

# 7

# PROFIT-DRAINING SAAS UX MISTAKES

Business Owners' Guide to Stop  
Wasting Money and Improving  
User Experience



CSABA HÁZI



Launching a new digital product is a daunting and tricky task.

In my years of experience as a UX professional, I have designed numerous digital products, collaborating with founders in early-stage startups, and contributing to large enterprise projects that served millions of users.

Unfortunately, this knowledge and experience are hard to come by. That's why I created this guide: to bring this knowledge to every startup and company looking to launch and scale their digital products.

By learning from the mistakes that cost other companies (including my own) several hundreds of thousands of dollars, you can focus your resources on making your product truly exceptional.

Let's dive in!



# Who am I and why listen to me

I'm a UX expert specializing in complex SaaS (Software as a Service) web applications and helping companies launch and scale their MVPs.

Beyond strategy and design, I am often responsible for managing teams or even stepping in as partner. I do whatever it takes to ensure the product's success.

I wrote two books, one of which became an Amazon bestseller in its category: 7STEPUX®: The Complete UX Process from Strategy to Design.

**115**

products designed

**15**

MVPs launched

**50M+**

Users served

I have witnessed teams falling apart, products failing to find market fit, and tons of investor funds, company revenue, and hard-earned cash of founders being wasted.

I wrote this guide to help you avoid these mistakes.

# Table of contents

No real understanding of the product development process .....	5
Functional, but not usable MVPs area not enough to go to market .....	13
Allowing poor frontend development sabotage your UX .....	19
Not building a design system .....	24
Not having a clear (and documented) product vision .....	31
Skipping documentation .....	36
Not investing in UX early on .....	43
360° UX Audit .....	49

## **MISTAKE #1**

**No real  
understanding of  
the product  
development  
process**



Recently, I consulted with an edtech (education technology) startup that had created a platform for educators in the US. The founders were incredibly talented and well-prepared, but **they had never built a digital product before.**




They opted for what seemed like a logical choice: they **hired a development company** to create the product based on their vision.

Several months and hundreds of thousands of dollars later, they ended up with a **barely usable product**, riddled with bugs, unintuitive interfaces, and lots of frustration.

The founders reached out to me and asked for an audit on their product to get an expert's opinion and guidance on how to move forward.

The more I audited the product, the worse I felt. I saw just how bad the implementation was. Leaving aesthetics aside, the navigation was incoherent, similar features worked differently on each page, and the whole product felt cluttered and cheap.

This was not the experience the founders wanted to present to the market.

Product strategy	
Product management	
Research	
UX design	
UI design	
Development	

*Many founders and business owners only know about UI design and development”*

## What was the mistake in this story?

Unfortunately, the founders did not know that a **development company would execute whatever they were told**, but they are not UX professionals, and they won't be able to design the product.

While some companies offer a full-range solution, most companies specialize. You might pick an excellent development company, but they still won't design the product for you.

# Product development process

Here's a more detailed process with methods and deliverables:



## Planning and kickoff

Gather the necessary information from stakeholders, set up a timeline, and define the process for the sprints. Create documents around business goals, users, ecosystems, and stakeholders to document our initial assumptions and ideas.

Project canvas

Kickoff workshop

Ask the experts

Journey mapping

Ecosystem map



## Research prep

Define the questions that we need to research, and create a plan to conduct both quantitative and qualitative research.

Research questions

Assumptions & hypotheses

Method selection

Define sampling

Research plan



## Research

Conduct research, collect and organize data, and report findings. For quantitative research, utilize data, primary sources, and secondary sources. For qualitative research, use methods such as interviews, observations, and diary studies.

User interviews

Field observation

Preliminary usability test

Card sort

Data analysis

### DELIVERABLES:

RAW data

Research wall

Research report





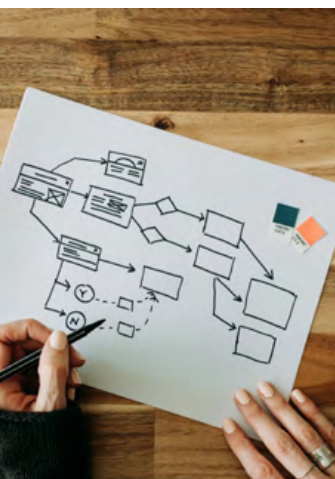
## Insights

Gather insights from research and turn them into key insights, user stories and Jobs to be done. Compare results with initial assumptions.

Key insights

User stories

Jobs to be done



## Ideation

Define the scope for ideation and employ various techniques to generate concepts suitable for prototyping, such as concept sketches and Crazy Eights.

Collaborative sketching

Future state journey map

Service/product blueprint

Brainstorming, brainwriting



## Low-fidelity prototype

Create low-fidelity sketches and wireframes from the selected ideas. The goal is to flesh out concepts quickly to be able to test them early on.

Low-fidelity sketches

Wireframes

Clickable prototypes



## Usability testing

Conduct a usability test on the low-fidelity prototype to identify potential flaws and issues early in the process. Iterate based on the feedback until you are confident enough to transition to high fidelity.

Moderated live usability test

Moderated remote usability test

Unmoderated remote usability test

### MEASURE

Task success rate

Completion rate

User happiness

Lostness

Completion time

Ease of use

Feedback on features



## High-fidelity prototype

Create the final UI and clickable prototypes using the design system. Address mobile screens and responsive design. Conduct design review to get feedback on the designs.

UI design

Design system

Clickable prototype

Design review



## Usability testing

Conduct a usability test on the high-fidelity prototype to identify potential flaws and issues early in the process. Iterate based on the feedback until you are confident enough to transition to developer handoff.

Moderated live usability test

Moderated remote usability test

Unmoderated remote usability test

### MEASURE

Task success rate

Completion rate

User happiness

Lostness

Completion time

Ease of use

Feedback on features



## Developer handoff

Make sure the handover happens smoothly with engineering addressing all the questions and designers following up the process and providing QA to ensure quality of implementation.

Supporting developers

Design QA

Updating designs

Updating design system

Design documentation



## Code

Engineers and developers should implement the designs using agile development methods.

Frontend development

Backend development

Staging/QA

Architecture

Agile



## Measure

Measure the feedback on the implemented solution and conduct tests.

A/B testing

Usability testing the live product

Hotjar

Analytics

Data

Customer support tickets

### MEASURE

Conversions

Signups

Time of completion

Churn rate

Feature adoption



## Iterate

Implement feedback and create new versions of the features and the product and test again.

This education startup learned this lesson the hard way.

We had to redesign the product to make the features work and prepare a roadmap to go to market. **In retrospect, the money spent on the initial “designs” could have been saved**, and the product could have launched six to eight months earlier.

### **Lesson learned:**

Founders and business owners must have at least a high-level understanding of digital product development. They need to be familiar with the different roles involved in the process:

- Developers: Frontend developers, backend developers, and architects, QA engineers.
- Designers: UX and UI designers
- Product development: Understanding methodologies like Agile and Scrum, as well as the roles of a Project Manager (PM) and Product Owner (PO).

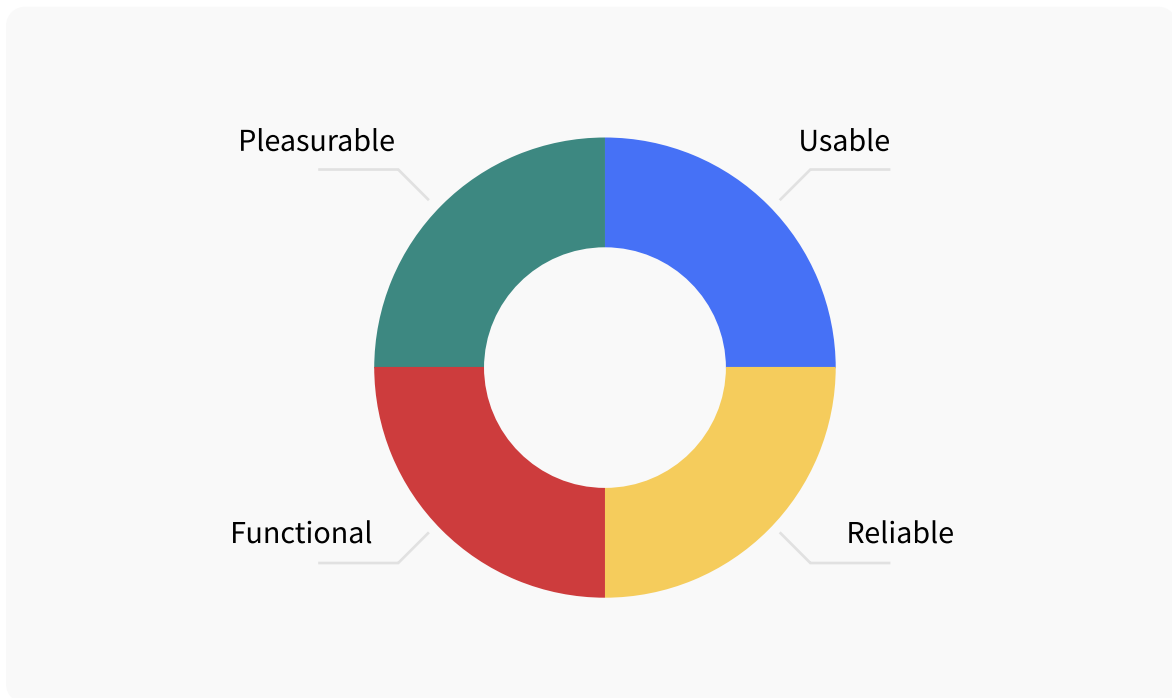
By acquiring a basic understanding of this landscape, you will be better equipped to navigate challenging decisions. Furthermore, it will prevent you from being subjected to incorrect advice from service providers.

## **MISTAKE #2**

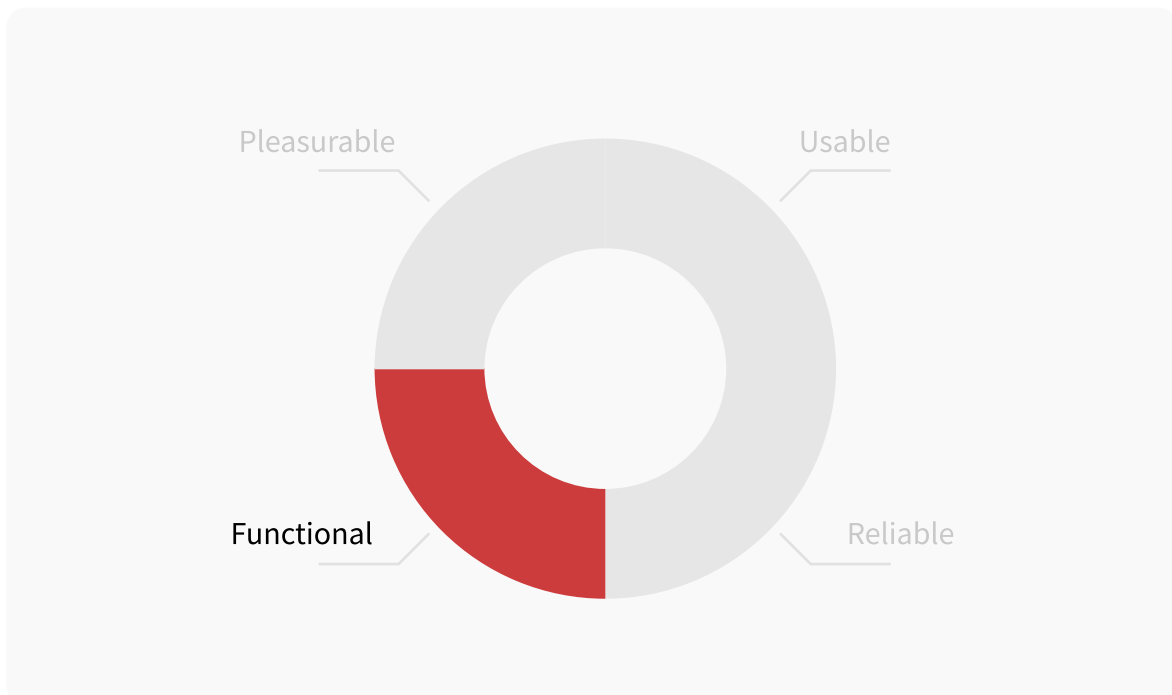
**Functional, but not  
usable MVPs area  
not enough to go to  
market**



MVPs should look like this:



Instead, most of the time I encounter MVPs that look like this:

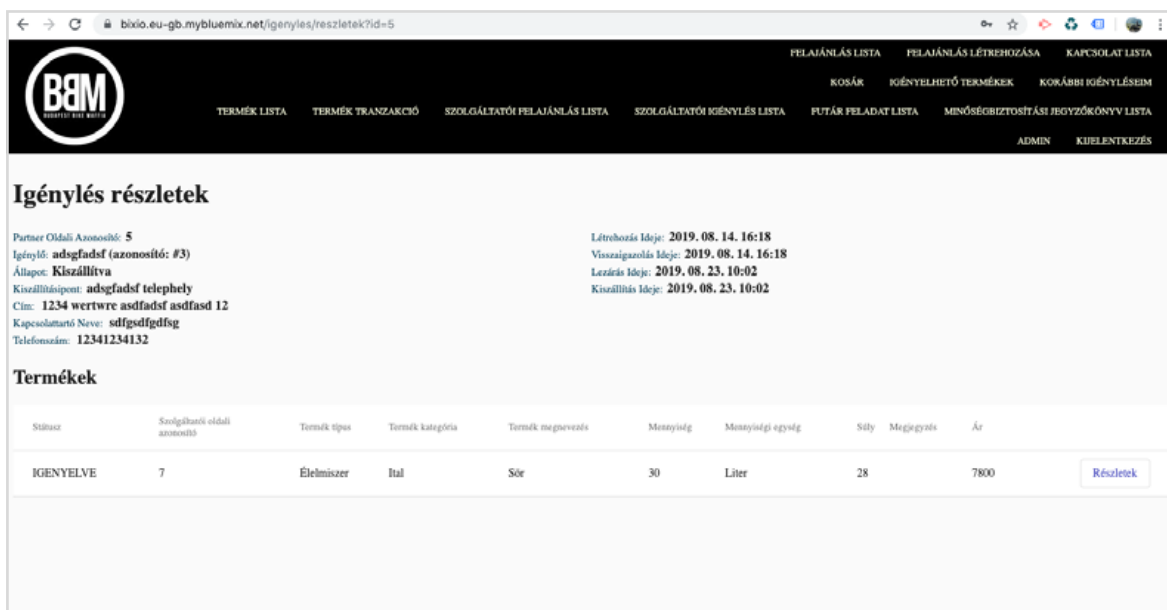


In the past decade, the focus was on shipping fast and learning on the market. People said it was okay to **deliver a lousy MVP** to the market because people would still use it, and you could refine it.

Well... that story is **no longer true**. And it definitely won't work for most of us.

Technology is changing rapidly. Ten years ago, it took six figures and months of development to get an eCommerce site up and running. Now it takes 30 minutes and a \$20 subscription.

People now expect more because we see high-quality products everywhere. Slow websites are unacceptable. Crappy user experiences make us switch to the competitor.



The screenshot shows a web browser window with the URL `bivio.eu-gb.mybluemix.net/igenyles/reszletek?id=5`. The page features a dark navigation bar with the BBM logo and various menu items like 'PELAJÁNLÁS LISTA', 'KOSÁR', and 'IGÉNYELHETŐ TERMÉKEK'. Below the navigation, the main content area is titled 'Igénylés részletek' (Order details). It contains metadata such as 'Partner Oldali Azonosító: 5', 'Állapot: Kiszállítva', and 'Kiszállítás ideje: 2019. 08. 23. 10:02'. A table titled 'Termékek' (Products) lists one item: 'IGENYELVE' (Quantity: 7), 'Élelmiszer' (Food), 'Ital' (Beverage), 'Sör' (Beer), with a quantity of 30, unit 'Liter', weight 28, and price 7800. A 'Részletek' (Details) button is visible next to the product row.

*The functions work, but it's unusable and not user-friendly. It's not enough to go to market and convince customers to pay for it.*

Unfortunately, too many new products still start like this:

- ❏ The owner/founder has an idea.
- ❏ They get a development team to create it.
- ❏ They create a functional but unusable product that looks terrible.
- ❏ They try to sell it.
- ❏ They realize nobody buys it (as is).

The problem is, when people think of MVPs, they assume that minimal means functional.

💡 A product needs to be functional, reliable, and usable and pleasurable. Your product should work without bugs, be reliable, and easy to use.

Another example is when the product is unreliable. This means the software has bugs, sometimes works, but sometimes doesn't.

This is a HUGE problem and can easily make you lose clients because it undermines trust. One of my clients in security intelligence, we learned when talking to customers, is that they only give us one chance.

If the software seems unreliable or gives them false information, they are out. They don't have time for unreliable products.



I've consulted with several companies that started with a development-first app and got stuck. One recent example is a social media startup that **created the web app without talking to any designer or UX professional.**

The product worked, and you could access the features, but it was so complicated to use that you didn't know how. When they showed the product to investors and customers, they didn't understand it. Even worse, they loved the sales pitch, but then they stared at the product, saying, "I don't get this."




**“ ... they loved the sales pitch, but then they stared at the product, saying, “I don't get this.”**

**We had to revamp the design for the MVP**, correcting flows, rewriting copy, and redesigning the pages to make it easy to tell what needs to happen.

This way, the startup managed to raise funds and onboard the first round of customers. Similarly to Mistake #1, the startup lost eight months and around \$120,000 redesigning and re-implementing something that was already built.

Luckily, they were able to recover, but not all businesses can.

## Lesson learned:

-  Don't do development-first MVPs. They don't work.
-  Invest in UX early on (if you don't believe me, Google this topic).
-  A good MVP is functional, reliable, usable and pleasurable. If any of these is missing, you are risking serious money.

Speed is important, and timing is crucial for a product. However, understanding your priorities is paramount. If you believe you can hasten your entry into the market by compromising on quality, only to find later that you must return to the drawing board, then you've lost much more than you could have hoped to save in time.

## **MISTAKE #3**

**Allowing poor  
frontend  
development  
sabotage your UX**



One myth about UX is that it's something only UX professionals are in charge of. This could not be further from the truth.

Imagine giving me your secret cheesecake recipe and clear instructions on how to make it, but I mess up some of the steps and end up with an ugly and suspicious-looking cake. Is it your fault or mine?

Ideally, you work with a **UX professional to plan your product, conduct research, and create detailed designs**. Developers then come in, take the designs, and create the live, coded version of your product.

“ **UX is that it's something only UX professionals are in charge of. This could not be further from the truth**

Frontend developers play a HUGE role in user experience. They're the ones who take the designs and turn them into code. If this isn't done properly, **you could have the best designer from Apple's core team, but the output still won't be good.**

Unfortunately, there are simply a lot more bad frontend engineers than good ones. It's true, and you can tell everyone I said so. Even today, frontend engineering isn't looked at as a UX position, but it should be.

I know brilliant frontend engineers who think like UX pros. They think through interactions, care about the minor details of the designs to be pixel-perfect, smooth, and awesome to use. They raise flags if the designer missed something.

Bad frontend developers just implement the designs somehow (more or less) and make absolutely no effort to go beyond what's in the designs. In my years of designing and QA-ing developers' work, I've identified the most useless way to spend money: giving feedback to frontend developers to fix stuff that could have been avoided.

**“ ..most useless way to spend money: giving feedback to developers to fix stuff that should have been avoided**

For example, there's a design that clearly shows how buttons and input fields should look, and then when it's implemented, it doesn't look (or work) that way.

What I tell engineers is a simple rule: you can change my designs; I don't care. I'm not Michelangelo, as long as it looks and feels great to use.

You'd think that, given clear instructions, developers would do a good job.

Unfortunately, that's not been my experience or that of my professional network.



It doesn't matter how the designs look if they're not implemented correctly. Because your customers won't be looking at pretty design files; they'll be looking at the real thing.

Here's a tough pill to swallow: it's **not financially worth** trying to fix bad frontend work.

Just do the math: a **bad implementation costs you management time**, a **designer's time** to **QA** and provide feedback, and the **developers' time** to fix the issues.

My friend, who leads engineering teams in Switzerland, told me they don't allow this to happen. A frontend developer needs to implement designs perfectly because frontend engineering is so much more than that.

The sad part is that you're **fixing the basics and never get to actually improve your product**. Implementing the designs is just the first step. Then you need to fine-tune it, really make it shine, test it, and release new features that your customers will love. You can't do all this if you spend your time and money fixing the first version so it doesn't look like crap.

## Lesson learned:

- ❏ If the designs are not implemented correctly, UX will suffer
- ❏ Developers are also responsible for the overall UX of the product
- ❏ If you catch yourself fixing development issues most of the time instead of moving forward, it's a bad investment

Here's an additional insight: Interestingly, there are more backend developers, architects, AI engineers, etc., than there are frontend developers. While many engineers do engage in frontend development, they often lack passion for it, and unfortunately, this is reflected in the final results.

**A telltale sign of an exceptional frontend developer** is their attention to detail. They aspire for the product to look fantastic and are willing to go the extra mile, akin to a designer, to ensure it both looks and feels amazing.

## **MISTAKE #4**

# **Not building a design system**





One of the ways companies can waste money is when the design of a product is inconsistent.

For example, having different types of buttons on similar screens when they should have the same design. But these inconsistencies are not just bad for the user experience; they also increase development time because everything needs to be designed individually, without the ability to reuse design components, layouts or other elements. Imagine how much time and money is wasted by redoing the same components and pages over and over again.

**“ One of the ways companies can waste money is when the design of a product is inconsistent ”**

If the design is inconsistent:

- ❏ Designers often find themselves designing the same elements over and over.
- ❏ Developers frequently implement components that have been created before.
- ❏ Consequently, users may become confused while using the product, leading to a less user-friendly application.

The solution is to introduce a design system. A design system is like a library that features the most frequently used styles (colors, typography), components (buttons, input fields, date pickers) and layouts (headers, footers, popup modals) of your product, all neatly organized and defined to the level where it's super easy to reuse them in the future. The rule is simple: if you use something multiple times, add it to your design system. This way, it will reduce development time and make your product look and feel consistent.



*Here, you'll see foundational components that show up again and again in the designs. What's great is that you just need to design these once, pin down all the interactions, get them coded, and then you're set. You can reuse them as much as you want.*

Usually, people don't think about creating design systems early on in the product development process, but only in the later stages of the business. I've consulted many startups that are just entering the scale-up phase. This is where the company has enough money to start hiring top-notch people, building proper design teams, and putting processes in place. And this is when people realize they could move faster and save costs with a design system.

But it shouldn't just be the "bigger" players who benefit from design systems. For startups or new enterprise products, it can make all the difference in the world. You can get the benefits of a design system at any stage of your product:



*Buttons in a design system to keep things organized, clean and well-documented.*

# Benefits of a design system

## Faster development

Design systems can speed up the development process by providing reusable components and patterns. This helps to reduce the need to recreate common elements, saving time and resources. It also enables designers and developers to focus on more complex, unique aspects of a project.

## Better user experience

Design systems promote consistency, efficiency, and accessibility, resulting in an improved user experience. Users are more likely to enjoy and engage with digital products that offer a seamless, predictable, and easy-to-navigate experience.

## Cost savings

The efficiencies gained through the use of a design system can result in significant cost savings. By reusing components and reducing redundant work, organizations can allocate resources more effectively and potentially reduce the overall cost of design and development projects.

## Future-proofing

Design systems provide a solid foundation for future growth and change. They can be adapted and extended as technology and user needs evolve, ensuring that organizations remain agile and responsive in a rapidly changing digital landscape.

## Fostering innovation

Design systems streamline routine design tasks and provide a solid foundation for collaboration. This empowers teams to think more creatively and innovate more effectively, leading to the development of new ideas, features, and products that better serve user needs and set organizations apart from their competitors.

## More scalable product

As organizations grow, it becomes increasingly challenging to maintain consistency across numerous projects and teams. Design systems serve as a central repository for design components and rules, allowing for efficient scaling of design efforts while maintaining a unified look and feel.



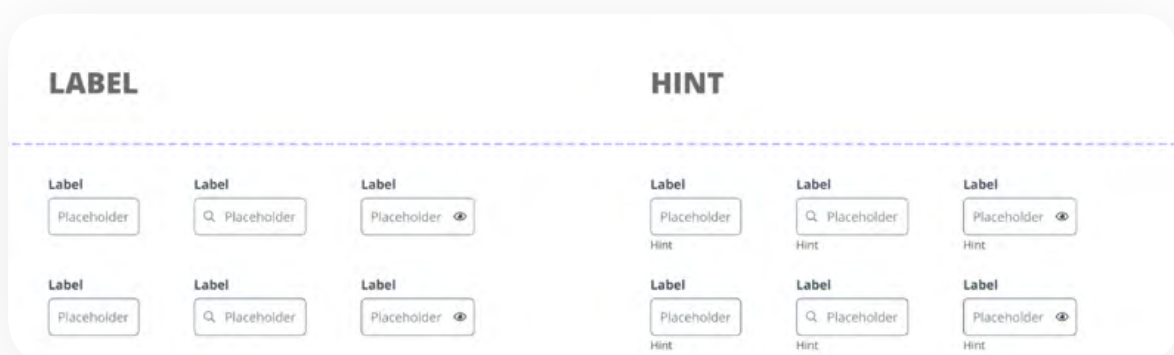
I've architected several design systems, some from scratch, and others in collaboration with development teams using existing libraries like Material by Google.

There is no one definitive way to create a design system; it ultimately depends on the preferences of the team and what suits the product best.

For one of my favorite clients, Edraak, which provides online education in the Middle East, we created a multi-lingual design system that contained all the components and styles for both their B2C and B2B products on the web and mobile.

It was a huge time-saver for design and development, as we only had to create each component once, despite having two versions of every component due to the RTL nature of Arabic text.

We were able to pull out the components and design for each project, enabling us to ship products faster and focus on serving users instead of worrying about pixel perfection.



## **MISTAKE #5**

**Not having a clear  
(and documented)  
product vision**



Have you ever experienced any of the following issues?

- ❏ Struggling to communicate ideas clearly to developers
- ❏ Developers/designers not understanding business goals and how the product should work
- ❏ The team lacking a “North star” and nothing tying the work together

To avoid these costly problems, you need a plan and ensure everyone is on the same page of what you want to build. This is easier said than done, but it’s an absolute must.





In recent years, I noticed a significant shift in my role: **now, it's more about getting everyone to understand what we want to build and minimizing communication issues.**

Most of the clients I've worked with had a clear vision for the product, but where was it documented?

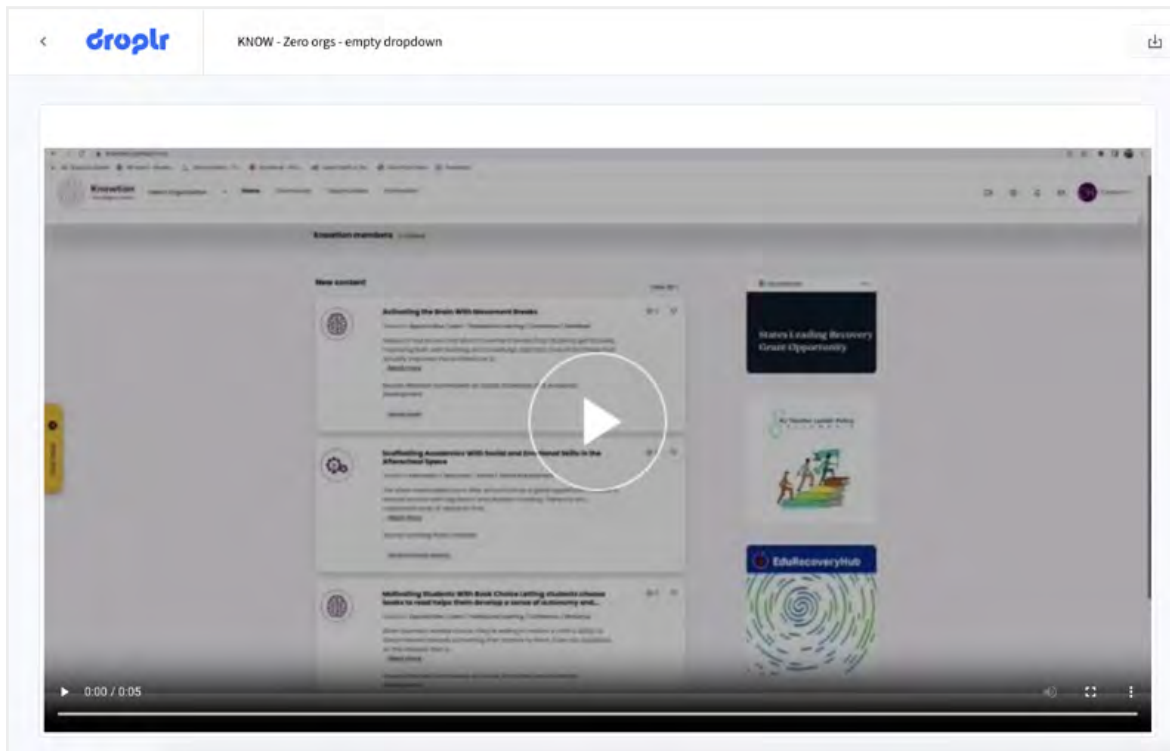
Mostly in their heads, maybe some high level thoughts written down somewhere in a document. But there was no central and clear documentation stating, **“This is what we want to build, and this is how we need to break it down step by step.”**

The issue becomes more challenging when a lot of business logic is required.

I've been working on several AI projects lately, and the designs don't always convey how the logic works since most of it is invisible. ChatGPT's interface is simple and would only require a Figma design file of 3-4 designs. However, is it enough for everybody to understand the product? Of course not.

As a product owner or business owner, **it's essential to ensure everyone understands what we want to build, regardless of what we need to create to make it happen.**

Usually, a product vision documentation includes written parts explaining how an algorithm should work or a payment process, visuals like designs, mockups, timelines, and charts, and even some videos.



*I always record walkthrough videos when presenting designs, explaining functions, or describing product behavior. This method makes it hard to misunderstand.*

I love videos because they help us get as close to reality as possible. Not too long ago, I was working on an AI product for social media.

Even with tons of documentation in Notion and lots of detailed Figma designs, the development team had a hard time understanding how the different parts of the product would come together.

**“ If it’s not written down, it does not exist ”**

To address this, I created a **video demonstrating how the product would work**. It only took five minutes, but it was the best investment of my time.

The video was so convincing that the Team Lead got worried I had another team create the product on the side (while it was just a design prototype).

We could have scheduled several meetings, added more documentation and designs, but a short video that demonstrated how the product should work was all it took to achieve clarity.

**The power of visual storytelling** is undeniable, and it's a great way to communicate complex concepts in a way that everyone can understand.

### **Lesson learned:**

- ❏ It's not enough to have the product vision in your head or telling it to somebody during a meeting. It has to be written down.
- ❏ Create a document with the goals of the product, target audience, every feature with detailed descriptions of what they should do and how.
- ❏ If you make a decision with your team, make sure to write it down, so it's easier to reference it later.
- ❏ Try recording quick videos, sharing your screen as oppose to long emails.

**MISTAKE #6**

# Skipping documentation



Documentation is key. Make sure everything is documented. Done, bye.

But seriously, one of the ways to lose time and waste resources is through inefficient communication.

Do you know what the first thing I do when I start working with a new team is?

- ▣ We set up Slack for communication.
- ▣ I create a Notion space to collect all documentation (Notion is hands down the best for this).
- ▣ We gather everything that has been created so far.
- ▣ I introduce meeting notes.
- ▣ We start documenting decisions.



Many people feel they don't have time to document things. But here's the catch: software development is complex and is usually not done by a 1-2 person team who can quickly update each other.

So, the more time you think you're saving by not documenting things, the more time you'll lose on speed, having unnecessary meetings, solving the same problems multiple times, and doing research on something that has already been researched.

**“ ...the more time you think you're saving by not documenting things, the more time you'll lose on speed...”**

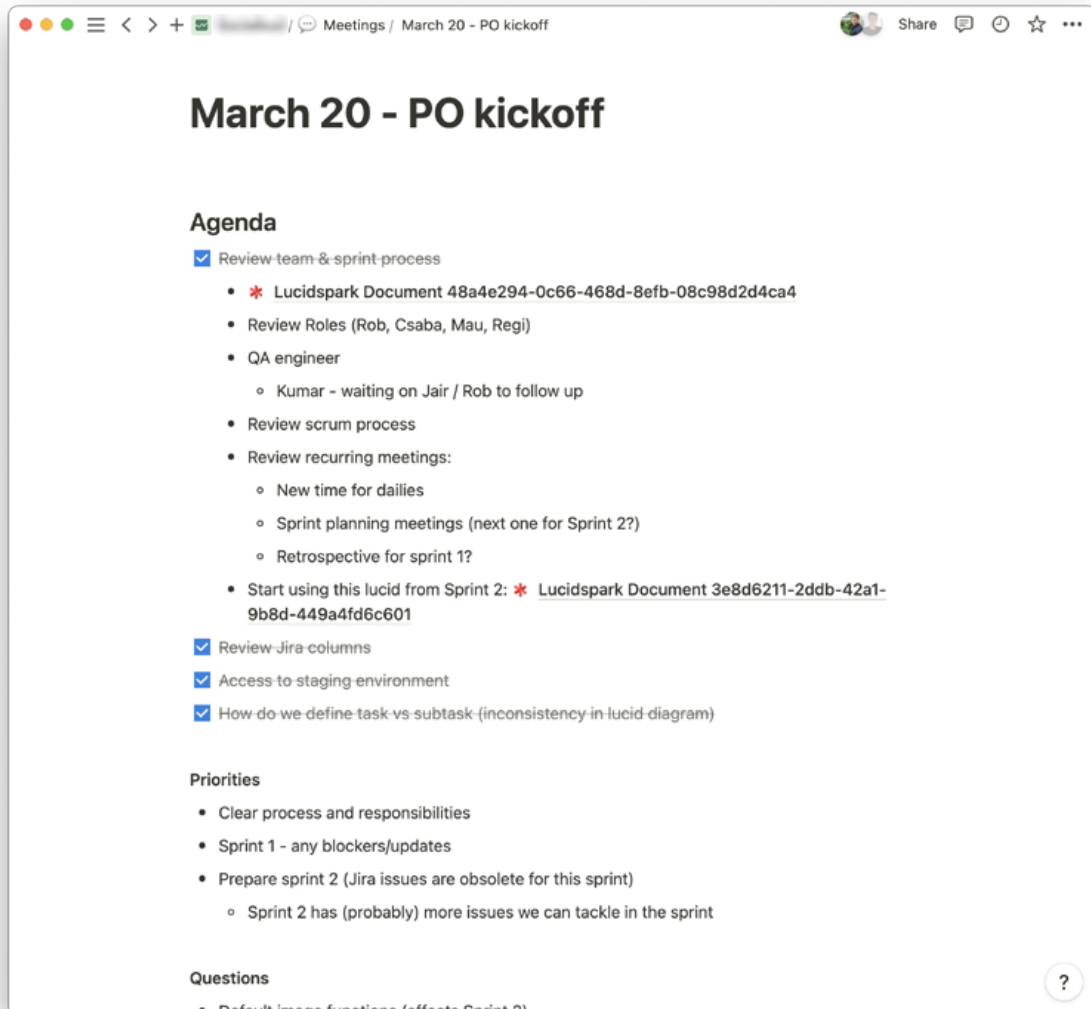
So here are a few things I think everybody needs to start doing ASAP. If you do a few of these, great! Way to go! If you do all of these, you are awesome!

### **Meeting notes:**

Writing an agenda before the meeting cuts down on meeting time and makes the team more prepared.

There is nothing worse than entering a meeting where people don't know what's going on. Just do the math on how much money that costs the company.

During the meeting, record all decisions and action items. That way, if you need to refer back to a decision made earlier, you can do so. It's also great for looping in people who couldn't make it to the meeting.



*Here's a simple meeting notes template. For the agenda, prepare it before the meeting. Include priorities, questions, and delegate tasks. I request everyone to review and contribute to the agenda prior to the meeting. With adequate preparation, we can save approximately 60-70% of the meeting time.*

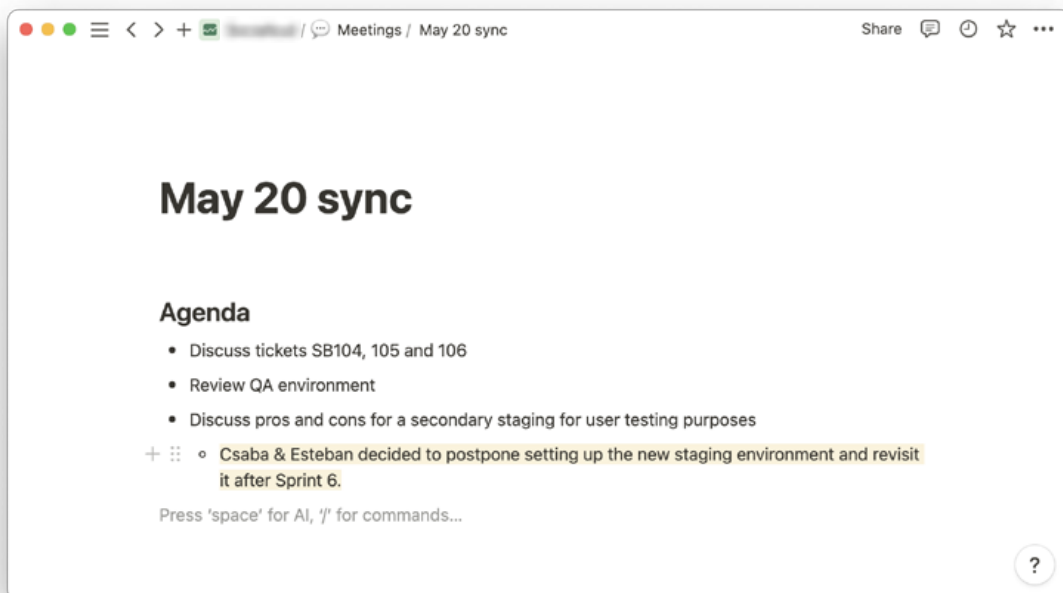
## Document design and development decisions:

It's super important to record if a decision has been made, such as "Steve, let's skip the onboarding for now and focus on the other priority tasks. We'll circle back to this later."

If it's not recorded somewhere accessible for the entire team, I guarantee there will be a meeting in a few weeks where someone will ask, "But why is the onboarding not moved forward?!"

The worst is when people are stressed out because of a delay or something that's not working out, and then you try to figure out what happened.

People get defensive because they worry they'll be in trouble, but in many cases, things just got lost in translation. **The easy fix: if it's an important decision, make sure it's written down.**



*Sync meeting notes in Notion. Decisions are clearly stated, making them easy to refer back to if needed later.*

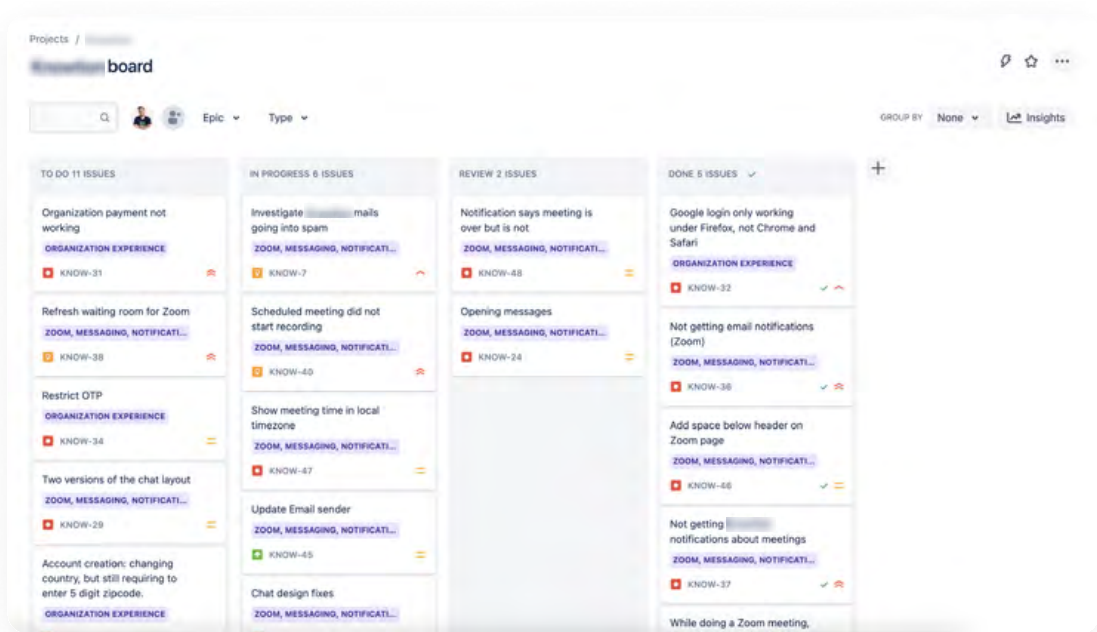


## Keep track of tasks:

Maybe you think, “Yeah, Csaba, you’re really saying we should keep track of things we do?” Well, I wouldn’t put this on my list if I didn’t see this issue with almost every client I work with. It’s so trivial, but too many people miss it.





When it comes to documentation, I love simplicity. There’s no need for fancy software or overly complex logic. The more change you impose, the less likely it is to have follow-through.

Going with something simple that doesn’t feel overwhelming is always best to drive change.



*Simple Kanban board in Jira to keep track of tasks. Jira is excellent for managing digital products.*

## Lesson learned:

-  Documentation is key
-  Write down decisions, plans, strategies
-  Ask developers, designers, and marketers to write everything down, especially when it may be important to revisit a decision in the future.
-  Documentation should not be hard. Pick a tool, such as Notion (or Google Drive), open a document, and take a few minutes to capture important decisions and notes from meetings.

## **MISTAKE #7**

# **Not investing in UX early on**



I'm sure you've heard that spending \$1 on UX can yield a \$100 return.

When I talk about UX, I don't just mean the "design" part. As we discussed in Mistake #3, poor frontend development is a UX mistake.

Do users care about whether the developer or the designer messed something up? No. They simply won't like the product that much and might cancel their subscription.

“ **...spending \$1 on UX can yield a \$100 return** ”

Some people I've worked with in the past have asked at the beginning: *“Csaba, I'm not sure if we're ready to start working with a UX professional yet.”*

It's funny because **involving a UX professional early on can actually save time and money** in the long run.

Recently, I worked with a startup in the retail industry. They wanted to create an app for supermarket chains, and they had a sort-of MVP done by developers, mostly to test out technology. Needless to say, the MVP wasn't good enough to go to market. They tried, but supermarkets just didn't understand how to use it and the value.

Recently, I collaborated with a retail startup aiming to develop an app for supermarket chains.

They had an initial MVP mostly for tech trials, but it failed market tests due to user misunderstanding and lack of perceived value.

On collaborating, **we initially mapped out customer journeys without and then with the app.** We identified the supermarkets' challenges and how we could ease them.

The founders had implemented a voice assistant, expecting uniqueness. **However, examining the supermarkets' and shoppers' actual issues revealed voice was not the appropriate fit,** and there were more urgent problems.

Unfortunately, resources had already been spent on the now redundant voice assistant. This process alone saved significant resources by preventing unnecessary design and implementation.

Had we conducted this earlier, the startup **could have launched sooner** and saved resources on the voice assistant.

The retail startup example I mentioned earlier serves as a powerful illustration of how investing in UX early on can save time, money, and resources. But what can you expect when you involve UX professionals in your project?

## **Prioritize what problems to tackle:**

UX professionals help you identify and prioritize the most critical pain points for your users. By focusing on the most pressing issues, you can create a more effective product that truly addresses users' needs.

## **Use research to provide data for the process:**

UX professionals can conduct user research to better understand your target audience, their preferences, and their pain points.

This data-driven approach ensures that your product is built on solid insights, increasing the likelihood of success.

## **Keep the team on track and aligned with the product vision:**

A UX professional can serve as a mediator between different departments, ensuring that everyone is working towards the same goal.

By keeping the team focused on the product vision, you can avoid wasted time and effort on features that don't contribute to the overall user experience.

## **Help you document and manage the product:**

UX professionals can create documentation, such as user flows, wireframes, and prototypes, to guide the development process.

These resources help ensure that everyone involved in the project has a clear understanding of the desired end result, reducing miscommunication and promoting efficient collaboration.

## **Validate and iterate on ideas:**

UX professionals can conduct usability testing and gather feedback from users to validate and refine your product.

This iterative process helps you fine-tune your offering, ensuring that you're delivering a product that meets user needs and expectations.

## **Optimize for accessibility and inclusivity:**

UX professionals can help ensure that your product is designed with accessibility and inclusivity in mind, making it more usable and appealing to a broader range of users.

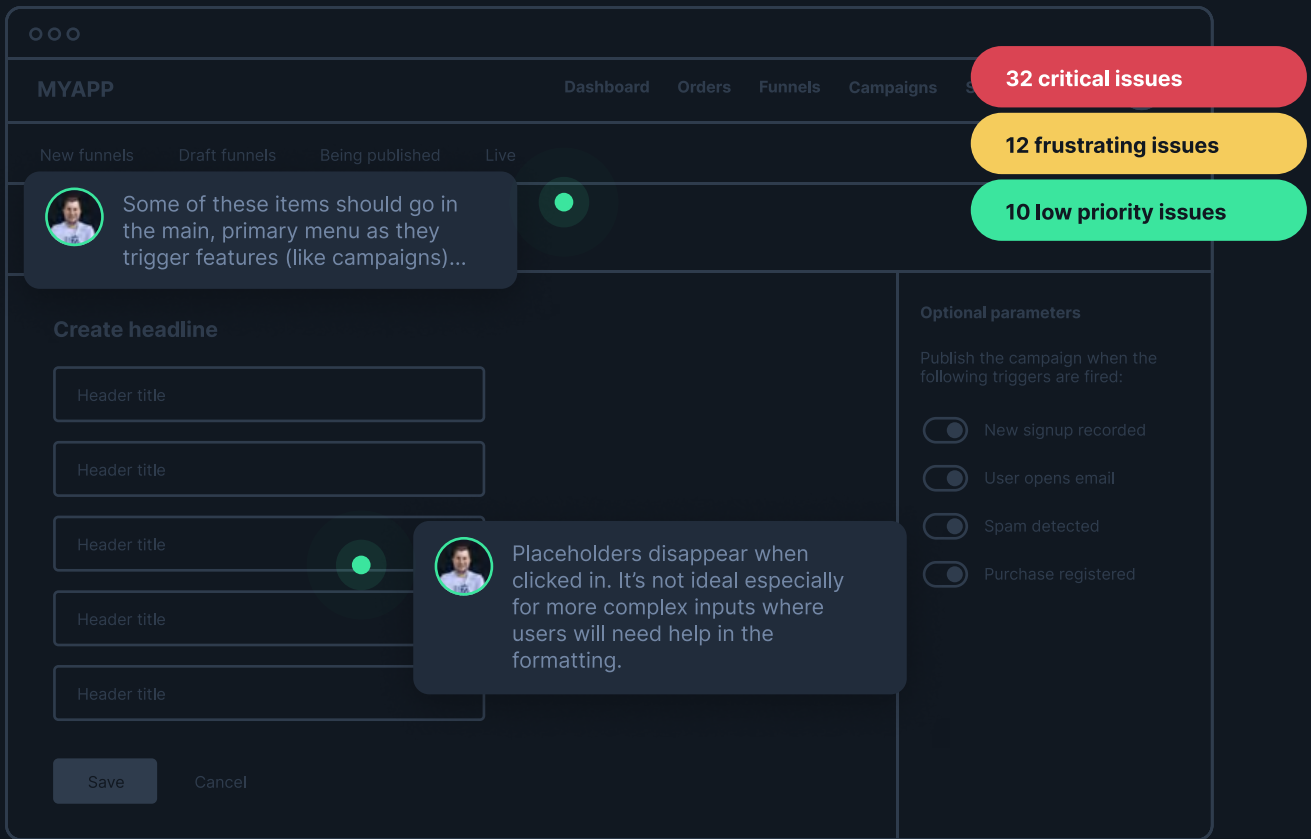
This not only benefits your users but also helps your company meet legal and ethical obligations related to accessibility.

## Lesson learned:

- ▣ The earlier you involve a UX professional, the more money you can save on development
- ▣ UX professionals will help you launch your product sooner to the market
- ▣ A good UXer will not start with UI design but with strategy, research and a gameplan to launch your MVP to the market







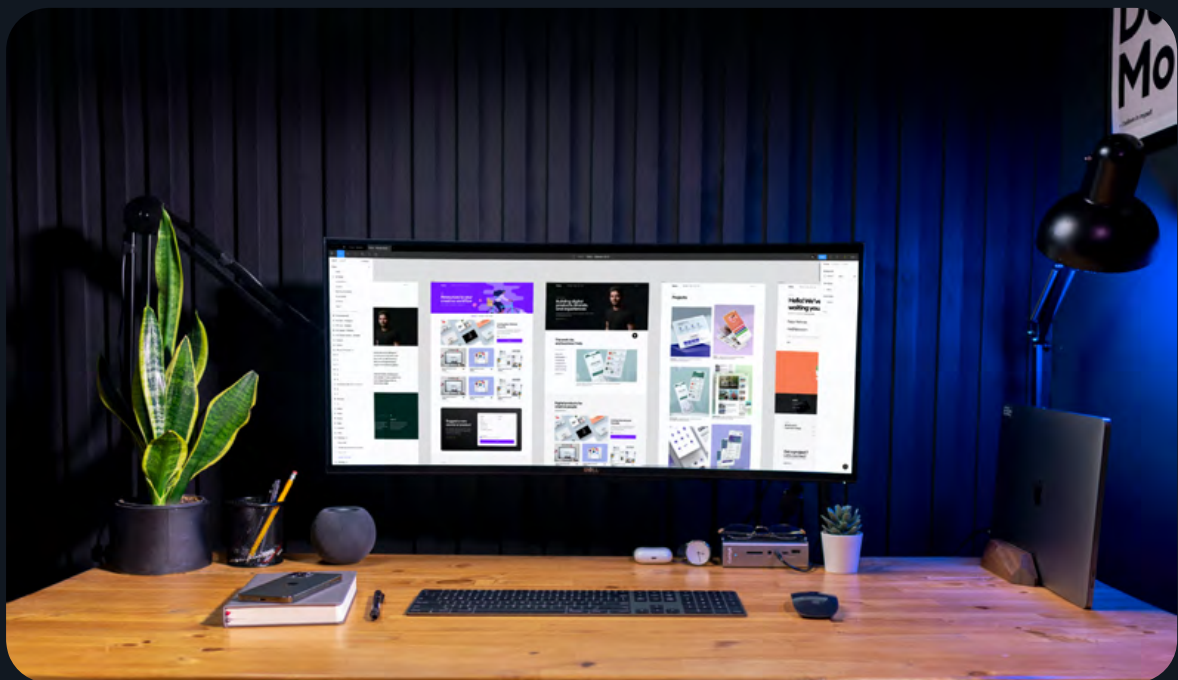
# 360° UX AUDIT

# 360° UX AUDIT

Let's put your product through a rigorous review and find out what could be improved.

During the audit, I will get to know your business goals, your team, and analyze your product to provide actionable feedback and recommendations on how to improve it with valuable insights:




- 📊 Testing your product and documenting the findings
- 📊 Detailed insights on UX issues (both design and engineering)
- 📊 Recommendations on what to focus on
- 📊 Recommendations on next steps to solve the problems



# 360° UX AUDIT





## Actionable insights

Everything I do is actionable. Even if you have a designer or want to work with someone else, anyone can implement the findings of the audit. My simple formula is always:

-  What's the issue?
-  Why is it a problem (and how severe)?
-  How can it be solved?

## Fast-track to success

This audit is great if you...

-  Struggling with your current product
-  Want to go to market faster, but not sure how
-  Want to create better user experience
-  Want to reduce development costs

[Learn more about the 360 degree UX audit](#)





# Book consultation

Do you have a project where you need my help?  
If you are ready to seriously uplevel your  
product's UX, let's have a chat!

Book a call with me, and let's get to work:

[Book a call >](#)

MATCHi

 click funnels

ERSTE 

  
O P E L

 CSABA HÁZI