points.

Most of it is made up and hidden under industry jargon, because it’s not really supposed to be beneficial to the client. It’s supposed to look like it was a lot of work to prepare and it’s supposed to look “smart”. The client then feels like they’re in the good hands of true professionals and gladly pays up for all the inflated hours.

But things started changing around 2013.

The iPhone, in 2007, pushed the digital revolution to the masses. It took another six to seven years for a large portion of the adult population to suddenly become tech-savvy. The use of technology have skyrocketed, and with it the actual knowledge of the users. That, around 2013-2014, led to clients becoming a lot more self-aware with their digital dealings. They simply started to understand what a good user experience is. After all they've been using digital products for many years now. They knew what was a pleasure to use, and what was extremely frustrating.

And because technology was so ubiquitous they started to notice the "ux patterns" themselves.

That knowledge boost led to many clients simply starting to see through the bullshit of UX agencies. They started reading industry articles and UI trends, and found out that some parts of the process were completely unnecessary for their products.

They felt cheated.

Many big agencies went agile seemingly overnight, because clients simply didn't want to be abused that way anymore.
With knowledge came the realisation, that some parts of the UX process can even be done in-house.

Sketch, Figma and Adobe XD provided a familiar, similar to powerpoint, interface for clients to sketch out some of their initial ideas. Material design, and later other libraries, gave them high-fidelity building blocks to put together the first wireframes - before talking to the UX agency. They started installing tools like HotJar and looking through user sessions - learning.

And you know what? Most of those wireframes were actually quite good. Sure, they did require some changes, and some additional research but the kickoff workshops suddenly became all structure and no filler.

The company from the first story of this chapter actually also reversed their ways. I met the CEO at a conference and he said they go straight to mid-fidelity now, with a much faster, streamlined approach because that's what the modern customer expects.

Those years of stretching out the process to build more suddenly disappeared. And sure, some agencies still write about practically useless parts with pride - like proto-personas - just to show that their process is "full of stuff" but luckily the clients often can see through this.

What's the moral of this chapter?

While the dishonesty of UX consultancies is mostly a thing of the past, be aware to never abuse your design powers just to bill more. If you value your work, charge more per hour, and do great work, but don’t stretch the budgets on purpose just to add unnecessary stuff to it.

That always eventually backfires.

Be honest.

1.4

Evil UX?

Evil UX?

Don't go into UX if your main goal is to do good in the world

There's this notion that UX design is all about the users. We're solving their problems and making their lives easier and better. That empathy towards the users is often cited as the most essential thing.

The reality is often very far from that. The real reason anyone cares about the user experience is that if the experience is bad users will take their dollars elsewhere.

Yes, it's all about the money, 99% of the time.

That means many ux designers are forced to make dark patterns and mislead the users on purpose. A dark pattern, is a way to design something that is making the user do things they don’t really want to do, but without knowing.

They can range all the way from making a popup closing button too small, all the way to misleading privacy labels with double negatives, so the users have no idea what they're agreeing to.

One popular education company is hiding the auto-renewal date from their website - you need to dig really hard to find it. Canceling is also a process full of unnecessary steps.

This is of course because they don’t want you to cancel. And like a gym membership that you stopped going to months ago, it simply bills you every month or every year - automatically.

But when a UX oriented educator is doing a dark pattern like that - making it harder for you to cancel, you can see where the industry is heading.

The bigger the company, the more probable is that you will be doing slightly evil things.

There are good companies out there, mostly small ones, but the majority of them are profit driven and your job will be simply about abusing the user to spend more cash, not helping them.

So basically you may be forced to be evil by your employer. You can of course say no and get fired. Or quit. Or fight over every little decision until your entire office hates you.

1.5

UX Tropes

UX tropes

They say UI is how it looks, and UX is how it works. That's bullshit! Everything, including the way an interface looks, is a big part of how it works. If it's not readable, aesthetically pleasing and clear, people will be frustrated by it and if they can they'll quit using it.

Most of these "UX memes" are simply cliché attempts to gain a lot of likes. I know - I tested some of them on Twitter. For a few months, every now and then I wrote something completely obvious, including the "how it looks vs how it works" one.

These posts are universally praised and liked because many of these tropes are repeated over and over so much, that when you see them again you feel smart for confirming to yourself.

Yes, UI is how it looks. Yes, UX is how it works. Yay, UX!

Don't fall into this trap and try to see through those obvious, clickbaity statements online because by liking something like that you're not learning anything. You're just falling into the trap of group-think which in turn will dial down your innovation levels drastically.

Want another example? Let's talk ketchup!

The two ketchup bottles are often shown as an example of UI vs UX. The glass bottle is UI - because well, it looks better. The Plastic bottle is UX because you can squirt out the ketchup from it a lot easier. No need to tap on the bottom of the glass bottle to get it out, just squeeze it and it’s there.

So what is wrong with this scenario? Let’s start with the obvious, the squeeze and squirt of the plastic bottle most often results with the ketchup droplets being sprayed in all directions, not just onto your sandwich. It’s just as messy. But there’s another, long-term problem with it too. Small plastic particles from a plastic bottle tear out and go into your ketchup. We eat so much plastic because of that, it amounts to one credit card per month right now. There are no studies of how unhealthy it is, but because it’s basically an oil-compound it’s definitely NOT healthy. So that great UX turns out to be a really bad experience for the users in the long run - like increased medical costs.

The moral of this story is to try and always evaluate everything from all possible angles. Don’t accept a simple explanation that it’s supposedly easier to squirt the ketchup from a plastic bottle as fact.

The results speak for how good the method is. If you get a positive result it doesn’t matter if you followed a process or you completely winged it.

2

Design

Design

Let's start with design. When we hear the word design we imagine an architect hunched over a desk, frantically sketching perfect blueprints with an insane amount of detail.

The reality, once again, is slightly different. That architect is solving a problem of a building not existing, when it needs to exist. We're doing something similar in the digital space.

However, our design process, while most often visual, is not fully about drawing things. So how can you design without putting anything to paper (or virtual paper?)

Well imagine your app is missing a feature, and your developers need to know how to code it. But because the feature is easy, in this case it could be as simple as adding one button. You get that idea in your head, and without putting it on paper, it leaves your head as words directed at your developers. Design happens and nothing has been drawn.

But of course in most cases, things we do are a little more complex than adding a button. This is where we start drawing our ideas. They can range from just a couple jiggly lines on paper, all the way to a final, beautiful design done in a design tool like Figma or Sketch.

What do we do?

Let’s start with establishing our own little niche. We are digital product designers. That basically means we’re working on apps and websites most of the time. Those apps and websites run on Smartphones, Smartwatches, computers, tablets, smart TV’s, set-top boxes, self-service kiosks, gaming consoles and other, similar devices.

There is a chance for a UX designer to work on something really crazy like a thermostat, a parking meter, or a ticketing machine. And sometimes, these devices don’t have interactive screens,so you design around a simple, low quality display and a couple of buttons. That is also UX design, but this is very rare. I have worked on a couple of projects like this, but still - 99% of the time we’re doing apps and websites.

And even these fringe devices are getting touch screens now, so their experience is becoming that of an app. In a few years they may not even exist in that form anymore. In the future we’ll be doing Augmented and Virtual Reality apps as well, so it’s definitely good to explore these experiences too. But apps and websites are not going anywhere.

So if you’re wondering what you’ll be working on in the future, it will likely be apps and websites. Just add some AR glasses or a VR helmet to a list of devices and you’ll be fine.

From problem to solution

Design is about finding a solution to a problem in either a visual or non-visual way. That kind of problem solving gets easier and easier the more you do it, because you'll be able to quickly find the right pattern for the solution. UX Design may see some true innovation in AR and VR, but for apps and websites we're at a point that almost everything has been tried and tested. Learning these good patterns will help you quickly identify which one, after a few tweaks, will fit the problem you're facing.

Here's an example
Nobody comes up with login or registration on their own anymore. The good practices for these essential steps of almost every product are already established. We need a couple of form fields - the less the better - and a button. The main difference will be how the form looks, but it always needs a "Forgot password" link, a Login button, and an email (or login) and password combination.

It works in a similar way for almost everything ranging from dashboards, data tables, photo upload and more. It's all about the patterns. Once you learn them, you'll simply pick your brain and find the right mix of those patterns.

True design skill comes with the ability to do this very fast, and also to not overcomplicate the experience.

The rule of clicks

Once you take those patterns and combine them into an interface - a set of steps the user has to take, it’s important to remember one thing.

You should design almost everything with the least possible pages or steps.

Let’s say your user needs to go through 5 pages to fully create their profile. Each page requires them to readall the text, process all the visuals, and figure out what to do.

The longer the process, the more steps it has, the bigger the strain on your users brain. So keep it short, don’t waste the user’s time and brainpower :)

Some may say that if you go above 3 clicks, or 3 pages then you’re already doing it wrong. Don’t listen to them.

It’s not about cramming everything on 3 pages and creating a lot of visual chaos just so you can have those 3 magical clicks. In some processes there’s simply no way to do it in three clicks and that’s fine. Just make sure to break down the process into smaller, digestible chunks.

The best way to do it is to only show things that belong together on one page. One page - one category of information. As an example, in longer forms don't mix the user's address form with uploading their profile photo. These are two completely different tasks, and it's best that they're separate.

We, humans, work better when we have a single task ahead of us at a time. It's simply a lot more efficient and faster. So yeah, in some cases you'll go well above the three clicks and that's okay.

What you should prioritise is the right structure of information and **not overloading your user with too much data at any one point.**

One thing to remember about long processes, however, is to **always show the user where in the process they are - which step are they on and how many steps are there in total.**

I worked on some very long, complex, government forms and what helped the most was when each step was between three and seven form fields and everything complex was also explained with an info box right under the field.

Sure, it made the process longer, but we avoided the cognitive overload and got a 24% form completion increase compared to the previous version of the form that had just four steps but each of them was super long and full of form fields.

The reason for this is simple - when you see a form that's super-long, you're already stressed by the size of it. You expect that huge amount of work ahead of you, your palms get sweaty and you're like - "Nah! I don't wanna do it!"

But if the first thing you see is "enter your email and create a password to proceed" it suddenly feels a lot less taxing. Seeing that, you of course expect subconsciously that there will be more, but you're relaxed. All is fine.

You're thinking to yourself "Yeah, I can do this!".

This approach requires testing in your specific scenario, but generally design should be about removing complexity.

The simpler, the better.

Or Less is More.

3

Design Roles

Design roles

You've probably heard someone introduce themselves at a conference by saying they're "Senior Design Empathy leader and customer journey facilitator".

Here's my advice for you if you ever tried to decipher what it even means. Don't do it! It doesn't mean anything and it's purely an ego thing to have the longest possible title and feel important.

Big companies like to create narrow, niche roles because it's easier to find people who can do one, single job right. That led to the UX world exploding with roles, most of which didn't really make sense.

Let's make it clear, shall we?

There are three main roles in design, so I'm going to tell you about four. You'll get one as a bonus.

Remember, that if you plan to become one of the first three, you'll still need a little bit of an overlap with the remaining too.

As a designer you'll have to understand a bit of writing and research. There is no way around it.

3.1

Designer

Designer

This one can be obvious, but sadly it's not. As stated before, we can design in some high-level ways, be more specific with low fidelity, or be a lot more specific with high fidelity.

Many people in the UX industry don't want to put in the time necessary to get good at UI design, and stop their work at low fidelity, or even simple paper sketches.

I believe a designer should be a person who does all the design, starting with flow diagrams, and ending up with the polished, high fidelity prototypes. That often also includes some basic motion and micro-interactions.

Industry tip

If you want to become a designer, don’t limit yourself and **try to learn ALL the skills of a designer**. You don’t have to be amazing at each one, but you need to know them anyway. That’s the only way for you to be able to communicate with another designer and work well as a team.

3.2

Researcher

Researcher

Researcher is the person coming up with ways to process research data. The job of a researcher is to both create and analyse the results of a research sprint. They also often look at usage data and analytics and provide context to what all those numbers mean. A good researcher can work with numbers and know how they correlate, but can also ask the right questions during user interviews. It's half statistician and half psychologist.

To be able to communicate the research with designers - so they know what to do - it's best if a researcher understands enough design to be able to do some quick sketches to illustrate a potential solution. Then the designer can take it further.

Independent research roles mostly exist in larger organisations. Their goal is to make the company earn more by finding every little potential issue or roadblock on the user journey.

Industry tip

If you want to become a researcher, look at big companies. Most small and medium ones don’t look for research-only roles.

3.3

Writer

Writer

UX writers are the people writing the messages in digital product. That ranges from micro-copy (like text on buttons) all the way to full feature descriptions or onboarding information.

In small and medium companies designers do that part completely, sometimes with some help from the marketing team.

In larger brands there can be a dedicated role just for that. If you decide to take this route, you'll likely limit yourself to a lot less potential employers.

A good UX writer should also know the design tools, to be able to write some of the copy inside the actual designs.

Industry tip

This is a very limited role - most companies don’t need or look for a dedicated UX writer. If you believe this is what resonates with you the most, look for work at the big companies only. If you want to be a UX designer, learning a bit of creative writing can be quite helpful, as it’s all about good communication.

3.3

Generalist

Generalist

And yes, at four out of three I place generalists. That's because it's a role that combines all of the above, and that is a role most small and medium companies are looking for.

A generalist can conduct research, and then proceed to design the entire experience all the way to the final, high-fidelity UI. Oh, and they write the copy too.

It actually makes sense to become a generalist, because that gives the job some variety and reduces communication noise.

If you've done the research and analysed the results, you'll understand how to create a design based on that. If you tell another designer about it, they may lose a little bit of information because people process knowledge differently. It's like a game of telephone - the more people involved, the more distorted the message.

Generalists are especially rare in big brands, but when they're good, they're also the best paid job in the industry. Those true unicorns are what companies are *really* looking for, even when they pretend they want many different, smaller roles.

I’d suggest you try to become a generalist - it’s the best path to success for most people. And it’s NOT that difficult.

4

Design Thinking

Design thinking

The design thinking process is quite famous, with plenty of people from all jobs saying they're doing it to sound more informed and in the know. So what is it? It's a way to add structure, or a plan, to your design process. By no means is it the ONLY way of doing design or something that you always have to do. It's just one of the structured ways of doing it that got the most popular.

Do we use the design thinking process at HYPE4? No.

After years of working on digital products we have outlined our own processes and approaches, so we don't follow the same set of the same guidelines for every project. We do design, test, repeat instead. It is good to know about it however, as it can help you communicate better with potential employers.

It has 5 main steps and we'll walk through them one by one. Chances are you've seen those steps before. They are:

Empathise, Define the problem, Ideate, Prototype and Test.

Or in other words:

Understand the user, find out what the problem is, come up with ideas to solve it, prototype a solution and test if it works.

Empathise

The first step is all about understanding the problem. Empathy means trying to feel what others feel. It's all about stepping out of what we're thinking and really trying to feel the other side's point of view.

So in order to solve a problem, you need to first understand the people having the problem. Why are they struggling with the product? What are their goals?

What is often suggested is to imagine yourself being in the shoes of your user. So stop being you, be your user, think like your user, feel like your user. That's it.

When reading about this step online you can be overwhelmed by lots of complex phrases like getting insights into the user, and phrases that don't really mean anything. Or they're repeating empathising again saying that empathising is about feeling empathy for the user. Duh!

You don't need to suddenly become a psychologist or a psychic, simply try to look at the product from the perspective of your user, however well you can. That MAY (but it also may not) help you define the actual problem better. You know, kind of like asking your actual users about their problems and distilling it into the most important parts.

Define the problem

You cannot solve things that are not defined, right? So we take the information gathered during the first stage and try to merge it together to find patterns. If you listened to some users, maybe there were some things they kept repeating? You take those repeating issues and use them to define what the problem is. If most of your users said they don't know why they should register for the app, for example, you know that either the value is not communicated well, or there simply is no value in it for the user.

They say you should define the problem as a human-centred problem statement, but in reality it doesn't matter. Those human centred problem statements can look good in a case study, but it's only phrasing - and we all know the reality - business always being above people. So you can well pretend it's "human centred" but you don't have to. The ultimate goal is more money for the business anyway. Phrase it however you like.

The goal of this stage is to find out what the problem is. You may already know it before empathising too, because a lot of these problems are found automatically whether you like it or not.

The definition should be as simple as possible: "We need more user signups" works quite well. Then you can create variations of the problem by being more specific like "We need more user signups in our mobile app".

Ideate

Ideating is coming up with ideas. Some people simply call it thinking about the solutions.

We have our problems defined so now we ask ourselves “how do we increase the user signups in our mobile app?”

Ideating is coming up with ideas as a list. There are supposedly no bad ideas, so they’re saying you should write out ALL your ideas, even the really stupid or silly ones. This is actually true, because if you’re working in a team, someone can get inspired by your outrageous or silly idea and use it as a base for something good.

There are many ways they say you can use to ideate, but in reality it doesn’t matter as it’s just about the quantity of ideas. By now, however, I hope you are starting to see that the mystical design thinking process is nothing more than the completely logical way of solving any problem, only wrapped up in some pretty UX words. Yes, you need to understand the users, then you need to define the problem and then think of some potential solutions.

This happens automatically in most industries, but in UX it had to be specifically split into processes, just so it’s a tiny bit more complex.

Prototype

Prototype means taking some of the best ideas from the previous step, and making a quick, dummy version of them to see if they make sense.

This part can be quite confusing.

After a few years of doing design, you will be able to throw out most of the bad ideas before you prototype them, but initially you may end up doing a lot of prototypes often seeing the uselessness of this action.

While in most cases it is said to test this stage within the design team, I would suggest to go with as high of a fidelity as you can in your prototype and show it to real users. It's a lot faster and more agile, and you get more relevant feedback from it in the process.

Each prototype is a potential solution to the problem from step 2. Once you test a couple of them, the team can then decide which one should they go with as their main proposal. It is encouraged by most UX practitioners to create a lot of prototypes, and once again - the more you do it, the less you will actually create as most of them are pretty useless. The more you grow as a designer, the easier it is to own your choices and make the right ones.

Test

Remember Design-Test-Repeat? Well here we are.

Design thinking ends with a test stage in which you see how users respond to the prototype you picked.

Chances are your prototype will be an improvement, but it won't be great yet.

This is how the test stage feeds back into all the other stages before it. You show your prototype to users and you note out what they say. Then you use what they said to come up with changes to your solution, or in some rare cases a different set of problems.

And this goes on and on. Round and round.

4.1

Should you use Design Thinking?

Should you use it?

My take is - don’t worry about it. What you should do is to simply understand the problem, propose a solution and test whether it works.

You can use Design Thinking for that, but you don’t need to specifically name your process because it’s only a way of getting to the solution. The solution itself is what matters, not whether you tick a couple of boxes from some process. So if you like it because it gives you a mental structure to tackle the issues - go ahead!

Just remember that it’s basic logic and at some point - the more you do problem solving - many parts of this process will be done automatically without thinking how some specific process step is even called.

Do I use it? No. I can’t remember a single instance when working with a client where we had to specifically say “we’re doing empathising now, and then we’ll do defining, followed by ideating”.

Also remember, that to many people (including your clients) some of these words (like ideating) can be quite meaningless so you can re-name the steps of this process to plain english, because it’s all about what they help you achieve.

5

Research

The truth about research

“If you don’t do research, it means you’re building a product for yourself and not your users.”

Back in 2015, when I was running a UX research company in Dublin, Ireland we used this as a slogan. It used to be difficult to convince businesses to do research, as they were mostly focusing on outcomes - i.e. some tangible designs they can look at and see where their money went. It’d be a fun twist to say that research is now well understood and widely accepted, but it’s not. Still many small and medium clients will push you towards design right away, dismissing research as a waste of time.

It used to make me really angry, until I realised a couple of things.

First off, it’s impossible not to do research, so the problem may lay in the way you portray it to the client. The other thing is that unless you’re working on a truly innovative, one-of-a-kind product (which is very rare) chances are you can use the collective knowledge of all the apps and websites that came before and still make it work.

So while research IS important and is definitely worth doing, it’s still considered less desirable and if you plan to do it you need to learn some tricks to be able to push it. But first lets explore what UX research even is.

What is UX research?

There’s this notion, pushed by many senior UX people, that UX research is a complex set of processes and you need to be a super-genius to be able to comprehend it all. The reality is far from that.

UX research should NOT be placed in the same row as rocket scientists and researchers curing diseases. Those are real research, with lots of math, logic and creative experimentation.

Moving a rectangle that says “Log in” to another spot to see if more people click it is baby research at best. Once you stop thinking of UX research as if it was a super-difficult science, you’ll be liberated to use it successfully for what it was made to do - making good design decisions.

There are many kinds of UX research and we’ll go through most of them in the following pages but start with the mindset shift. Research is ASKING your users some questions, and then interpreting their answer to learn what you should improve in the product and how.

It’s a similar kind research to a waiter asking how you want your steak cooked. Medium-rare? That’s perfect!

Research is asking questions

Once you start thinking of UX research as simply asking the right questions you'll be able to both better understand it and use it.

If you take just one user, let's call him Bob, and you ask about the problems with the app, you're already doing research.

Sure - it helps to ask more people, as just one point of view can distort your vision - but even just saying "Hey Bob, tell me why you're not happy with the app?" is research enough.

That simple way of asking questions and talking to the users as equals is the biggest UX research superpower you can get. Once you understand it's all about simply asking - human to human - you won't be afraid to come up with all kinds of questions.

No need for a white lab-coat, just be yourself.

Research is just asking questions. The more you do it, the better you'll get at it because those user interviews are surprisingly similar.

Qualitative vs Quantitative

In your design journey you'll come across Qualitative and Quantitative research types. What they are is simple - quantitative means that it's possible to quantify the results - i.e. it's all about numbers and counting them. That's what it means to quantify.

A good example of quantitative research is a series of simple YES or NO questions, because you can then calculate how many % of users answered what.

QUANTITATIVE, or number-based research is useful because most products already have some sort of simple statistics you can use. If the data is there, you can use the numbers to uncover a story. But of course being confined to just numbers severely limits what you can understand because they don't give you a lot of context. Sure, you know 37% of users find the login process confusing, but you don't know WHAT is confusing about it.

To solve this problem you need **QUALITATIVE** research. What it means is that you ask people open questions, so each answer is unique to the person. They can just write a few words, or share pages upon pages of their thoughts and insights.

The main difference at the end, is that quantitative data (numbers) you can group into data sets and analyse together. With qualitative data it's not as easy because each answer is a data set of its own.

There are many specific research methods in each of those two main categories, but what's really important to know is not the methods but rather what are they often used for.

Quantitative (number-based) research can show you some high-level trends but won't likely guide you directly towards a potential solution. Having people SAY what their problem is (qualitative research) is a lot more helpful but also time consuming as you need to analyse those answers one by one.

Quantitative is good to start. Qualitative is where most of the actual problem solving happens and I believe it's the most important part of user research.

How to ask the right questions?

Qualitative research takes time to learn and - like everything - happens with practice. If you want to get good at it, normalise asking your users questions as often as you can. If you're in a startup, do open-ended Twitter surveys weekly, or even daily. Just practice it! And make sure to use the most interesting responses you got as a base to ask further question and dig deeper! It all comes with practice.

Talking to users

If you’re not a fan of research and your main goal is to do design, you should still try and just talk to your users. The less of a scientific chore research becomes, the easier it gets to perform. Because in the end that’s all it is - a soft skill of talking to people and finding out what their problem is.

The line between user research and market research has blurred quite a lot in the recent years. It was a normal thing for a dedicated marketing or product team to research if a new product or feature is needed. Now it has merged with UX research which adds to the confusion. Yes - sometimes UX researchers come in before a product is build and determine whether your company should even attempt to build it.

But don’t worry, it’s all really very similar so stay curious and ask people questions for as long as you need to get answers that will help you make a better product.

Enjoy research, because that kind of research is not science.

It can even be fun.

6

The things we do

The things we do

In this chapter I'll briefly go through many of the UX methods and processes - often with difficult to understand names - and show you whether they're actually used beyond your junior portfolio.

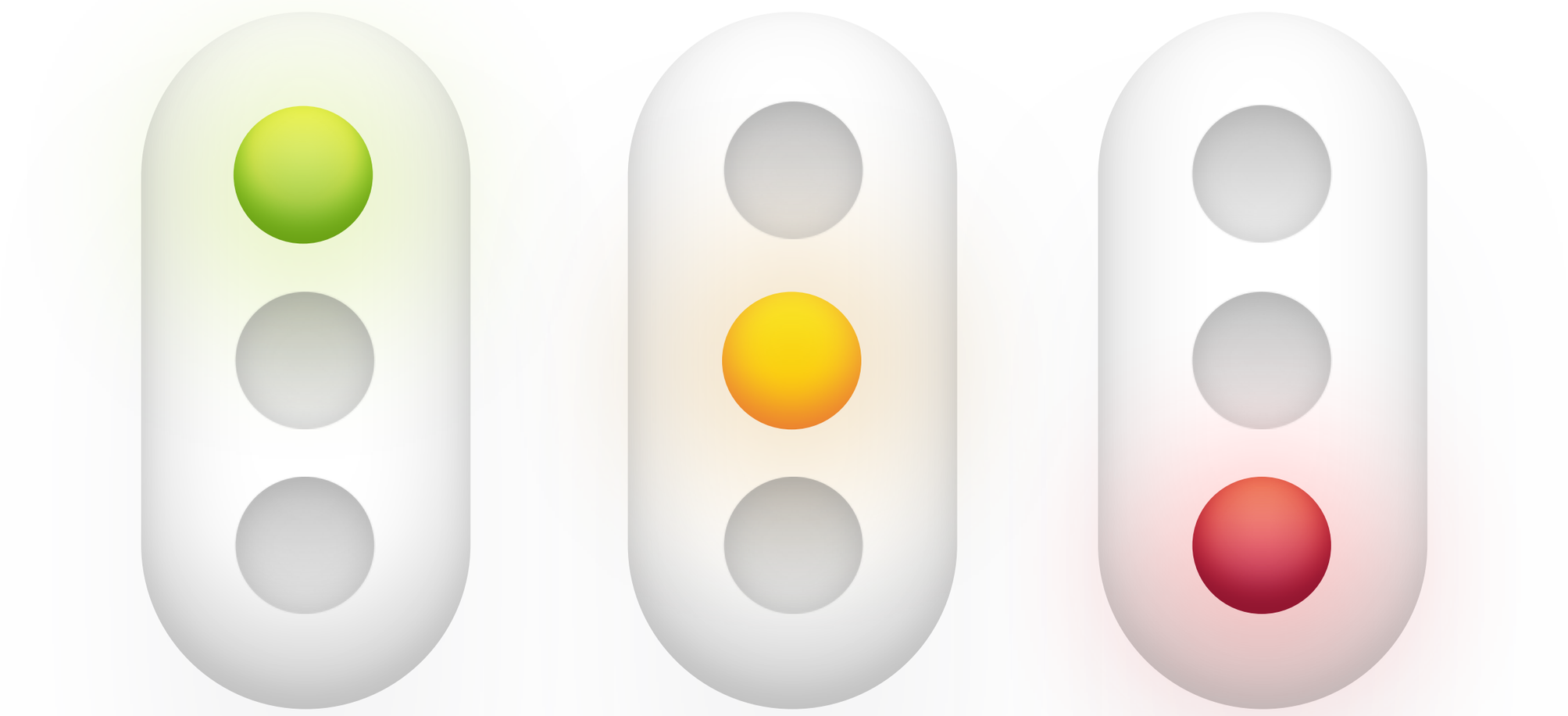
Keep in mind, that the fact that we don't use something, as a company, doesn't mean it's not used at all - but I tried asking a lot of similar sized businesses and some larger ones, so in many cases the traffic-light rating is pretty accurate.

Of course that doesn't mean you shouldn't learn these things when you're a beginner - but some of them definitely require less focus and energy, as they likely won't be used in any real-world scenarios.

Treat it as a guide, but make sure to check what processes your future employer uses and how because it's not 100% universal - if anything ever is.

Note: This is the first batch of methods and processes that was requested by my readers. Based on future requests I will be adding new content to this chapter regularly!

Traffic lights

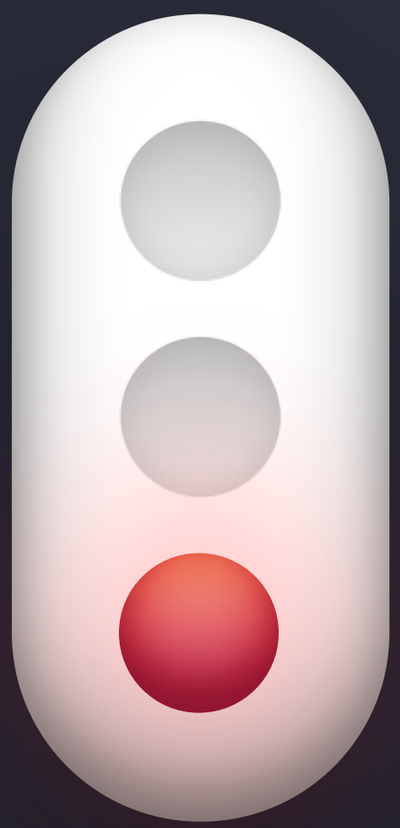


To differentiate them, I decided to use a familiar traffic light pattern. Next to each term there will be a colorful dot. If it’s **green**, it means that this technique or concept is essential and commonly used in daily work.

If it’s **orange**, it’s something that’s sometimes used, but not essential for many projects.

If it’s **red**, it means that it’s a part of the process mostly used in UX Courses, but rarely used in any real work. Many of the things in red, however, are necessary for your junior portfolio.

Ready? Let’s go!



Personas

Rarely Used

Christopher Evans

27, Sales Manager

About Christopher

Context of use

His goals

His frustrations

Personas

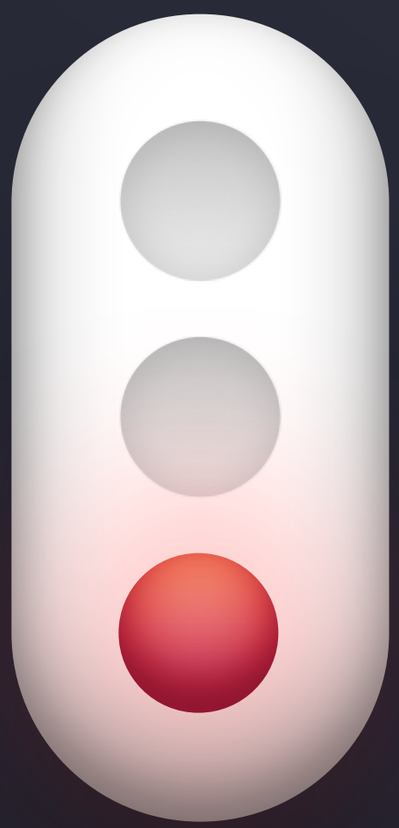
Personas are a great way for junior designers to learn how to empathize with their users. And by empathize I mean how to try and feel and think what the user feels and thinks.

Yes, that makes sense at the early stage, and especially when building a portfolio. Personas to many recruiters are an important “check” on a list of things to look for in a junior portfolio so at this stage definitely make sure to use them.

But ...

They are very rarely used in any real projects, however, because most of the data on a typical persona sheet is irrelevant to any real findings. And the time you spend thinking about names, looking for photos or writing a description can be better spent actually analysing the market segments and finding out who your users are.

In our work we rely on customer models, which are data sets of only the relevant information. There’s no need and no time to create “artificial people!”



Proto-Personas

Rarely Used

Chloe Nielsen

26, Product Manager

About Chloe

Context of use

Her goals

Her frustrations

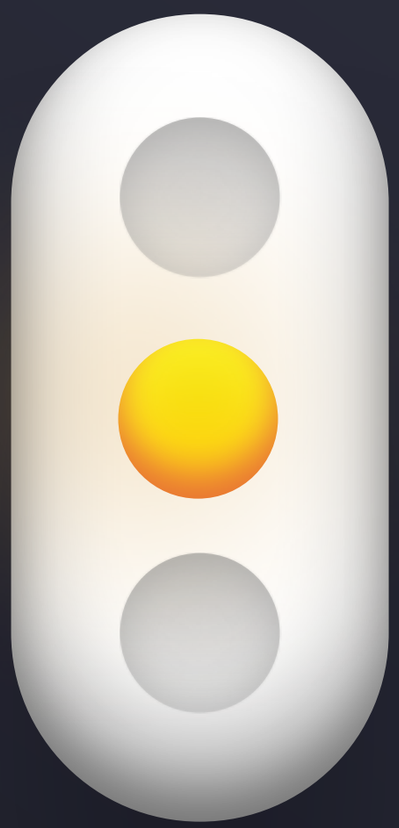
Proto-Personas

Proto-personas are fake-people generated for the initial client meetings. They're not research based, and their main role is for the client presentation to look better. It means they're just slides, nothing more.

While regular personas are devised based on what you know about your user groups, these are just hypothetical assumptions at best, and filler-content in most cases.

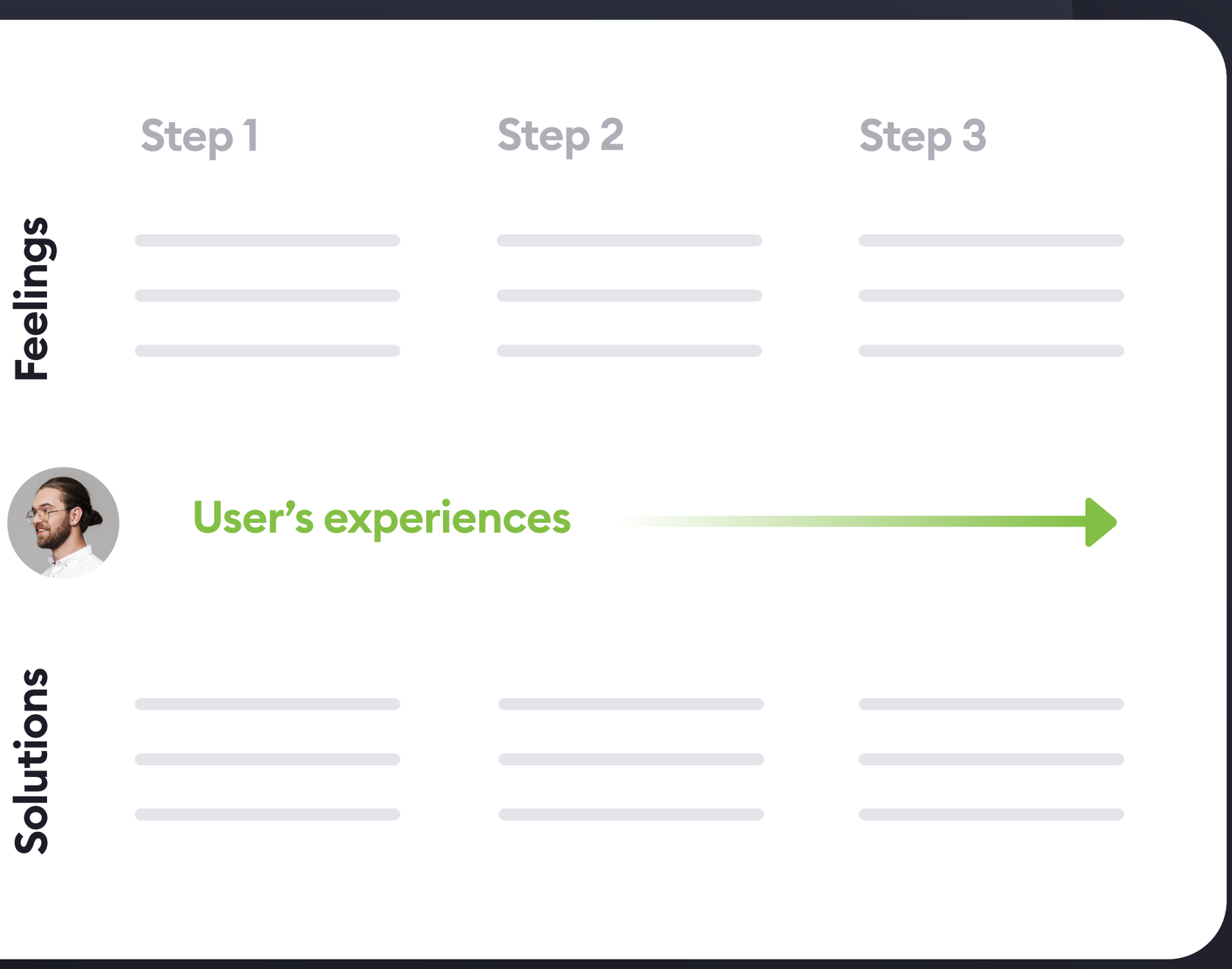
The often used reasoning for creating these, is that they're so vague they're likely to fit an actual user group, even if accidentaly.

It's one of those elements that gave UX a bad reputation among clients, as more and more clients are starting to see them as tactics to "pretend to do more work" and simply charge more, while discarding all that extra work as useless.



Customer Journey Maps

Sometimes Used



Customer Journey Maps

Customer journey map is a timeline, often presented as a table, showing the entire interaction of a user with your product.

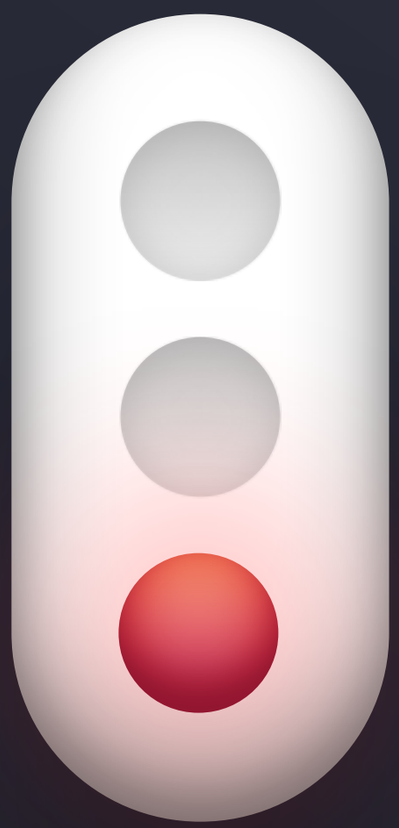
It usually starts with how they find out about it, and ends with user achieving their goal (or not achieving it and complaining to customer service).

The problem here is similar to the one with personas - most of the time hypothetical journeys are either already well known (especially for simple products) or a market segment group is established.

It can be helpful if the product never had a journey mapped in any way (including as a simple set of bullet-point steps) or to make sure that everyone is on the same page.

In my career, we've mostly worked on very simplified maps based on market segments / user categories. For example we can create a typical scenario of a step by step how a new user approaches the product vs a second option where a power-user is using it. That can help with getting a perspective and understand we may need to modify the messaging to cater to all kinds of users.

It's not a very popular technique in real projects though.



Customer Experience Maps

*Rarely Used in UX**

** They are often used by sales and product teams and delivered to the UX team*

Customer Experience Maps

Customer Experience Maps are similar to the Journey Maps we outlined before. The main difference is that they try to talk about the entire experience, and include marketing, branding and sales in the evaluation.

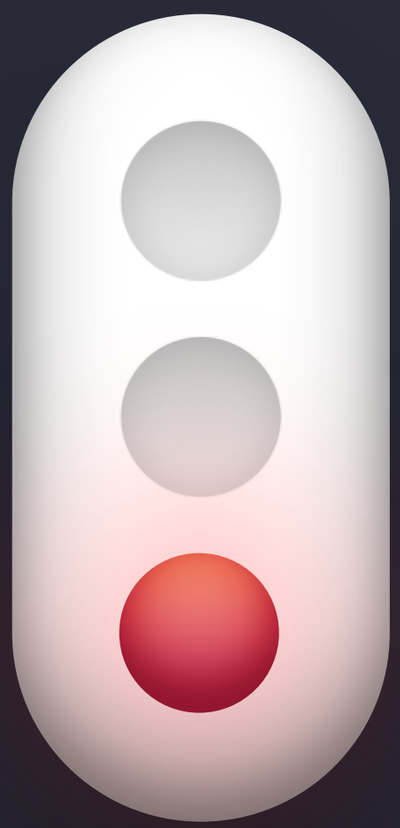
That basically means it’s a much wider scope of a journey map and often something that is done by marketing or product teams and not UX people.

This process is good to have, but it’s not our responsibility to do it. Some UX agencies add it to be able to bill some extra hours while involving the marketing and sales teams anyway.

While useful to UX designers, don’t think that you need to actually be doing these yourself.

However, breaking down customer experiences into a timeline style easy to digest steps is something you will be doing quite a lot as a designer. We just don’t call them “maps” of any kind and they don’t have to follow any specific procedures. Their goal is to clarify and simplify sets of data so you can see the bigger picture and faster find solutions to the problems.

What you call some of these processes is less important than how you understand what the user does and why.



Empathy Maps

Rarely Used

Says

Thinks

Does

Feels

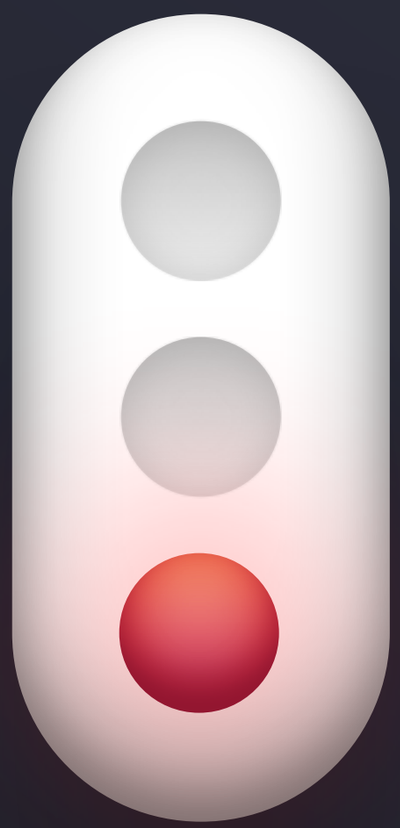
Empathy Maps

Empathy maps are usually a part of the design thinking process and are based on previously created personas. You create one map per one persona, often during a workshop with clients and stakeholders to get them engaged in the process.

The idea is to define what the persona sees, hears, says and does and later also thinks and feels.

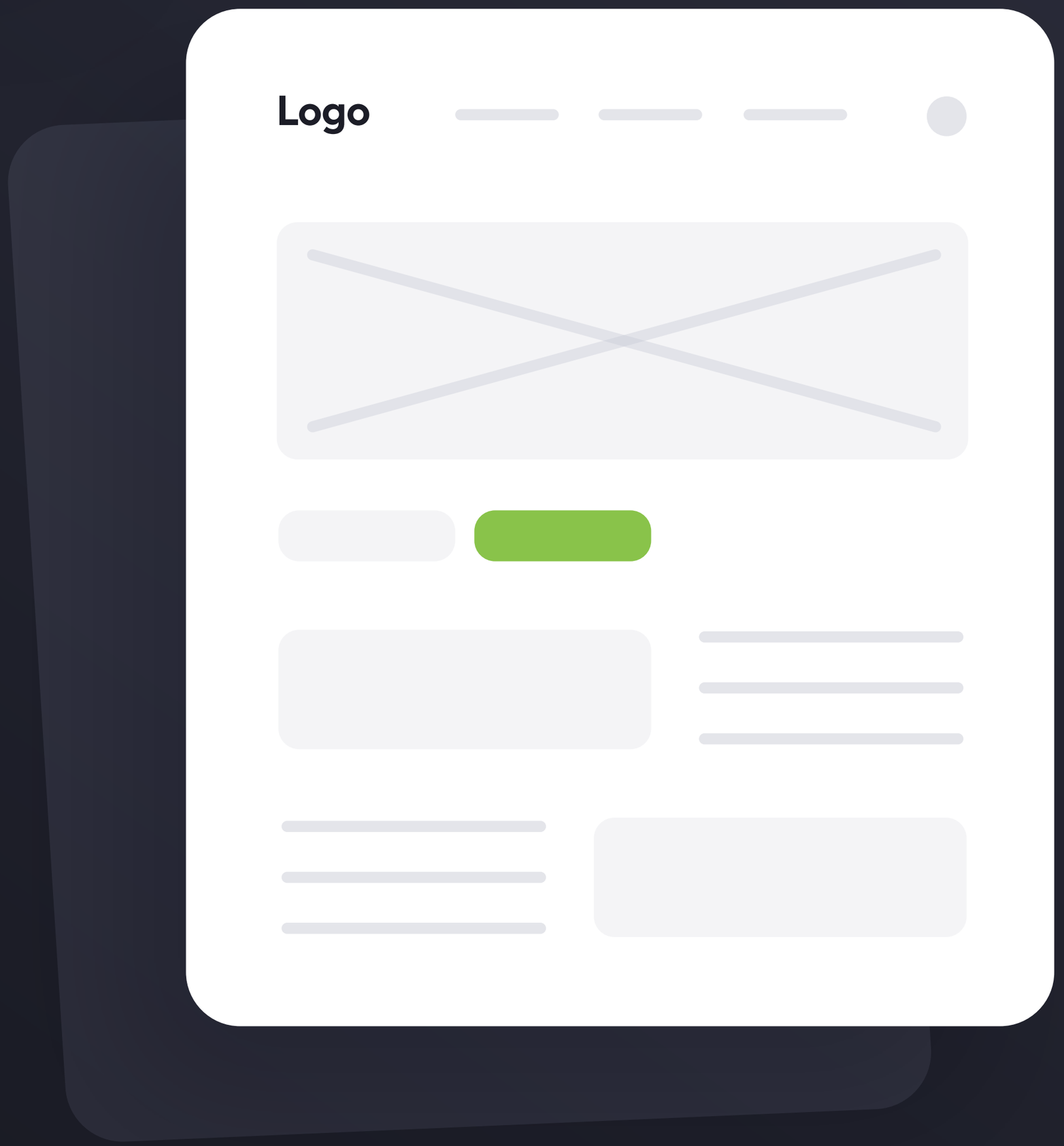
It’s a purely theoretical exercise that after a while turns to most of the six fields having the exact same answers - like the person **feels fear of making a decision** and if you look at examples of empathy maps online, you’ll find out that most of it is the same stuff over and over again. While they can be helpful to establish some initial goals for the product, this entire process (of both personas and empathy maps) is now replaced by creating customer segments and simply asking the users how they feel about particular features. Because then you’ll know what they really feel, not what you assume they feel during a workshop.

We’ve worked on over 500 digital products since 2011 and haven’t used empathy mapping even once in that time. This entire process can simply be done faster, and as a result much cheaper to the client. They are “good to have” in your junior portfolios.
In the real world? Not so much.



Low Fidelity Wireframes

Rarely Used*



** We have done typical lo-fi wireframes about four times in the last five years. There are some cases in which they are necessary, especially with complex projects but in most cases they're not needed anymore.*

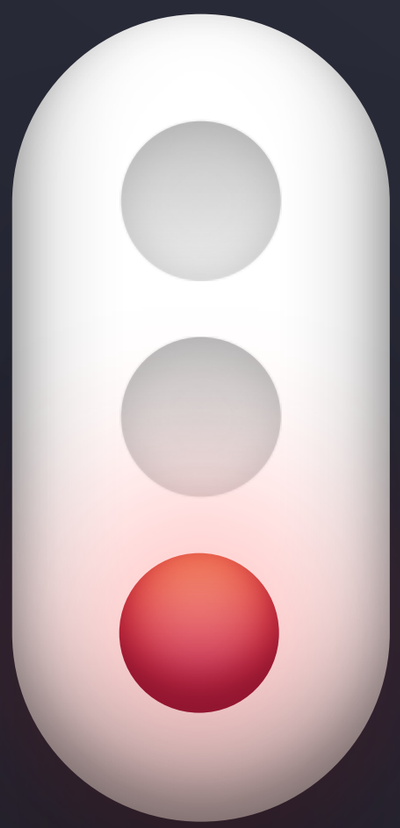
Low Fidelity Wireframes

When you search for images related to UX, a low-fidelity wireframe is often the first result you see. They main advantage of lo-fi wireframes was that they're quick (and thus cheap) to create. The problem however was that most users and a lot of stakeholders and clients don't understand the fidelity decrease.

Even when it's explained to them, they still perceive them (often subconsciously) as "ugly project" and that turned out to skew their overall experience to be a bit more negative.

Testing Low Fidelity Prototypes on users (clickable flows composed of Lo-Fi wires) used to be a very popular initial phase in most product design sprints, but over the years people realised that the results we get from low fidelity can be off by way too much to justify doing them. Another reason for doing low-fidelity was that people don't focus on "fonts and colors" as much, and can pay more attention to the flow - but that argument lost with human nature - we don't care about ugly wires.

With modern design tools and easy to use design libraries, going mid-fidelity is just as fast right now and yields much better results. Quick low fidelity sketches can be a good idea for very complex projects and to determine whether we understand it the same way the client does, but they're not really a thing anymore.



Low Fidelity Sketches

Rarely Used

Low Fidelity Sketches

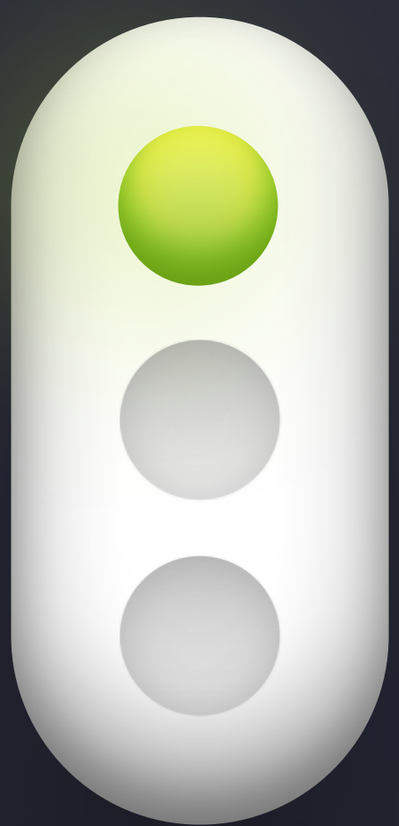
This is another relic from the past and a way agencies used to be sneaky and charge clients more. The flipchart meeting scribbles that barely resemble wireframes were used as a way to explain some basic navigation concepts to the client and other meeting participants. But because in most cases there was no text on them, just scribbled lines, they were pretty useless after the meeting.

Taking a photo “for later” generally meant you come back to a series of badly drawn lines with no context.

While they may be useful to the person doing them, as it may help them to think visually, they’re generally a time-waste and filler that is practically never really used in any real work.

It can be used to add some more billable hours to a meeting, if that’s your goal.

Making slightly higher fidelity (with real text on buttons and cards for example) wireframes can be done faster and much clearer on a computer than on a flipchart.



User Stories

Often used in complex projects

Chris, 27
Sales manager

As a (role)...

I want (goal)...

So that (gain)...

Chloe, 26
Project manager

As a (role)...

I want (goal)...

So that (gain)...

User Stories

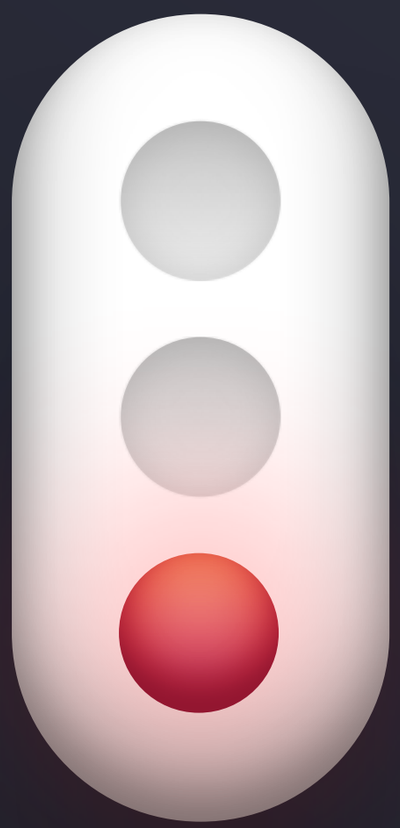
User Stories sound like another complex UX thing, right? Well, luckily the structure for them, that we've been both using and seeing in many projects is super simple.

It's a little bit like a customer journey map, but written as a short story - literally a couple of paragraphs. You create them for all types of users - so for example the regular customer, a moderator, an administrator and so on.

In the user story you simply write, in easy to understand language, what the user does in the product step by step. Don't focus on how they get or install the app, just outline the actual interaction. For example:

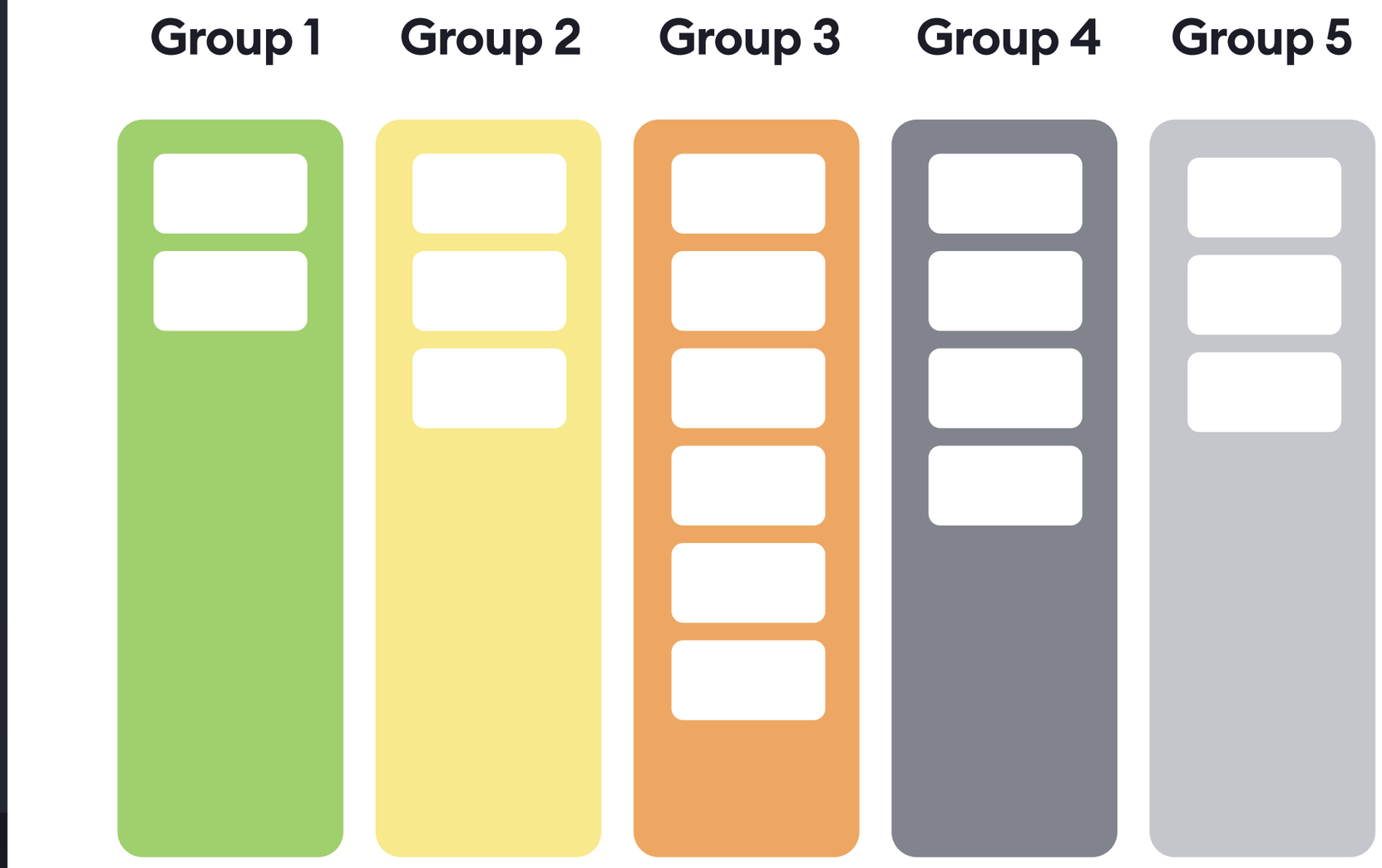
"The user enters their email and password and creates a free account. They have to click on a confirmation link in the email to activate it. After that's done, they can fill in the rest of their profile right away for extra discounts, or start shopping right away. They're asked to select a couple of product categories they like, so we can tailor the products that they see".

You usually request these from the client, just to see how they understand what they think their product should be doing. Then, together, you fill in all the extra details to iron out the experience.



Affinity Diagrams

Rarely Used



Affinity Diagrams

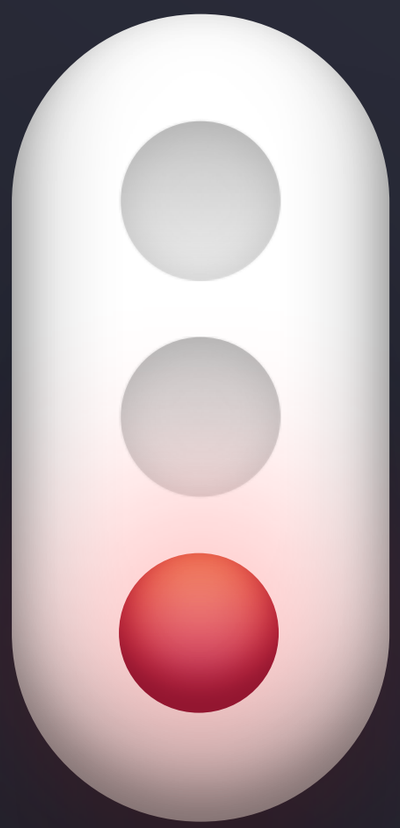
Affinity diagram, or Affinity mapping, is a way of sorting data into a couple (usually five or six) main categories. It’s often presented either as a data table, or as clusters of square post-it notes in various colors.

As with most other methods in typical design thinking processes, it serves as a way to make the information coming from many different people easier to process.

In real work this process happens automatically and is not called by any specific names. Grouping information is usually as simple as pasting some bullet points into a couple of sections.

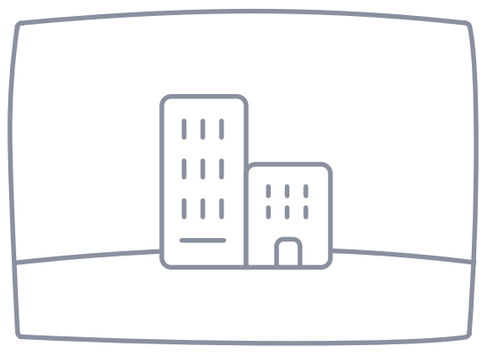
Affinity diagrams are almost never a client-deliverable, which means you don’t make them to show to the clients. They’re a way to make your ideas clearer and potentially pick better ones. Because of that they’re not really a part of any real work. The process of combining data into categories does occur, but happens without thinking about it, and often it happens in your mind, without creating anything visual.

They are, like many other processes, a good thing to showcase in your junior case study to show how you evaluate ideas. Outside of case studies they’re not really used.



Storyboards

Rarely Used

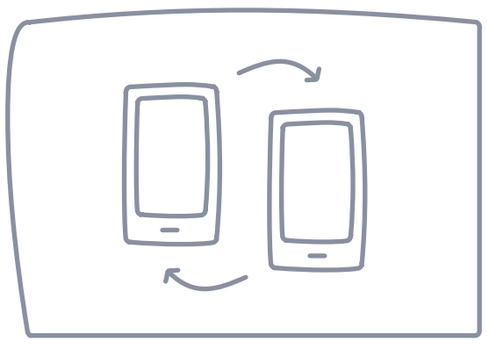


Picture 1

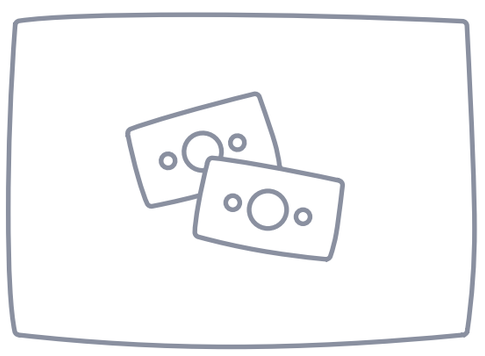
Picture 2

Picture 3

Picture 4



Picture 5



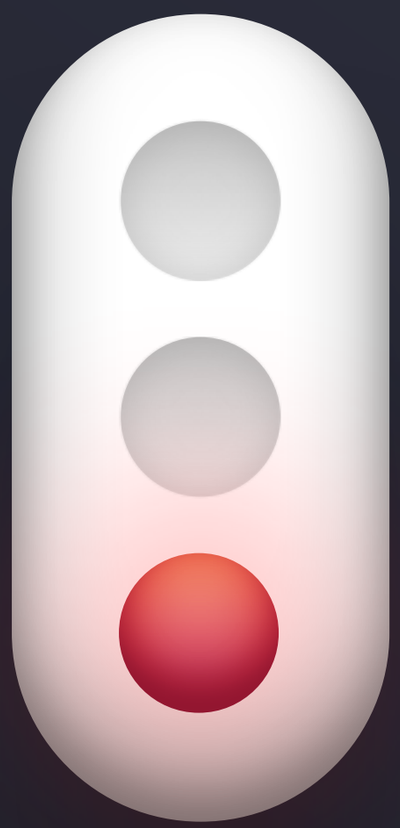
Picture 6

Storyboards

Storyboards are a concept coming from movie sets. They're like a simple, usually 8-frame comic book. In those four or eight little squares they try and show a specific process. In UX it's usually the user trying to achieve something.

It's basically a user story, but in a visual form. It can be a good idea for a case study, but avoid even doing it in every case study you have, as it doesn't really contribute that much. Unless your illustration skills are really high, in most cases it ends up being a bunch of not-that-pretty images with a story that could've been a few bullet points.

Are they used in the real world? Not really. On client meetings a story can sometimes be presented, or acted out, but almost nobody really creates those drawings as it's considered a waste of time and resources.



Crazy 8’s

Rarely Used

1

2

3

4

5

6

7

8

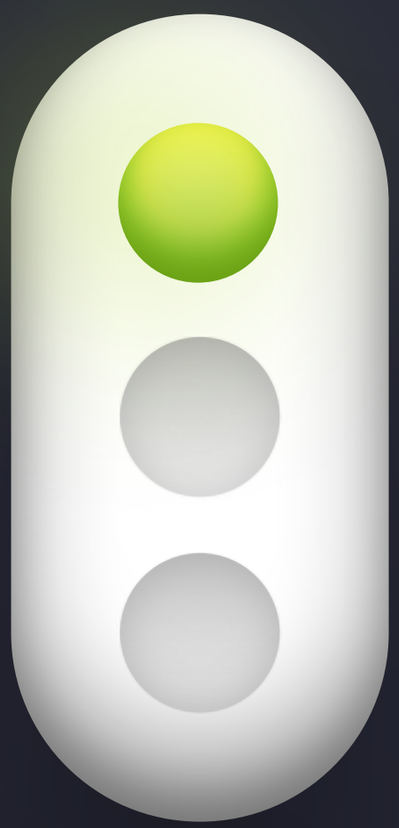
Crazy 8’s

You take a piece of paper and fold it a couple of times first. Then you unfold it again and have it naturally divided into eight sections. Then with only one minute per section, and a running timer, you sketch out a solution to a problem or a product flow.

The idea is a little bit like storyboards, but because of the limited time you don’t have the time to overthink things. We already know that our first ideas are often the best ones, and this exercise relies on that. Often the ideas sketched out with such limited time are somewhere between bad and interesting. When they’re interesting, and when it’s done in a team so we have a lot of the ideas, we can then try to merge the best ones into a single concept.

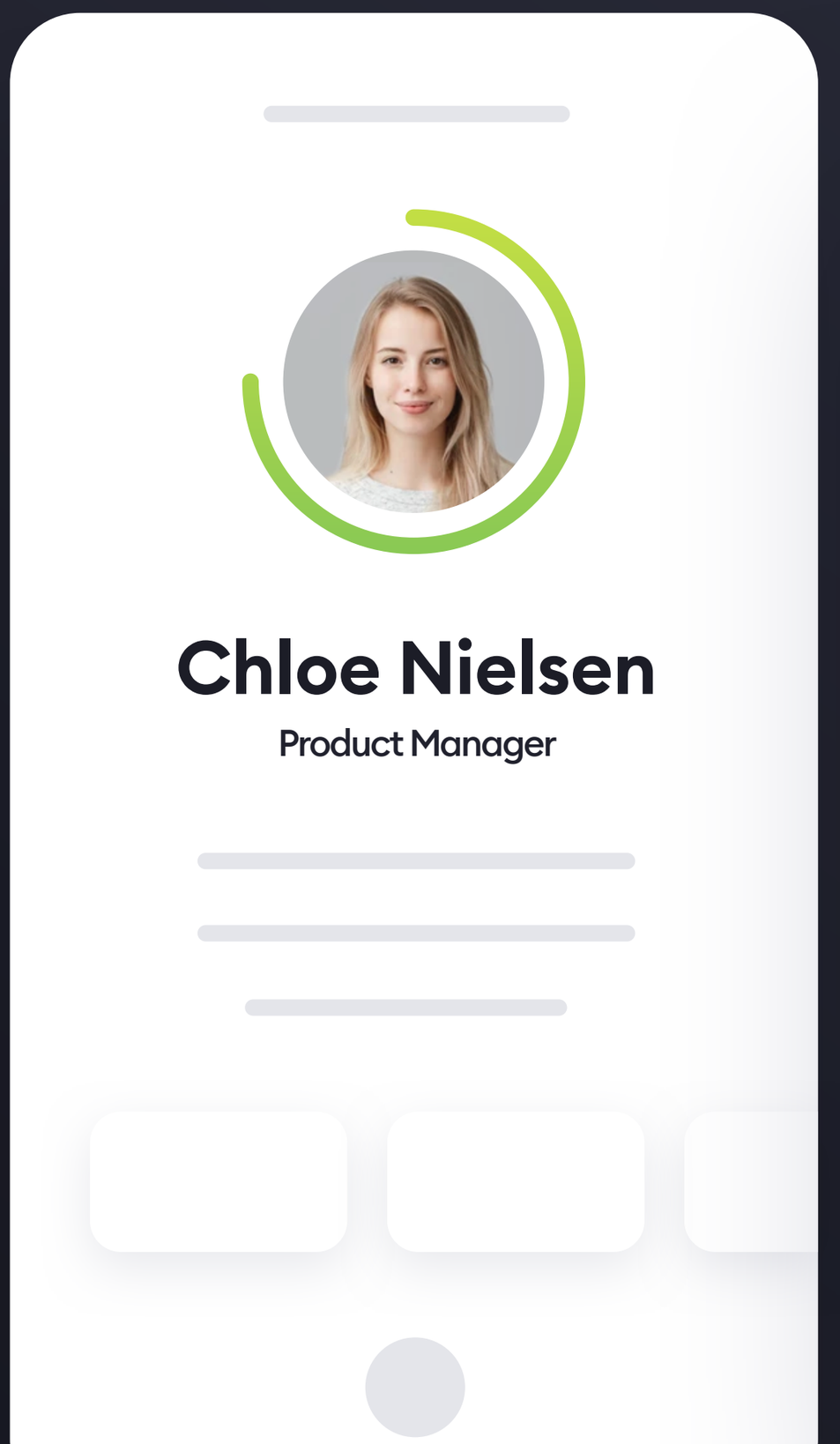
At least that’s what this exercise is on paper. In my career we have never, ever worked this way because, especially now, you come up with a solution to a problem by quickly finding the right “pattern” in your mind - most UX problems have already been solved somewhere.

You can add crazy 8’s to your portfolio case study, but don’t add it to each one, as it’s not really something that brings anything of value to the table. Just another way to quickly generate ideas, which after a while simply happens automatically.



Competitive Analysis

Almost always used



Competitive Analysis

Competitive Analysis simply means looking at other products from the same category and seeing “how they do it”.

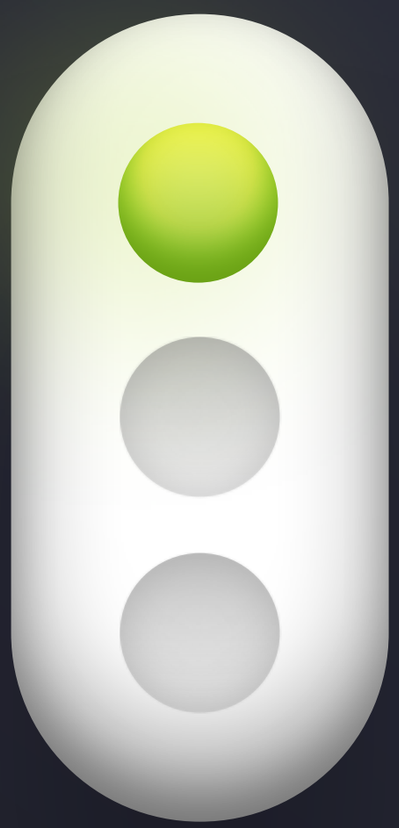
We do this with practically every project, to figure out what others do well, and where we can improve on what they do and do it a lot better.

Evaluating the competition is a very useful process!

Also, at this point most products are simply sets of features from other, existing products - combined in a new way. That’s why using the collective knowledge of others is always a good idea - no need to reinvent the wheel.

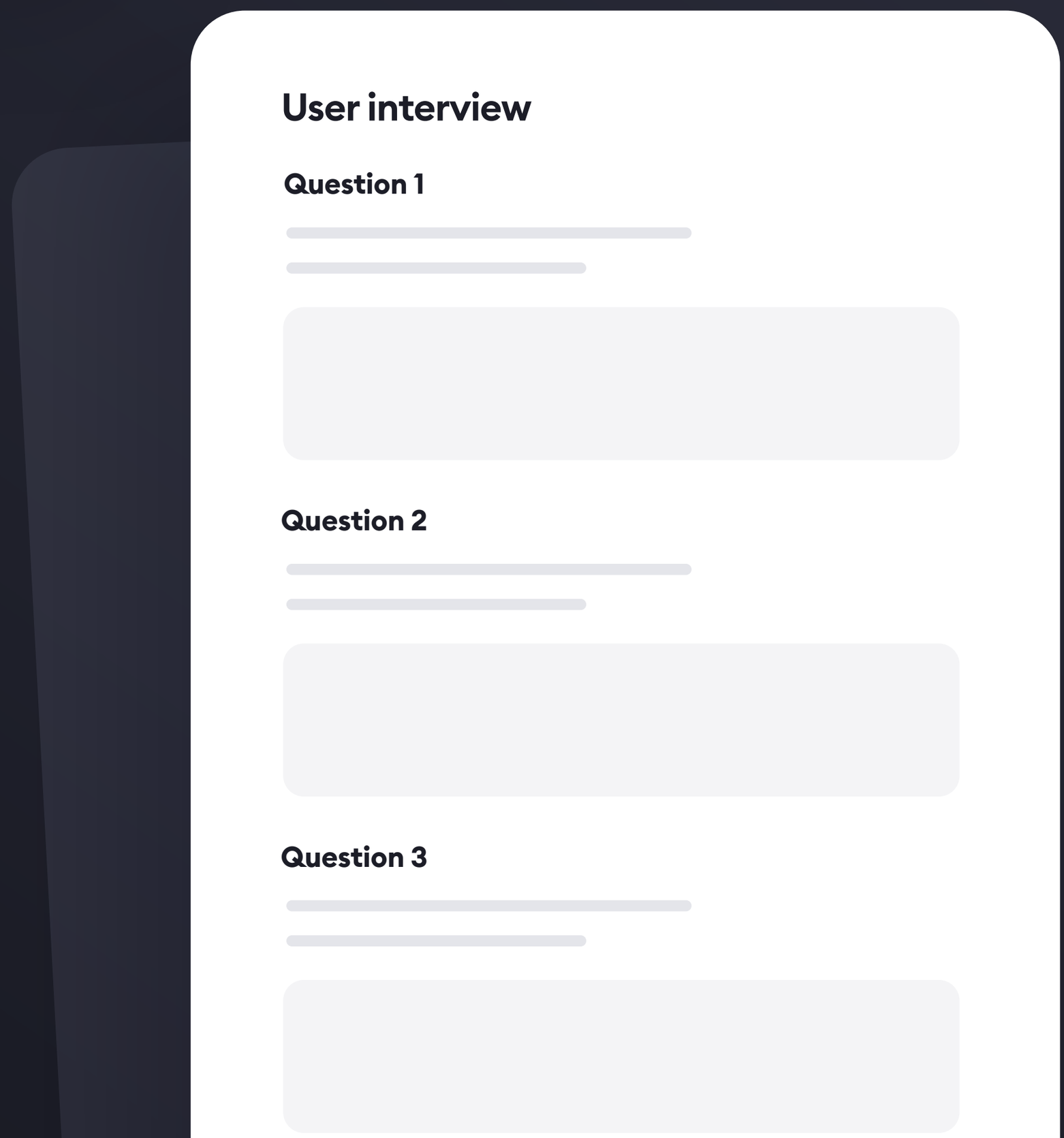
You can both analyse HOW other apps or websites accomplish a task (what steps do they use to guide the user to completion) or look at customer feedback in the app stores.

This is a qualitative approach, meaning you need to read the detailed comments as they often show what issues the product has. If there are many repeating comments that all talk about the same problems, you know what to avoid in your product. It’s as simple as that!



User Interview

Almost always used



User Interview

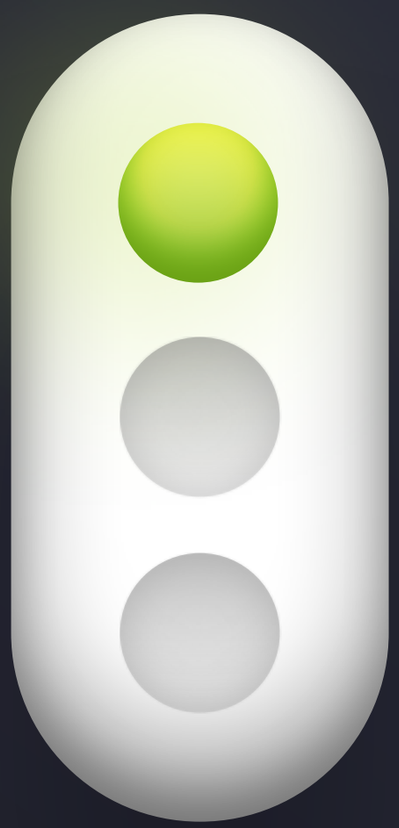
It is exactly what it sounds like. You find a group of users and ask them questions. To make it easier to process afterwards, it's best for the questions to be the same, so you can focus on the answers. It helps you to find out about the user's needs, experiences, attitudes and goals.

There are many templates on how to do these interviews, whether they're in form of an online survey, conducted in person, or over a conference call.

The most important part, as with any user research is to guide the users but never suggest anything to them. They either know the answer to a question or not - and if they don't it's also a valuable insight for us. It means we may have to rethink that product or feature so it's clearer.

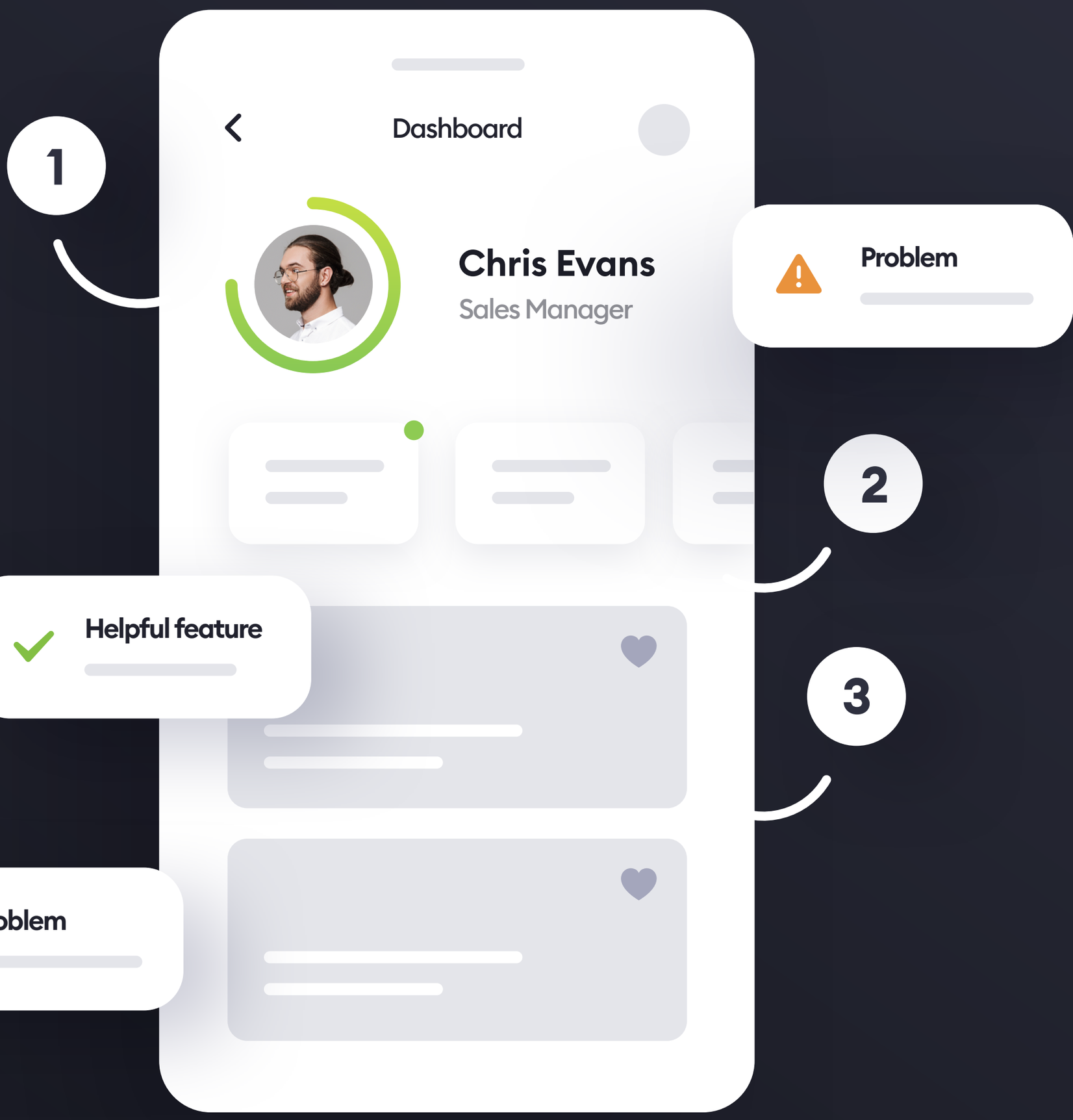
We do this quite often, but mostly as an online survey with up to five open questions. With time, you'll be able to refine those questions so that you'll get the most useful information from them, so don't worry if your first interviews don't bring a lot of useful insights. You'll get there.

Keep them short, under 5 minutes to complete, as those tend to yield the most submissions.



Usability Testing

Very often used



Usability Testing

This is a very useful method, but when described it’s unnecessarily complicated.

A usability study is a one on one session with the user and (most often) a prototype of the product. The interviewer has a list of a couple of tasks they want the user to complete. Those tasks can be completely separate user flows that don’t need to connect - you can just switch the prototype for each task to a new one.

They’re usually quite short, for example: how would you invite a friend to the service, starting on your profile screen. That is ONE action. A full study can have a couple of simple actions like that.

When planning which ones to check, make sure they’re actually relevant to the product success - it barely makes sense to test whether they know how to agree to the terms and conditions. Registration on the other hand is almost always a must!

It’s important to record AND evaluate mentally what the user does and what they click/tap, but never suggest anything. If they don’t know - they don’t know. Redesign and come back for another study.

It’s a very useful exercise and we try to do it with at least half of the consumer facing products, but there are a couple of things that I found that can negatively impact the experience.

Whenever you can, use a high or at least mid-fidelity prototype for usability studies. Even when the users understand what a low-fidelity wireframe is, their peception is subconsciously skewed by “using an ugly product”. It’s just how our brains work.

If you’re making drastic changes, and testing them on the same user, you can easily fall into a bias trap. It means the user will be so used to the previous prototype, they’ll say it’s better even when it’s clearly not the case. A good practice to resolve that is to test both on the same users you tested before, AND add a new control group of a similar size to every radical redesign study.

Don’t test obvious or not quite necessary things like “how would you change your profile picture” because the users feels like it’s a functionality that’s not that important in most products and it gets reflected in the results. Try focusing on the most relevant flows for the product to serve it’s purpose. If you’re making a ride-sharing app, test the actual booking of a cab as the most important process.

Usability testing works great if it helps you improve the really important parts of the product. Only do it when you feel it will contribute, never to just “add it” to the project scope.

Also, don’t get too attached to the name. You can call it anything. I usually call it a “Product vs User” test. What’s important here is not the industry jargon, but rather whether the user actually knows where to click or tap to go forward.

There’s always some bias, so try to minimize it but don’t feel bad after the fact - we’re only human and you should treat it like watching another human ride a bike (use your product) - observe, take notes and get over with it.

This method also works best when the prototype is as close to the final product as possible. Sure, you can test in a Sketch or Figma prototype, but from my experience it’s actually better to code an HTML prototype. It doesn’t need a database, all the content can be hard-coded into it. But with that you can add something like hotjar to gather some extra heatmap data from it next to your regular observations.

The more it looks and feels like a real product, the better the outcome. Of course if the product is already out, you can also test on it. Just make sure the user doesn’t feel like they’re using a low fidelity demo.

Eye Tracking

Rarely Used

Eye Tracking

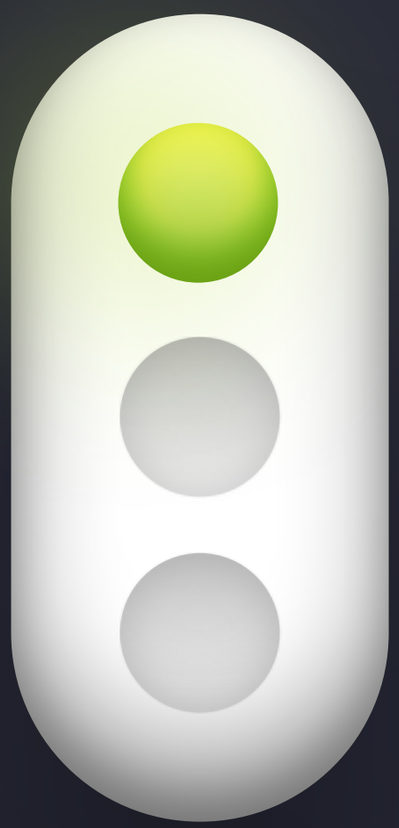
Eye tracking was one of those features companies that could afford a set of expensive cameras and glasses always showed off to the clients at meetings.

In short: the cameras and glasses allow us to create a heatmap showing what the user is looking at exactly at any given time. This can let you know whether they noticed an important button at all, or have they read the entire paragraph of text or just skimmed through it quickly.

While it definitely looks impressive and high-tech, we only used this three times in our career. The main issue I see with this method is that after a while it brings very similar results based on the good practices of how our brain processes what we see.

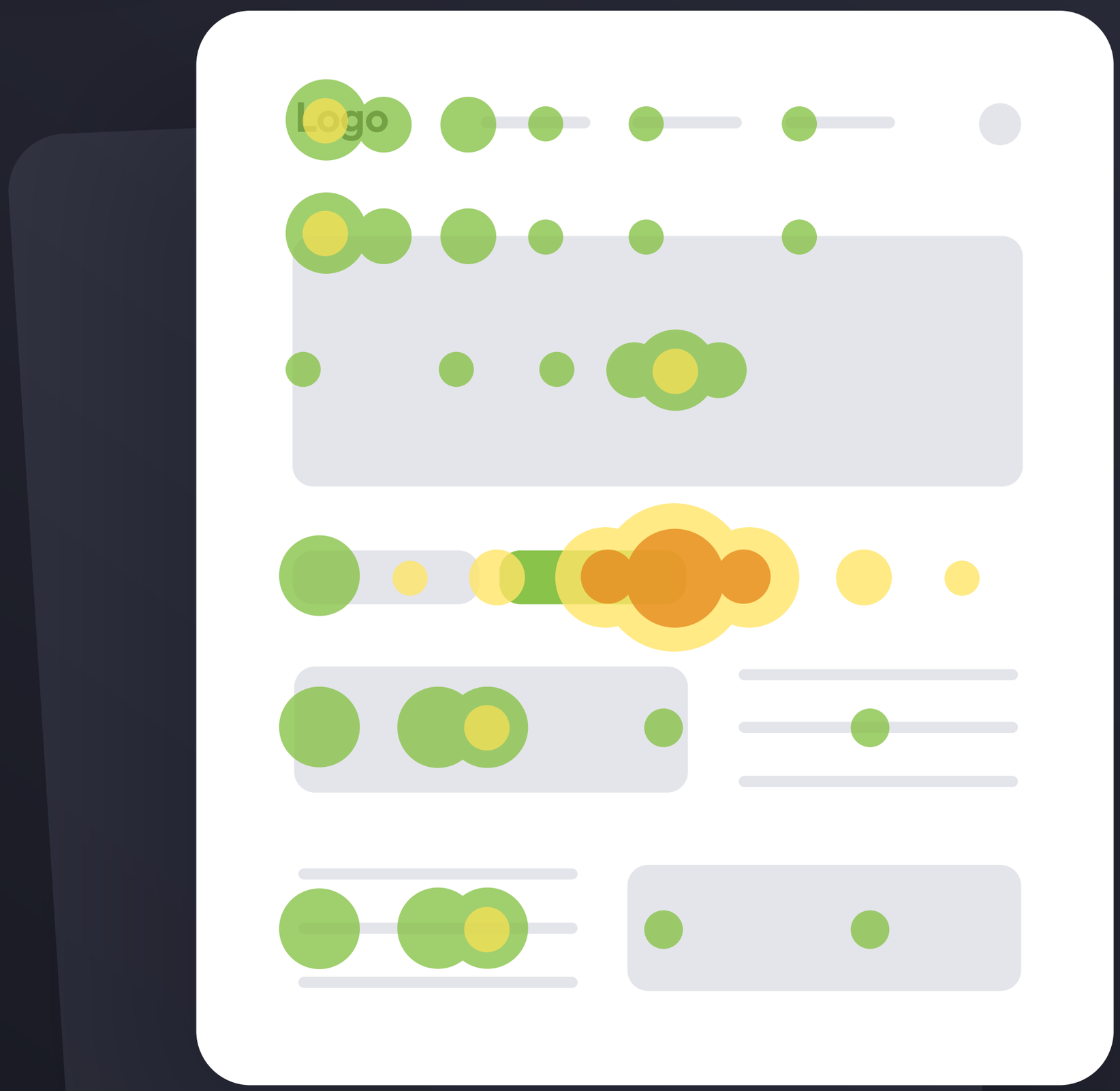
Once you know how to design a good UI hierarchy, that's clear and readable, the results will be very similar. For now I don't believe this method should be considered by almost anyone. However, with the upcoming Apple AR Glasses, and likely a bunch of other companies making their own - eye tracking can have a comeback, but it will likely work in a completely different way that we have yet to define. Don't worry about it today.

On a sidenote there was a free tool called EyeTato that used AI to predict where the users would most likely look.



Movement Tracking

Often used



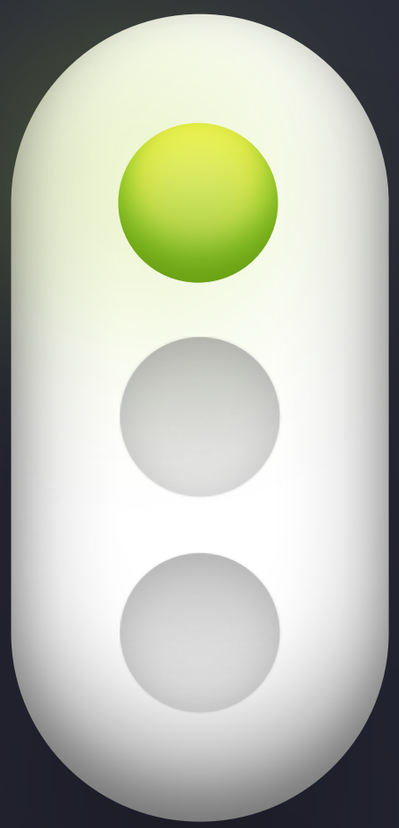
Movement tracking

While we don’t track the eyes of our users as often anymore, there’s another type of tracking that is actually done very often. Mouse movement and scroll tracking can be done mostly on the web, but it allows you to record the actions of your users and create heatmaps of what they click the most and where they hover their curor most often. That is super useful!

We try to do it for every single web project, because these recordings show us how users interact with the current website
- what are their problems, at what stage do they quit the process (like abandoning the cart mid-way) and what’s often neglected
- what are they clicking on exactly.

This is helpful because very often you’re able to notice people clicking on things that are not links or buttons while completely missing the actual clickable objects. Recordings like these are super helpful to better understand the problems with the website and how to optimize it to perform better.

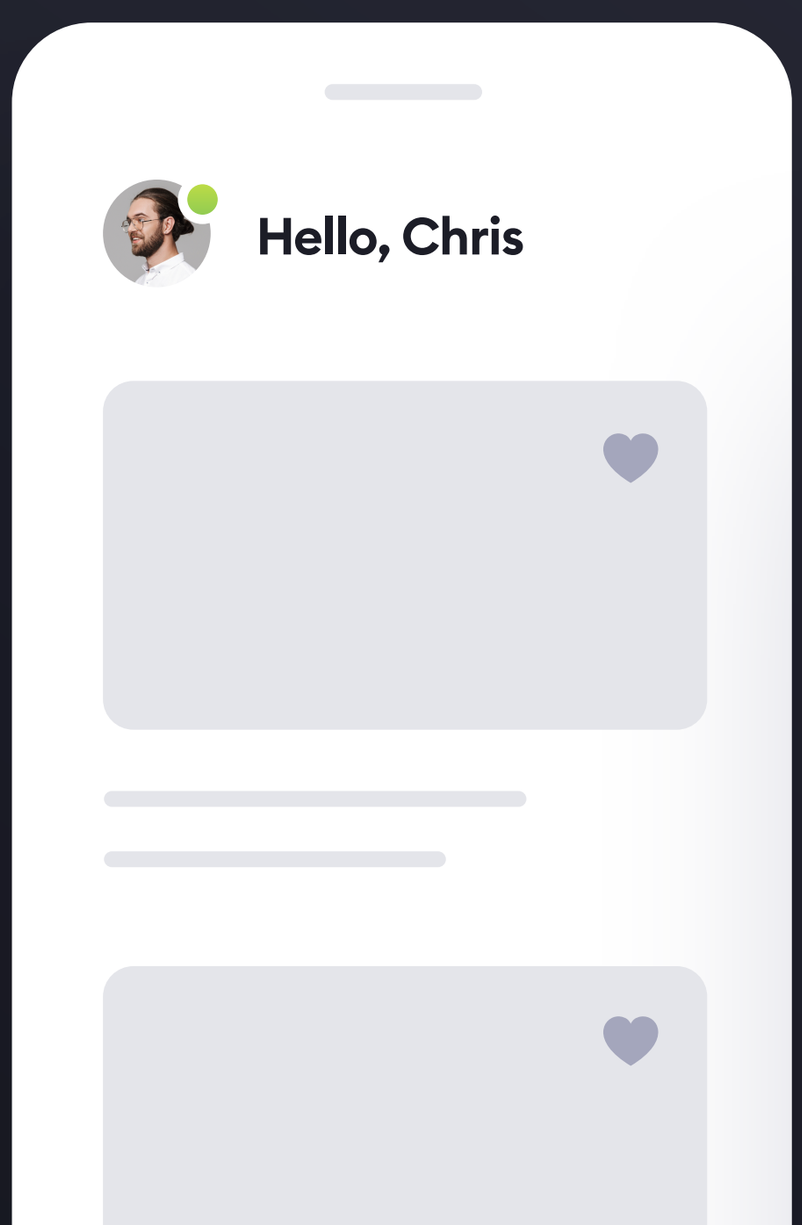
There are many tools you can use to do it, including ones that have free plans. We mostly use HotJar, Mouseflow, SessionCam and more.



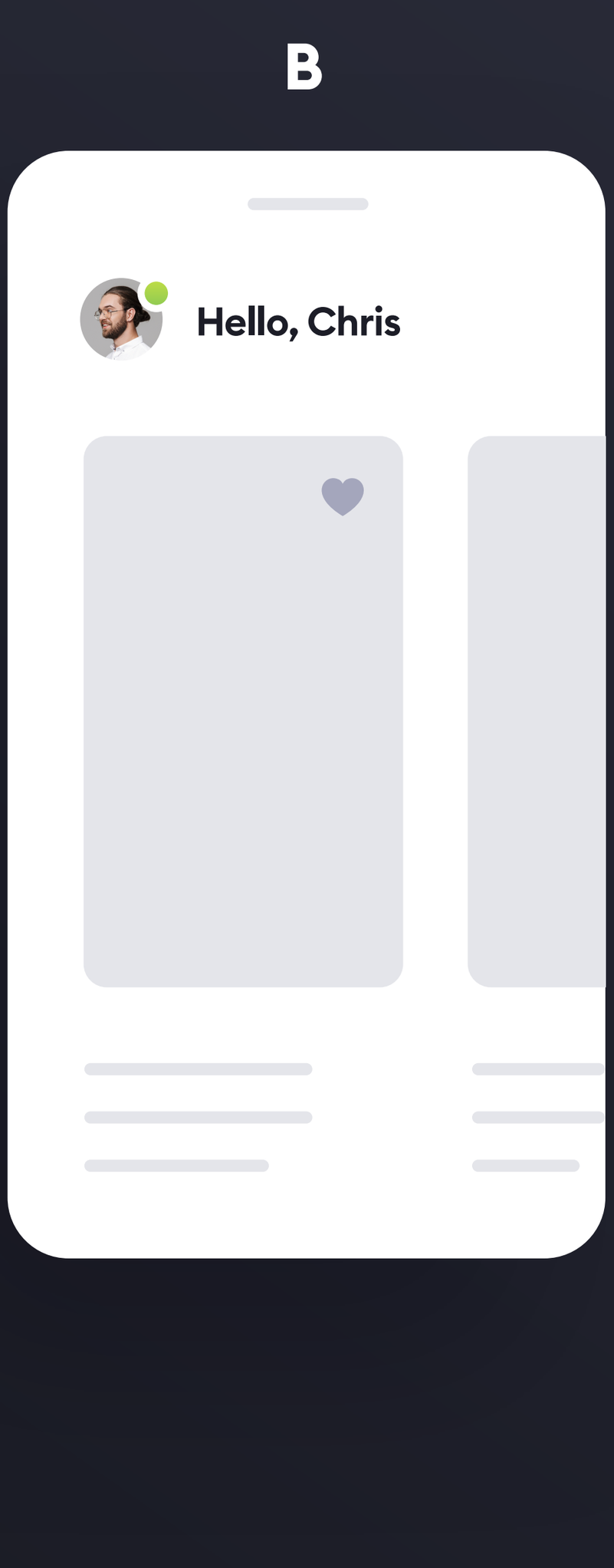
A/B Testing

Often used

A



B



A/B Testing

A/B Testing usually means a test on a live product that's available to all the users. We split them randomly into two groups, and each gets a slightly different interface. It is used to see which option converts better.

Big companies like Netflix or Google are using it all the time to improve their product.

We are using it about half the time. The main problem with those tests is that they're very often not necessary, or you can get better results from a set of usability studies.

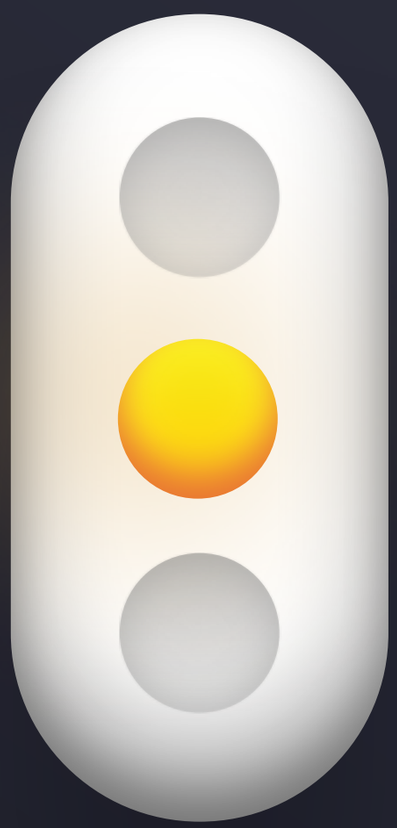
Another common problem is, that when the change is too revolutionary, the conversion can actually drop for the first month or two, before users get used to the new way. Even if the new version is better, it will initially show as worse, and many product teams don't have the patience to run these tests long enough.

If you're working at a small startup, chances are you'll do a couple of these in the very initial stages of growth to figure out what works. The next stage in which A/B testing happens a lot is when you've reached a point of having so many users the growth seems to slow down - there simply aren't enough humans on the planet. In that case it can be a good idea to resume A/B testing because even small improvements can lead to big results.

I personally really like doing A/B tests, but it’s important to understand that they can also lead to revenue over user - often pushing for shady, dark-pattern practices just because they’re sure to increase revenue at the expense of the users.

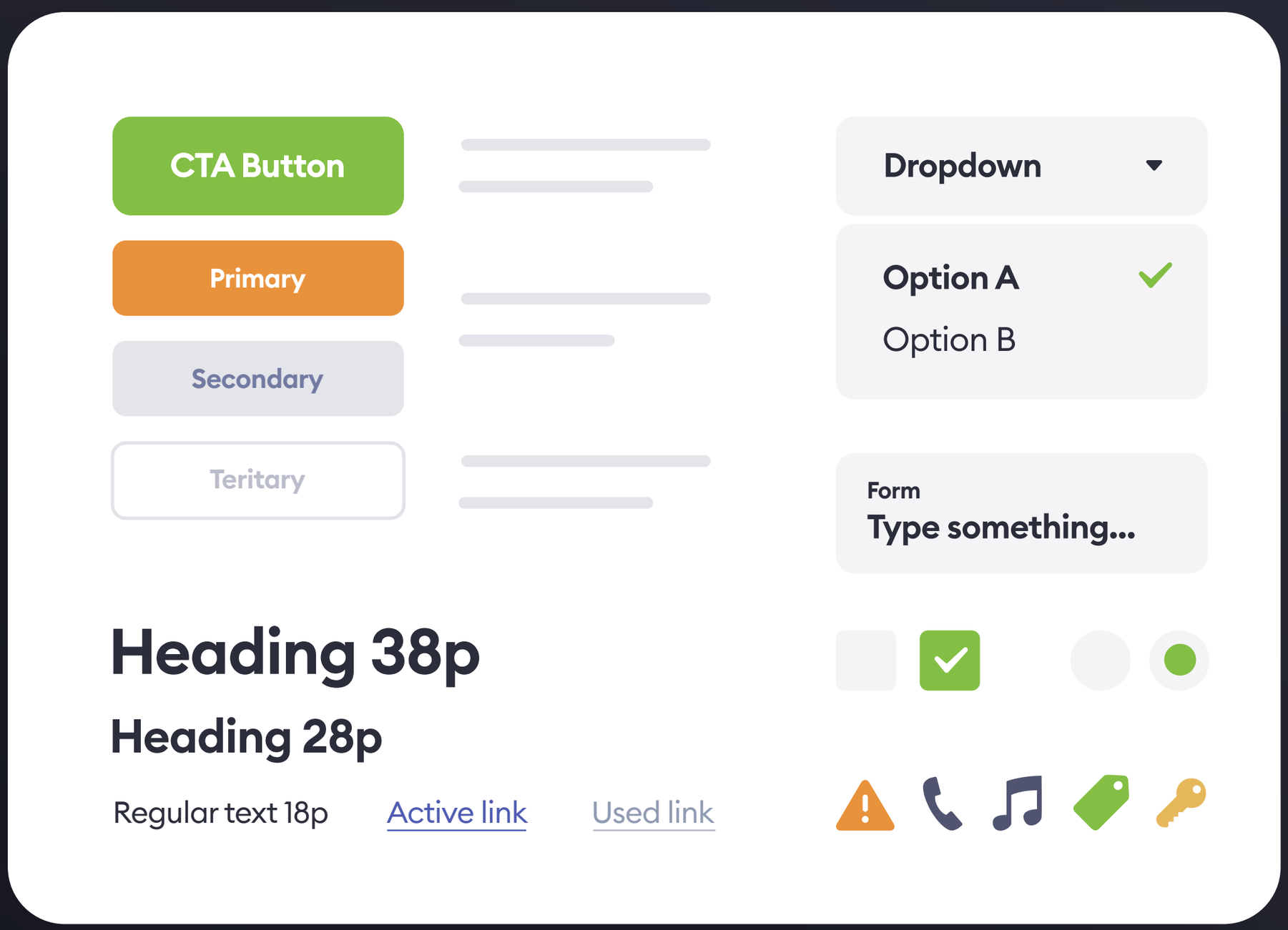
I’ve been to a presentation, where a big company showed how they used A/B testing to upsell to 37% of their existing users a low-quality product that nobody really needed and the company wanted to get rid of it.

When you’re planning an A/B test about an important feature of the product (for example two different ways to sign up for it) do a quick usability study first to iron-out as many small issues as you can before placing the tests live.



Design Systems

Sometimes Used



Design Systems

Remember when design systems were all the rage? Before we dive in to when you should use them, I want to share an important fact about them. A design system is good when it benefits the company that uses it. It sounds obvious, but it's not.

What it really means is that even when there are general good practices and patterns, if you do everything completely "wrong" but your company benefits from that design system, it becomes "right" automatically. Which leads to it being a bit of a pseudo-science within UX.

There are many experts on Design Systems, often talking about contradictory stuff. I actually decided to live test it, and with limited knowledge (we had only done three DS at that point) I pretended to be an expert and got invited to a couple of big conferences. There, on stage, I spoke to thousands of people about Design Systems and nobody questioned me. Anybody can become a design system expert with the magic words "it works good in my company" as their shield from critique.

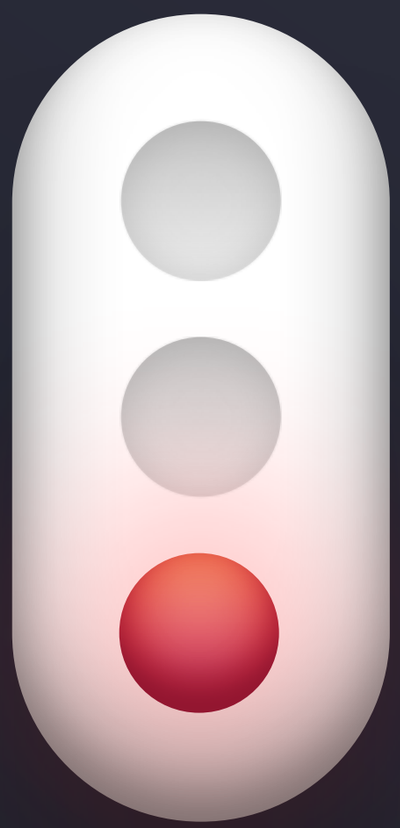
Now, after we got this out of the way, let's explore when design systems are actually beneficial and when you should definitely use them.

In short, a design system is a way to organize your designer and developer collaboration. It often consists of a design library made in your design tool (Sketch or Figma), a wiki-style documentation showing all elements (components) and how to use them, and code snippets (HTML, CSS, JavaScript and others depending on the project). In some cases it has more, but this is a good starting point.

While organising your design is a great idea, not all products need a full design system. They mostly work well for very big products that already have a lot of users and a lot of people working on the product. They also require to have a dedicated design system team - usually 2-3 people - that update it, make sure everyone is using it correctly and generally oversee it as if it was an internal company product (because in a way it is).

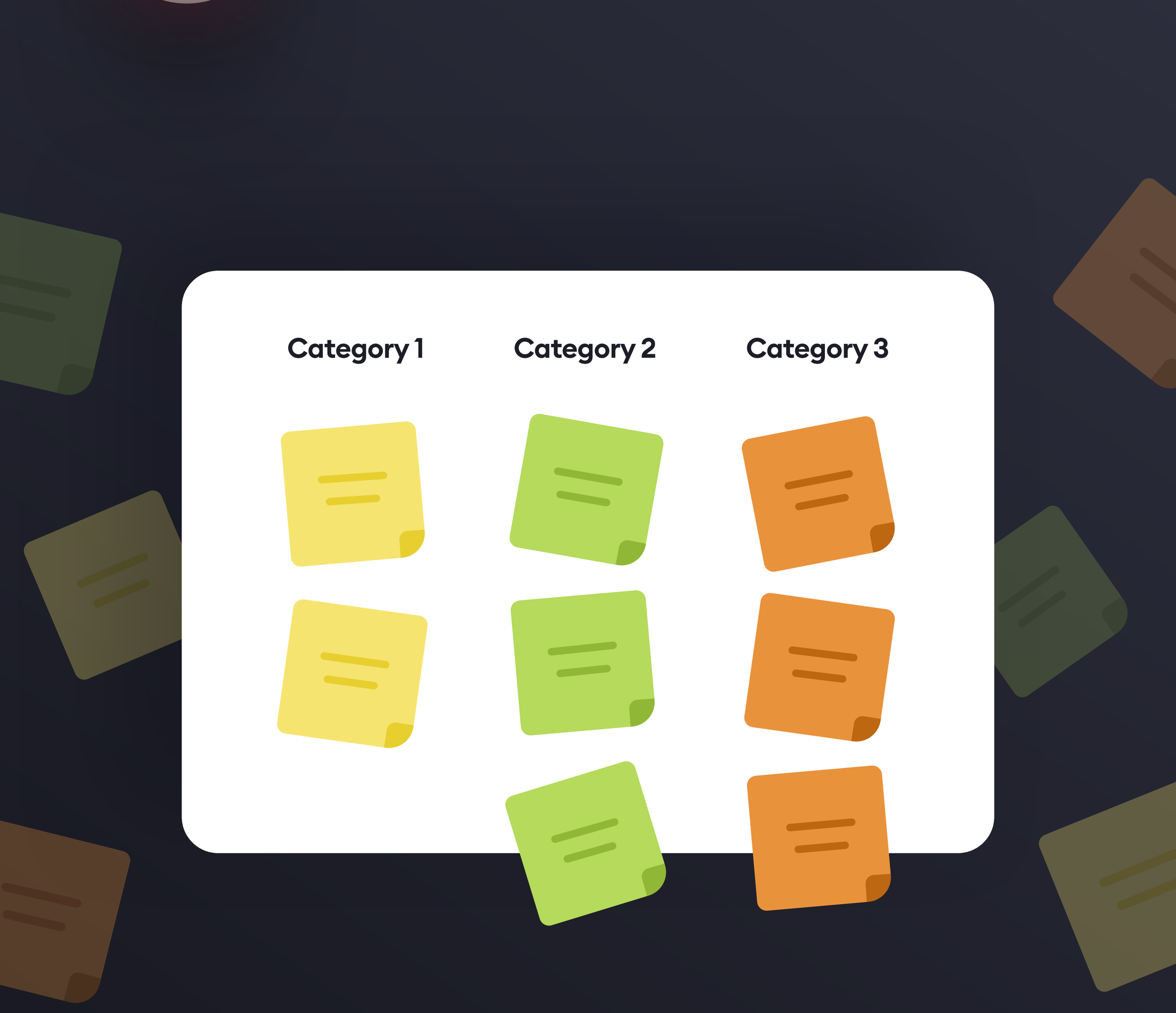
That means maintaining a design system can be a lot of work, and in some organizations you simply need to ask yourself whether adding a design system will benefit you at all - it may turn out to be more work than without having one.

For smaller product, a well annotated design library is often enough, especially when your developers know enough about design to not make an ugly mess when out of it when coding ;-)



Card Sorting

Rarely Used



Card Sorting

The colorful post-it cards are one of the first thing people think about when they hear UX. They are used in a couple of different workshop exercises, but the most popular one is card sorting.

The goal is to define the sections (or a sitemap / information architecture if you prefer this term) by arranging those sections into logical groups. For example you may put cards that say “notifications”, “friend requests” and “edit profile” in the Profile category.

As more than one person does the sorting, the result allows you to see whether most people understand what belongs where in a similar fashion.

There are two types of card sorting, Open and Closed. In open the participants get the cards but they must write the categories for them themselves. In a closed version the categories are already pre-defined and the participants simply place the cards next to the category they believe it fits.

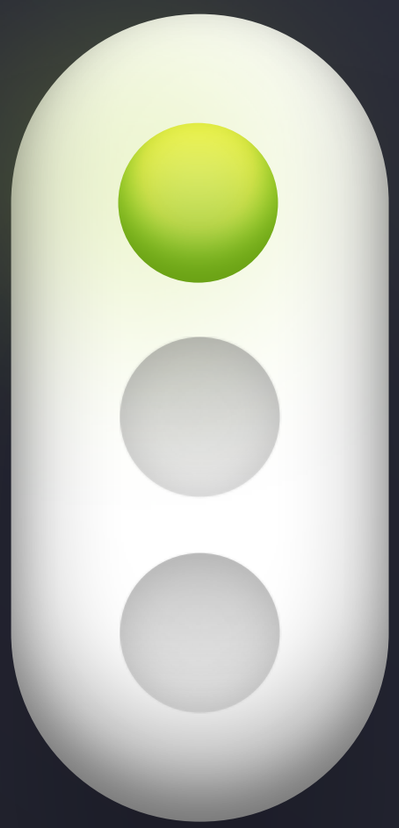
It is one of those “engaging” activities many UX agencies used to do with clients at meetings and workshops, especially back in the early days of UX. Right now it’s barely ever used anymore.

There are a couple of problems with card sorting. First of all, we already have enough good practices and safe patterns established for most kinds of products, so a lot of the information architecture is pretty obvious (and can later be user tested).

Next to that, the fact that each card only has a line or two of text means the users can understand them differently than you think, making the whole exercise completely pointless.

If you listen to your users closely, you may hear them mention where they expect a feature or page to be during an usability study. In many cases that combined with the collective knowledge of how to plan sections is more than enough.

No need to reinvent the wheel. Card sorting is almost always a waste of time and money, unless you're aiming for a photoshoot after a client workshop that shows that you "did something together".



Heuristic Evaluation

Almost always used

Guideline 1

Guideline 2

Guideline 3

Guideline 4

Guideline 5

Guideline 6

Guideline 7

Guideline 8

Guideline 9

Guideline 10

Heuristic Evaluation

Heuristic is a popular word in UX. It sounds smart when you're telling your clients that you're going to do a heuristic evaluation of their product. But what it actually is, is taking a couple of general good practices and rules of thumb and seeing if the product meets all the requirements.

There are ten general guidelines outlined by Jakob Nielsen, and we'll go through all of them here, but keep in mind that they're not set in stone. After a while your list of what and how to evaluate will grow and it will be a longer list than just the ten.

They are however a good place to start. This is something we ALWAYS do when auditing products.

Visibility of system status

That means the product needs to tell the user what's going on with minimum wait time. The user needs to know if it's loading, or it crashed, froze, where they are on a map (with a pin) - generally informing the user.

It's basically communication so the user doesn't feel lost or confused. If they are taking an important action in the product, it's good to guide them through it all the way, so they know what they're doing.

Match between system and the real world

I sometimes wish UX as an industry adopted this one too.
It means that the design should be easy to understand by the user. Follow what people do (and how they talk) in the real world and avoid both language that's difficult, but also confusing visuals.

Make sure - for example - when making an icon that your target audience understands what it does. We were once making a grocery store loyalty app for rural areas in Ireland and when asked what the heart icon next to the product means, a whopping 67% of users said that it means the product is good for the heart. We changed the icon to a star AND added a label that said add to your favorites. In short: avoid complex messages - both in text and visual so people have it easier to understand and use.

User control and freedom

While cutscenes are nice and good in videogames, in user interfaces it's good when the user has as much control as possible. If they click or tap on something by mistake, they need a quick way to go back or cancel their choice right away. They need to feel in control at all times.

Consistency and standards

This connects strongly to Jakob's Law - which says that people spend most of their time on other apps or websites than yours (Probably excluding TikTok ;-). That means they expect your product to do most of the things in a similar way - this is why most registration forms have the same fields in the same order. Maintain that consistency and the users won't have to "learn" your product as much - it will feel familiar to them. No need to reinvent the wheel.

Error prevention

Nobody likes to make mistakes. But they do occur, whether they're just accidents (called slips) or mistakes (because the user didn't understand how to do something) it's up to us to prevent as many of them as possible.

You can prevent errors by having clear messages, warnings (where necessary) and not allowing the user to do things that don't make sense - like changing their email to a random character set and locking them out of their account. Problems like that are surprisingly common in digital products so try to spot them early on.

Recognition instead of recall

Our short-term memory is pretty limited so try not to hide important information from the user. A good example of this is a form field with the label inside that disappears when you start typing.

If the user is forgetful, or switches to another task, they can completely forget what the field they're filling in was for. That's why labels above fields should always be there - never disappear.

Flexibility and efficiency of use

Allow the users to use the product their own way when possible. A good example of this is adding keyboard shortcuts to apps - beginners will just use the cursor to do stuff, but more experienced users will be able to use the keyboard to accomplish their goal much faster.

In some small way you can also allow the users to customize the product, but don't go too far. A method of choosing which 5 tabs you want in your apps bottom menu is generally a bad idea (and some larger apps still do it) because it creates chaos in navigation. Personalizing content to the user is another good practice - by showing them stuff they're likely going to need you decrease the time it will take them to accomplish their goal.

Aesthetic and minimalist design

UI design should generally be pretty minimalist. It means we shouldn’t overload it with elements that don’t contribute to the goal. While I believe decoration can work well in some cases, it definitely isn’t something you should use everywhere.

Extra visual objects can be good to have in the onboarding process, or on the login page, but with a longer form (like registration) it’s best to keep it clean, simple and readable - so the user can clearly focus on the task at hand. The design should also follow the general layout, hierarchy and consistency rules - so it simply has to look good and readable.

Help users recognize, diagnose and recover from errors

Even with the best efforts, sometimes we cannot prevent the user from making a mistake, or the system from crashing. The solution is in how we communicate those errors. Avoid things like “Error 126216C” and instead always use simple language to explain what exactly happened.

There should also be a suggested solution right within the error message. To make it clear that it’s an error, you can use the color red (most often associated with errors) and bolder font so it’s visible.

Help and documentation

A help section in your product is good to have, but with an approach that it's best if the product is so easy to use almost nobody has to go seek help. So always try to make it easy to use first, but provide documentation and how-to's when necessary. Make it searchable if it's a lot of documentation and divide it into clear, easy categories.

In the actual solution to a problem, within the help section, try to use step-by-step walkthroughs whenever possible - with visuals showing individual screens and examples on how to work with them (like a form pre-filled with an example of what kind of information you should add).

"Stuff we know"

When making a report, you don't need to precisely follow these specific rules and the structure of the report can be anything that you're comfortable with, as long as the recipient can understand it too. In the report we always use plain, easy to follow language and we never use the word heuristics because there's no point in making the client Google it.

We usually call it "stuff we know" - suggesting that it's the best practices and good patterns that are already established in the industry. It's a useful technique, I just wish it had a more human name.

Discovery Workshop

Sometimes Used

Discovery Workshop

This used to be a lot more popular, but with customers often doing a lot of the pre-work themselves is now roughly happening around half the time - mostly with bigger clients. The purpose of the discovery workshop is to understand what the client's product is struggling with. For that we analyse the user feedback (if there's any), the pain points (what doesn't work for the users) and confront it with what the business needs (business goals).

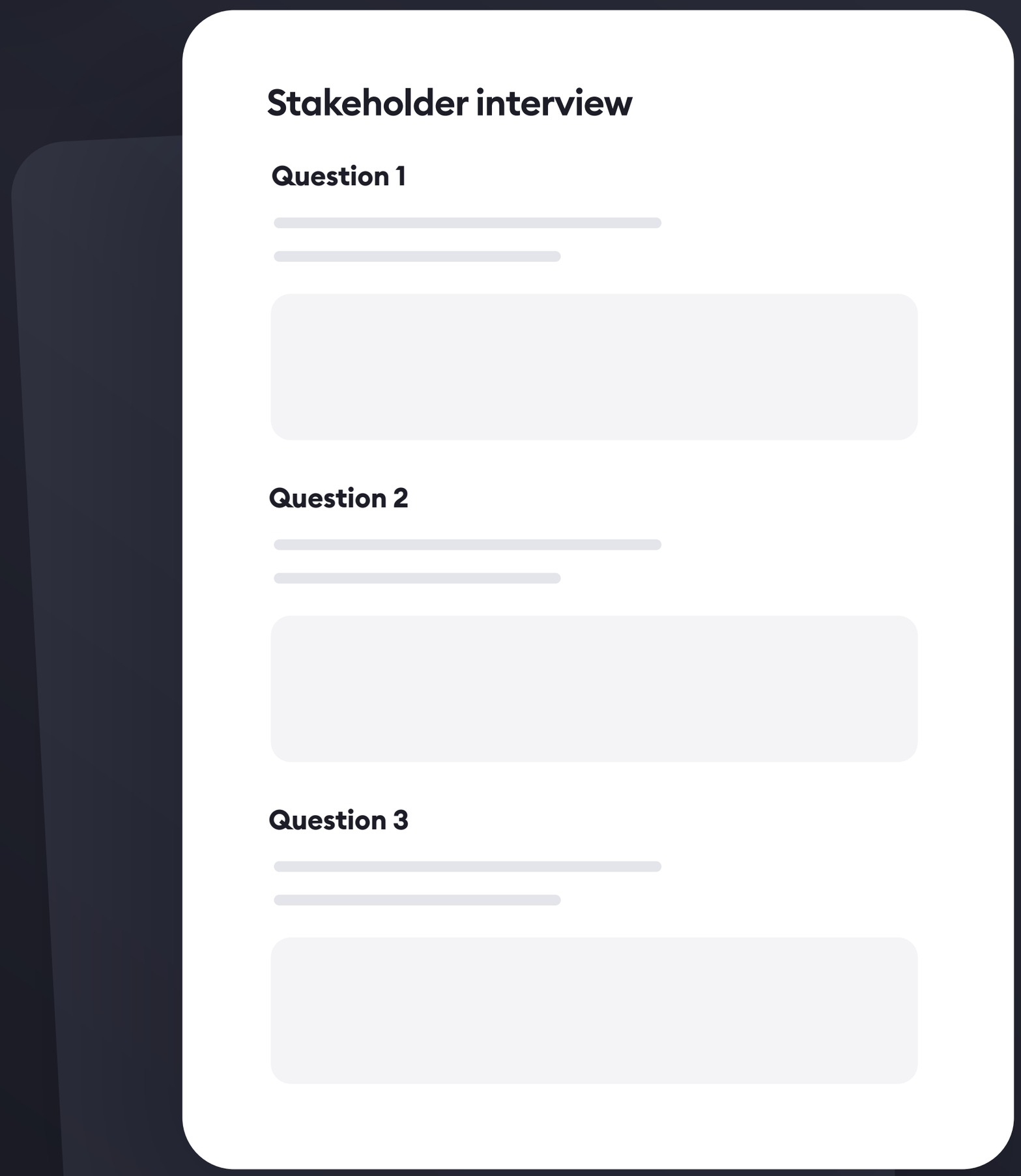
This is also the time when you meet the **stakeholders** - a popular term for people from the client side that make decisions about the product. It is those people that you need to convince that you know what you're doing.

Right now, with many clients having internal research teams (sometimes of even just one person) they can provide enough context and data upfront, that there's often no need to run a dedicated workshop. What we usually do in that case, is to evaluate everything the client brought in (on our own time) and send a set of questions and things we need them to clarify further.

Sometimes it's good to make some basic user journeys on a kickoff meeting, but most of that process is left to large agencies billing large clients for a lot of hours - not much real value.

Stakeholder interviews

Often used



Stakeholder Interviews

Stakeholders are basically members of the client team that are part of the product team. These people worked on the product before and have the best knowledge of it. It is important to get to know them, and try to get as much information about the product as possible. Based on what you get from stakeholders you create a plan on how to proceed with either future testing or a redesign.

The main goal from talking to different stakeholders is to find the sweet spot between what the users need and what the business wants in order to make more money.

Stakeholders can range from internal testers (low influence) all the way to the CEO (high influence) but it's good to be on the best of terms with all of them. Listen to what they have to say until the end, don't interrupt with suggestions because they may actually confirm what you wanted to say and that is really annoying.
Be respectful and come prepared with a list of questions that help you make a proposal that will truly benefit the company and the product.

Stakeholders are human beings so remember to treat them as such. They should be your friends, as having stakeholders who hate you will likely lead to a horrible work environment. If they're hostile from the start, you may want to reconsider doing the work.

We usually do them as simple conversations - mostly online calls, where we ask them individually to tell us about the product and what they think are the biggest problems they're facing.

Based on what they say we then ask follow-up questions and ask to clarify anything that appears interesting or can help us prepare a plan of action.

7

Laws of UX

Laws of UX

“If UX has laws, does it mean we’re all lawyers now?”

This question has been on my mind for years, because while very useful in the real work, most of those UX laws are stated like real law - with complex, industry language that makes it really hard to grasp even simple concepts.

I decided to slightly rewrite some of them, with simpler language and whenever possible - examples from our approach as design practitioners.

After all, for laws to be followed and respected, people need to understand them first, right?

I’m going to start with a couple basic ones and grow this chapter with future updates.

Minimize choices

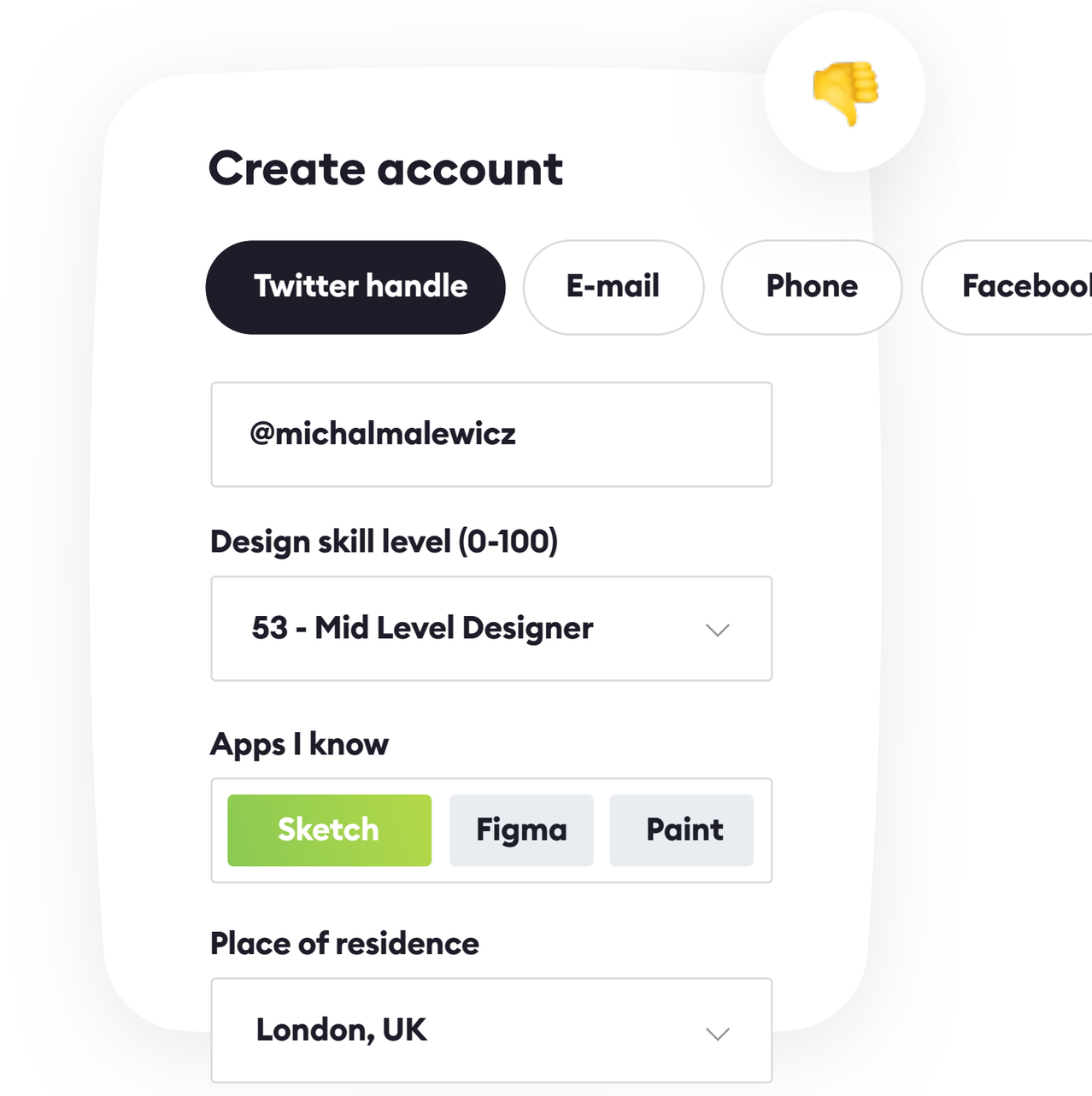
Also known as Hick's Law

The more choices, or option a person has, the longer they need to think about what to do next. There's also a bigger chance they'll be confused.

Example

The form on the right side is the first time a person decides to sign up for a service.

Right from the start it offers **WAY too many options** which will discourage most people from ever completing it. Registration should always be about the minimum number of fields possible.



Always try to only show the necessary options and hide everything that is not needed - because all those extra things add to the cognitive load. A high cognitive load means our brains need to work really hard to understand all the stuff we are looking at. Too much information to process = high cognitive load.

Don’t be too original

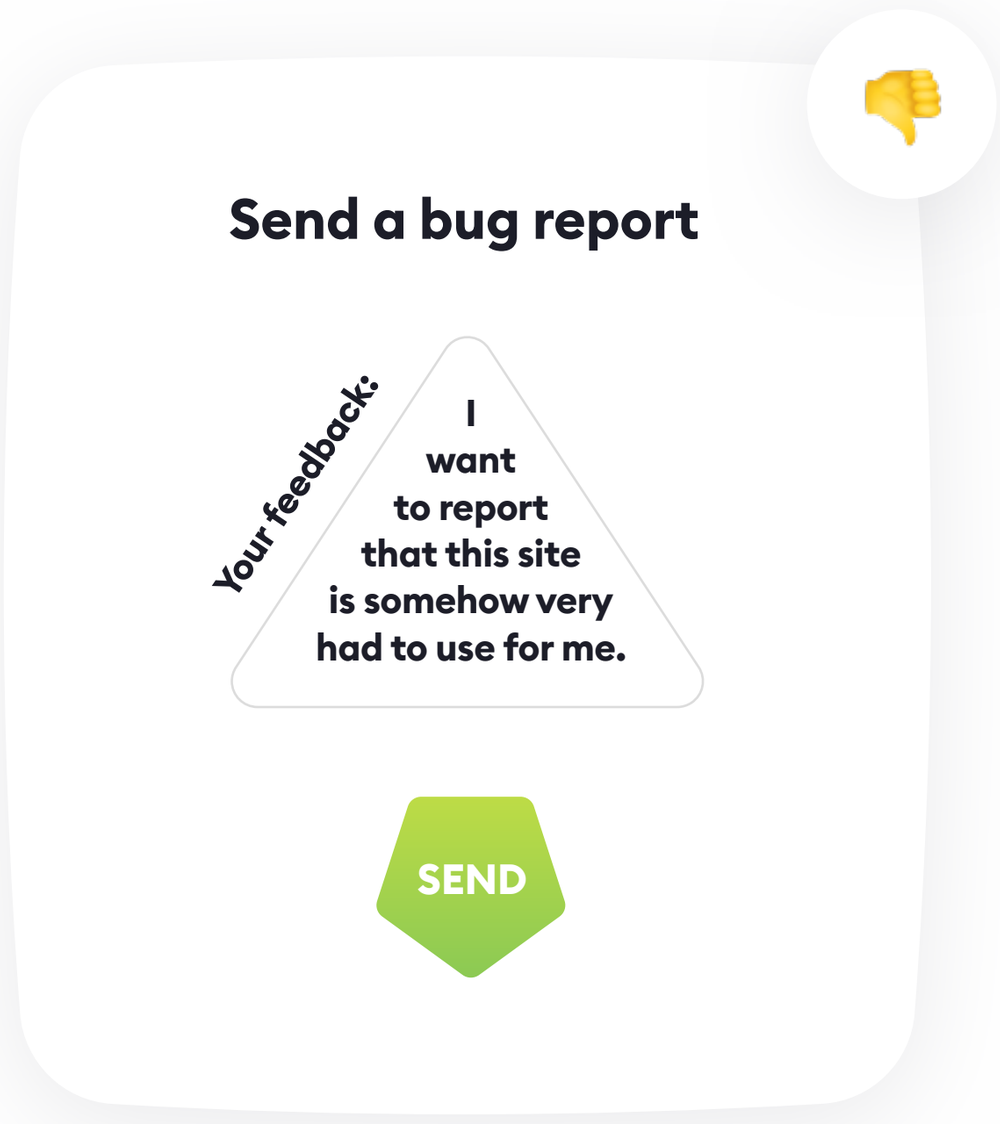
Also known as Jakob’s Law

People mostly use other apps and sites, so they expect familiar patterns in your product.

Example

While a triangular form field can seem like a cool new idea, most people will be confused by it because they expect a form field to look in a specific way - usually a rectangle. Same with a polygon-shaped button.

Avoid originality for the sake of being different.



That right now can be half-true, as there are some people that use TikTok 99% of the time, not other sites or apps, but still - it’s valid. You can still play around with how you present the interface - using different colors, shadows, fonts or decorating it with a cool, animated illustration that follows your typing progress. Breaking established patterns, however, always ends with the user not knowing what they’re looking at.

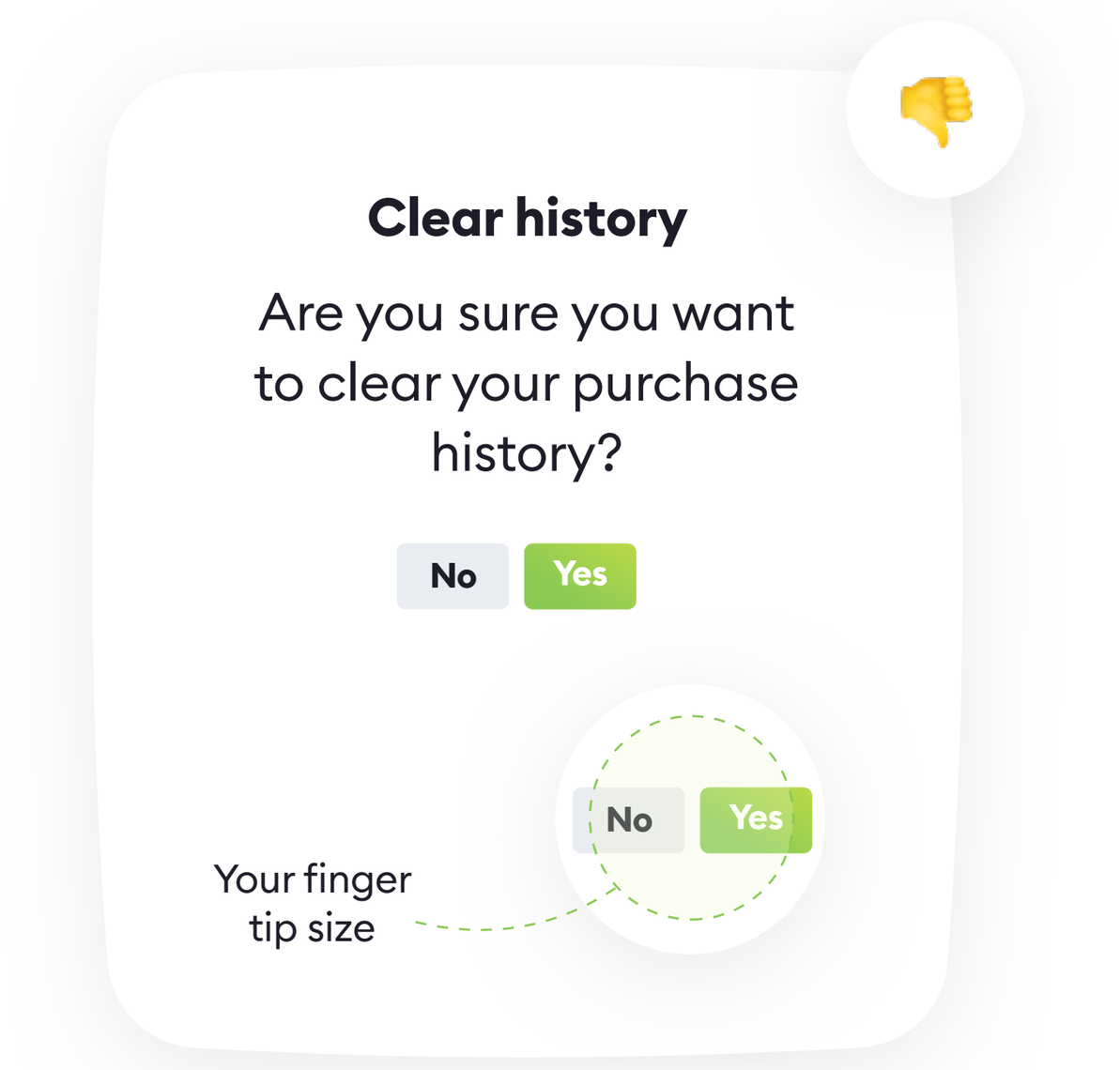
Big (Touch) Targets!

Also known as Fitt's Law

Bigger buttons with enough space between them are easier to access (aim). We click and tap things all the time now - but both with our fingers and with our mouse cursor, our target needs to be big enough to comfortably aim at.

Example

When your buttons are too small and very close together, they can both fit under your fingertip at once. If you're trying to make a choice, there's no way you'll know what you actually pressed in this case.



Make sure all UI elements can be accessed individually.

Follow the interface guidelines from the platform you're designing for, or to be safe always use at least 44x44 mobile touch targets and 24x24 (and preferably 32x32) on desktop. The cursor is more precise but it also moves faster - so even on desktop try to make all clickable elements as easy to aim at as possible.

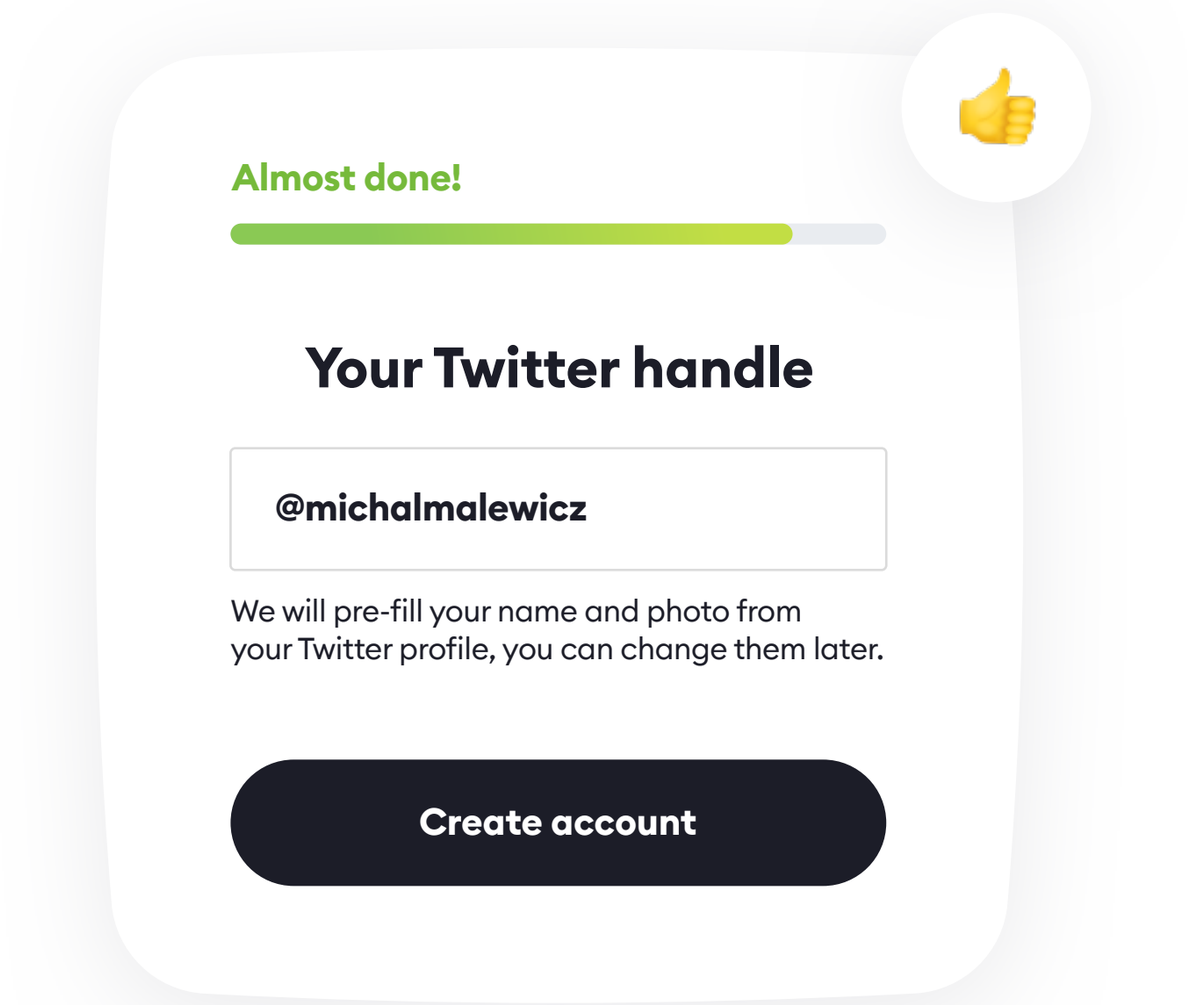
Almost done!

Also known as Goal Gradient Effect

The more you feel like the task is close to done, the faster you work to complete it. This means it's important to show progress, as the closer that progress bar is to 100%, the more inclined people are to fill it.

Example

This is a little lie many products do, showing you the first step as 0% and then the second step (out of two) as 95%. This approach motivates you to finish as you know that you're practically done already so it's not worth it to abandon the process.



In all longer processes show progress, avoid unnecessary elements or fields and whenever you can, pre-fill the forms for the user. If you can get their name from their Twitter handle, pre-fill it and let them change it if they need to. By showing the progress go fast towards completion, you'll influence your users to stick around and finish their task.

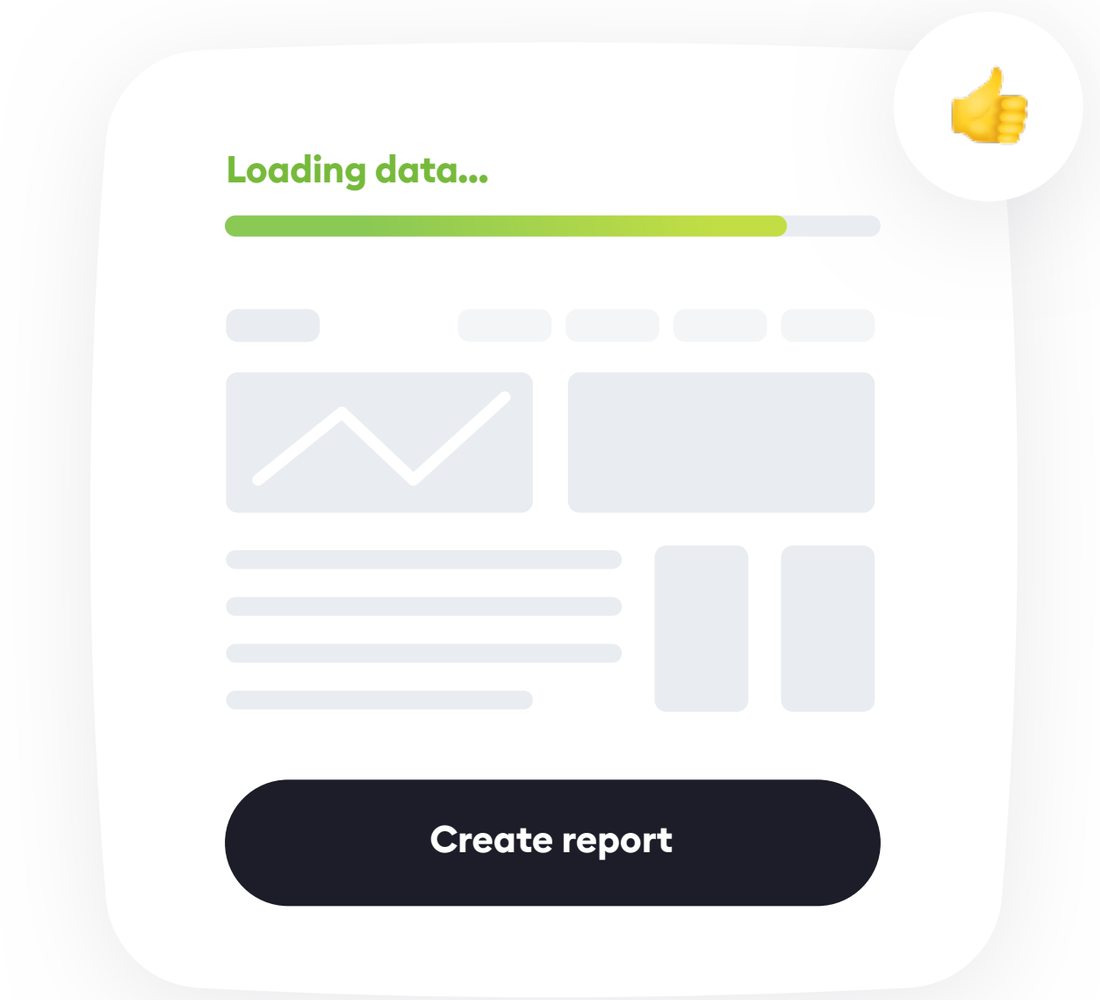
Instant access

Also known as Doherty Threshold

The user should not wait for the system to respond. When loading things keep them engaged! Our attention spans are low and our devices and internet speeds fast. We don't have the patience to wait longer than a few seconds for anything.

Example

When loading a lot of data you can use both a progress-bar (so something is "moving" while the user waits) and skeleton screens that roughly show the structure of what will load and where. In this example we know there will be two graphs on top and some text.



Skeleton screens are a great way to trick users into thinking the product is working faster than it is. The worst thing you can do is to have an empty loading screen that doesn't have any movement or indicator that it's actually loading. People can accept a progress bar, or even a spinning wheel because when things move we understand that something is going on.

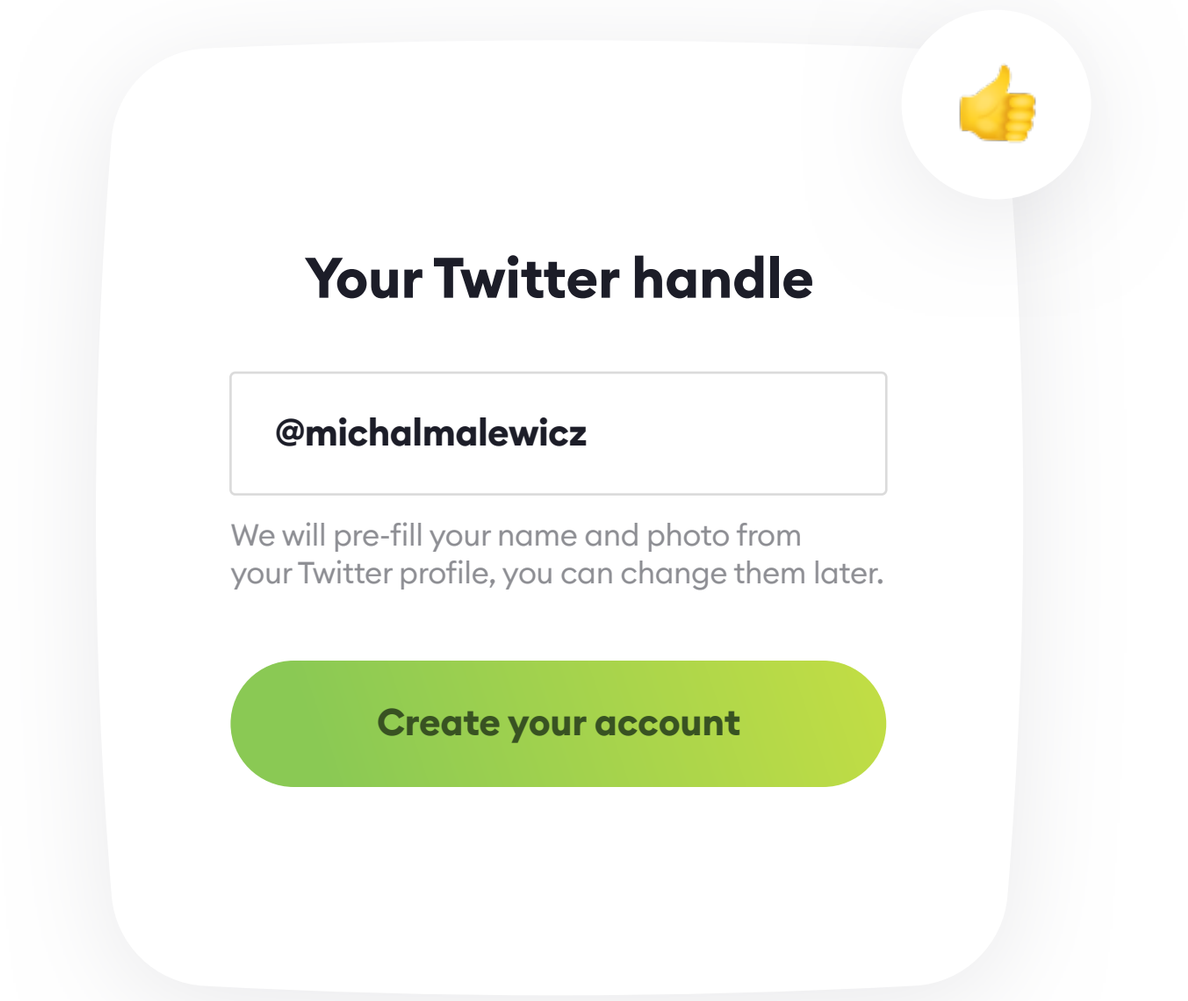
Reduce complexity

Also known as Miller's Law

Our memory and processing is limited, so don't overload people with too much stuff to think about. Our short term memory can only store a limited number of items - that's why it's best to divide all complex tasks into smaller, well categorized steps.

Example

Split tasks into logical, easy to process steps so users are not overwhelmed. For example, don't ask the user for their address right next to their email - they can often fill in the address later - after they create the account.



Don't mix together elements that can't logically be a group. That's often a problem with longer online forms - they really work best when you group information into sections and don't mix your profile picture with your address. Separation like that make it clearer and a lot less stressful for the brain as we simply process things faster as logical groups.

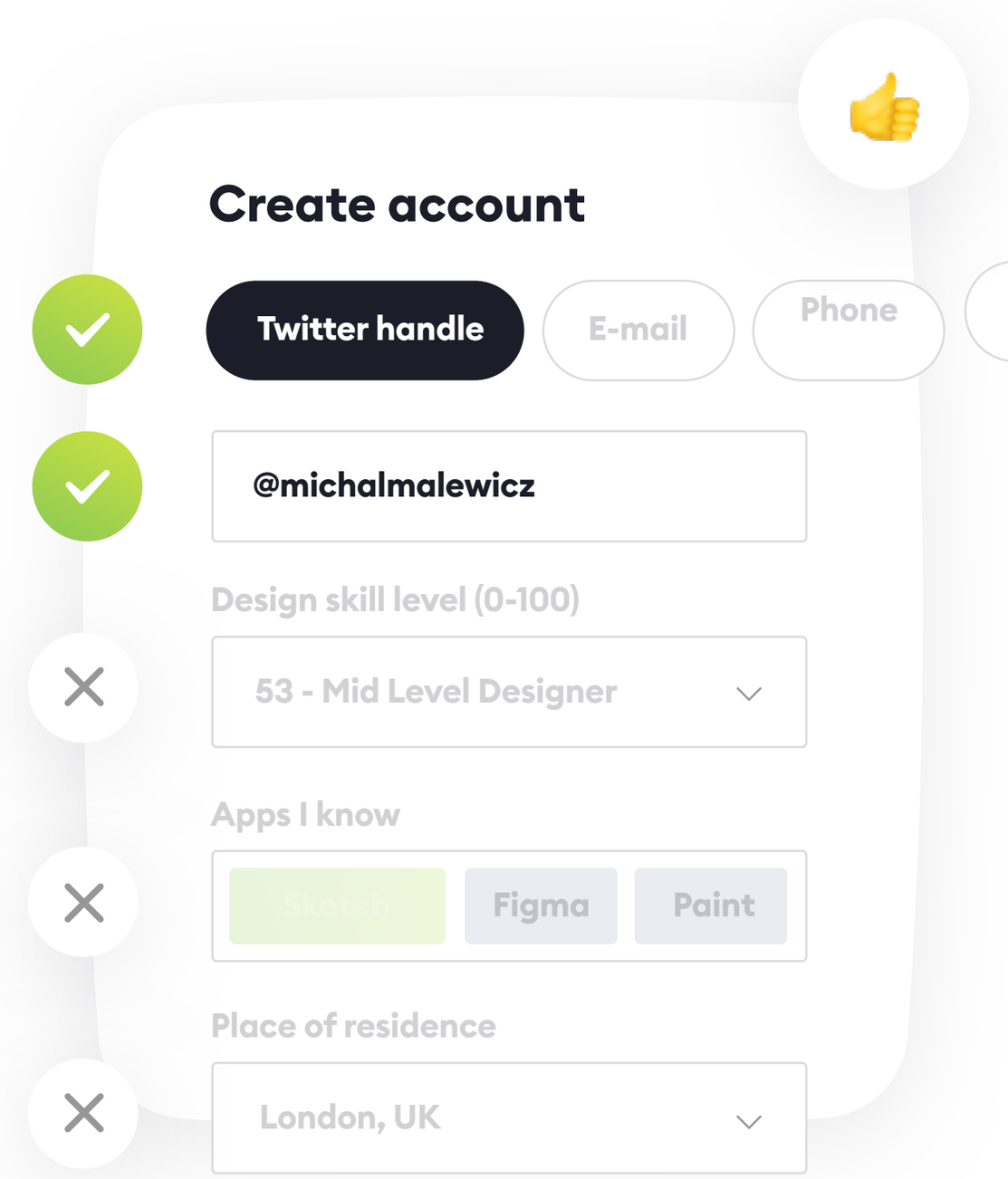
Simple is best

Also known as Occam's Razor

Less is more. Keep only the design elements necessary to complete the task. When you complete your initial design, ask yourself: what can I remove without sacrificing functionality? What is NOT important here?

Example

In the example from the first law we can easily remove a choice of different login methods and just use Twitter handle. We can also remove the apps and design skills because users can add it later, in their account settings if they really need to. Registration should be simple!



The Occam's Razor says, that when you see two solutions to a problem, you should pick the simpler one. You can solve many design problems with very complex, demanding solutions, or you can just make things simpler.

Pretty feels better

Also known as Aesthetics Usability Effect

We perceive beautiful products to also be better, easier and more fun to use. That can be an advantage!

👎

Aesthetics are not important

Who would care about aesthetically pleasing UI's anyway?

Yes i completely agre

What are aesthetics?

👍

Aesthetics are important!

We naturally perceive well structured visuals as more usable.

Good to know this!

I will do that

Minimalism, clarity and readability are all a big part of aesthetics. If your design is all over the place with chaotic layout, out of place typography and colors that hurt the eyes, it will be perceived as "worse" than a similar product that only looks better. Even if the functionality is exactly the same, the better looking product will get better scores for functionality from users. Focus on the visual quality and readability at every stage of the user facing design.

8

What’s next?

What's next?

A good testing scenario for most products is to start with Competitive Analysis, then, if it exists go through user comments and how (statistics) they use the product.

Based on that create a quick high-fidelity prototype for the usability study - preferably in HTML for best results. Once the results are in, either iterate on another version, or if it roughly does the job go into coding. If you had more than one idea, you can also do a live A/B test - just avoid doing it on very important product features - in that case do it in closed, smaller research groups.

Then, as the live product is being used by more people plan to interview them - either through user survey, or through app-store comments to find out if they're having issues with some part of the product. You can use analytics to find which parts of the product are confusing (users are lost) and either just redesign them for an A/B test again, or do another usability study to find out if you can improve it.

This whole thing repeats. Over and over and over and ...

Welcome to the world of UX.

Of course that's just one scenario, but I will try to update this book quite regularly based on your questions, suggestions and requests.

It's a work in progress, because I want it to grow with the needs of the community, not be a pre-made set of ideas to digest.

I hope you found my experience with UX useful and inspiring enough to start experimenting and making great products yourself!

All the best!

How you can contribute?

If you found any issues with the book (typos, mistakes, omissions - let me know via email!) I'll be happy to fix them and add your name to the list of contributors at the last page of the book. I want to grow this project with your feedback, so it actually meets and exceeds the needs of junior designers worldwide! Thank you!

You can reach me at ***designingui@hype4.com***

Contributors

Version 1.2

Special thanks to:

Mohamed Elghamry

Cara Oberfoell

Roger Fung

Natalie Taylor

Brian Paul Smith

Khrystyna Goeijenbier

Dario Cianciarulo

Our other resources

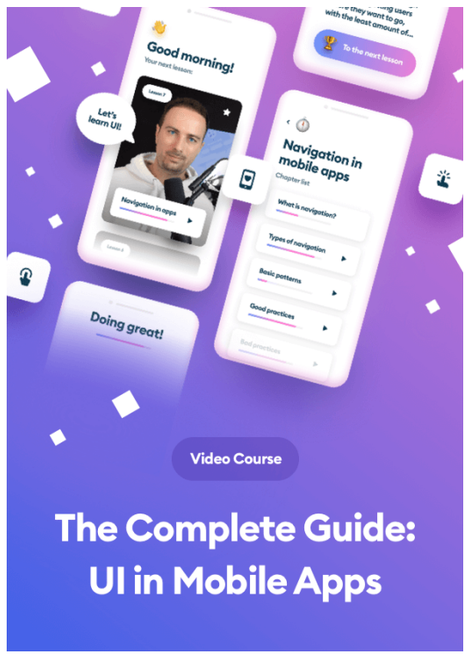
Thanks for supporting me with the purchase of this eBook.

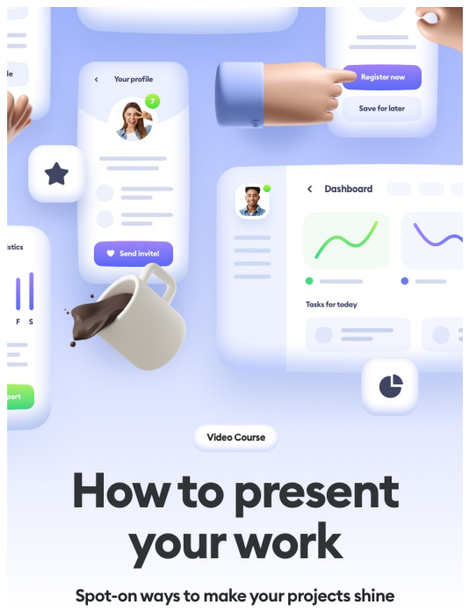
If you want to see more of the things we do, including over 600+ pages of UI knowledge in “**Designing User Interfaces**” or our popular video courses, you can use the code:

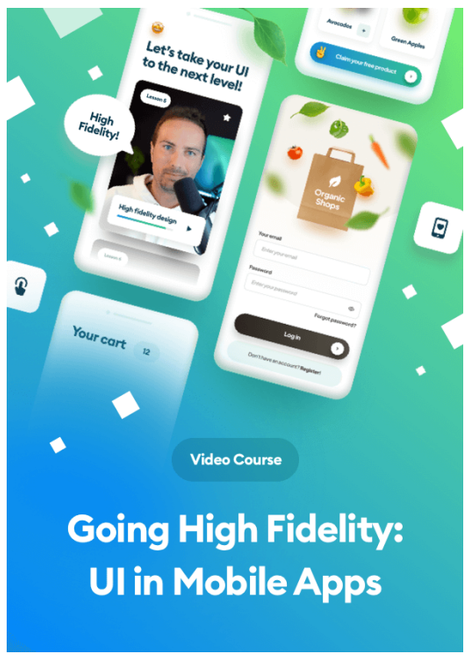
uxiseasy3127

To get any of our products with a **25% discount.**

Thank you!







Version 1.2, April 8th 2022

hype4 academy