

UX DESIGN FOR STARTUPS

Marcin Treder

UX DESIGN FOR STARTUPS

Marcin Treder

UX Design for Startups

By Marcin Treder

Published in 2013 by UXPin

On the web: www.uxpin.com

Please send errors to hello@uxpin.com

Copyright © 2013 by UXPin.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording or any information storage and retrieval system, without prior permission in writing from the publisher.

CONTENTS

<i>Foreword</i>	8
<i>About Marcin Treder</i>	11
THE AGE OF USER EXPERIENCE DESIGN	12
<i>What is user experience design?</i>	18
<i>Users as the centre of UX design</i>	21
<i>Lean canvas as a design tool</i>	26
<i>The road to success</i>	29
GET TO KNOW YOUR USERS	30
<i>Getting out of the building</i>	31
<i>Guerrilla Research</i>	37
<i>How to do Guerrilla User Testing.</i>	39
<i>Further research</i>	44
<i>Communicate!</i>	51
EFFICIENT DESIGN TECHNIQUES	52
<i>Design techniques are just tools</i>	57
<i>The power of analog</i>	60
<i>The true nature of wireframing</i>	64
<i>Misunderstandings around mockups</i>	69

<i>The real power of prototyping</i>	71
<i>Getting out of the silly deliverable business</i>	74
<i>Iterate, iterate, iterate</i>	75
GROWTH AND DESIGN HACKING	78
<i>Crossroads of art and science</i>	81
<i>To measure or not to measure?</i>	85
<i>Economic metrics</i>	88
<i>Behavioral metrics</i>	89
<i>Mirror mirror on the wall...</i>	91
<i>Do it over and over again!</i>	93
<i>Quality comes from conversations</i>	97
<i>Growth and Design Hacking Tools</i>	99
GET IT OPTIMISED	102
<i>The golden rule</i>	103
<i>Technically and actually working stuff</i>	107
<i>Seeing design through metrics</i>	109
<i>Enough is enough</i>	114
TOOLS, TOOLS, TOOLS	118

FOREWORD

Not everyone has the inclination to go spend time learning more about potential customers. Some people believe so fervently in their idea; the thought of spending time on anything else than building it is inconceivable. So these people focus 200% of their energy breathing life into their idea, staying up late, working when everyone else is taking a break. Like Jeff Veen, founder of Typekit--now part of Adobe, said to me the other day, "It's hard to persuade someone to go spend time understanding users. I completely believe in research up front; I did it for Analytics. But I didn't do it for Typekit, because it was an idea I totally needed myself." Then he said, "But you know, research would have made it easier to explain the concept to people who didn't understand it." (Those people being the folks with the money who were hopefully going to fund the effort.) No matter what, there is always an aspect of development that can be made easier by understanding the people you are building for.

I always ask entrepreneurs, "Who is this for?" Before

I learn anything about their ideas, I want to have specific behavioral and marketing segments (personas) in mind. I want to know the real world in which the idea might be used. I used to always hear the answer, “Everybody!”. These days, entrepreneurs are smarter. They have a better idea whom they are creating something for, but it is still a sketchy idea. Spending a day or two putting meat on that user is powerful. It guarantees that you have no illusions about the things your idea will solve and the things it will not affect. And that word, “illusions,” is one to contemplate. Ask yourself if you’ve completely wiped away the fuzziness around the edges of your idea. Those fuzzy edges are the places that the monsters live; that’s where the problems come from that you hadn’t anticipated, and that can kill your effort before it is successful.

So, put a little time into making sure you have no illusions. Protect all that energy that you are investing in your idea by defining and directing it to the right place. Know your customers.

Indi Young

ABOUT MARCIN TREDER



Marcin Treder is a design enthusiast that literally lives for creating the best user experience possible. After years working as a UX Designer and UX Manager he focused on his own start-up UXPin that provides tools for UX Designers all over the world. UXPin tools are used by designers in companies like Google, Apple, Microsoft, IBM, Salesforce. UXPin was recently voted the best start-up in Central and Eastern Europe. Marcin enjoys writing (e.g. for UXMag, SmashingMagazine, DesignModo, SpeckyBoy...), blogging (Blog UXPin, UXAid, Startup Pirate) and tweeting (@uxpin, @marcintreder).

THE AGE
OF USER
EXPERIENCE
DESIGN

Like many of my contemporary UX Design peers, I started my career as a so-called usability specialist. Fascinated by ergonomics and cognitive science, I was working to make sure users were able to actually use interfaces. Armed with user research, heuristics and a little bit of prototyping, I was trying to find my place in the ‘developer-oriented’ world. This wasn’t easy.

For dev teams, an interface was considered to be an addition to great technology, and usability was even less important than that – a kind of nice-to-have option.

It was a time when binary logic ruled. Actually having a product that worked was important in contrast to not having a product at all. Delivering anything functional was seen as a success. Whether users could easily use it was often outside the picture.

Business people didn’t get it either. The term ‘usability’ was on everyone’s lips thanks to the work of Jakob Nielsen and Steve Krug (their popularity was skyrocketing!), but executives believed it was more

important to have a product with tons of advanced features, rather than something highly usable but technically limited.

No wonder my ‘usability specialist’ position was a struggle. But the real suffering was felt among users – this is how it was at the dawn of the age of technology.

You might have witnessed its rise. The time when engineers started to really rule the world. The Woz (Apple), Bill Gates (Microsoft) and Bill Joy (Sun Microsystems), were among the first stars of that age.



Photo credit: "Macintosh" by hsigmond. Creative Commons 2.0.

Internet startups that survived the dotcom bubble of 2000 were run by tech bright minds. Think of Google's Larry Page and Sergey Brin, eBay's Pierre Morad Omidyar, Max Levchin and Luke Nosek of PayPal, David Filo from Yahoo – these guys know how to code. And in even more contemporary times developers struck again: Jack Dorsey (Twitter) and Mark Zuckerberg (Facebook) shaped the social media with their tech expertise.

But then, suddenly, the age of technology ended. Fierce competition among similar (at least when it comes to technology) products forced executives to look for more vivid differentiation. Technology became easier and cheaper than ever. The world started to look for a new idol. Luckily for all of us this can be found in user experience design.

To make an app that can be launched has never been so easy. To succeed in a highly competitive market full of consumers with cognitive overload and an extremely short attention span ... that's another story. I shifted from usability to the much larger concept of UX

design a couple of years before the revolution, inspired by the work of Don Norman (father of the term ‘user experience design’, psychologist and former VP of Apple). I understood that great products create a great end-to-end experience: they shouldn’t be just usable, but seductive, pleasurable and inspiring.

Working as a UX designer, UX manager and finally creating UXPin – a set of tools for UX designers – I soaked up the design industry. Even so, the revolution came to me as a surprise.

When, together with my team-mate, we visited Silicon Valley to discuss UXPin’s strategy with our clients, investors and great UX designers, I was surprised to hear, “This is the decade of user experience design” from one prominent business angel.

“Design and marketing aren’t just as important as engineering: they are way more important.” says Dave McClure, founder of 500 Startups – one of the most important startup incubators in the world, and he’s got a point. The world has changed and products now

succeed if they provide stunning UX.

YouTube, Airbnb, Flipboard, Square, Pinterest, Etsy, Path, AboutMe, Slideshare – all these well designed, successful products were co-founded by designers.

Just think how Samsung and Apple fiercely fight over design patents. They want to conquer customers' emotions with unique designs. Remind yourself of Microsoft, who surprised the design world with a coherent, beautiful system across devices – Windows 8. Google, the former engineers' kingdom, redesigned all its significant products and employs UX designers all over the world. And of course Apple, the most valuable company in the world, built its success on well-crafted designs. These are all signs of a change of paradigm.

An incident that emphasized the growing importance of UX design was O2 UK's rejection of the sale of the Blackberry Playbook, because of “issues with end-to-end customer experience”. Take care of user experience design, or you'll kill your product before any user touches it.



Blackberry Playbook. Photo credit: ["Blackberry Playbook Screenshot 1"](#) by The GameWay. [Creative Commons 2.0](#).

WHAT IS USER EXPERIENCE DESIGN?

User experience design is not a niche anymore. It's easier to find an internet company without the SEO guy than without a UX designer on board. According to LinkedIn there are more than 800,000 people somehow connected to UX design and almost 2,000 open job positions as of September, 2012. There are conferences for thousands of people, great books, magazines,

webinars, courses... but still, I doubt if the understanding of UX design is very common nor well-spread.

This is what usually happens to words that become hype. Everyone talks about a term, believing it's self-explanatory, and in no time it loses its meaning.

I assume you're an entrepreneur. Most probably you're super busy making your dreams come true. You want to get the job done. You want results. Let's focus then on clearing the air around the definition of user experience design. It's really important that you understand the nature of UX Design, which unfortunately gets easily confused with visual design, usability, wireframing and a bunch of other stuff.

User experience design (abbreviation UX, UXD)
– A discipline focused on designing the end-to-end experience of a certain product. To design an experience means to plan and act upon a certain set of actions, which should result in a planned change in the behaviour of a target group (when interacting with a product).

A UX designer's work should always be derived from people's problems and aim at finding a pleasurable, seductive, inspiring solution. The results of that work should always be measurable through metrics describing user behaviour. UX designers use knowledge and methods that originate from psychology, anthropology, sociology, computer science, graphic design, industrial design and cognitive science.

When you're designing an experience, you are in fact planning a change in the behaviour of your target group. You've found out their problem and you're trying to destroy the burden using design methods.

User experience lies at the crossroads of art and science and requires both extremely acute analytical thinking and creativity.

Let's consider an example: we're about to create a door handle. As a usability specialist your task will be to make sure that the person faced with the need to open doors will be able to perform the task using your newly designed door handle. You conduct a series of user

tests and iterate on the best solution. As a UX designer you're not only interested in a usable door handle. You want to create something that will encourage people to open doors and will provide a unique experience. You want people to open doors twice as enthusiastically as before. Again, you'll iterate on the best solution, but the approach will be broader and the measured result should focus on the user's behaviour.

User experience design at its heart is an optimisation: an iteratively improved solution to a general problem. UX is the air successful startups breathe.

USERS AS THE CENTRE OF UX DESIGN

If the heart of UX design is the concept of constant iterative optimisation, then the problem is the blood that the heart is pumping. The problem of your future users. Spot it, define it, feel the pain it causes and eliminate it. That's the highway to great user experience.

To stay on the right track you'll need a lot of empathy and analytical skills, because the tricky thing with problems is that we sometimes have difficulty defining them – even if they trouble us.

When traveling by train on a hot day, I'm never sure if I'm irritated by the heat, the crowd, or – as I usually grumble – by the fact that I need in fact to travel to work via train. Give me an office closer to home and I'll find another reason to complain on hot days. Eliminate the heat in the train and I might even enjoy the ride to work.

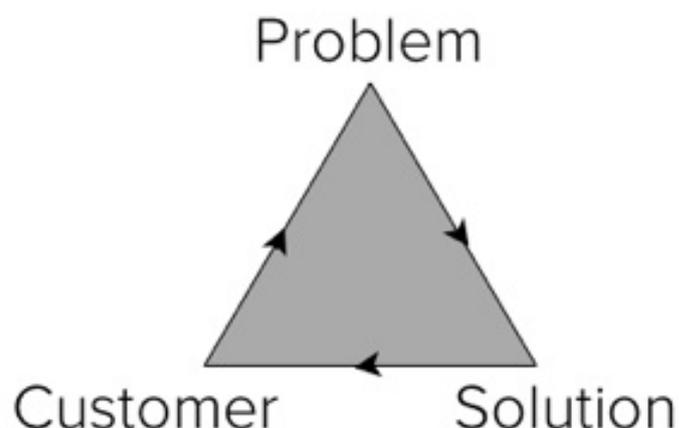
The key to success is to actually get to know your clients. Arm yourself with empathy and talk to them. Get out of the building and face the problems that might be the foundation of your business.

When we started working on the UXPin app we crossed the ocean from Poland to California to talk to our customers and check what troubles them the most. Several in-depth interviews later, we had completely new ideas about what product we should create.

There's nothing more refreshing and more crucial to your business than having an actual conversation with your customers. UX design is human-centric: it doesn't exist without interaction between people.

C-P-S hypothesis

If you've reached your target group and interviewed its members looking for a serious problem, it's about time to define the basis of your product in a triangle:



Before the launch of a product and thorough measurement of user behaviour, everything is a hypothesis.

The C-P-S hypothesis is a basic description of any product. It reaches the core of any successful endeavour in a neat, minimalist way. Define who exactly your customer is, what problem they have and what the solution is that you offer. Do it in one sentence. For example:

“For people who are trying to design products with great user experience and are having problems with documenting their ideas quickly and clearly and sharing them with their teams, UXPin provides an online, fully collaborative app that helps them to go through the UX Design process together with their teammates.”

As you can see, I described UXPin’s target group as anyone who is trying to design products with great UX. I also defined the problem that was observed during research on a target group and I briefly described the solution.

Simple as that – my product is specified. The C-P-S hypothesis forms the backbone of the whole product. It

not only helps me and my team focus on what's really important, but also gets us ready for an optional pivot.

Each part of the C-P-S hypothesis is questionable on its own. I might wrongly describe the target group. I could misunderstand a problem. Or I could create a product that doesn't address the problem. Any of these mistakes gets your business into trouble.

No worries though! If your product doesn't fly you can always come back to the initial C-P-S and re-form it to test new assumptions.

A great UX experience can only be achieved iteratively, and the C-P-S hypothesis is a powerful tool that helps you draw a meaningful conclusion from each phase.

LEAN CANVAS AS A DESIGN TOOL

Alex Osterwalder, in his great book *Business Model Generation* shows and explains an amazing method

of describing any business idea: the Business Model Canvas (BMC). This eight-field table can do what an extended business plan struggles to achieve: it can explain your business. It revolves around the Unique Value Proposition (a single, clear sentence describing the way you're different from your competitors and why you're worth buying) and the canvas depicts your idea, key partners and resources, and your model of revenue. This is the single most productive page you'll ever come across in your business endeavours.

No wonder clarity of information and ease of use made Osterwalder a killing. The Business Model Canvas is extremely popular and it's not a rarity nowadays to be asked by a VC to prepare a BMC.

But how is it all connected to user experience design?

As you already know, designing UX means taking care of the whole product. The end-to-end experience is what you're aiming at. A BMC enables you to grasp the whole product in one place and understand at a glance both the business and users' side. This is

absolutely crucial.

Always remember that in any commercial project UX design cannot be separated from the business model of a product. Designing UX without any knowledge about the business side of the product is a futile and stupid thing to do. A product that doesn't bring home the bacon will soon cease to exist and the whole effort will be a huge waste of time.

If you look closely, you'll find a Business Model Canvas template in the UXPin app. We made it part of our process and our users often use a BMC as a basic description of just about any project.

A variation on a BMC, even more focused on a product and its users, is a Lean Canvas, created by Ash Maurya:

It's divided into two parts: Product and Market, and it clearly shows exactly what your product is and who will benefit from its use. A Lean Canvas should be developed iteratively (just like your product) so



Lean Canvas. [Photo credit: Ash Maurya. Creative Commons via Wikimedia Commons](#)

get used to updating it whenever you have a new hypothesis.

Both a Business Model Canvas and a Lean Canvas should be used as a collaborative tool. Encourage your team to discuss canvases and question all assumptions. Great UX design stands upon efficient collaboration.

THE ROAD TO SUCCESS

In the age of user experience design your startup needs to focus on users' problems rather than on technology only. Iteratively test the C-P-S hypothesis and fill in a Business Model and Lean Canvas together with your team. That's the start of the road to success.

Competition is fierce and only those who are able to provide a stunning end-to-end experience will survive.

GET TO KNOW YOUR USERS

User experience design is deeply human centric - it dies without a decent amount of interaction between human beings.

When was the last time you talked to your users? Hope it wasn't too long ago! Even if it was though, don't worry - it's never too late to get to know your users. Today we'll learn how to befriend them and use their unique perspective on a product to your benefit.

All right! Let's befriend our users!

GETTING OUT OF THE BUILDING

How do you imagine people designing a product? If you're anything like me, you're picturing people in front of desks full of papers, watching a huge screen with tired eyes while drawing some part of an interface.

The solitude of the designer.

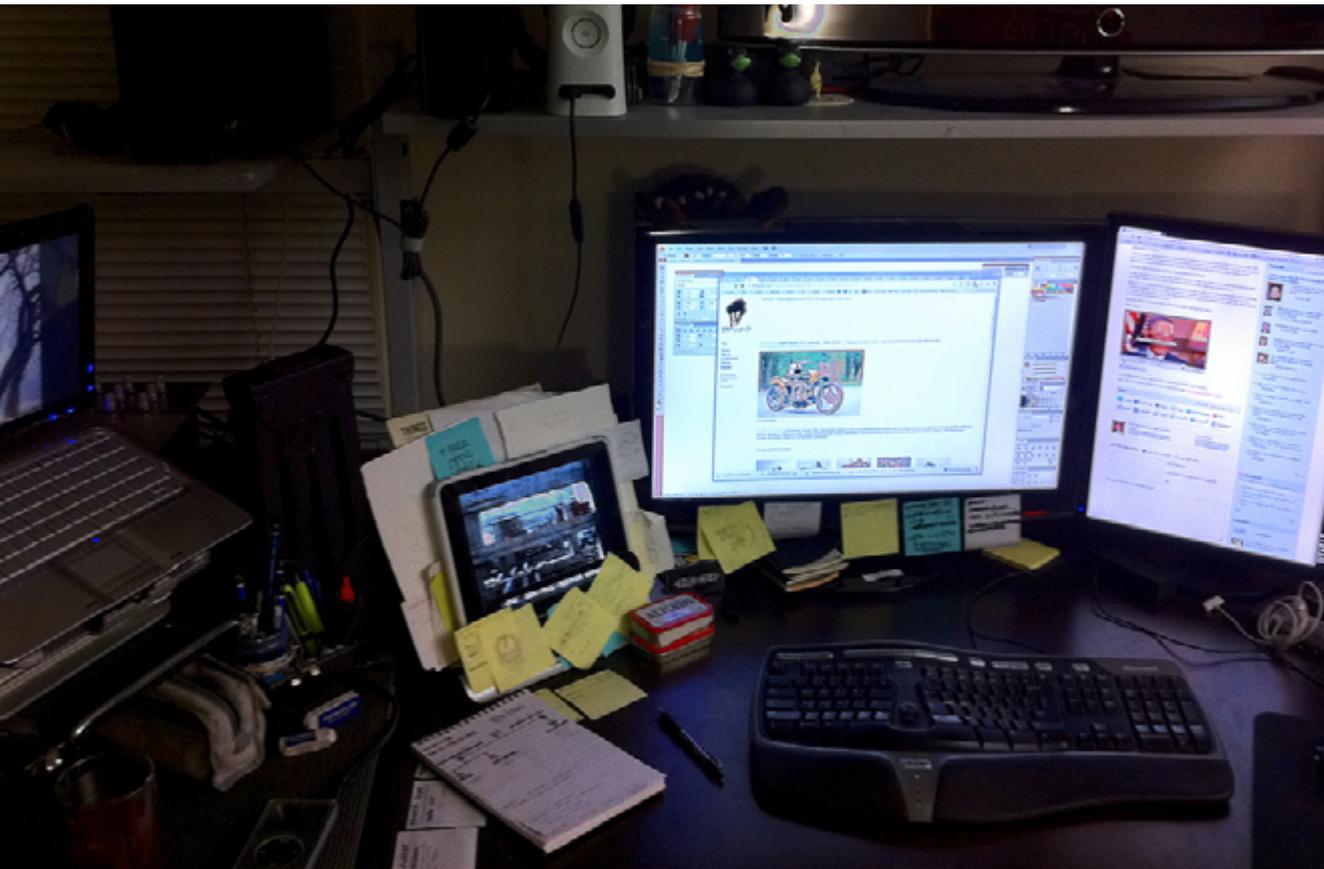


Photo credit: "[128/265 C](#)" by Joseph Nicolaa. [Creative Commons 2.0](#).

Damn, it's depressing! Luckily, the image above is true only for one phase of the design process, the least important one - drawing. In the drawing phase you need to sum up your previous design efforts in a set of sketches - either on paper or as a digital wireframe/mockup. You might even go further and simulate interactions in a prototype made in a dedicated software, or in html; it won't change the fact though,

that drawing is not where great User Experience Design happens. Great User Experience Design happens when you talk to your customers and solve their problems.

Surprised? Let's remind ourselves of Steve Jobs' words: "Design is not just what it looks like and feels like. Design is how it works".

To succeed on the "how it works" side and create something really valuable, you need to focus on the whole experience, not just the aesthetic value. In design, thinking precedes drawing, so think through any design decision and always refer it to your C-P-S hypothesis (Customer, Problem, Solution):

- **Does your design address your target group?**
- **Does it resolve their real problem?**
- **Is your solution an accurate response to your target group's needs?**

Remembering about the C-P-S triangle is the first smart thing you can do while designing. The second thing is to actually discuss your strategy with potential customers.

Let me share the single most important thing that I've learnt about customer development: you won't meet your customers in the reflection on your screen. You have to get out of the building and really talk to people. It doesn't matter if you reach them by Skype (you might get out metaphorically) or in-person - the important thing is to transcend the boundaries of your ego.

Don't try to hack it, or your design will be lost.

If I'm asked to give one piece of design advice, it's always: Root your design in the actual knowledge about your customers and execute mercilessly based on this. Don't daydream, don't say "my mom wouldn't get it", or "well I would use it!" - reach out to your customers and ask them what their thoughts are. Back up your design assumptions with knowledge to

minimize the risk of failure.

When we started to negotiate our founding deal with our investors at UXPin, their first piece of advice was: “Pack your stuff and go to San Francisco to talk to your customers!” (We’re lucky to have wise investors, who were successful as entrepreneurs). We didn’t need much persuasion – a 14-hour flight and we were where the majority of our clients are – the USA.

And yes, it wasn’t easy to break our comfortable habits and start to have three meetings a day for around 2 weeks, instead of dilly-dallying in front of a computer for the whole day... but we did it. We put our introversive natures aside and fought for the sake of the people who trust us - our users.

We needed to know what they think about us, how they work, what they really need... there’s no other way to learn that than getting out of a building, approaching the users and asking the right questions.

When we got back home, we continued talking to

customers via Skype and till today this is a part of our product development process, a routine which we do to make sure we're on the right track.

This whole trip and our conversion to “customer centric madcaps” was the smartest thing that we could ever do for UXPin. Conversations with customers led us to change to a strategy which accelerated our growth to a rate of 50% (minimum) in sales each month.

This is amazing. You can actually be successful by creating a product that's truly valuable for your customer! Captain Obvious strikes back? Kind of. How many start-ups do you know, though who seem to try hard to avoid talking to customers? I'm sure I know plenty of them.

You may wonder what the cost is of this kind of user research? If you've heard anything about professional user and market research, it's probably their price counted in thousands of dollars. Well, that's true for professional lab research. Methods that we use cost us \$0.

GUERRILLA RESEARCH

Guerrilla Research methods were firstly used in the market research field following so-called Guerrilla Marketing introduced by Jay Conrad Levinson in 1984 in the book “Guerrilla Marketing”. Guerrilla stands for atypical, cheap and somehow aggressive methods of achieving goals.

An example of Guerrilla Marketing would be a graffiti or a flash mob used for promotional reasons. The crazier and more buzz-generating - the better.

In the User Experience Design world Guerrilla Methods somehow became known after the famous book “Don’t make me think” by Steve Krug, who encouraged designers to do research even if the only subject they tested their design on was their mom. It’s better to check your product with one person than not check it at all - argued Krug. Today we can call such a research method - Guerrilla User Testing.

There was a time in my career when I strongly

disagreed with Krug. I believed that only methodologically valid research may lead to meaningful results (no wonder, I'm kind of a statistics nerd). When I started my own company I quickly re-learned Krug's old wisdom though. Whether the results of a study are meaningful or not depends on your definition of meaningful. And you should always do research that can be most economically valid - create the biggest value for the least amount of money.

Of course, the methodology of Guerrilla User Testing isn't right. You can't extrapolate the results achieved by one, two, or even ten people on the whole targeted population, but it doesn't make it meaningless. It just makes it meaningful in a different way. Judge its meaning by the results it brings to your company and you'll see the benefits in a brighter light than the flaws.

Quick & dirty research is an amazing way to explore your product. You'll find out more possible problems that you ever bad-dreamed of. An additional perspective on your project is a lever that may be crucial for the whole endeavor. Each time we make a

quick usability study (usually on around 7 subjects) we learn so much about our own mistakes, which is just overwhelming. And bear in mind that two of UXPin founders are experienced UX Designers.

Each tested person increases the probability of your success, so I strongly encourage you to make it your routine. After all, it's free and all you're risking is a couple of hours.

HOW TO DO GUERRILLA USER TESTING.

Reach out to your users *(or any approximation of the target group)*

Think where you can find your users. Local Starbucks? Walmart? Perhaps a park? It all depends on your target group. In our case, it's easy - since UXPin provides tools for UX Designers, we just invite local UX Designers to visit us in our office (it's a small

community and we all know each other somehow).

Wherever they are - grab your laptop and go and talk to them. Show them what you have and check if it's usable. If finding your users is a problem (e.g. there's no cafe nor shop in your neighborhood and extreme weather plus polar bears make it tough to go outside) - try with your neighbors. In the worst scenario - use your family. Just talk to somebody! Go outside your ego and check the value of your work.

You must remember though that your closeness to subjects will affect the feedback. Your family probably don't want to hurt you.

Prepare a testing script

This is the single most important thing while preparing a usability test. The script guides your testers and shapes the whole interaction. Plan it thoroughly thinking about specific parts of the product that you want to test.

I always prepare a short story that provides a context for research. It lets subjects use their empathy and kind of role play (like in an old-fashioned RPG). Thanks to the context, people soak up the research and the whole situation is closer to real-life.

For example, in the last research we did at UXPin, we provided a brief from a client and a story out of a UX freelancer's life. It helped our subjects stay on the right track and feel the pain of the problem they were trying to solve.

The results of the research were stunning.

Get your gear ready

The one thing that's surprisingly easy to break during guerrilla usability testing is your gear. Here's my short checklist, originating from my personal list of shameful mistakes:

- Always check your screen recorder before the session (recently, I didn't notice that Silverback -

which I strongly recommend, stopped working after a Mac OS X update).

- Always use ethernet or cable network, rather than WiFi (which likes to disconnect you during a study - been there, done that).
- Check that the mouse works properly (a customer sitting on my mouse almost ruined the research).
- Check the power source (oh yes, I've run out of power).
- Avoid any non-standard keyboard shortcuts (again custom settings on my Mac made it really awkward).
- Turn off all the unnecessary software (taking extra care with Skype and Messages).

Avoid these lame mistakes and you'll be good.

Prepare the participants

First of all, accept the fact that people may be scared of your research. Give them confidence and treat them like experts in the field. Say clearly that the results of the tasks don't matter; what matters is their opinion. Encourage them to speak aloud while performing tasks and justify this by the fact that you need to know what they're thinking while interacting with the product.

Don't let them blame themselves if something goes wrong. Let this experience be pleasurable.

Record the session

Record the screen (in the case of mobile app testing - a whole mobile), the face and voice of your research participants. It'll let you easily assess what was going on during the interaction.

Analyse the results immediately

Always take some notes during the study. Write down only catchwords, which will help you remind yourself of a specific situation and won't take too much time

and attention during the test. As soon as possible after the study, change your notes into specific tasks.

In my experience - if you don't do this in the first two days after each session, you'll never do it.

FURTHER RESEARCH

So that's guerrilla user testing. When to use it? As a cheap and quick method of research, guerrilla user testing is suitable for any phase of the product's development in which you have at least a prototype that you can check.

And what if you don't have a prototype? Or you want to test something other than an interface, like a general strategy of future development, or you want to check how people are dealing with some specific problems?

Use customer interviews

A customer interview is a simple, straight-forward, research method. You basically talk to people according to a script that you've prepared and you try to get as much from the conversation as possible. Ask in-depth questions, stretch your empathy to its limits, pay attention to every tiny detail, ask your participants to sketch solutions to their problems and sketch your own ideas. Just do whatever is needed to really dig in to the true nature of the problem.

The interaction should be natural. It's like talking with a friend about their work problems. You need to encourage them to share at the start and when it blows, just listen and carefully dig deeper to really get to the core of the problem.

Though a customer interview may seem to be a little long-winded, it's super efficient.

A conversation with customers is a truly refreshing experience and if you play it right, every minute will be super valuable.

Skype as a research tool

What if your customers live far away from you and you can't visit them every couple of weeks? You might do your start-up thing from a remote forgotten land, like, say, Poland (oh yes, that's where we're based). It shouldn't stop you from doing a decent amount of guerrilla style user research. Make no compromises when it comes to your product's user experience.

Just use Skype. People are more willing to talk via Skype than have a coffee with a stranger. The awkwardness is kept to a bare minimum and the interaction feels somehow safer. You can stop it at any time, easily postpone, follow-up, etc.

Secondly, there's no distance barrier. It makes the research even cheaper and eliminates any possible excuse for not having a conversation with a client.

Finally, you can easily record the conversation with your user and even see their screen using the "screen share" option.

Could it be easier? Don't think so – I've done Skype research dozens of times and I have only positive experiences. Try it!

Systematic surveys

When you have all your guerrilla research methods in place, it's time for creating a systematic way of gathering large amounts of feedback. This is an important addition, which is much closer to proper research methods than guerrilla style usability testing and customer interviews.

Why? First of all, systematic surveys are, well, systematic. You can see how any changes made in a product affect the survey answers of your users. Every gathered reply is placed in a specific place in time. It's ongoing, unstoppable research.

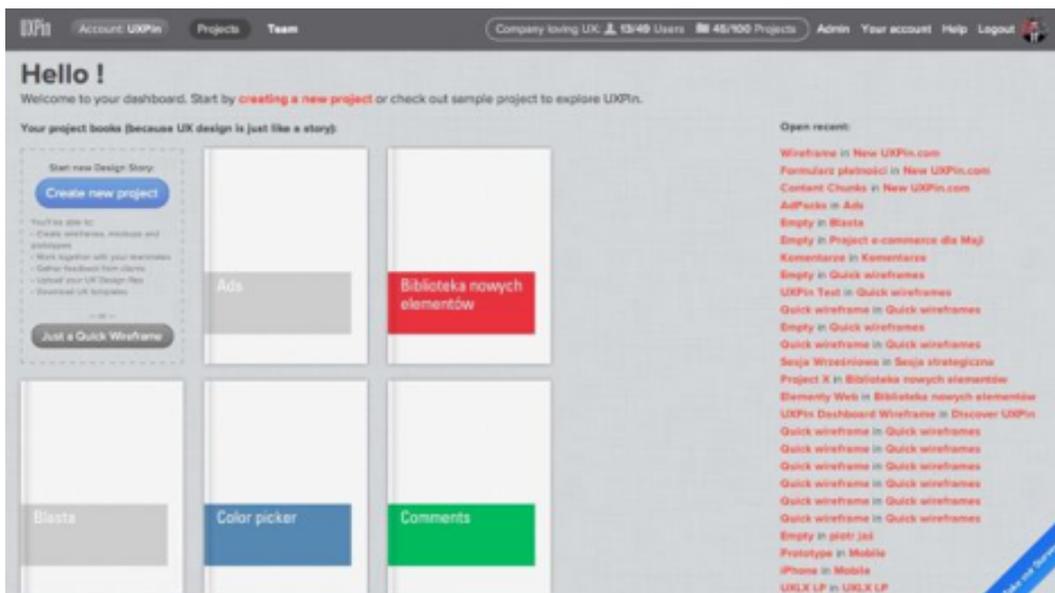
What's more, you have a chance to reach out to a statistical representation of your target group. You can easily gather dozens of results and sum them up using statistical analysis.

All right, but how can you create an efficient survey? It doesn't seem to be easy. What questions should you ask? How can you gather answers? How should you distribute the survey? I'm sure you have tons of questions.

Don't worry about them. Seriously. Just use Survey.io supplied by KissMetrics and create a full-blown Customer Development Survey in two minutes.

We did it (leaving behind the surveys made in Google Docs) and honestly, this simple tool is one of my personal favorites.

The results of using Survey.io are stunning. Applied in an unobtrusive way (take a look at the screenshot's



Link to a survey placed unobtrusively in the bottom right corner

bottom right corner), it attracts lots of users. People want to share their thoughts with you. Just let them do it without any extra effort.

Users can click (whenever they are ready) on the blue ribbon (bottom right, on the screenshot above) and fill in the form that's created automatically out of the template crafted by Sean Ellis, one of the start-up marketing gurus.

Could it be easier? Doubt it. Get ready to use a survey created by an experienced marketer.

Start by
UX design

UXPin Survey

Thanks for taking the time to complete our survey. There are **8 questions** below, which have been taking most people **5 minutes** to complete.

1

How did you discover UXPin?

Blog

Friend or colleague

Search engine (e.g. Google, Yahoo!)

Facebook

Twitter

Other (please specify)

2

How would you feel if you could no longer use UXPin?

Survey generated by Survey.io

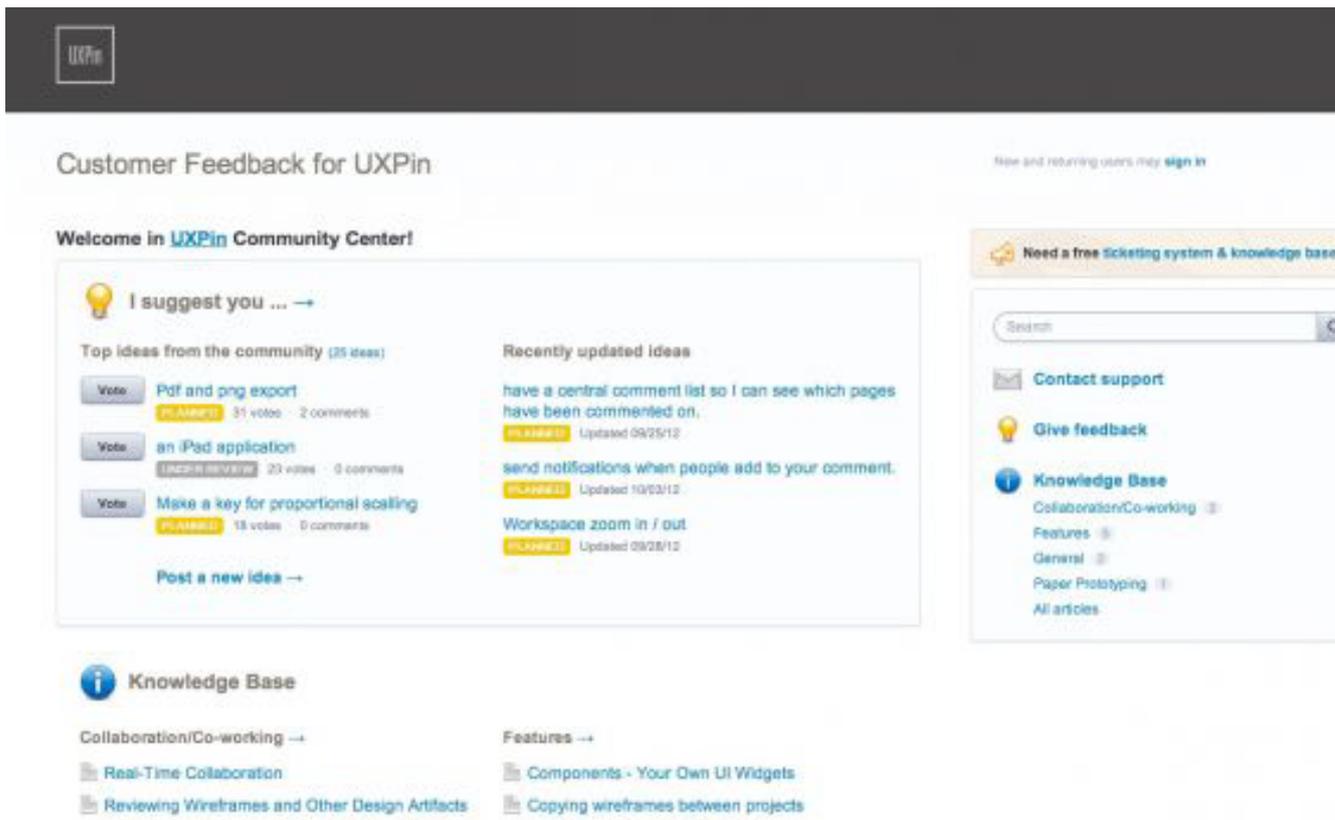
Set up a feedback forum

What's that? It's a special forum optimised to let people easily share their opinion about your product and suggest changes.

Sounds simple right? And it is. It's a conversational space where you not only gather super valuable information, but also build your community. By creating a feedback forum you show your customers that you really care about their opinion.

There's just one rule - talk with people. You should be the most active part of the community. Be with your users and listen to them.

There are many options on the market for setting up a forum, but the one I strongly recommend is User Voice. It's an easy-to-use and highly efficient tool with a ticketing system, knowledge base and suggestions forum. Try it!



Setting up a forum on Uservice is easy. In the photo: UXPin Uservice Forum

COMMUNICATE!

There's one way to build a great user experience: active communication with customers.

Without the voice of your customers, your design will be an empty shell. Don't let it happen.

EFFICIENT
DESIGN
TECHNIQUES

Ah, design techniques ... the exact epicentre of longstanding battles in the user experience design world. Should we wireframe? How do we create the real-deal prototype? How many deliverables are too much? Should we create any deliverables at all? Should designers know how to code? The fight goes on endlessly. It's easy to get lost in all the arguments, especially if you're an extremely busy entrepreneur, trying to fire up the design and amaze users.



Endless Battle. Photo credit: ["Toy Soldiers \(silhouette\)"](#) by Kyle May. [Creative Commons 2.0](#).

In my experience, only two things are certain in all these discussions: they are boring and they are futile. You don't need to engage in them to design a stunning user experience. Most probably you even shouldn't.

The best designers (judging by the results!) that I know are not bothered by the above-mentioned battles. They are agnostics on the subject of which design technique is the holy grail of design. The best designers I know are strategists. They thoughtfully choose the best design techniques for each of their projects. Because they're very serious about creating an amazing experience for their users.

I assure you they know each of the techniques well, just as you know your old friends. You know that you can take Sammy fishing, but he's not a party animal (Jeff is) and they also know that sometimes we're fine with a sketch, sometimes a good old wireframe would do and sometimes we need a reusable prototype written in HTML.

Simple, right? Nothing to get over-excited about and

fight over for hours.

To really nail your design, you need a strategy – and to form a strategy you need to get to know and befriend classic design techniques. As always, you should base your decisions on knowledge, rather than on assumptions, and I’m here to help you. In this chapter we’ll thoroughly discuss what to use and when to use it.



The best designers are strategists. Photo Credit: [“Conflict \(Chess II\)” by Christian V. Creative Commons 2.0 Generic.](#)

However, before we talk about the advantages of paper and the true nature of wireframes, mockups and prototypes, we need to build a strategy framework that will help you in every design project you'll be working on from now on.

Whenever you're faced with the problem of which design technique use to guide your product from the idea stage to the execution stage and beyond, try to honestly answer these questions:

1. What speed level is appropriate for your project? Are you in a hurry, or do you have some extra time that can be used for design? Remember, it's always about the quality-time transaction.
2. What accuracy level is appropriate for your project? Are you planning to test prototypes with users? Are you designing a complex product in the health industry? Greater accuracy equals more time spent on the documenting phase of the project.
3. How did your team respond to different design

techniques that you used in the past? Did they understand it? Were they able to proceed without tons of questions? Were they happy with the way you worked last time?

Speed, accuracy and past experiences are three determinants that are absolutely crucial for the right choice of design technique. Try to learn this by heart and you'll never get lost nor distracted by the endless quarrels that trouble the design industry.

DESIGN TECHNIQUES ARE JUST TOOLS

You must remember, though, that tools are just tools and that user experience design shouldn't be confused with merely drawing a wireframe, creating a prototype, or forming a diagram.

To be honest, I have a feeling (based on both my UXPin and UX Manager experience) that many

people consider UX designers as sort of wireframers. By wireframers I mean unfortunate people who just do wireframes and perhaps clickable prototypes from time to time. People who are focused on using design techniques, not on designing.



Tools are just tools. Photo credit: ["Chocolate Tools" by Janne Moren. Creative Commons 2.0.](#)

The true nature of user experience design exceeds the techniques that we use to express our design ideas. A UX designer's work should always be derived from people's problems and aim at finding a pleasurable, seductive and inspiring solution. The results of this work should always be measurable through metrics describing user behaviour. UX designers use knowledge and methods that originate from psychology, anthropology, sociology, computer science, graphic design, industrial design and cognitive science.

User experience design is a complex set of activities and, in fact, a way of thinking. If you start to breath the air of UX design and really devote yourself to it, your startup will surprise you and amaze your customers. In the age of user experience design, your users expect your product to be well designed.

And it's really important for the sake of your company that you remember: UX design does not equal wireframing. Anyone can wireframe – it's a rather simple activity; many people can write simple HTML code, but not everyone can design experiences.

Before you start thinking about the right design technique to use in your project, get back to your C-P-S hypothesis. Great user experience design is all about the problem of a specific target group that you're just dying to solve. Without acknowledging and in fact deeply understanding the problem of your users-to-be, any design technique will be just an empty shell.

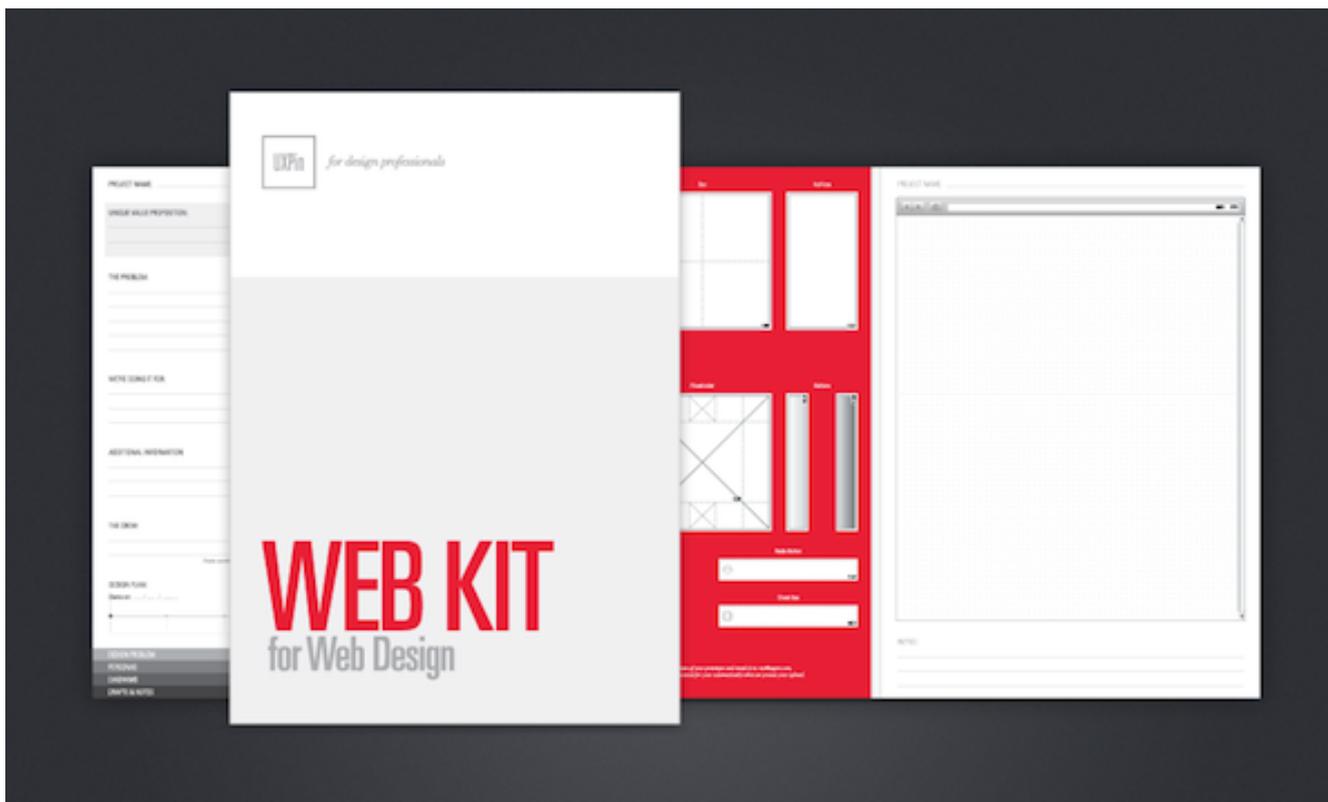
If you haven't considered your C-P-S hypothesis yet, do it now.

THE POWER OF ANALOG

Back in 2010 together, two of my friends and I started to think about a solution to the internal problem of the company we used to work in the R&D department of: how to engage non-designers in design activities. They couldn't be bothered with professional software (we were using Axure at the time), because it was way too hard to use and too far removed from fun (fun is actually an important factor here). Just the sight of a

complex interface scared them to death and we were trying to get these ideas out of their minds and engage them in the UX design process.

We came up with the idea of using sticky notes as design elements and creating workshops in which anybody could express their ideas. It was a blast, and not long afterwards we commercialised the method as UXPin Paper Prototyping Kits, selling it to designers in 41 countries.



Our success was possible because people love to interact with physical products. It's much easier and less stressful to move a UI design element on paper than doing this in the software. Try it with your team and be amazed.

Generally speaking, the advantages of paper include:

1. No limits, because paper has an extremely simple user interface with no predefined styles, rules, or guidelines
2. Inherent collaborative qualities; it's easy to share and easy to pin on the wall.
3. It's easier to throw away what you only spent five minutes designing.
4. It teaches designers that their ideas are more important to the design process than the tools they use.

This is well understood in the UX community. Todd

Zaki Warfel, in his research for Prototyping, found that paper prototyping is the most widely used design technique in the world. Bill Buxton, in his great book *Sketching User Experiences*, brings up lots of great real-life stories of the use of paper in design.

The great question is, when should we use paper prototyping? I can come up with two particularly useful situations:

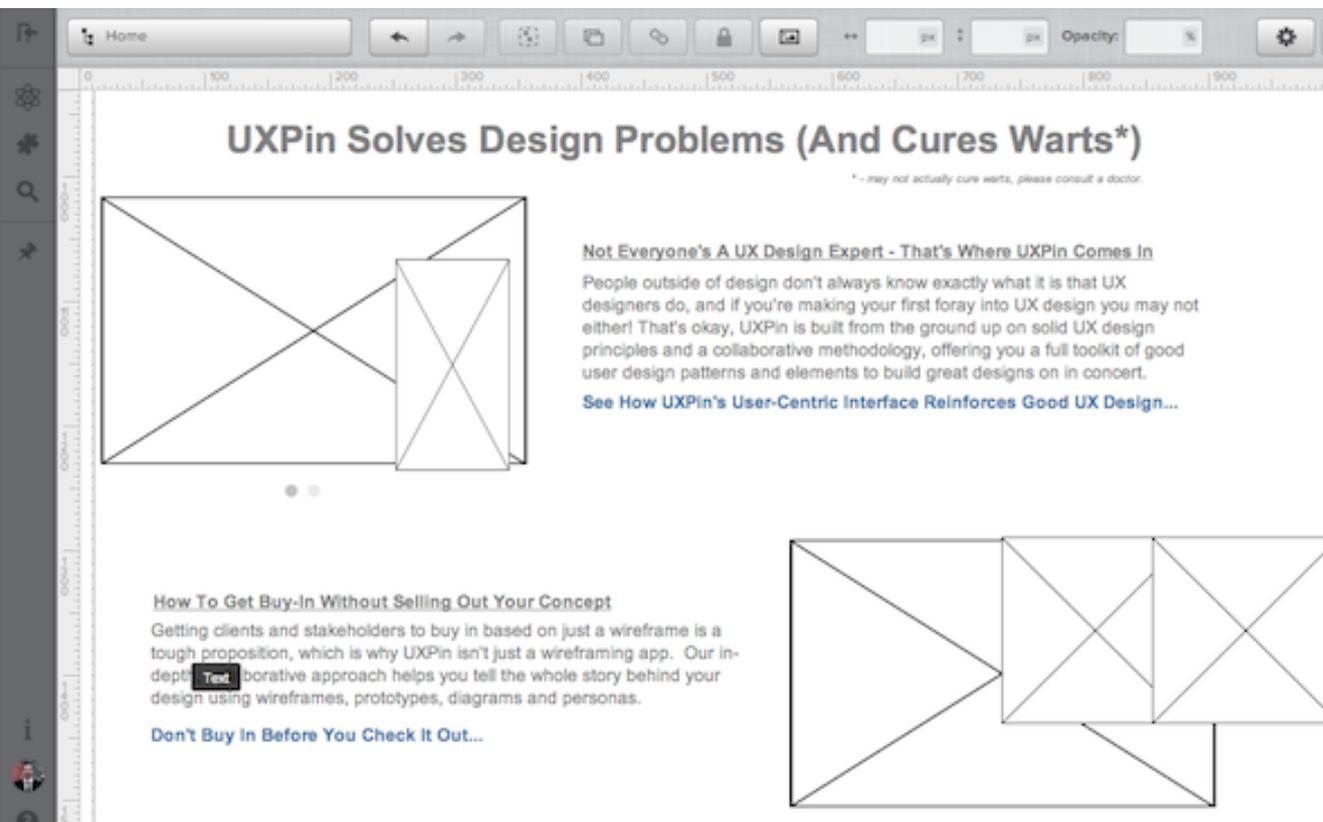
- Always, as a quick and dirty ideation tool that you use before using other, more complex, design techniques;
- In small, not complicated, projects, as a primary design technique, on one condition: you must have a good relationship with your team who can work on scrappy documentation and, in this particular project, you must prefer speed over accuracy.

You can either engage in sketching, or use sticky notes as we did. Try to be as quick as you can and don't over-design your paper prototypes. They don't need to be

works of art, unless you plan to test them with users (which is an interesting way of quickly testing your design ideas!).

THE TRUE NATURE OF WIREFRAMING

I suppose wireframing is the second most popular method of rapid documentation of design and one that you have probably seen many times throughout your career.



A simple wireframe in UXPin App

A set of gray boxes that hardly look like a design, but somehow get everyone excited during ‘buy in’ meetings. Sounds familiar? That’s what a wireframe looks like if you don’t consider its powerful communication skills.

A wireframe is a low fidelity representation of a design. It should always clearly show:

1. The main groups of content
2. The structure of information
3. A description and basic visualisation of interactions between users and the interface

Consider the wireframe as the backbone of your design, and remember that it should contain a representation of every important piece of the final product.

A wireframe is for web and mobile projects what a blueprint is for the architecture of buildings – a representation.

‘Representation’ is an absolutely crucial term here, which will help you find the right fidelity-speed balance. You can’t go into too many details but, on the other hand, you need to create a solid representation of the final design that doesn’t miss out any important piece of it. You’re setting a path for the whole project and for the people that are working with you (developers, visual designers, copy writers, project managers – they all need well-created wireframes). You might compare it to the creation of a map of a city. Every street is represented on a map, but, obviously, it’s vastly simplified. You can sense the architecture of a city if you look at a map, but you can’t perceive its beauty.

Wireframes should be created quickly and almost the whole time should be spent communicating with team members and ... thinking. The mere activity of wireframe-creation should be really quick.

The visualisation should be aesthetic, but this is vastly simplified. Black-grey-white are the typical colours you’ll use (you may add blue to specify links).

If something takes too much time to prepare (such as choosing icons or uploading images), you have to represent it in a simplified way (for instance, using placeholders – crossed rectangles for images, plus an appropriate description). This is why we call wireframes low-fidelity deliverables (lo-fi).

It doesn't need to be pretty, it has to be functional. Remember – a well-created wireframe communicates design in a crystal clear way and sets a path for the whole team.

When to use wireframes?

- In any huge, complex project that has an alarmingly close deadline.
- In any small project with a limited budget.

In both cases, wireframes are typically used as the documentation of the project. Since they are static and fix interaction with an interface at a specific point in time, they should be accompanied by the written

word (from short notes explaining interaction to, when needed, complex technical documentation).

However, they might be also used in a less formal way. Since they are quick and simple in form, they serve well as clear sketches for inner communication in the team. If developers ask how something should be done – the answer can be provided as a rapidly created wireframe.

To give you an example, UXPin is a startup with really rapid development cycles (releases every couple of days). We use wireframes to quickly visualise tasks (even small ones!). It eliminates misunderstandings and is really cheap.

Wireframes are seldom used as a testing material, although they may help to gather feedback in initial, guerrilla-style, research, in which you don't care about methodological purity, but rather try to get some quick insights.

Wireframes placed in the context of the whole design

story can be surprisingly effective and, though in recent years they've received some bad press, are still indispensable as an initial stage of complex projects.

MISUNDERSTANDINGS AROUND MOCKUPS

The term 'mockup' was used for years in regard to high fidelity, static, design representation. It's a kind of draft (or even a final version!) of visual design used to get a buy-in from stakeholders.

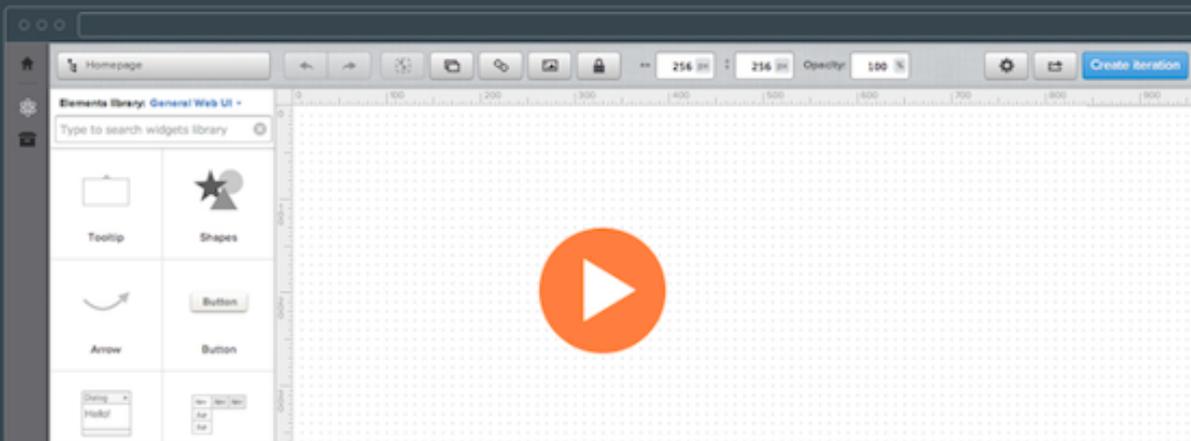
Suddenly the term started to lose its meaning and become close to wireframes. The reason is trivial. A couple of companies, founded by non-designers, which created wireframing software, confused mockups and wireframes and started to advertise themselves as 'mockup software'. It's a pity really, because right now a lot of people don't see the difference between these completely different ways of expressing design ideas.

UX DESIGN TOOLS BUILD BY UX DESIGNERS

UXPin's Collaborative Approach Cuts Cost & Confusion, Making Work Fun Again

Start Using It Now!
14 DAY FREE TRIAL

Get started in under 10 seconds,
with no credit card required!



1 Person, 100 or Fortune 100, UX Design Always Counts

A mockup is a visual representation

Remember, a well created mockup:

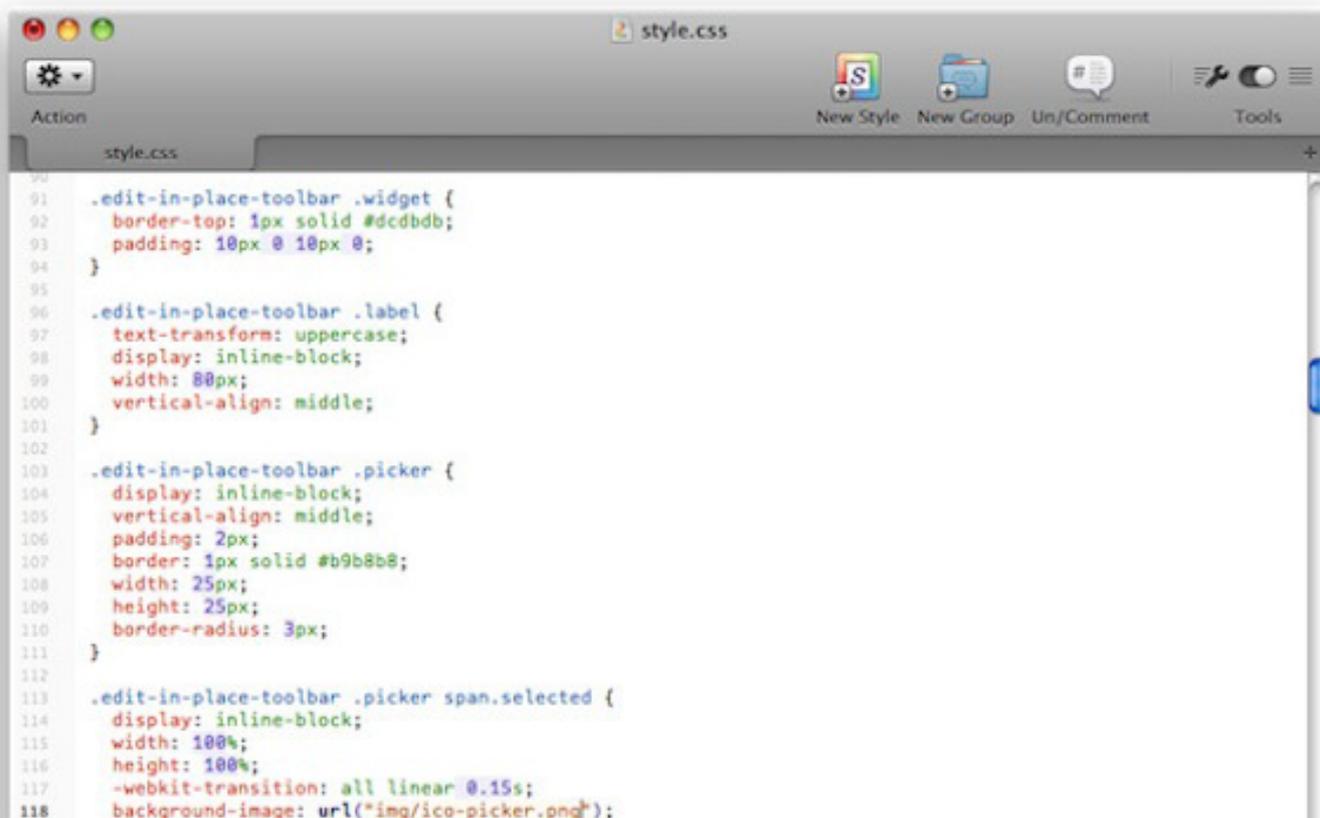
1. represents the structure of information, visualises the content and demonstrates the basic functionalities in a static way

2. encourages people to actually review the visual side of the project.

Mockups are particularly useful if you want to get an early buy-in from a stakeholder. Thanks to their visual nature, mockups don't have the resistance of low fidelity deliverables and are much quicker to create than prototypes. They are a good feedback-gatherer and, if placed in the context of the whole design story, can form a great chapter of your documentation.

THE REAL POWER OF PROTOTYPING

A prototype is often confused with both a wireframe and a mockup. In actual fact, though, it's totally different. A prototype is a middle-to-high fidelity representation of the final product, which simulates user interface interaction. It's interactive and dynamic.

A screenshot of a web browser's developer tools CSS editor. The window title is 'style.css'. The code is as follows:

```
91 .edit-in-place-toolbar .widget {
92   border-top: 1px solid #dcd bdb;
93   padding: 10px 0 10px 0;
94 }
95
96 .edit-in-place-toolbar .label {
97   text-transform: uppercase;
98   display: inline-block;
99   width: 80px;
100  vertical-align: middle;
101 }
102
103 .edit-in-place-toolbar .picker {
104   display: inline-block;
105   vertical-align: middle;
106   padding: 2px;
107   border: 1px solid #b9b8b8;
108   width: 25px;
109   height: 25px;
110   border-radius: 3px;
111 }
112
113 .edit-in-place-toolbar .picker span.selected {
114   display: inline-block;
115   width: 100%;
116   height: 100%;
117   -webkit-transition: all linear 0.15s;
118   background-image: url("img/ico-picker.png");
```

Prototypes are often written in HTML and JavaScript

It should enable the user to:

- experience content and interactions with the interface
- test the main interactions in a way similar to the final product.

If your prototype is static, it's either a mockup or a wireframe. A prototype is not the final product though. It's a simulation of the final interaction between the user and the interface. It might not look exactly like the final product, but should be vastly similar (it's definitely not a greyish, sketchy thing). Interactions should be modelled with care and have a significant resemblance to the final experience. Interdependence between the interface and backend mechanisms is often omitted to reduce costs and speed up development cycles.

Prototypes are used to their full potential in user testing. Such a simulation of the final interactions forms great material to check the usability of the interface, before the development actually begins.

Prototypes usually aren't the best documentation you can imagine, since they force the 'reader' to take some effort to understand the interface. On the other hand, a prototype is the most engaging form of design documentation, as the interface is tangible and straightforward.

Beware that prototyping is rather an expensive and time-consuming form of design communication. I'd suggest rather creating prototypes that can be reused in development (yep, it means that you need to code some HTML, CSS and probably JS on your own). It's especially effective in relatively simple projects.

Done right and combined with user testing, prototyping can pay for itself.

GETTING OUT OF THE SILLY DELIVERABLE BUSINESS

You might hear from time to time a call to get out of the deliverable business and focus on the action. It seems a little bit strange and unnecessarily rebellious. Should we always get out of the highway to try our luck at finding a shortcut through the woods? It might work from time to time, but just watch the famous (and painfully simple) horror movie *Wrong Turn* to see where it can lead you if you are out of luck.

Oh yes, it may hurt.

I might agree with this call, though, if we added ‘silly’ to the sentence. Get out of the silly deliverable business – now that’s the task we all should accomplish as soon as possible. It makes a lot of sense.

What’s a silly deliverable? It’s the one that’s unnecessary for success.

You should keep unnecessary work to the bare minimum and always remember that your job is to design a product, not documentation. Do only what helps you create a stunning user experience.

ITERATE, ITERATE, ITERATE

We live in dynamic times in which we need to constantly adapt ourselves and our ideas to the changing requirements of the (business) environment. This means that all the design techniques that you

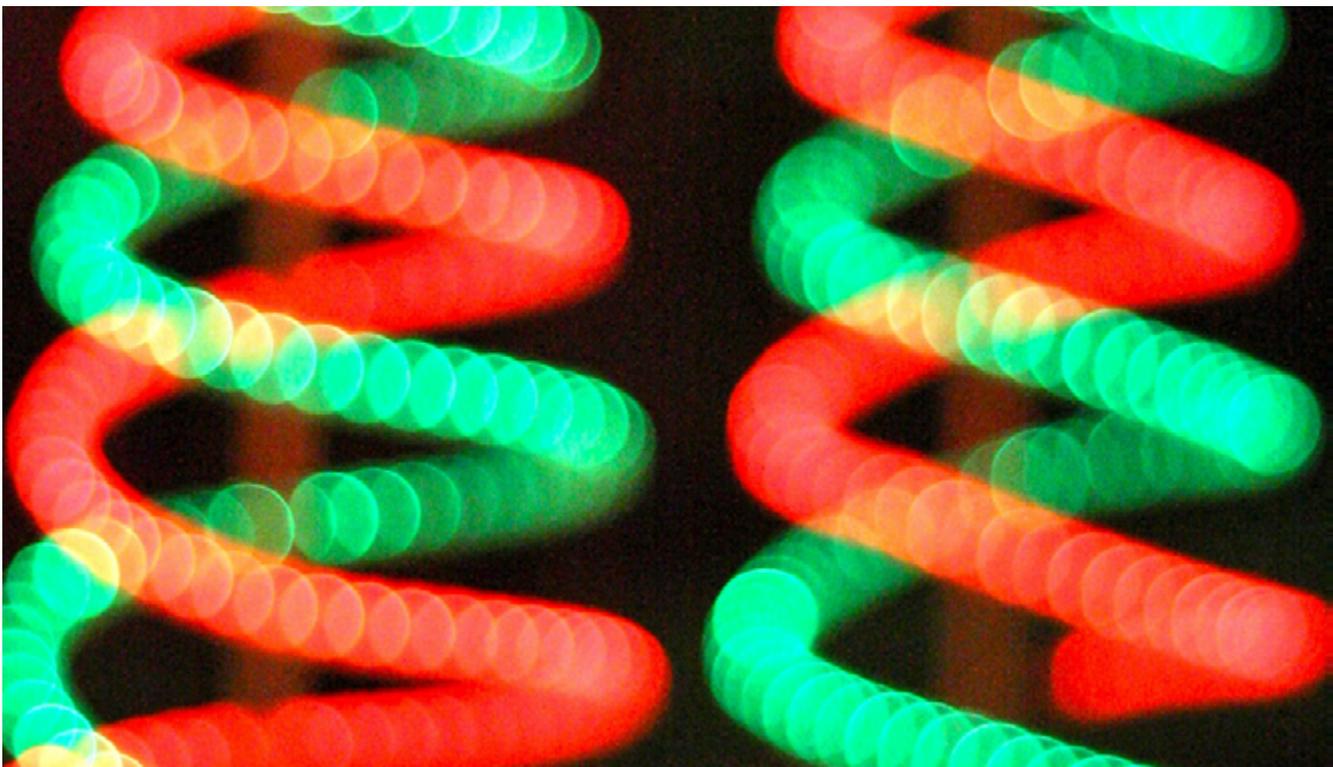
use should leave you just enough space and time to constantly iterate on your ideas and those of your team.

Don't totally devote yourself to one concept and one design technique. Design needs richness and you shouldn't feel limited.

GROWTH AND DESIGN HACKING

I love the smell of metrics in the morning.

I spent years in a metrics-driven organisation with analysis deeply rooted in the company's DNA... and it was just great. My user experience design team was constantly occupied with lots of small tasks focused on the optimisation of the UI and though this is not something you can brag about during a family dinner (hardly noticeable things aren't particularly admired), it let our users perform better every day and damn... it was bringing the company money! We did our job well.



Metrics Driven Organizations got Analysis deeply rooted in their DNA. Photo credit: "Christmas DNA" by Kevin Dooley. Picture cropped to fit layout. [Creative Commons](#).

Despite all this experience, I struggled to put up an analytical framework for UXPin - my own startup. By not measuring things properly, We were pushing our company into the void of lost opportunities, money and users' trust. It was a nightmare for me both as an entrepreneur and a user experience designer.

I started to wonder: if we're creating The UX Design App, shouldn't we be an example of a great approach to user experience design? Why do we keep failing? What's the difference between our startup and my previous job? The answer was simple: the difference is fundamental. Startups are just different.

It's not the difference in team size and revenue numbers, it's the difference of dynamics, uncertainty and your personal feelings.

The latter is probably the most influential. In your own startup some things are unnaturally hard, because you care too much. Caring too much brings chaos on board. Chaos makes the simple complicated. Next thing you know: you're in trouble.

Besides, in a startup you're so busy building that you sometimes forget about thinking. Ain't that shamefully right, fellow entrepreneurs? Who doesn't get caught by this vicious trap from time to time? I know I do.

UXPin came a long way – from a fresh product with problems to current solid traction and amazing growth in sales. I'd like to share with you how we used specific metrics to stay focused and really accelerate our business. Hopefully, it will spare you a couple of sleepless nights and give your project a proper push.

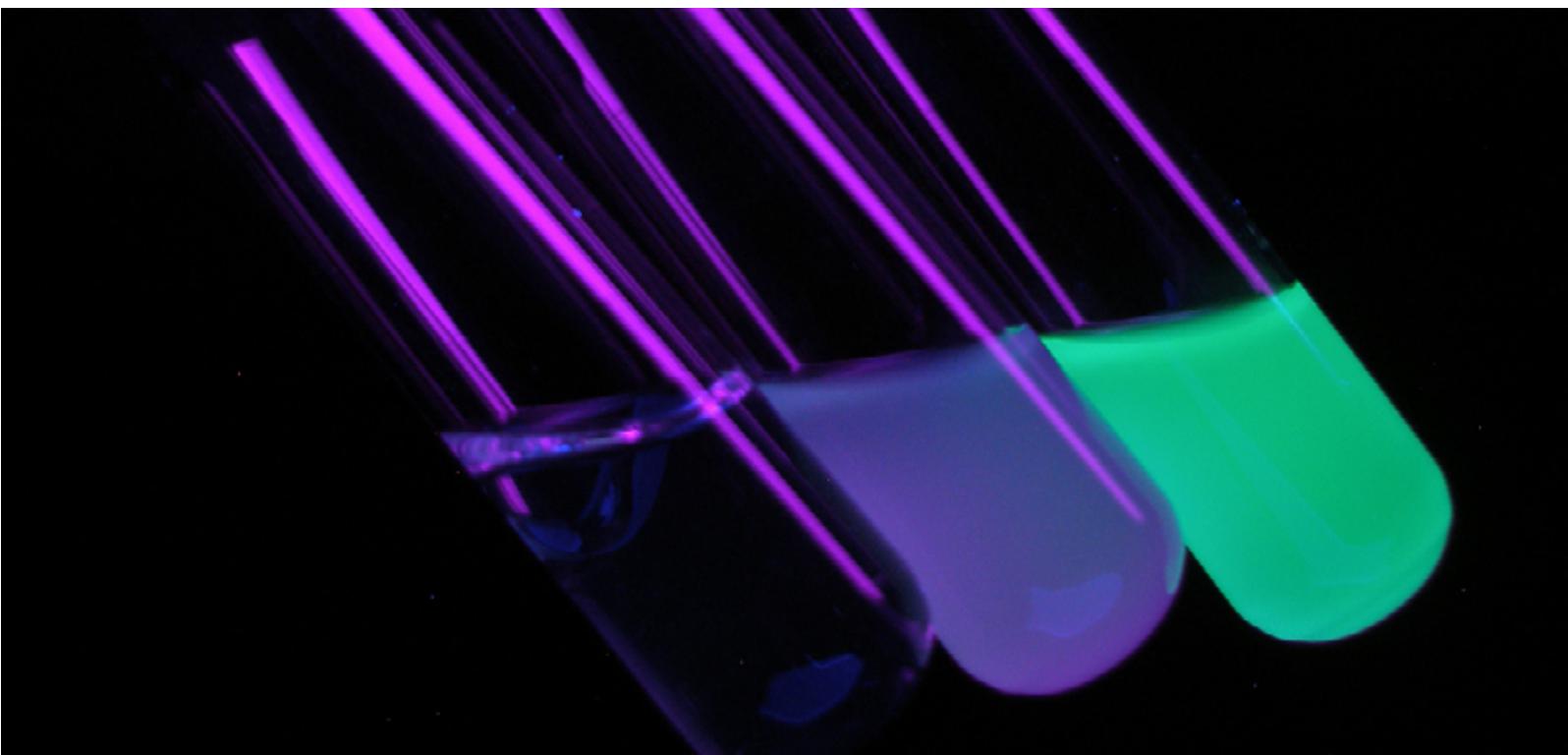
CROSSROADS OF ART AND SCIENCE

User experience design lies at the crossroads of art and science. It's a magical mixture of visual art, hard-boiled psychology and numbers. Drink it, click your heels and you'll soon be in the right place, Dorothy.

User experience design is powerful, but honestly, there's not a lot of mystery here. On a very general

level, successful UX designers do just three things:

1. Measure human behaviour and act upon metrics;
2. Come up with **solutions** to well understood problems, basing their ideas on psychological knowledge and data gathered in research (solutions are visualised as prototypes, wireframes, sketches, diagrams etc.);
3. Communicate with other members of the team to **facilitate design collaboration.**



UX is a mixture of visual art, psychology and analysis. Photo credit: "E. coli GFP" by Ryan Kitko. Creative Commons 2.0.

Do the same and your startup will flourish and grow rapidly. Sounds simple right? Unfortunately, sometimes the road is unpleasantly rough. Measuring human behavior in a startup is hard to do and easy to forget.

I've recently had a conversation (not the first one of its sort) on why the results of the work of a very talented designer don't bring home the bacon (happy users and money). The design looks great, most of the decisions are backed up with reasonable argumentation, it's shiny, personal and seems to be clever. What could be wrong? Why doesn't it simply fly?

It's very easy to lose faith in the designer's talent, the users, or, God forbid, the design itself. Too easy. We have this inner urge to blame, but believe me - that's not the right path to take. This shiny design might have a certain value, it just doesn't perform well enough. Blaming the designer would only obscure the picture. Perhaps we're just one small tweak away from a great-looking, high-performance interface. How could we know this, if not by carefully measuring performance, gathering the right data and drawing a

valid conclusion?

Make sure that you know what your design is supposed to do (choose one main thing to start with), choose one metric that can tell you if people succeed and measure it. The numbers don't look too good? Try to figure out what's going wrong (classic usability testing might come in handy) and correct it. It's almost always that easy.

Measurement is a habit that you need to grow and in time you'll get better and better at choosing the right things and ways to measure them. Your startup will flourish.

In our story, the particularly talented designer, didn't measure and didn't optimise his designs. No wonder there was no bacon at the table. He remained unsuccessful because he forgot about one ingredient of our magical mixture of user experience design - the numbers - a measurement of user behaviour. That's the easiest way to fail.

You don't want to copy his approach. Especially when your business is at stake.

TO MEASURE OR NOT TO MEASURE?

Big players measure a lot. Every step a user takes, every tiny business occurrence, cash flow... no doubt they gather powerful data and it costs them a lot. Dozens of analysts are using every working hour to measure everything that's measurable.

I assume, as an entrepreneur, you can't afford an army of analysts. I'm pretty sure you have a lot of things worth measuring, but not nearly enough people and time to measure them.

Don't worry. That's not the problem.

Measuring too many things is paralysing for just about any company and it's a death walk for a startup. You measure to validate decisions and decrease the risk of

failure. The minimal amount of information necessary for a certain decision is good enough. Over-thinking decisions doesn't further decrease the risk of action, as Daniel Kahneman pointed out in his recent book. Just listen to your data and make a decision. Do it.



Measuring to many things may lead to decision paralysis. Photo credit: "ampel" by iwamp. [Creative Commons 2.0](#).

Testing adds more value to your company than over-thinking. It might sound ridiculous, but only

testing lets you operate on real data, not on a set of assumptions. The lean startup methodology draws a lot from this approach, which is pretty common in the world of science. For example, psychology relies heavily on experimental methods to test theories about human behaviour.

Measuring only the right things is one of the competitive advantages that you can have over bigger players. They have way too much money and too many resources to stay focused! When it comes to analysis, being small actually helps! You can't allow any waste, because it may put you out of the business. When big players dive into data, postponing the decision for months, you can test a couple of assumptions based on your small, but accurate, set of metrics. Isn't that just great?

Agility is your greatest power. Use it wisely and may the force be with you.

All right, but how do you decide what to measure? There are two sets of metrics that you need to take into

account: economic metrics and behavioral metrics.

ECONOMIC METRICS

Company Stage	Metrics
Pre-revenue. No traction.	<ul style="list-style-type: none"> * Track the engagement of users. How many of them start to use the product on a regular basis? * Analyze the sign-up conversion funnel, without the last step (paying customer). Optimize each of the steps.
Pre-revenue. Traction.	<ul style="list-style-type: none"> * Continue to track the engagement of users. If possible start testing whether customers are "willing to pay". * Continue to optimize the conversion funnel, without the last step (paying customer).
Revenue. No traction.	<ul style="list-style-type: none"> * Pray to your sign-up conversion funnel. That's the single most important thing right now. * Track the number of paying customers * Keep an eye on the Churn Rate (number of people who are leaving your product). * Analyze your CPA (cost per acquisition)
Revenue. Traction.	<ul style="list-style-type: none"> * Track the recurring revenue per month, * LTV (life-time user value - how much do you earn on an average user as long as he uses your tool?), * ARPU (average revenue per user), * Churn Rate. Focus on optimization. * Analyze your CPA (cost per acquisition)

They must clearly show the state of your business. The choice of this set of metrics strongly depends on the stage of your company. Take a look at following table:

Right now UXPin focuses strongly on the number of paying customers, as we're vastly interested in tracking our progress in encouraging users to join us. The number of people making the decision to use UXPin and overcoming the obstacle of reaching for the credit card is currently more important for us than monthly revenue. The number of paying customers lets us know if our target group responds to UXPin products in a positive way. Luckily it does!

We're getting to the point at which the whole business model will become scalable and we'll have enough data confirming that we're on the right track. It will be the time of LTV (user's Life-Time Value) and ARPU (Average Revenue Per User) optimisation, which should elevate our business to the next level.

BEHAVIORAL METRICS

Behavioural metrics are meant to track very specific actions of your users. Whenever you're about to launch a new feature or product, consider:

- What's the main use case? (It should be derived from your C-P-A hypothesis)
- How do you measure whether users are able to succeed in the main use case?

Your goal is to gather data that will let you assess the new feature or product's performance. For example, if it's a new sign-up form, track:

- The number of successful sign-ups
- The conversion rate
- The number and type of errors

The number and type of behavioral metrics depends on

the project and the hypothesis that you formed during the design process. Remember - less is more. You just need metrics to validate your design hypothesis - don't track everything or you'll be lost in data.

When you're analysing behavioural metrics, you must always take into account economic metrics as well. Most of the features, and certainly all of the products, must add value to the company and you need to make sure they do. This is why you actually track economic metrics, right?

If, after the launch of a certain feature, sales suddenly drop, you'll need data to check what happened. That's why it's particularly important to precede any launch with the implementation of appropriate analytical tools.

MIRROR MIRROR ON THE WALL...

We all love to brag sometimes, right? OK... at least

most of us do. Numbers are one of the greatest bragging tools. Their meaning always depends on the context and they're just so easy to manipulate. If a SaaS application brags about 4 million pageviews, but they don't have any paying customers, would you call them successful? I wouldn't.

The number of pageviews is a typical vanity metric for almost all SaaS applications and many other web startups. What's a vanity metric? As Brad Smith nicely put it: "Vanity metrics are things people love to quote and obsess over, even though they're almost entirely useless to your business."

Vanity metrics make the naive among us feel good, but at the same time they push the whole business into an endless depression of idleness. Vanity metrics are absolutely unactionable and therefore useless. They're a waste of time that can destroy your startup.

To give you a couple more examples: time on site is a vanity metric, so is the average number of pageviews per user, or the percentage of new visitors.

Some vanity metrics are more tricky. In UXPin the “number of projects with comments” was one of them. It seems to be a reasonable behavioural metric that was supposed to let us check the engagement of users in commenting on a feature. Well... it didn't. The number itself didn't tell us anything. Some users don't have people to share a project with, some rather like to export a PDF and attach it to a project management tool, etc. This metric couldn't tell us about those cases and overall it just failed to provide us with the appropriate knowledge to make any decisions. We killed it to stay focused on what's really important.

That's my recommendation to you: keep up with the important metrics and kill the vanity ones. Less is more.

DO IT OVER AND OVER AGAIN!

After several weeks of madness in UXPin we managed to get off our knees and start to properly measure

the right metrics. That was a relief! Finally, I didn't feel completely stupid and we started to learn from our users. Great! We used Google Analytics and everything that was important was right there; we could see all the metrics with our own eyes.

Did it cause the necessary change? Nope.

Nobody seemed to care about our shiny, super sexy metrics, apart from two UX designers (including me), who cared a little, but not nearly enough. Our approach wasn't actionable. Metrics were separated from product development cycles, which should never happen!

How can you expect people in your company to care about metrics if you don't let them see the influence they have? Every product development cycle should result in a positive change of metrics.

Then we came up with a ridiculously obvious idea: why not set goals based on metrics and check if we're on the right track weekly? This single thought set our minds

on fire and we started weekly measurement cycles with monthly and quarterly sum-ups.

How could we not have come up with the idea of measurement cycles earlier? I have no idea. When we had them up and running they seemed so obvious. After all - you measure to optimise your business, not measure for the sake of mere measurement, and a weekly control of metrics forces the whole company to focus on business optimisation.

That was the shift that we were looking for.



Measure in cycles, over and over again. Photo credit: "look downstairs into stairwell whirl" by Karl-Ludwig Poggemann. Picture cropped. [Creative Commons 2.0](#).

Suddenly, the whole company started to care about our metrics. Goals helped us focus on really important things. They clearly showed where we are and how our work influences business. Metrics became powerfully actionable. If we started to fall short of our predictions, we could take almost immediate action and correct

	Week 1	Week 2	Week 3	...
Unique Visitors				
Registered users				
Conversion Rate #1 (%)				
Activated users				
Conversion Rate #2 (%)				
Number of sales this week				
Conversion Rate #3 (%)				
Canceled accounts				
Churn rate (%)				
Total number of paying customers				
Weekly growth in sales				

ourselves based on knowledge gathered weekly.

Here's the table that we use:

Your table might look different - it depends on your business model and the current stage of your company. We're a SaaS company with steady growth and decent traction, so this kind of funnel makes sense for us.

QUALITY COMES FROM CONVERSATIONS

We've got the numbers figured out, but there's another equally important part of analysis that can't be undervalued if you really aim at designing the best user experience possible. Qualitative testing.

The methodology of science empowers us with a whole range of qualitative research methods (case studies, participant observation, direct observation, unstructured interview, individual in-depth interview,

focus groups...), which deserve their own chapter, or perhaps a whole book. Let's not focus on each specific research method, but rather the general approach in a startup perspective. After all, again - we're not aiming at complex knowledge, but actionable results.

Qualitative methods are best for broadening your perspective and filling your mind with fresh (often surprising) ideas, derived from your target group. They might give you completely new feature or product concepts, or point out lots of bugs in existing ones. Either way you'll get unique knowledge that will let you work on the quality of your product.

At UXPin we believe that the quality of the product comes from conversation. Constant, on-going conversation with users. Only by having a proper dialog can you work outside of the box formed by your product, get compared to your competitors and know the real problems of your target group. It's a powerful type of research.

The important thing is to be consequent. Just as with

economic and behavioral metrics, you want to perform qualitative tests regularly.

Choose a method and implement it in your measuring cycles. At UXPin we do:

1. Classic usability testing once a month (thanks to the helpful local UX design community! Cheers guys!)
2. Individual in-depth interviews with customers every two weeks.

Each session is always extremely refreshing and has a strong share in our current growth.

GROWTH AND DESIGN HACKING TOOLS

You know how we approached measuring crucial things in UXPin; you know where it has led us. I sincerely hope that this knowledge will help your

startup reach a high peak of user experience design.

To help you start, here's a short list of tools that we find really valuable.

Economic and behavioral metrics

- Google Analytics
- KissMetrics
- MixPanel

Usability Testing

- Silverback

A/B testing

- Visual Website Optimizer
- Optimizely

GET IT OPTIMISED

At UXPin, we're trying to be as agile as we possibly



*Celebrate only when you really improve the user's experience. Photo credit:
"Weekend Warriors" by JD Hancock. [Creative Commons 2.0](#).*

can. We launch new versions of our user experience design app every couple of days. Some features stay

with us for good, some are killed soon after the launch, and some are improved in time to meet our (and our users') expectations. We accept constant change and a never-ending quest for improvement. Let me explain the basic tenets of our approach and tell you how to use it in your startup.

THE GOLDEN RULE

When I used to work as an in-house UX manager, my UX team had one golden rule: “a designer never leaves a project”. We meant that the launch of a new product/feature doesn't authorise anyone to have a crazy celebration. No champagne, no running about naked, no dancing around. There's nothing to drink to or jump for joy. **Launching a product is just an opportunity to measure your users' behaviour, learn from it and optimise the design.** It's a time for crucial work and a feeling of anxiety rather than relief.

It sounds like we were no fun, right? But the point of

this approach wasn't to become party poopers. It hung on one thing: the definition of success.

UX designers always try to influence the behaviour of users and improve their experience. That's the ultimate goal. We succeed if the users succeed in using our product and their experience is magnificently good.

Our golden rule was meant to constantly remind ourselves about two things:

1. A UX design is only as good as its measured performance
2. Optimisation is constant. We just postpone it when the costs are higher than the assumed gain.

Interestingly, this approach proved itself to be even more important in a startup than in an established company. Accepting the constant effort to improve the user experience is a 'make it or break it' factor for every young organisation. UXPin would never have survived its initial stage if not for constant measurement and

optimisation.

Most entrepreneurs I know are not eager to accept this. We all live in an extraordinarily tough and stressful environment. When we launch a new product or feature, we're chasing our dreams. We believe that from now on, finally, everything will be just fine.

Usually, it isn't. In many cases, nothing changes after a launch. In some situations, things might even get worse. You must be ready for this, have the knowledge to improve your situation, and convert losses into gains. Remember, as an entrepreneur, **you must be goal-oriented, not relief-oriented**. Relief after deploying code to production servers is not a success: a change in metrics is.

I know this is hard. Startups are hard. Startups that care for user experience are even harder.

But don't worry: what doesn't kill you, makes you stronger. Constant improvement of your users' experience will lead you to overall success. It might not

happen overnight, but consistency will pay off.



Technically this car is fine, but actually it might fail to provide a great driving experience. Photo credit: [“Old cars around woodford,10-09-2013 \(10\)”](#) by [bertknot](#). [Creative Commons 2.0](#).

TECHNICALLY AND ACTUALLY WORKING STUFF

“A design isn’t finished until somebody is using

it,” said Brenda Laurel, a great designer who used to work for Atari, among others, in the 80s. This is a condensed truth. Great designs don’t just look beautiful: they let people succeed in their endeavours.

A design actually works if people keep using it and their experience is great. This is the difference between ‘technically working’ and ‘actually working’. Your team might claim that something works just because functionally it’s more or less all right. If you know how to perform a certain task, you can succeed – they say. It’s according to the spec – they say. It’s just as we discussed it – they keep saying.

The problem is that the users don’t know, or care, whether the feature is as you discussed. They only care whether they like using it.

So does the feature work or not? In a company focused on user experience, it doesn’t. If users fail to use a certain feature, it doesn’t work. It just works technically – and technically is not nearly enough.

Most successful startups are not in the programming business: they are in the ‘creating a great user experience’ business. So work until your startup will actually enable users to succeed – and beyond.

SEEING DESIGN THROUGH METRICS

To cross the chasm between ‘technically’ working and ‘actually’ working, you need to see your design through metrics. You need knowledge to build a bridge between reality and your dreams. Metrics are your building material, sight-enhancement and super power!

We discussed what to measure and how in fourth chapter; now it’s time to focus on optimising the design.

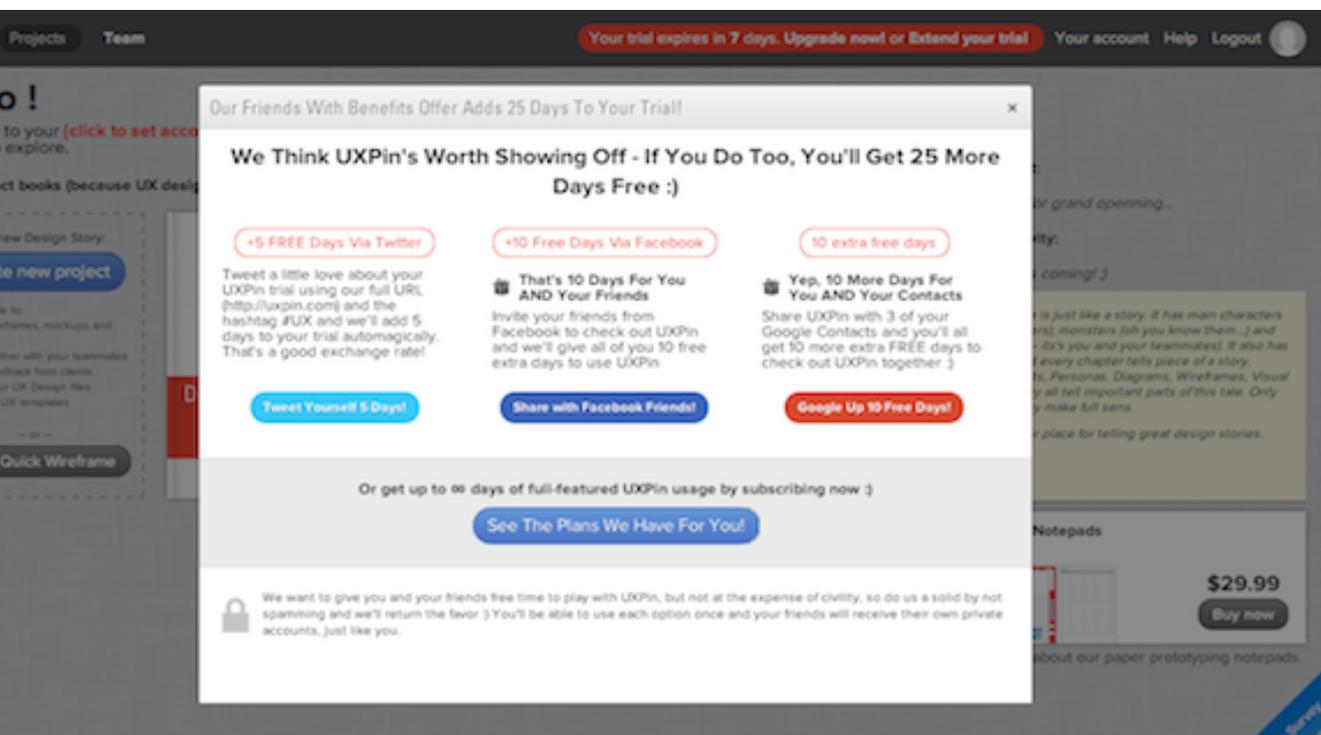
Let’s start with an example. A couple of weeks ago, UXPin launched a new feature. We were trying to implement a kind of viral loop inside our system. Our goal was to encourage people to share UXPin and as a

result help us accelerate the growth of sign-ups.

We couldn't give away server space as a reward for sharing as Dropbox did (in UXPin, it's unlimited), so we came up with the idea of giving away trial days. You could extend your trial by up to 30 days if you gave your friends free trial days as a present.

Awesome, right?

We implemented the whole feature in just three days and launched it quickly with a feeling of excitement about the upcoming huge shift in our company.



An object lesson in optimisation: UXPin's incentive scheme

Still aware that our work might need some optimisation, we carefully planned what to measure. To remain actionable, we limited ourselves just to the most important metrics:

1. Number of newly registered users who invited their friends to UXPin for free trial days
2. Average number of invitations sent per user
3. Number of invited friends who visited UXPin and registered
4. Number of users who invited their friends and bought an account
5. Number of invited users who bought themselves a UXPin account

Soon after the launch, people started to use our feature. After one week, the results looked promising: 36% of newly registered users extended their trial using our new feature, inviting an average of three friends

to join UXPin. Unfortunately, only 5% of the invited friends visited UXPin and registered. After one week there were still no sales, but we weren't surprised as everyone was still on the trial.

Then suddenly our sales dropped. We immediately started to check what was going on and which of the new features could be changing our users' behavior. All the signs pointed to the viral loop thing, but since a lot of people were still on the trial, we decided to wait.

After another week, we noticed that the activity metrics (which show us the engagement of people with the app) decreased as well. We assumed that they had probably started to get bored with their long trial. There was not nearly enough stimulation to keep them interested in UXPin without converting them to customers.

We started to talk with our users to find out what was going on. Unfortunately, we were right. The longer trial made them postpone the decision whether to purchase for an indefinite time. They didn't feel rushed

to decide.

That wasn't too good, but since we had all the numbers and we had talked to our customers, we knew what to do: we just changed the point at which the box offering the extension of the trial shows up, and we introduced wireframing templates to engage the users in the app right from the start. After two days of optimising and one week of waiting, our sales started to grow again. Amazing!

But if we hadn't measured things properly, we wouldn't have enough knowledge to act on. Though technically, our beloved feature was brilliant, we could actually just have sat and watched our sales dropping day by day. Awful.

As you can see, viewing design through metrics is your insurance. Whenever you're about to launch something, think carefully about how are you going to measure success and what knowledge you will need to optimise the feature later.

This is the secret of successful UX design – and one which you can implement in your startup straight away.

ENOUGH IS ENOUGH

Fair enough, you may say. But if you never leave your projects, you will either drown in a sea of projects, or constantly optimise just one, both of which will hold your company back.

Nothing could be further from the truth. You just need to know when to stop.



You always need to know when to stop. Photo credit: "Stop Eating" by Rich Anderson. Creative Commons 2.0.

Whenever you're optimising your product, make an assumption of how big the possible gain is, how big the cost is and how high the probability of success is. Try to specify both cost and gain in dollars (or whatever currency you use).

Let's say you need three hours to design an optimised version of a certain feature and another five hours of development to launch the new version. And let's say that your time is worth \$100 per hour to the company. That means the cost of optimisation is \$800. What's the assumed gain? Will it be \$200 per month, for example, which means you'll start earning on this optimisation after one month? What's the probability that it will actually work? What will happen if you don't optimise it?

All these questions are equally important. My rule of thumb is to deal first with anything that creates a loss. If something irritates your customers and makes them leave your app, that's the first thing to optimise. If you don't, you'll always have a hole in your product.

The second step is to look for the cheapest solution that offers the highest probability of the highest gain. (Obviously, in practice, it's not quite that simple. Choosing the right optimisation requires solid knowledge about your users and a little bit of experience in making design decisions. But if you properly measure your users' behaviour, you'll get this with time.)

Finally, in a startup, try to avoid expensive solutions. Think what else you could be spending that money on. If you expend your entire development budget on a feature that will take three months to optimise, your competitors might gain such a lead on you that you'll never be able to catch up.

TOOLS,
TOOLS,
TOOLS

UX Design App:

UXPin <http://uxpin.com>

- Wireframing
- Interactive Prototyping
- Responsive wireframing/prototyping
- Live collaboration
- Full project view (personas, research results etc.)
- Communication in the design process
- Reviews
- Iterations

Wireframing:

- Balsamiq <http://balsamiq.com>

- Mockingbird <http://gomockingbird.com> (free)
- Moqups <http://moqups.com> (free)
- Mockflow <http://www.mockflow.com>
- UXPin <http://uxpin.com>
- Pencil Project <http://pencil.evolus.vn> (free)

Prototyping:

- Axure <http://axure.com>
- FluidUI <http://fluidui.com>
- Hotgloo <http://hotgloo.com>
- iRise <http://irise.com>
- Just In Mind <http://justinmind.com>
- Pidoco <http://pidoco.com>

- Proto.io <http://proto.io>
- Protoshare <http://protoshare.com>
- UXPin <http://uxpin.com>

Diagramming:

- Cacao <http://cacao.com>
- Creately <http://creately.com>
- Draw.io <https://www.draw.io> (free)
- Gliffy <http://www.gliffy.com>
- Omnigraffle <http://www.omnigroup.com/products/omnigraffle>
- Lovely Charts <http://www.lovelycharts.com>
- Lucid Chart <https://www.lucidchart.com>

Web Analytics:

- Adobe Analytics <http://www.adobe.com/solutions/digital-analytics.html>
- Google Analytics <http://google.com/analytics>
- KissMetrics <http://kissmetrics.com>
- MixPanel <http://mixpanel.com>
- Piwik <http://piwik.org>

Feedback tools:

- GetSatisfaction <https://getsatisfaction.com>
- Kampala <http://www.kampyle.com>
- Survey.io <http://survey.io>
- UsabilityTools <http://usabilitytools.com/tools/feedback-form/#tool-description>

- UserVoice <http://uservoice.com>
- Qualaroo <https://qualaroo.com>
- 4Q Survey <http://www.q4survey.com>

Session Recording:

- Clicktale <http://www.clicktale.com/default.aspx>
- GhostRec <http://www.ghostrec.com>
- MouseFlow <http://mouseflow.com>
- OpenHallway <http://www.openhallway.com>
- Tealeaf <http://www.tealeaf.com>
- UsabilityTools <http://usabilitytools.com/tools/visitor-recording/#tool-description>
- UserReplay <http://www.userreplay.co.uk>

ClickTracking:

- Clickheat <http://www.labsmedia.com/clickheat/index.html>
- Clicktale <http://www.clicktale.com/default.aspx>
- CrazyEgg <http://www.crazyegg.com>
- UsabilityTools <http://usabilitytools.com/tools/click-tracking/#tool-description>
- UserZoom <http://www.userzoom.com>

Remote Usability Testing:

- BagelHint <https://www.bagelhint.com>
- Chalkmark <http://www.optimalworkshop.com/chalkmark.htm>
- Ethnio <http://ethn.io>

- Feedback Army <http://www.feedbackarmy.com>
- Five Second Test <http://fivesecondtest.com>
- Keynote <http://www.keynote.com>
- Loop11 <http://www.loop11.com>
- TryMyUI <http://www.trymyui.com>
- UsabilityTools <http://usabilitytools.com>
- Usabilla <http://www.usabilla.com>
- Userlytics <http://www.userlytics.com>
- User Testing <http://www.usertesting.com>

Offline Usability Testing:

- Camtasia <http://www.techsmith.com/camtasia.html>
- MediaCam <http://netu2.com>

- Morae <http://www.techsmith.com/morae.html>
- Silverback <http://silverbackapp.com>

A/B testing:

- Adobe Test and Target <http://www.adobe.com/products/testandtarget.html>
- Artisan <http://useartisan.com>
- GlobalMaxer <http://www.globalmaxer.com>
- Optimizely <http://optimizely.com>
- Visual Website Optimiser <http://visualwebsiteoptimizer.com>