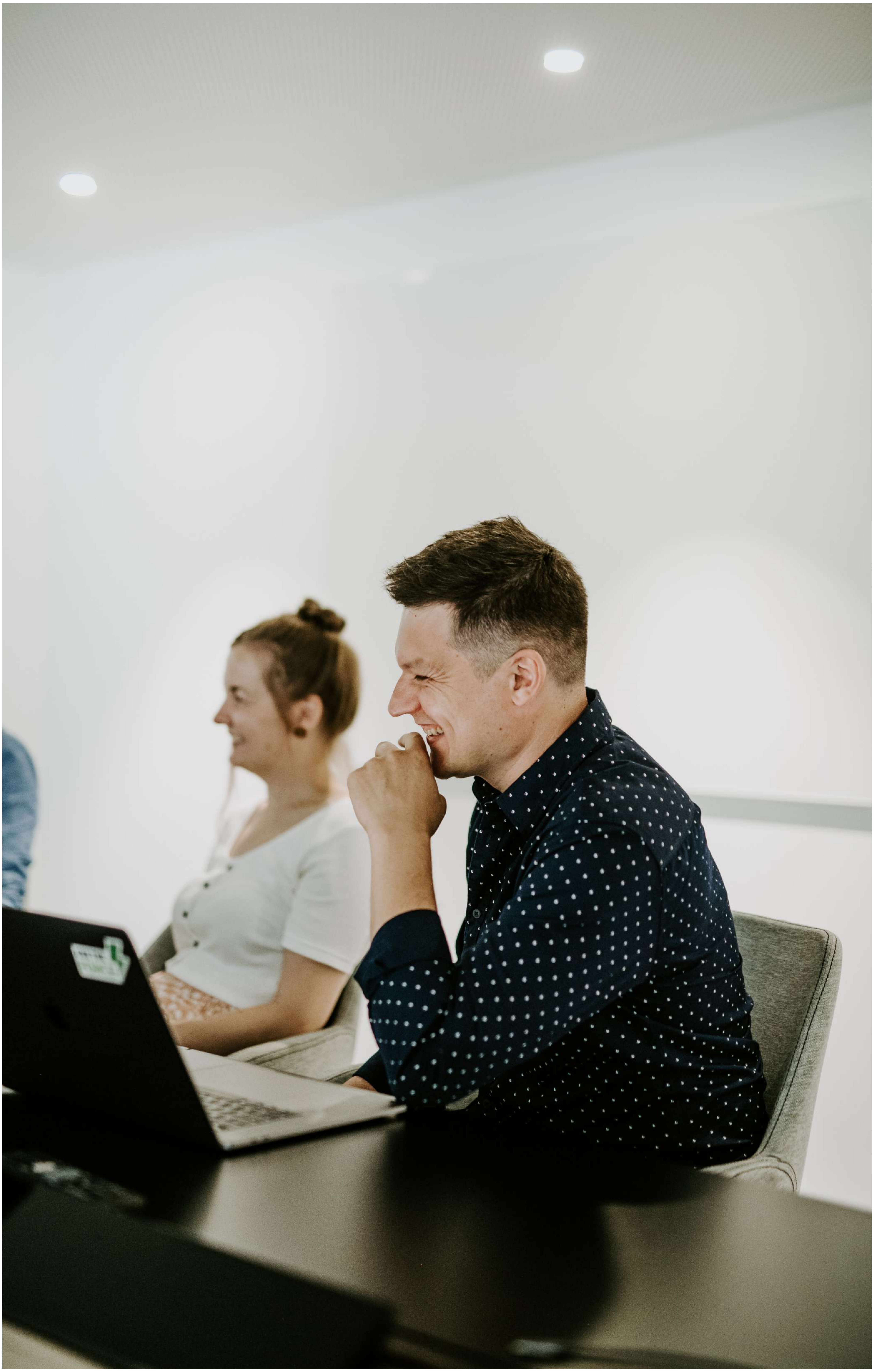
The Complete 2022 User Testing Handbook for Mobile Apps

MOONCASCADE

Content index



What's in this handbook?

In this handbook you will find out...

- Why user testing is key for mobile applications
- How user testing and compliance are connected
- Why user testing is more about quality than quantity
- How we've solved specific user testing issues in the past
- What to keep in mind if you're currently developing a mobile application
- What the numbers tell us about user testing
- ...and more!

This handbook is for you if...

- You're thinking about developing a mobile application
- You're already developing a mobile application and want to avoid pitfalls
- You've launched mobile applications before and are looking for explanations for past issues
- You want to find out more about user testing in general
- You're interested in how we do things here at Mooncascade

This handbook is not for you if...

- You think user testing isn't essential for developing mobile applications
- You aren't interested in mobile applications
- You aren't interested in user testing, UX/UI design or product development

Ready? Brilliant – let's dive in!

Why is user testing important?

The First-to-Market Race

“Apple killed Blackberry, because Blackberry was first.”

Remember Blackberries? The hugely popular smartphones of the early 2000s, with their miniscule plastic keyboards and awkward, rectangular shapes?

Back then, no one would have predicted that, come 2022, Blackberry devices would be (almost) unheard-of; whereas iPhones – definitely “second-movers” in the smartphone market – would make up 22% of all smartphone sales in Q4 2021, making iPhone producer Apple the top smartphone vendor globally!

Why is this important?
Because in today’s race to develop new, innovative mobile applications, there’s an overwhelming “first-to-market” mentality that leads well-meaning founders and creators to rush product development, just to make sure their idea is the one that sticks in users’ minds by being first (and not necessarily best).

Paired with a potential lack of experience and knowledge to make actionable plans for their UX and UI goals (which is understandable: no one’s born a UX or UI designer!), this rush will inevitably lead to a roadblock. Maybe before their application launches, maybe after – but the fact that certain key steps have been overlooked will have a significant impact on the application itself.

Imagine it like this:

”First-to-market” mentality + lack of experience

↓

Diving straight into designing the app (without research or **testing**)

↓

Development team missing key information

↓

Unproven onboarding flow, custome journeys etc.

↓

Several issues post-launch (needing real-time fixes)

What Sets You Apart

Letting go of this racing attitude, on the other hand, and striving to be BEST, not FIRST, will ultimately lead to a much stronger (and more viable) product. (Like the iPhone, for example!) One that lifts out from the competition by giving users exactly what they need, when they need it.

But how does anyone know what their users will want with such precision?

Two words: USER TESTING.

It's not rocket science, after all: the reason we develop digital products (like mobile applications) is that **real-life people will use them and enjoy them**, isn't it? So it makes sense to include these people already at some stage of development, to make sure the product we're creating for them is in tune with their needs, their likes, their pain points.

(And – as we mentioned earlier – the insights won from eagle-eyed user testing will help avoid typical roadblocks and pitfalls your competitors will definitely tumble into!)

Here are some of the key findings we can gather from user testing:

User-centric design and compliance

Regulations and Requirements

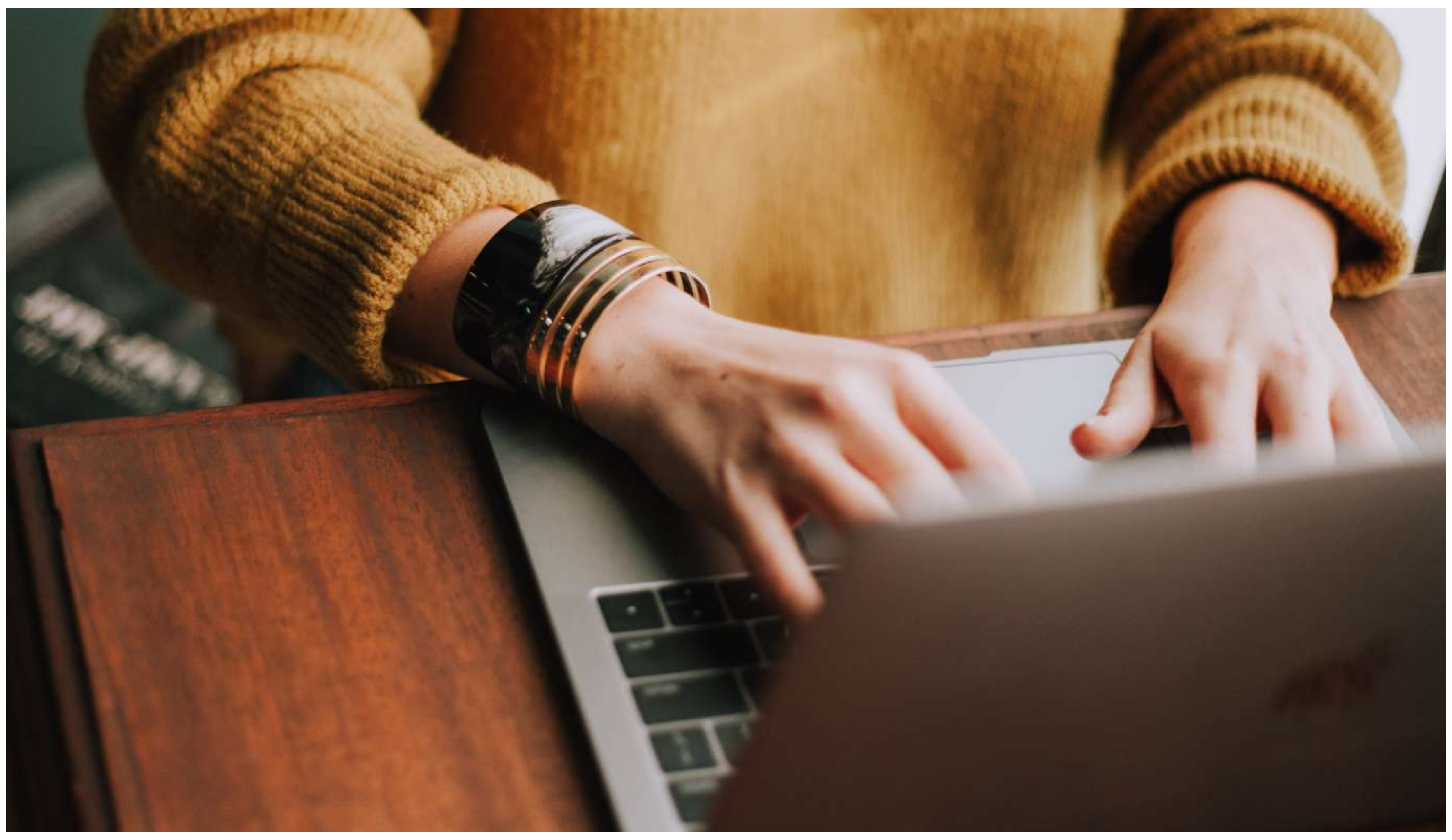
As if user testing wasn't complicated enough, mobile app creators have to keep yet another thing in mind:

Data privacy.
A term that has, all of a sudden, become ubiquitous for anyone interested in digital interactions. And while (as users) we've all become desensitized to cookie pop-ups and GDPR banners showing up on websites around the world, other areas of our digital landscape are also heavily influenced by the constantly changing rules, laws and protocols around data privacy and compliance – like user-centric design, and thus of course, user testing too.

The relationship between user-centric design and compliance becomes even clearer when you look at the intent behind the new privacy legislation. Because, let's face it, no one is thoroughly reading the smallprint of the data processing activities they're agreeing to on websites. Which is quite problematic, not only from a compliance standpoint... but from a user experience standpoint!

Fintech companies, for instance, can experience high conversion and user retention challenges while trying to create a user-centric mobile application that also complies with the (usually quite rigid) Financial Services Regulations – especially since they're notoriously likely to change.

This is one of the main reasons why companies will have to start being more upfront and clear about what they do with their users' data. Not only will it make them more compliant in general: it also ensures that all their hard work to make their product user-friendly isn't completely annihilated by "user-unfriendly" compliance banners and processes.



Here are six ways to be compliant and user-centric in your designs:

01

Keep your language plain and conversational

When you ask your users for consent to process their personal data, make sure that the language you use is precise, clear, and unambiguous. No generic statements or heavy legalese. Give it to them straight.

02

Separate your Terms & Conditions from consent requests

Too often, T&Cs and consent are still bundled together. While this is an easy way to get your users to agree to both things, it's not exactly compliant. Get them to agree separately to each thing, and make the value of each agreement clear to them (e.g. newsletter sign-ups, ad tracking and customization, T&C updates, etc.). Generally, users will wonder "what's in it for me?" — beat them to the punch by answering that question.

03

Avoid pre-ticked boxes

When you ask your users to give consent, encourage them to actively consider what they're saying yes to. That means no pre-ticked boxes, pre-filled in fields, or anything that'll make it unclear which permissions are optional or mandatory.

04

Name all third parties

We read it all the time, but does anyone actually know who these "third parties" are? Chances are, you've no idea. Wouldn't it be nice if you as a user were informed what third parties are making use of your consent? Start considering how you can create more clarity and transparency about who has access to your users' data. Your users (and the new GDPR regulations) will thank you.

05

Make your consent more granular

Different flows require different levels of consent. By separating your consent for marketing and communication into e.g. separate checkboxes (one for email, one for text messages, etc.), your users can see what data processing activities are saying yes to. And so can you! This is a great way to find out what your users are interested in (allowing you to make your marketing even more precise).

06

Make it easy to withdraw consent

Just as you should make it as easy as possible to give consent, the reverse is equally true and important. Let your users know they can withdraw their consent at any time, and give them clear and simple instructions on how to do so.

Remember: These are just six relatively easy ways of becoming more compliant in your UX thinking. It's important to understand that compliance in design is not going anywhere and isn't a "nice to have" anymore.

It's a "need to have".

For better or worse, compliance considerations will always create an extra layer whenever you design and build a new feature. Anytime you run a user test, it will basically function as an additional element that may have an influence on your results. It can have an impact on user conversion, engagement, retention, and so on.

This doesn't necessarily have to be a bad thing! In fact, for the companies who get it right, it can be a huge advantage and the thing that sets them apart from their competitors. Companies who find the balance between compliance and user-centric design also tend to succeed in creating a cohesive overall user journey.

Or, in other words: they tick all the proverbial boxes.

So, the next time you plan a user test for your mobile application, make sure you cover that all-important balance between compliance and user-centric design. Because the companies who manage to incorporate both? Are the companies who will end up winning.

The Safe Way Forward

Obviously, when it comes to user testing mobile applications, the most urgent issue comes up during onboarding or sign-up flows, when users need to submit data such as names, telephone numbers and/or other personally identifiable information (PII).

Luckily, there are workarounds: while some types of information (e.g. social security numbers) should **never** be collected, for non-sensitive PII such as names, email addresses etc., users can use “dummy” or fake entries, or the testing team can blur out all information before presenting it to all others working on the application.

This way, not only will you be compliant with all data privacy regulations, but your mobile application will also be tested, tried and user-centric!

So what exactly is the difference between sensitive and non-sensitive PII?

Sensitive PII includes information that isn’t available elsewhere or may somehow put the individual at risk if it’s made publicly available. Some examples include:

- Personal email address;
- Personal phone number;
- Official documents and data (social security, driver’s license, passport, credit and debit cards, bank account, tax ID numbers, biometrics, medical or criminal records, etc.);

Non-sensitive PII includes information already made publicly available (with consent!) on e.g. a website or social media profile, and likely won’t put the individual at risk as it only contains quasi-identifiers. Some examples include:

- Work email address;
- Work phone number;
- Zip Code;
- Gender;
- Religion;

Qualitative vs. quantative testing

From Data to Insight

Let's say you have an idea for a mobile application. You've done your research, you have your proof of concept, your brilliant design and development teams have put together a working prototype (well done!) – and, because you've read this handbook, you're committed to some serious user testing.

Brilliant! The only question is: how to do it?

By now you can pick and choose from an enormous offering of custom, out-of-the-box user testing applications (our favourites are Mixpanel, Heap or Fullstory). You can add any and all of these to mobile-based digital products to capture business insights; each of them lets the application's owner track users' in-app activity, with the aim of understanding them better.

So far, so good – but the real challenge: how do you turn all that collected data into actionable insights?

This isn't just a user testing dilemma, by the way: as more and more aspects of our (digital) lives become data-driven, each and every industry has to find their own answer to this question. Data science, data engineering, big data... There's a reason why these topics are top-of-mind for business leaders today.

“What” Data or “Why” Data?

How we turn data into insights, as any researcher can attest to, depends very much on the type of data we’ve collected: whether we chose to carry out quantitative user testing, leading to what we often call “what” data, or qualitative user testing, which yields more “why” data:

Qualitative	Quantitative
Answers the question:	
Why?	How many and how much?
Goal:	
To inform design decisions;	To evaluate the usability of an existing site
To identify usability issues and find solutions for them	To track usability over time
	To compare site with competitors
	To compute ROI
When is it used:	
Anytime - during redesign, or when you have a final working product	When you have a working product (either at the beginning or end of a design cycle)
Outcome:	
Findings based on the researcher’s impressions, interpretations, and prior knowledge	Statistically meaningful results that are likely to be replicated in a different study
Methodology:	
Few participants	Many participants
Flexible study conditions that can be adjusted according to the team’s needs	Well-defined, strictly controlled study conditions
Think-aloud protocol	Usually no think-aloud

Obviously, “what” data is more mathematical, perfect for putting together user statistics; in fact, we could argue that it’s less easily manipulated than “why” data, which has a far less tangible dimension. Even so, qualitative testing is every bit as important as its mathematical counterpart: live user testing offers a first-hand look into how our users experience customer journeys, user flows and the overall build of our mobile application, something that would be infinitely more difficult to derive from quantitative testing!

Getting back to our task of turning aggregated data into actionable insights: for an informed approach, gathering both “what” and “why” data is definitely best practice. However, if absolutely necessary to choose one or the other, we strongly recommend turning towards **live user testing**, instead of merely relying on statistics: it’s the perfect way to get to the “why” of everything. (Not just data.)

Tips for live user testing

01 Plan out user testing stages carefully

02 Try to create environments close to users own reality

03 Aim for a group of 5 people (studies show that this will highlight 85% of usability issues)

04 Screen users for suitability beforehand

05 If possible, let product teams sit in and watch first-hand reactions too

User testing 101

Now that we've covered why user testing is essential for developing mobile applications that stand out from the crowd, it's time to dig a little deeper into the practicalities of user testing itself! Definitions, testing methods, challenges, examples... It's all here, enjoy!

How do we define user testing? And how is it different from A/B testing and/or focus group interviews?

Here's a working definition for user testing: **a process where (digital) products are tested on targeted user groups to get feedback and gain insights into user behavior, expectations and needs.**

User testing is different from A/B testing because users are given tasks to complete while also asked to describe their thought process (if it's task-based testing), so it's slightly more complicated than just comparing one option against the other – which, in a nutshell, is what A/B testing asks you to do.

User testing also includes an interview session, in addition to the testing session. But, unlike focus group interviews (which are mostly reflections of product use, but don't involve actually using the product), it's not "just" an interview either.

Here's the difference between all three of them in a nutshell:

- A/B testing lets us know what happens when users interact with our product
- Group interviews let us know why users act(ed) the way they did
- User testing lets us know both :)

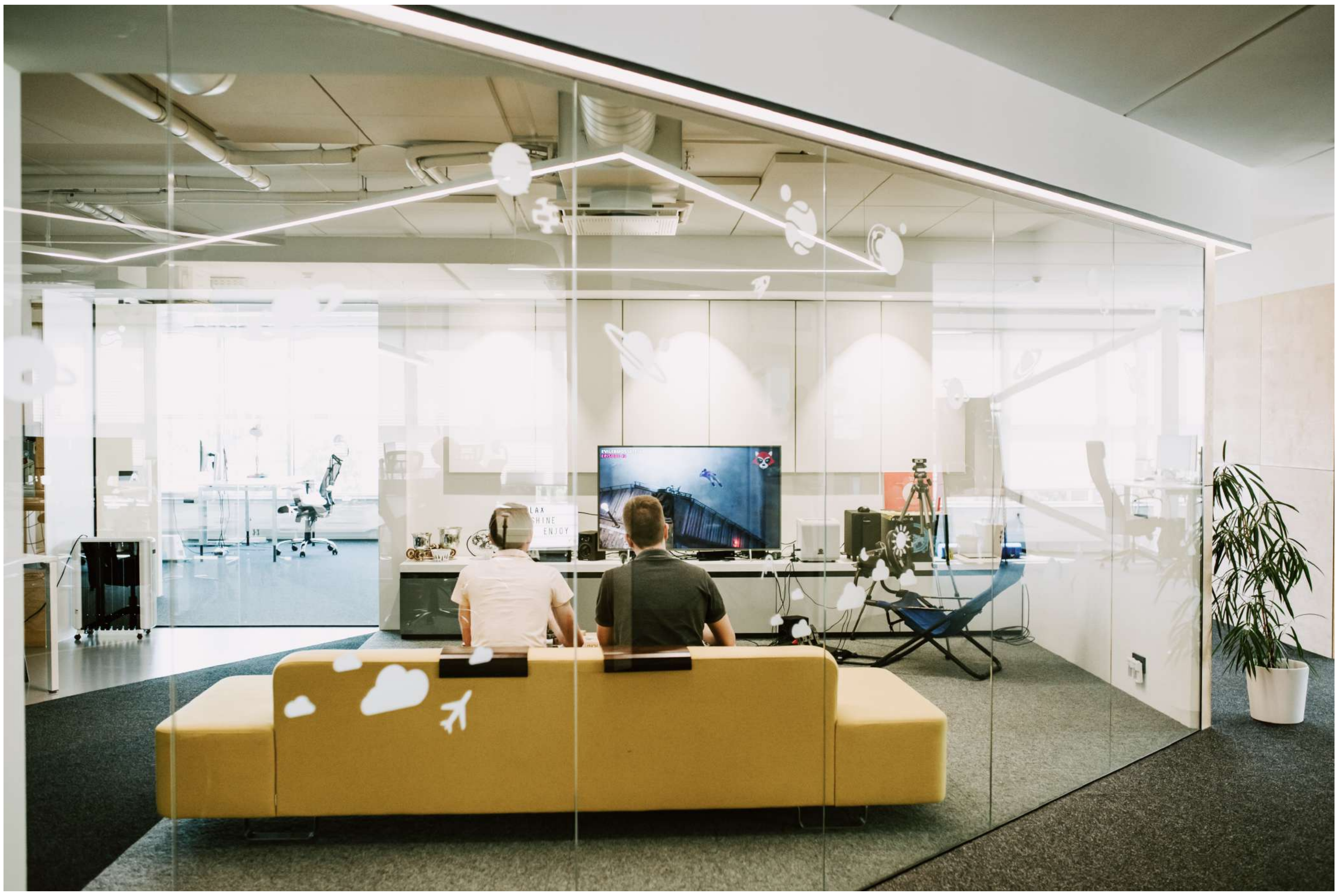
When should we use user testing, and when should we choose A/B testing instead?

A/B testing is great for smaller interactions in your mobile application, like whether an icon should have a text label underneath, or if its function is clear without the label. You should choose user testing when you want to gain more insights or make sure that the assumptions you made for your user flows (while building your mobile application) are actually valid.

Should I choose between user testing and A/B testing based on what my company does?

It's more a question of what your company's doing at the moment. Companies building a new digital product, or launching new features for that product, should definitely choose user testing. Companies that already have a working product and want to validate different ideas, on the other hand, should choose A/B testing.

...Generally speaking, all companies will need to do some user testing at some point. A/B testing is more of a quick and easy way to get large-scale feedback on small changes.



A real-life user testing session

Here at Mooncascade, we do live testing and in-person testing sessions too. Both work really well, but when some technical issue crops up – and sometimes it does! –, it's easier to fix it if we can just quickly reboot the prototype, rather than try to explain to our test subjects how to manage it all through a video call.

We usually use Lookback for screen recordings, live comments for the test session itself, and then use the recordings afterwards for analysis. During the session, we ask our test users to think out loud, so we can understand what they think and expect.

(We also usually give a small gift to our test users for the hour of their time they've spent with us – like an Amazon gift card or something!)

Then once we've analysed the outcomes and come up with new solutions, or just implement the necessary adjustments, it's basically a "rinse-and-repeat" situation to make sure the adjustments are actually valid.

Does user testing ever have bad results?

There's no such thing as bad results! Like this short case study: we once did prototype testing for a client who had an idea – something that might be relevant and valuable for potential users. Before we sent the product to the development team, we did some user testing and found out that actually, there's no real need for this kind of product on the market.

In a way, this was a bad outcome: our client didn't continue with the implementation. On the other hand, we saved them a LOT of time and money, because they didn't end up building a product that had no actual market value!

Or here's another example: one of our clients increased their conversion rates by 40% because we ran a few sessions of UX-focused live user testing. These sessions gave us actionable insights on how users viewed the client's existing services. This then allowed us to come up with quick wins and long-term solutions by designing onboarding flows that better targeted the needs and expectations of the client's user base.

What are the biggest challenges of user testing a product? And how to avoid them?

The biggest challenge is usually recruiting valid test users. (This often happens when the client doesn't have well-defined user groups.) This can be fixed by first creating user personas, and then submitting them as requirements for user recruitment.
Another challenge: guiding the test users too much, since this won't let them think with their own minds, which leads to a tampered outcome. The trick to this is making sure all user testing tasks are open enough (i.e. they can be interpreted freely, but they're also understandable).

And last but not least: do a pilot run before actually going to the first batch of external test participants! Pilot runs can be done in-house and are perfect for ironing out all the kinks in the process. Because there's nothing worse than an unexpected bug messing with the flow of a user test and influencing the overall outcome.

Is the environment of a user testing session really that important?

Yes, it is – it should be as close to our users' reality as possible. If we're testing a mobile application that will be used by farmers on a wheat field while driving a tractor, then it makes sense to actually do the testing on that wheat field as well, doesn't it?

Making sure test users are comfortable with the testing environment is essential because they need to feel unashamed and unembarrassed to express their most authentic opinions. It's important for people conducting the test sessions to remain open-minded and remember that all reactions and feedback improve the end product – even if test users say the mobile application they're testing is the most useless thing they've ever seen, their feedback is still valuable. (And if all test users say the same thing, it might be time to rethink the whole thing…)

How should we evaluate the results from user testing?

All insights from user testing should be written down systematically and analyzed. Then it's easy to pinpoint the bigger pain points, and craft suggestions for improvement. (Usually, user testing is followed by a comprehensive UX/UI review, which includes a full analysis of all design and UX-related product issues.)

One last thing...

We can't stress this point often enough: truly knowing your users and their needs is the wisest investment for your business, especially in this digital age. No matter the industry you're in, user testing is pivotal in ensuring that people will actually use and enjoy the mobile application you spent a long time developing. And while you may worry about the time and money user testing costs, think about the time and money you end up saving by getting it right from the start.

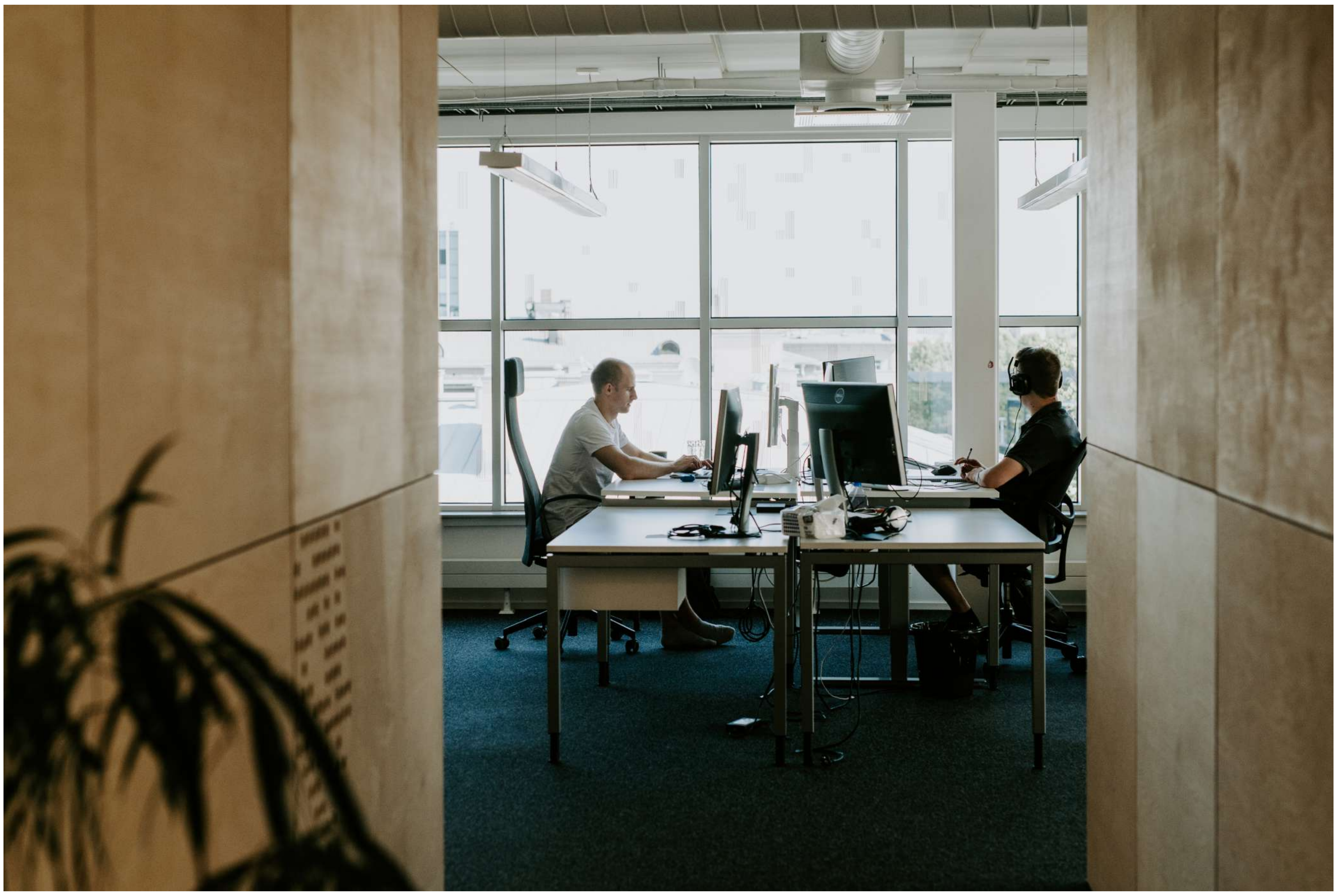
In that sense, it's really all about a shift in mindset when it comes to user testing. The ability to go from "how much will user testing cost me?" to "skipping user testing is going to cost us". The insights you gain from user testing will remove guesswork and gut feelings, meaning you won't waste money redoing things down the line (and lose out on users in the meantime!).

At the end of the day, user testing is not just user-driven and data-driven: it's revenue-driven. Knowing your users makes for a better user experience, and therefore a better product, and a healthier bottom line.

So stop striving to be the first.

Strive to be the best.

About Mooncascade



Mooncascade is an advisory, design, and software development company that guides businesses through the ever-changing digital landscape. Mooncascade's mission is to help small companies grow and established businesses move quickly.

If you work within the fast-paced and highly regulated FinTech industry, Mooncascade can apply a wide range of skills and experience to your mobile app development. Whether you need help with scoping a new product, adding a new solution to an existing product lineup, or making sure your back-end set-up can support your new product plans - Mooncascade knows the way.

Need an experienced FinTech mobile app development team to partner with? Get in touch with the Mooncascade team today.

MOONCASCADE

contact@mooncascade.com
+372 504 5119

mooncascade.com

LINKEDIN TWITTER FACEBOOK CLUTCH