UXPin

UX Design Process Best Practices

Documentation for Moving Design Forward

UXPin

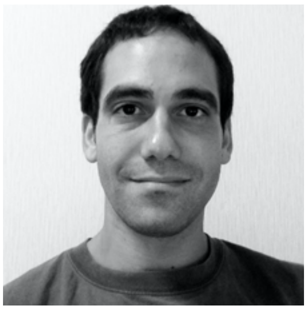
UX Design Process Best Practices: Documentation for Driving Design Forward

Index

Jerry Cao is a content strategist at UXPin where he gets to put his overly active imagination to paper every day. In a past life, he developed content strategies for clients at Brafton and worked in traditional advertising at DDB San Francisco. In his spare time he enjoys playing electric guitar, watching foreign horror films, and expanding his knowledge of random facts.
Follow me on Twitter

Ben Gremillion is a Content Designer at UXPin. Previously, he was a Design Writer at ZURB. He started in newspaper design, saw a digital future, and learned HTML/CSS in short order. He later earned an Adobe Certification, and learned up regex, PHP, MySQL, and other impressive-sounding acronyms. He also builds and maintains a CMS for webcomic artists, and participates in bi-annual NaNoWriMo challenges.

Co-founder and head of product, Kamil previously worked as a UX/UI Designer at Grupa Nokaut. He studied software engineering in university, but design and psychology have always been his greatest passions.
Follow me on Twitter @ziebak



With a passion for writing and an interest in everything anything related to design or technology, Matt Ellis found freelance writing best suited his skills and allowed him to be paid for his curiosity. Having worked with various design and tech companies in the past, he feels quite at home at UXPin as the go-to writer, researcher, and editor. When he's not writing, Matt loves to travel, another byproduct of curiosity.

Minimizing UX Design Busywork

The days of documentation for the sake of deliverables are over.

That's not to say that documentation is no longer relevant – quite the opposite, it's more important now than ever before. The main point to keep in mind is that design documents should complement, not supplement, the design process.

Photo credit: "Taxes." James Morris. Creative Commons.

Documentation must be actionable. It must have a purpose beyond creating a paper trail. The best design documentation both enhances the design process and communicates design thinking to others. If a UX document accomplishes this, its benefit to the project will be seen immediately. If not… then it's just busywork.

In this chapter, we'll give an overview of the design process and how documentation can make it better, instead of just busier.

A Quick Overview of Design Thinking

Design thinking is a strategy that uses the traditional tactics of design to solve problems. It accomplishes its goals from the inside-out, instead of trying to break in from the outside. As Tim Brown, the CEO of IDEO and stark promoter of design thinking, explains it:

> *Design thinking can be described as a discipline that uses the designer's sensibility and methods to match people's needs with what is technologically feasible and what a viable business strategy can convert into customer value and market opportunity."*

In essence, design thinking adheres the classic business adage, "build the right thing, and build the thing right." This means, first off, making sure you're designing a product that people want, and then making sure you design it in such a way that people like it. Notice that both of these goals revolve around the end user.

In design thinking, the product is designed around what the users want and need.

As a business ideology, design thinking is catching on. Even engineering-driven giants like IBM are abandoning their old ways and adopting a more design-centric method. They've hired thousands of designers, put designers in executive positions, and created their own design language.

Photo credit: IBM Design Language

Their new strategy comes from the general manager of design for the IBM Software Group, Phil Gilbert. This design thinker is steering the company towards what people actually want, not what executives think they should want. Gilbert believes that everyone in the process, designer or not, should "have the user as their north star."

3 Stages of Design

Despite its creative points, all design boils down to a methodical, almost scientific approach. While everyone's design procedure varies according to their own personal tastes or constraints, in general the process must cover three essential stages, with plenty of iteration in between:

1. **Research** – Analyze what your users want ("build the right product").

2. **Design** – Create ways to give them what they want ("build the product right").

3. **User Testing** – Confirm your results with users, or discover what you need to change.

Often this cycle is repeated, with each iteration bringing the product closer to perfection. Moreover, the stages are always one after another. Testing should be done intermittently with design to incorporate the results in later designs.

And as for research, ideally you'll want to know as much as you can before you start, so it should be the first step… but that said, it's never a bad time to learn more about your users.

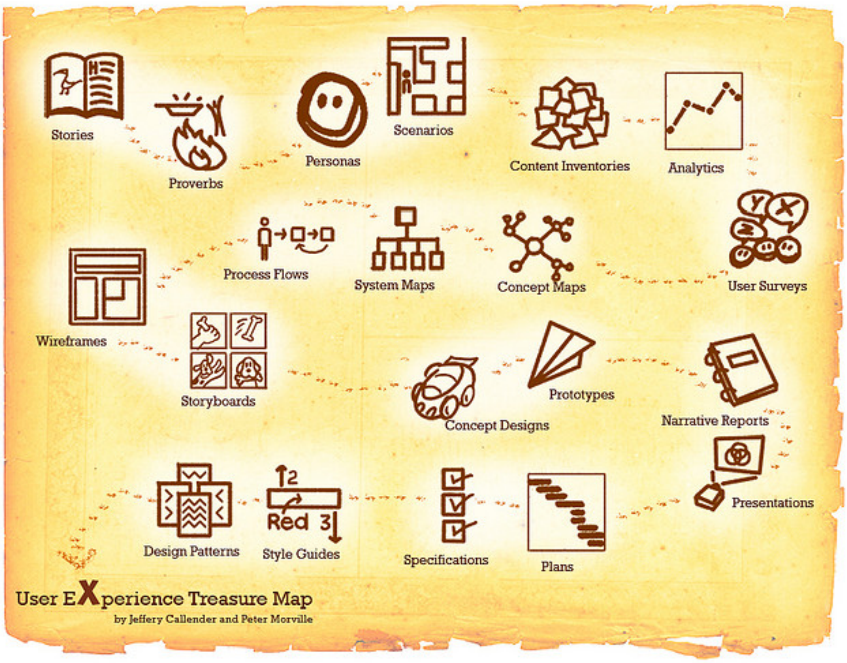


Photo credit: "User Experience Treasure Map." Peter Morville. Creative Commons.

Design Documentation

Now that we've outlined the process, let's talk about where documentation fits in.

As we mentioned above, design documentation should help solve a problem or facilitate communication among the team – ideally, both. If you're creating a document just to have proof that you did something or because that's the "standard process," then you're wasting your time.

While the Waterfall Model of design is outdated, there are some lessons we can take from its linear roots. Before we dive into the details of how to create effective UX documents, let's start with an overview of what to expect based on the three design stages.

1. **Research**

   The research phase is where you make sure you're going in the right direction instead of blindly sprinting ahead. Being prepared and having a solid plan means you won't have to backtrack as much later, so a little extra work at the beginning saves you a lot of time and effort at the end.

   This is the point of the design process where you really come to understand your users. Collecting data through user research (interviews, field studies, etc.), mixed with some old-fashioned empathy, will give you a good idea of whom you're designing for, and what they want.

Additionally, you'll want to understand the needs of the stakeholders, just as important as the end-users. You could design the greatest product in the world, but it would never see the light of day unless the stakeholders are satisfied.

2. Documentation

The research-stage documentation can be broken up into two sections: collecting the data and what you do with it.

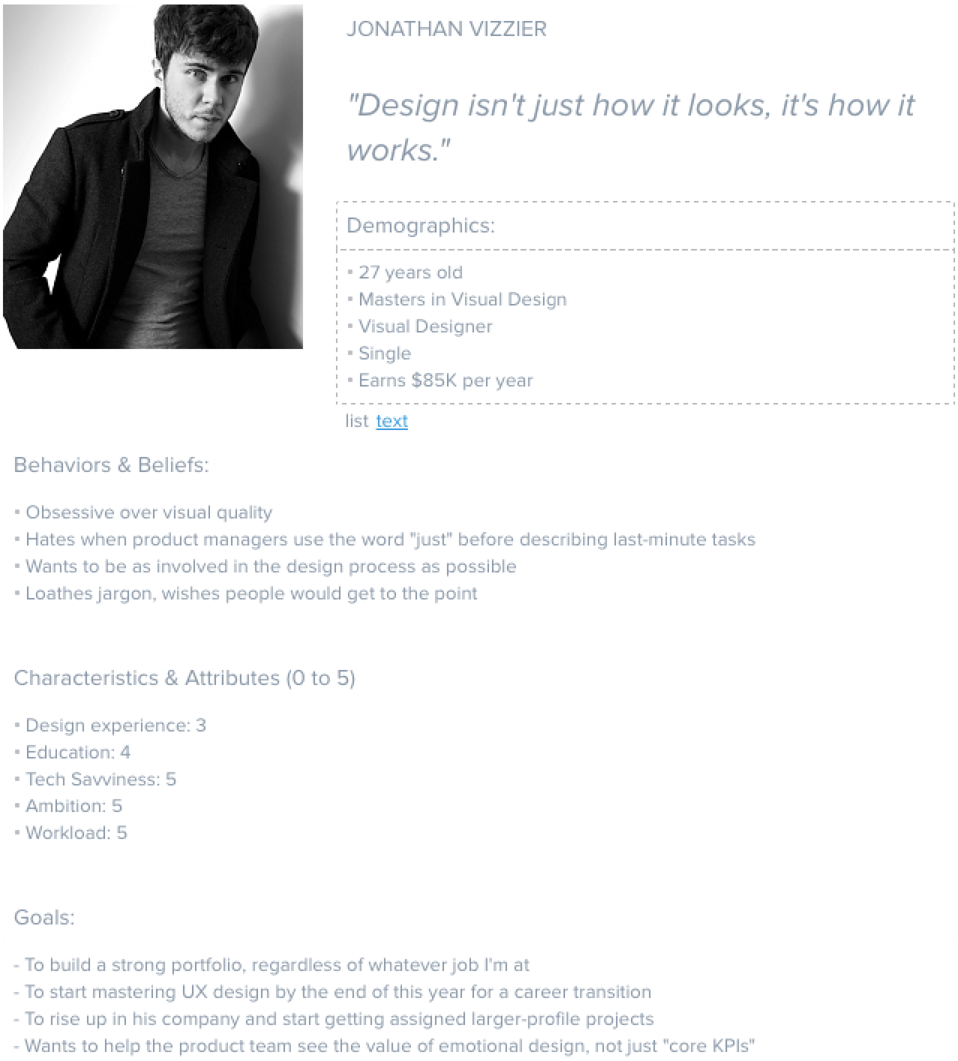
Collecting the data:

- **Stakeholder interviews** – The first thing is understand business needs and technological parameters, and the best way to do this is to ask the sources. Both the questions you ask and the notes you take on the answers will determine how close to the mark your final product hits.

- **User interviews** – Just like stakeholder interviews, you need to ask the right questions to get the most helpful answers. User interviews prove invaluable for understanding the people you're designing for, which is the cornerstone of the design process. Other options for collecting initial user data are field studies and diary studies.

- **User surveys** – While not as personal as user interviews, user surveys are easier to orchestrate and can cover more people – plus they are natural documentation that are easily forwarded to teams. Surveys are great for taking a quantitative approach to qualitative data. You can create these quickly with Google Forms, or use SurveyMonkey for something more complex.

- **Competitive Audit** – Examine strengths and weaknesses of your competitors using an overlaid heuristics diagram. Evaluate areas such as ease of form completion, clarity of navigation, accessibility, trust factors, etc.



JONATHAN VIZZIER

"Design isn't just how it looks, it's how it works."

Demographics:

- 27 years old
- Masters in Visual Design
- Visual Designer
- Single
- Earns $85K per year

list text

Behaviors & Beliefs:

- Obsessive over visual quality
- Hates when product managers use the word "just" before describing last-minute tasks
- Wants to be as involved in the design process as possible
- Loathes jargon, wishes people would get to the point

Characteristics & Attributes (0 to 5)

- Design experience: 3
- Education: 4
- Tech Savviness: 5
- Ambition: 5
- Workload: 5

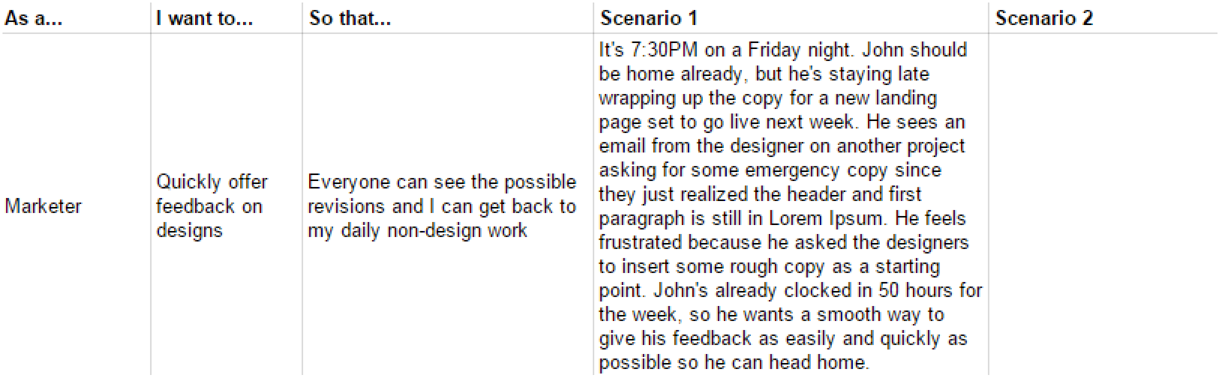
Goals:

- To build a strong portfolio, regardless of whatever job I'm at
- To start mastering UX design by the end of this year for a career transition
- To rise up in his company and start getting assigned larger-profile projects
- Wants to help the product team see the value of emotional design, not just "core KPIs"

Showcasing the data:

- **User personas** – Once you have adequate data on your users, you can build fictional user personas. These act as stand-ins for the actual user during the design process, focusing more on behaviors rather than demographics.

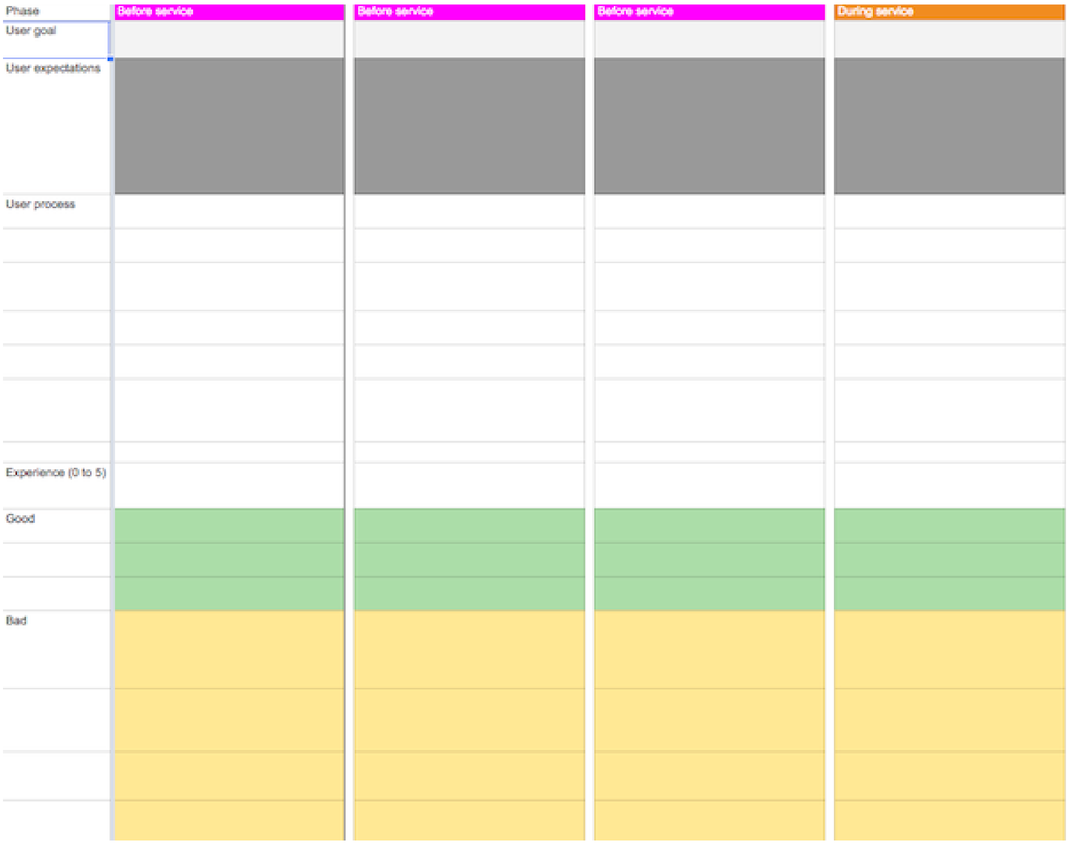
- **User scenarios** – These logic exercises take personas a step further – they outline how a persona would act in a specific situation, including what pages they visit and why.



As a...	I want to...	So that...	Scenario 1	Scenario 2
Marketer	Quickly offer feedback on designs	Everyone can see the possible revisions and I can get back to my daily non-design work	It's 7:30PM on a Friday night. John should be home already, but he's staying late wrapping up the copy for a new landing page set to go live next week. He sees an email from the designer on another project asking for some emergency copy since they just realized the header and first paragraph is still in Lorem Ipsum. He feels frustrated because he asked the designers to insert some rough copy as a starting point. John's already clocked in 50 hours for the week, so he wants a smooth way to give his feedback as easily and quickly as possible so he can head home.	

- **Customer journey maps** – The ultimate document for understanding your user, journey map out the personas and user scenarios at each step of the experience. User emotions, quality of experience, product weaknesses, and other factors can all be documented. Moreover, they cover customer touchpoints before, during, and after service so so you can gauge the lasting effect of your design.

- **Product documentation** – While optional in the age of prototyping, documentation like product requirement documents and functional spec documents consolidate market and user research into a unified vision. ProductHunt's documentation is a fantastic example. Meanwhile, style guides help ensure consistency and adherence to best practices during the design stage.



Phase	Before service	Before service	Before service	During service
User goal				
User expectations				
User process				
Experience (0 to 5)				
Good				
Bad				

3. Design

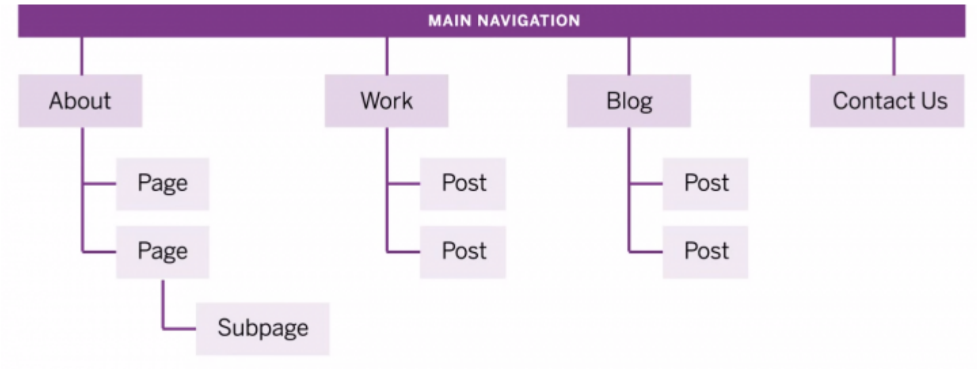
Design documents are often the physical design itself. They change in both form and complexity as the design process advances, and the types of documents, including their fidelity, vary greatly.

While there are many advantages to designing quickly and creating a final product as soon as possible, that's no excuse to cut corners. Some of these documents may feel like extra work, but each of them brings something unique to the design process. That doesn't make them all necessary, though – feel free to skip the ones that aren't applicable to your project.

4. Documentation

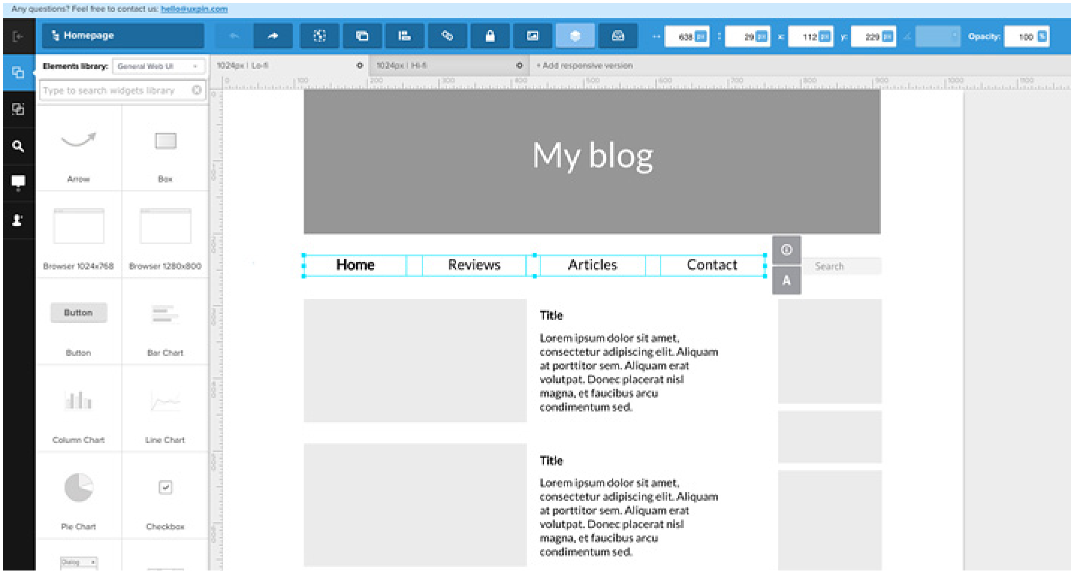
- **Sketches** – Classic sketches on paper are one of the quickest and easiest ways to get your ideas down and share them with others, especially if you're brainstorming.

- **Site maps** – Outlines of your information architecture, showing how your pages are connected to one another.



Site maps in UXPin

- **Wireframes** – These are the barebone structures of your product; dedicating time to building one allows you to flesh out your sitemap without more complex details distracting you.


Any questions? Feel free to contact us: hello@uxpin.com

Elements library: General Web UI

Opacity: 100

Title

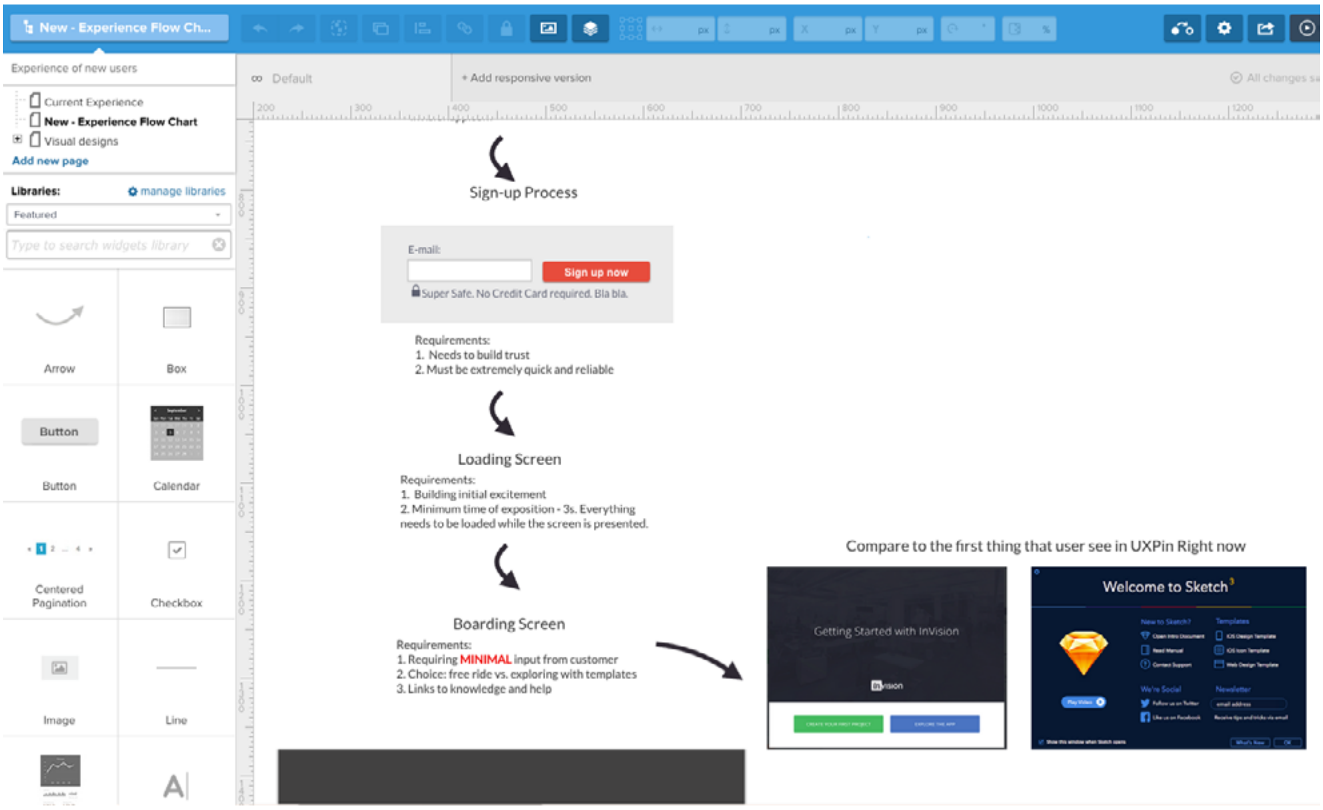
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam at porttitor sem. Aliquam erat volutpat. Donec placerat nisl magna, et faucibus arcu condimentum sed.

Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam at porttitor sem. Aliquam erat volutpat. Donec placerat nisl magna, et faucibus arcu condimentum sed.

Lo-fi wireframes in UXPin

- **User Flows** – Once you've done enough user research, you can start outlining how the pages in your design correspond with user actions. User flows are fast shorthand notes that help you improve the efficiency of your design. You can evaluate the amount of friction at each step and minimize steps when possible. We recommend Ryan Singer's shorthand user flows.



Super Safe. No Credit Card required. Bla bla.

Requirements:
1. Needs to build trust
2. Must be extremely quick and reliable

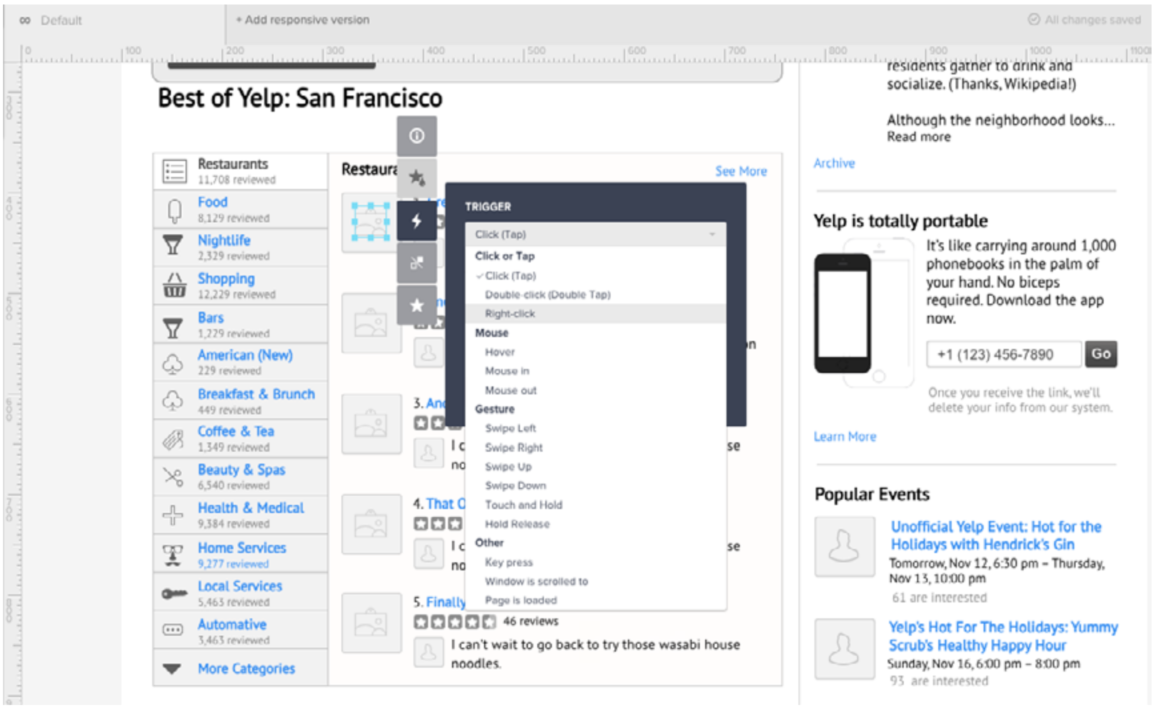
Requirements:
1. Building initial excitement
2. Minimum time of exposition - 3s. Everything needs to be loaded while the screen is presented.

Requirements:
1. Requiring MINIMAL input from customer
2. Choice: free ride vs. exploring with templates
3. Links to knowledge and help

Compare to the first thing that user see in UXPin Right now

User flows in UXPin

- **Interactive wireframes (lo-fi prototypes)** – Adding a little interactivity into your wireframe allows for early product testing – the earlier you can get feedback, the easier it is to implement. No need for complex interactions (save those for a hi-fi prototype), just make important elements clickable so people can actually use the design.



residents gather to drink and socialize. (Thanks, Wikipedia!)

Although the neighborhood looks... Read more

I can't wait to go back to try those wasabi house noodles.

It's like carrying around 1,000 phonebooks in the palm of your hand. No biceps required. Download the app now.

Once you receive the link, we'll delete your info from our system.

Tomorrow, Nov 12, 6:30 pm – Thursday, Nov 13, 10:00 pm

Sunday, Nov 16, 6:00 pm – 8:00 pm

Interactive wireframes in UXPin

- **Paper prototypes** – The most basic form of prototyping to explore the efficiency and usability of your design. Good for brainstorming and inviting feedback from others due to its simple format. Requires a coworker act as the "human computer" to manipulate the prototype for usability testing.

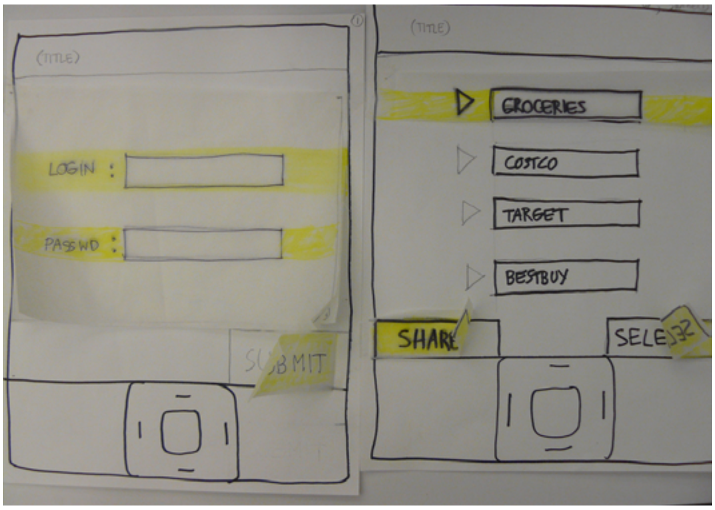
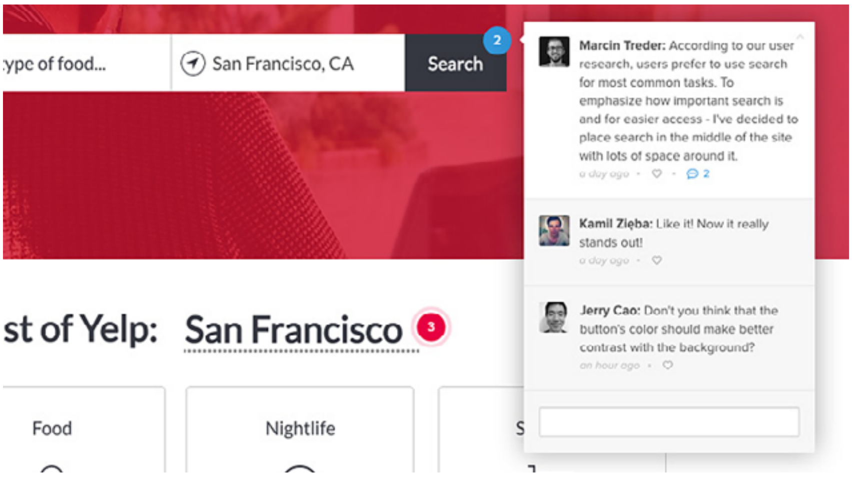


Photo credit: "Paper Prototype of Mobile Application" by Rodolphe Courtier is licensed under Creative Commons 2.0

- **Mockups** – These allow you to focus solely on the visual details of your product, creating a hi-fidelity reference for how it should look. Using UXPin's Photoshop and Sketch integration, you can easily transform static mockups into interactive prototypes (no layers lost).



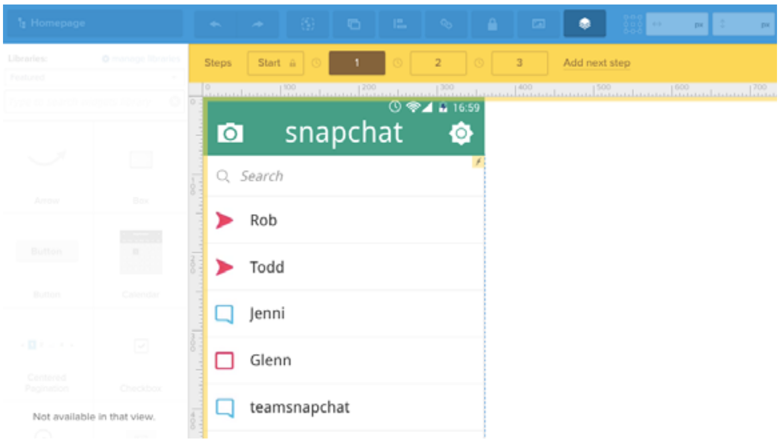
Marcin Treder: According to our user research, users prefer to use search for most common tasks. To emphasize how important search is and for easier access - I've decided to place search in the middle of the site with lots of space around it.
a day ago · ♡ · 💬 2

Kamil Zięba: Like it! Now it really stands out!
a day ago · ♡

Jerry Cao: Don't you think that the button's color should make better contrast with the background?
an hour ago · ♡

Collaborative Mockups & Prototypes in UXPin

- **Hi-fi prototypes** – The last iterations of your product before the live version. Hi-fi prototypes are ideal for fine-tuning the interaction design and animations. If you use UXPin, the custom animations editor lets you map out animated prototypes step-by-step without any code.

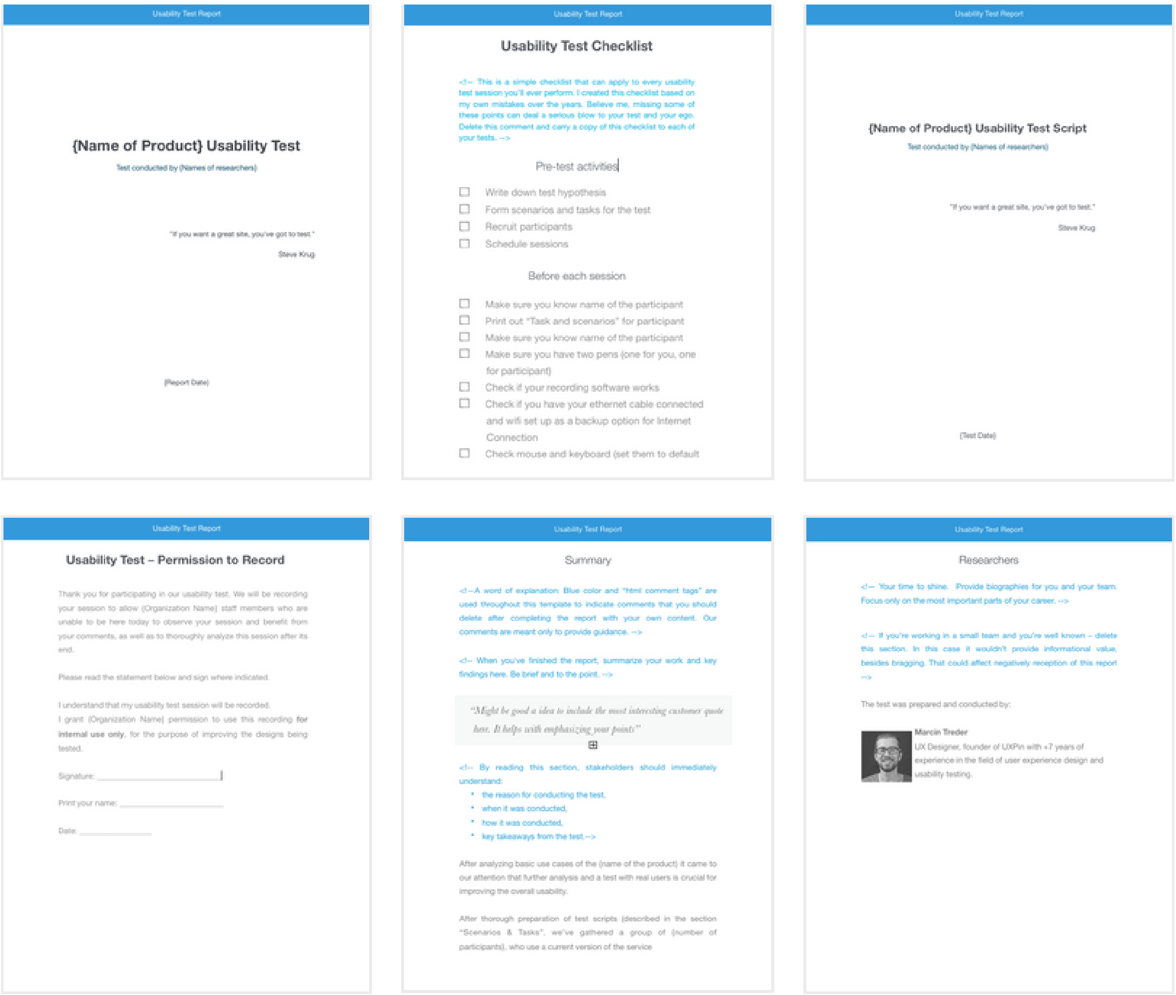


Animated Prototypes in UXPin

5. User Testing

Usability testing is the strongest influence on design decisions. It helps ensure you aren't forcing users into unnatural paths of behavior, and also counterbalance stakeholder feedback. Don't confuse user testing as the final stage – usually the results of the tests lead to further research and iteration.

Don't think this stage must only come after the designing phase. Testing should occur early and often – it should happen alongside of the design process at different intervals, so that the results can be integrated into further designs.



Usability Test Report

{Name of Product} Usability Test

Test conducted by {Names of researchers}

"If you want a great site, you've got to test."

Steve Krug

{Report Date}

Usability Test Report

Usability Test Checklist

<!-- This is a simple checklist that can apply to every usability test session you'll ever perform. I created this checklist based on my own mistakes over the years. Believe me, missing some of these points can deal a serious blow to your test and your ego. Delete this comment and carry a copy of this checklist to each of your tests. -->

Pre-test activities

☐ Write down test hypothesis
☐ Form scenarios and tasks for the test
☐ Recruit participants
☐ Schedule sessions

Before each session

☐ Make sure you know name of the participant
☐ Print out "Task and scenarios" for participant
☐ Make sure you know name of the participant
☐ Make sure you have two pens (one for you, one for participant)
☐ Check if your recording software works
☐ Check if you have your ethernet cable connected and wifi set up as a backup option for Internet Connection
☐ Check mouse and keyboard (set them to default

Usability Test Report

{Name of Product} Usability Test Script

Test conducted by {Names of researchers}

"If you want a great site, you've got to test."

Steve Krug

{Test Date}

Usability Test Report

Usability Test – Permission to Record

Thank you for participating in our usability test. We will be recording your session to allow {Organization Name} staff members who are unable to be here today to observe your session and benefit from your comments, as well as to thoroughly analyze this session after its end.

Please read the statement below and sign where indicated.

I understand that my usability test session will be recorded.
I grant {Organization Name} permission to use this recording **for internal use only**, for the purpose of improving the designs being tested.

Signature: ______________________

Print your name: ___________________

Date: _____________

Usability Test Report

Summary

<!--A word of explanation: Blue color and "html comment tags" are used throughout this template to indicate comments that you should delete after completing the report with your own content. Our comments are meant only to provide guidance. -->

<!-- When you've finished the report, summarize your work and key findings here. Be brief and to the point. -->

"Might be good a idea to include the most interesting customer quote here. It helps with emphasizing your points"

<!-- By reading this section, stakeholders should immediately understand:

- the reason for conducting the test,
- when it was conducted,
- how it was conducted,
- key takeaways from the test.-->

After analyzing basic use cases of the {name of the product} it came to our attention that further analysis and a test with real users is crucial for improving the overall usability.

After thorough preparation of test scripts (described in the section "Scenarios & Tasks", we've gathered a group of {number of participants}, who use a current version of the service

Usability Test Report

Researchers

<!-- Your time to shine. Provide biographies for you and your team. Focus only on the most important parts of your career. -->

<!-- If you're working in a small team and you're well known – delete this section. In this case it wouldn't provide informational value, besides bragging. That could affect negatively reception of this report -->

The test was prepared and conducted by:

Marcin Treder
UX Designer, founder of UXPin with +7 years of experience in the field of user experience design and usability testing.

Photo credit: Free Usability Testing Kit

As a minimum, test between every iteration. For example, if you just finished a lo-fi prototype and you're about to start a mockup, conduct a quick test first. You may need to tweak some functionality issues that will affect the site's visuals.

Testing documentation can come in the form of plans, the tests themselves, and the results. Share everything among the team, so a standardized form for these documents will streamline the process (you can find some in our free usability testing kit).

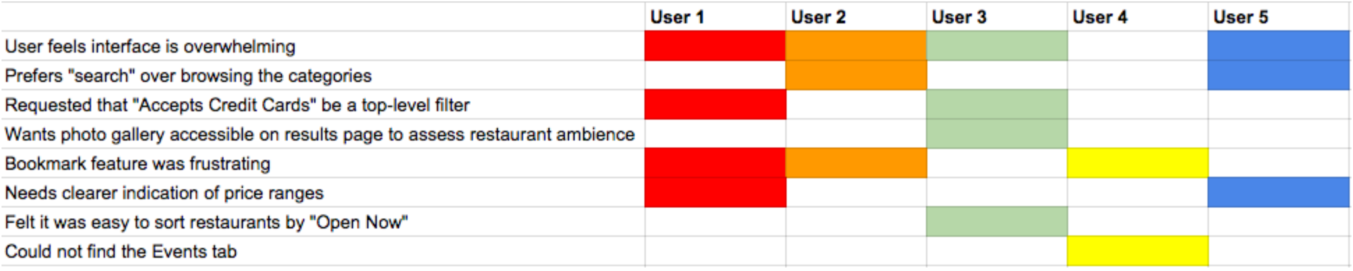
6. Documentation

- **Usability test plans** – These outline your test goals and procedural details such as the location and time, or even the specific questions/tasks. These are especially useful for stakeholders to understand what's being tested and why.

- **User tasks** – You need to describe to users the exact tasks you wish for them to perform. Be specific, avoid jargon, and don't provide too many details on the steps needed to accomplish the task (that's for the users to show you!).

- **Usability test script** – If you're moderating the test, you definitely need a script to ensure consistency.

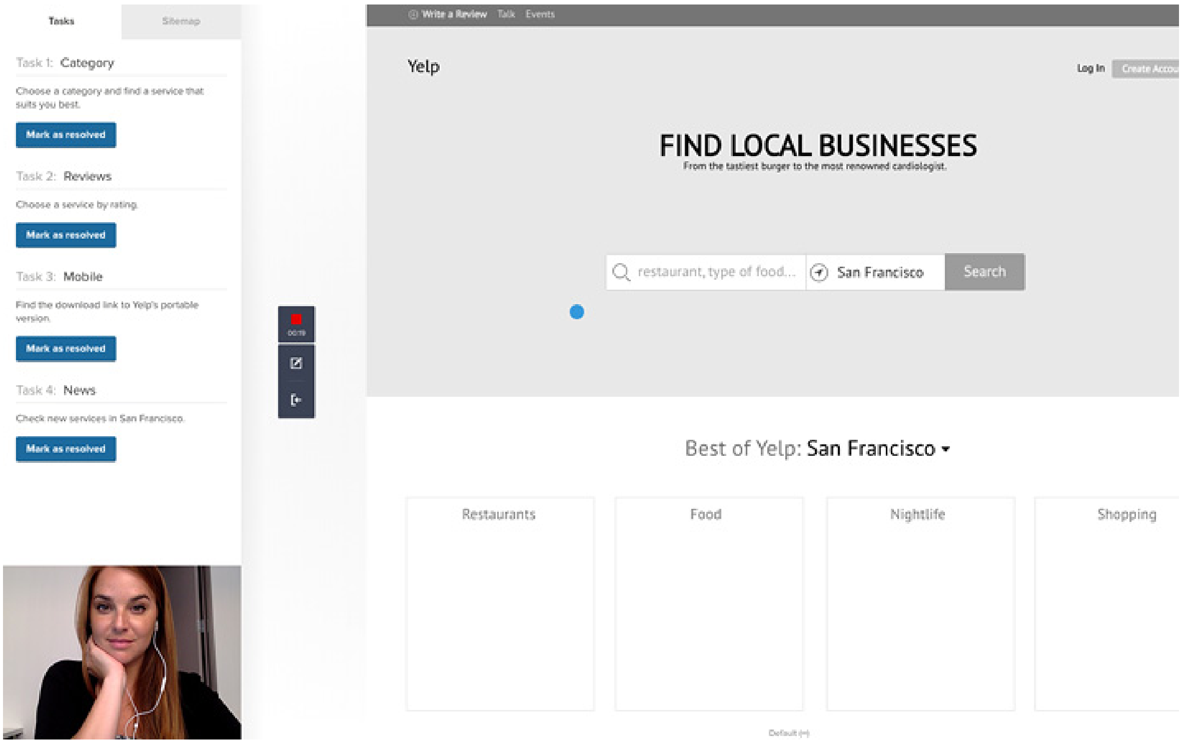
- **Usability reports** – Once you have the results of the testing, you'll need to make the data comprehensive to members of various departments who may not have your specific knowledge. The presentation of the data is crucial to ensuring that everyone interprets it correctly.



	User 1	User 2	User 3	User 4	User 5
User feels interface is overwhelming	🟥	🟧	🟩		🟦
Prefers "search" over browsing the categories		🟧			🟦
Requested that "Accepts Credit Cards" be a top-level filter	🟥		🟩		
Wants photo gallery accessible on results page to assess restaurant ambience			🟩		
Bookmark feature was frustrating	🟥	🟧		🟨	
Needs clearer indication of price ranges	🟥				🟦
Felt it was easy to sort restaurants by "Open Now"			🟩		
Could not find the Events tab				🟨	

Photo credit: UXPin for Yelp design usability testing based on exercise suggested by Tomer Sharon

If you're interested, UXPin actually features built-in usability testing. Write the tasks, invite your user, then moderate the session using your test script. The session records the user's clicks, audio, and facial reactions so you can send the clip to stakeholders afterwards.



Tasks Sitemap

Task 1: Category

Choose a category and find a service that suits you best.

Mark as resolved

Task 2: Reviews

Choose a service by rating.

Mark as resolved

Task 3: Mobile

Find the download link to Yelp's portable version.

Mark as resolved

Task 4: News

Check new services in San Francisco.

Mark as resolved

Takeaway

Any documentation that doesn't make the process easier is wasteful, and likewise any documentation that helps, no matter have frivolous it seems, is not a waste.

Documentation should come about naturally, as a result of necessity or best practices. Staples like wireframes, mockups, and prototypes are the most common, but less popular documents like customer journey maps or test plans can, when done properly, be just as useful.

As a basic rule of thumb, if you're writing up something just to hand it off to someone else, it probably wasn't worth your time in the first place. Only create documentation that moves design forward.

Informing Initial UX Requirements

If we were all mind readers, UX documentation wouldn't be necessary. We wouldn't need to decipher what our users want, and we could understand our teammates' ideas clearly. But, sadly, we're not mind readers, so we do the next best thing: make records about the minds of our users and our teams.

Photo credit: "Research." Neil Conway. Creative Commons.

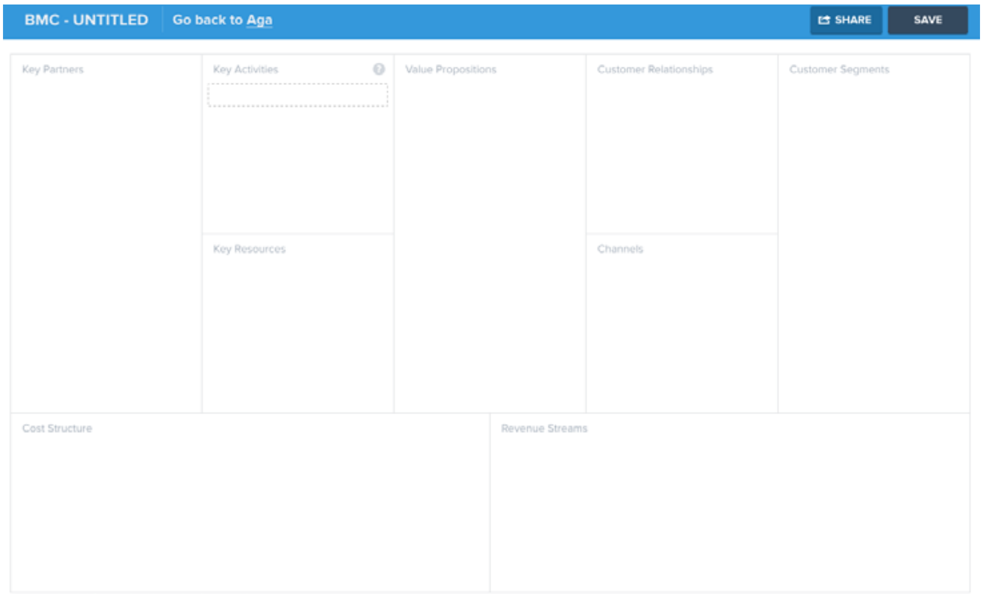
As we escape of the deliverables business, we become susceptible to the poor planning. In order to continue designing great products without the paper trail, we must make sure to create the *right* documents, instead of *more* documents – especially in the early stages. This chapter will explore the documents that help us build momentum without weighing down the project in busywork.

Early Requirements

As we mentioned in the last chapter, the first stage of the process should be research, which is itself divided into collecting data and organizing it. This chapter deals with how to collect data.

But even before that, you must know *why* you're collecting data. Every UX project has its own unique requirements, and research is used to determine what they are. We can divide the requirements into three categories: business, user, and technical.

1. **Business Requirements**
   What the product hopes to accomplish for the company/sponsors. Startups and smaller companies might be able to summarize these in a business model canvas available in UXPin.



Business Model Canvas by Alexander Osterwalder.

   Larger companies will usually have more requirements, sometimes completed by business analysts or product managers. In either case, these requirements can be better understood through stakeholder interviews.

2. **User Requirements**

   What the user wants is central to design thinking, so these requirements should have an enormous influence on the UX design. While using your own empathy will help, you shouldn't guess at user requirements – these should be gathered through testing like user interviews, surveys, field or diary studies, etc.

3. **Technical Requirements**

   The technical requirements can be separated into functional and nonfunctional categories. The functional specifications outline what the product should be able to do, such as authentication and administrative functions, and can be organized into a specs document like this one from Product Hunt. Nonfunctional specifications describe how well the product performs, such as usability, performance, data integrity, and maintenance.

The entirety of the design process will be shaped around these requirements, so the better you understand them from the onset, the more precise your final product will be in satisfying them.

To help inspire critical thinking about your goals for a project, Michael Margolis posted these questions to ask before starting user research.

Stakeholder Interviews

One of the best first steps to any UX project is to interview your stakeholders. They are the ones who define the basics of the project, especially the criteria for success. Kevin Hoffman of Goodkickoffmeetings.com even suggest doing this before the initial kickoff meeting so that you have a better "behind the scenes" understanding by the time everyone comes together.

Photo credit: *UXPin*

1. **Purpose**

   The point of stakeholder interviews is mainly to be clear on the business and technical requirements of a project. Getting a straightforward answer on these early on will set you in the right direction going forward, and making a record of these answers will give you ground to stand on later in defending some design choices.

   Additionally, stakeholder interviews make the stakeholders feel heard, validating that their opinions matter. You can't deny the reality of politics in any design process.

2. Best Practices

If your company has a Product Requirements Document, jotting down notes from this is a good starting point before composing the questions.

The questions you ask will vary depending on the type of stakeholder you're speaking with. As a starting point, we recommend following the templates of Kim Goodwin from her book *Designing for the Digital Age*, with excerpts published on Boxes and Arrows.

She breaks up the types of questions based department:

- **Marketing Stakeholders** – brand promoters looking for new, viable markets.
- **Engineering Stakeholders** – design engineer, system architect, or GUI, if applicable. For hardware, heads of manufacturing and electrical or mechanical engineering.
- **Sales Stakeholders** – in addition to a marketing head, it's advisable to talk to a sales head, as the two often have different goals.
- **Executives and SMEs** – the decision-makers with authority over the different branches.

Before beginning, ask the rest of the design team for their input. They have their own questions that they want answered, too. We suggest opening a collaborative document like a shared folder in Google Docs that everyone can add to on their own.

While conducting the interview, don't feel you need to be "all business." Creating a casual atmosphere and friendly rapport will produce better results, not to mention make the experience more enjoyable. It will also help when the time comes to ask the risky questions about their fears and personal opinions concerning the project.

Pay particular attention to the following:

- Assumptions
- Lists of requirements
- Prescriptive solutions

When these are brought up, probe with follow-up questions. These areas tend to be complicated, so extra effort is needed to clarify them.

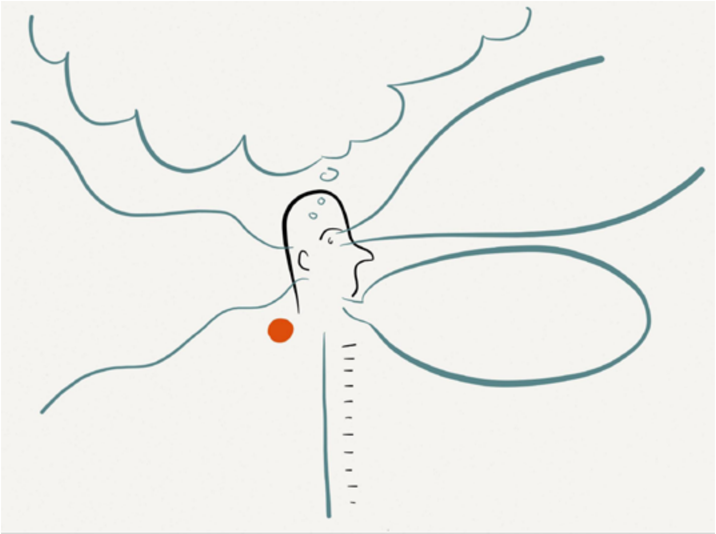


Photo credit: "Sketchy Empathy Map Template". Zak Greant. Creative Commons 2.0.

At the end of the interview, Chris Cashdollar (VP of Design at Happycog) suggests asking them if they'd like to mention anything.

Beyond just being polite, this opens up the door to any unanticipated information that could benefit the project.

Chances are there will be some areas of contradiction, whether in their own statements, or between different stakeholders. Be sure to clarify these in the moment, before they cause even bigger problems later on in the design process.

In terms of time, we've found that 45-60 minutes is usually sufficient. If you find this isn't enough time to answer all your questions, it's better to schedule a followup meeting than to exhaust someone with a two- or three-hour interview.

Recording the interview is critical – it's what makes this documentation, and safeguards against misinterpreting the data. The recording can then be shared with other team-members who weren't there, and can be referenced later to settle disputes.

Don't forget to ask the stakeholder to provide you with any relevant background documents that could help with the project. What the stakeholder sends over will also indicate where their priorities lie.

User Interviews

To understand the all-important user requirements, you need to understand the users themselves. While there are many different methods for connecting with your target audience, simply sitting down and talking with them is one of the most reliable.

1. Purpose

The main purpose is to narrow your design focus by solidifying what your users want. The responses from your user interviews will reveal which design ideas will work, which won't, and may even inspire new ones.

The data collected from user interviews will feed directly into the creation of personas, which dictate other documents like user scenarios and journey maps. For this, interviews should be as thorough as possible.

As described in *The Guide to Usability Testing*, user interviews also validate the other areas of research. Here you can either confirm or deny the problems and potential solutions that arose from stakeholder interviews or brainstorming sessions.

Photo credit: "2014-04-30 17.09.22." Nicholas Wang. Creative Commons.

2. Best Practices

Before the interview, we suggest creating an overview document (again, Google Docs) describing the event. This is the easiest way to keep your team informed and elicit their input – just like with stakeholder meetings, your other teammates will have their own questions and concerns they want brought up. Furthermore, it will keep stakeholders feeling in the loop.

The overview document should contain the following:

- **Goal** – What you hope to gain from the interview (keep it concise).
- **User Background** – The user's demographic information: job title, gender, age, web experience, etc.
- **Behavioral Questions** – Your intended questions about the user's particular habits and processes.
- **Product Questions** – Questions specific to a preexisting, related, or theoretical product, such as why they (would) use it or how it might be improved.

Every project will have different goals, which means the actual questions should be different. However, there are some general guidelines that can apply to all situations.

- Instead of only asking what users like and dislike, ask what they do, how they do it, and what frustrates them. Asking open questions, such as "why?" questions, give the user the opportunity to expand on their thoughts and feelings and yield data that simplistic "yes or no" questions cut short. An appropriate amount of silence on the part of the interviewer will also gently encourage the user to elaborate.

- Remember that the answers your user gives are merely opinions, so don't necessarily accept what they say as gospel. Cross-reference it with other interviews and studies first.
- Just as with the stakeholder interviews, keep it casual and light. Since the user is likely a stranger, helping them relax will give you better, more honest responses.
- Also like with the stakeholders, make a record of the interview for documentation to share with the team and as a reference later on.

Photo credit: "Expert Interview - Ruth Benny." *Nicholas Wang*. *Creative Commons 2.0.*

We suggest going a step further with collaboration than just sharing the list of questions. Try inviting along developers, product managers, or marketers so they can ask their questions in their own words. While you don't want to gang-up on the user, a small team is acceptable – just be sure to select a leader beforehand, so you don't confuse the user.

As for the location, difference options will suit different purposes. Conducting interviews in your user's natural environment (contextual interview) will put them more at ease, make it easier to schedule, and provide additional observational insights (such as distractions in their work or home, and clues to their personality). However, these require more effort and time on the interviewer's part.

If resources are an issue, you can even conduct interviews remotely with Skype or Google Hangouts, which makes it easier to record.

For more practical advice on user interview best practices, check out these resources:

- Interviewing Humans by Erika Hall
- My Best Advice for Conducting User Interviews by Whitney Hess

User Surveys

You can think of user surveys as "user interviews light" – if you're short on time or resources, they're the next best option for understanding your users. While their data is not as in-depth or reliable as interviews, they can still shed light on your customer base.

Do you like this question?
YES
NO
I DON'T KNOW
POSSIBLY
SPOILED VOTE
Carried out by some guys on the street on behalf of The Internation Survey Group.
Over 7 billion people were surveyed from October 2010 - October 7010

Photo credit: "survey." Sean MacEntee. Creative Commons.

1. **Purpose**

   In addition to a lighter version of the benefits of user interviews, surveys provide a few special advantages all their own.

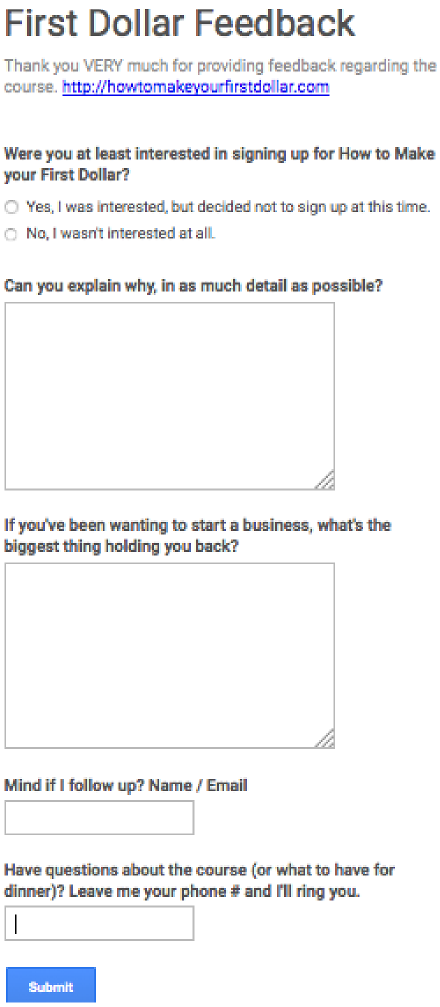
   As we mentioned last chapter, surveys take a quantitative approach to qualitative data. This allows you to recognize patterns more quickly. They also come readymade for graphs and statistical analysis, which create helpful documents to pass around.

   Surveys can also reach a wider range of people since they are cheaper and less intrusive. You can even choose from several testing sites (download our free *Guide to Usability Testing* for a complete list) to target a specific set of criteria for whom you reach.

2. Best Practices

Surveys are generally low maintenance – your involvement is limited to writing the questions and analyzing the data. While surveys are a time-saving option, you should still be diligent when writing them to get the results you want.

- First, shorter is better, so only ask the questions you need to. Users will give better data to a 3-minute survey than a 20-minute one. Try creating a list of all potential questions, then editing them down to 5. If you have 8 questions, that's fine, but any more and you should start cutting.



First Dollar Feedback

Thank you VERY much for providing feedback regarding the course. http://howtomakeyourfirstdollar.com

Were you at least interested in signing up for How to Make your First Dollar?

○ Yes, I was interested, but decided not to sign up at this time.

○ No, I wasn't interested at all.

Can you explain why, in as much detail as possible?

If you've been wanting to start a business, what's the biggest thing holding you back?

Mind if I follow up? Name / Email

Have questions about the course (or what to have for dinner)? Leave me your phone # and I'll ring you.

Photo credit: First Dollar Survey

- Pay attention to phrasing. Closed questions (yes/no, multiple choice) will give you data that's easier to analyze, while open questions allow you to discover new information – however, Jonathan Kochis, partner at Treble Apps, advises to be highly selective with your open questions. Regardless, you want your questions phrased as succinctly as possible.
- Consider the order of your questions, with the most important first. If you'd like to follow up, mention this at the end, after they're more familiar with you.

For more concrete guidelines, check out these addition survey resources:

- 20 Tips for Writing Web Surveys by David Travis, CEO of Userfocus
- Writing Surveys that Work by Diane Loviglio, UX Researcher at Mozilla
- Sample Survey Template from SurveyMonkey

Competitive Analysis

The level of competitive analysis can vary depending on your needs, from hiring a heuristic researcher, to simply browsing around related sites one night. Either way, it helps to know what you're up against.

1. **Purpose**

   A competitive analysis helps flesh out your business and (non-functional) technical goals, if not inspire new ones. What is your competitor doing that works, and where is their room for improvement – knowing these will come in handy when designing.

   Understanding your competition will also help understand your users. If every other product you're up against has a certain feature, users will expect that on your product, too, for better or worse. What you discover can also provide material for your interview or survey questions, such as how they felt about this or that aspect.

2. **Best Practices**

   If you want to take an analytical approach, we suggest this four-step method from our free ebook *Principles of Visual Consistency*:

   1. **Determine which fields to evaluate** – Common areas are visual impact, navigation, or clarity of copy – but customize your approach to the project (Leigh Howells' spiderweb heuristic above is a great starting point).

2. **Evaluate** – Try to get 5 people to critique your competitors based on the selected fields, but in a pinch you can conduct the evaluation yourself.

3. **Diagram the results** – Plot out the results in a way that's easy to visualize. Michael Hawley recommends the spider-web graph.

4. **Compare the results** – Compare your competitors results with each other, or with your preexisting product. What areas do your competitors do similarly? Where is the most effective area to break-in to the market?

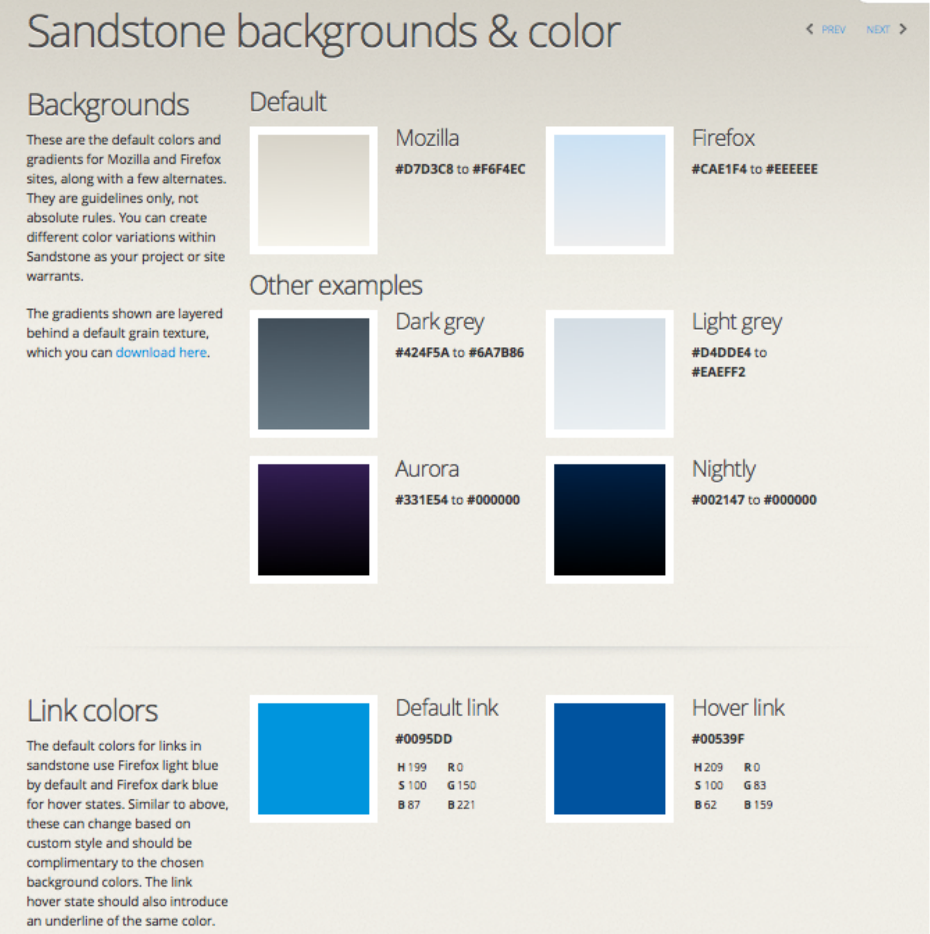
Note the consistencies and inconsistencies in the market, so you know when to follow suit and when to break the mold.

Reference Documentation

Some UX documents collect and officialize data for reference. These documents are optional, but recommended, meaning that if you're trying to streamline the process as much as possible, you can cut them out, but if you have the time, they'll prove valuable.

1. **Specs Document** – lists concretely the (functional) technical requirements. It's a useful reference for the entire team to have throughout the process, especially during brainstorming and stakeholder interviews. Check out this sample from Product Hunt.

2. **Style Guide** – organizes all the project-wide choices for areas like color, typography, brand usage, layout, etc., and can even include bits of code. Download the free *Critical Components of Web UI Style Guides* for best practices.



Sandstone backgrounds & color

< PREV NEXT >

Backgrounds

These are the default colors and gradients for Mozilla and Firefox sites, along with a few alternates. They are guidelines only, not absolute rules. You can create different color variations within Sandstone as your project or site warrants.

The gradients shown are layered behind a default grain texture, which you can download here.

Default

Mozilla
#D7D3C8 to #F6F4EC

Firefox
#CAE1F4 to #EEEEEE

Other examples

Dark grey
#424F5A to #6A7B86

Light grey
#D4DDE4 to #EAEFF2

Aurora
#331E54 to #000000

Nightly
#002147 to #000000

Link colors

The default colors for links in sandstone use Firefox light blue by default and Firefox dark blue for hover states. Similar to above, these can change based on custom style and should be complimentary to the chosen background colors. The link hover state should also introduce an underline of the same color.

Default link
#0095DD
H 199 R 0
S 100 G 150
B 87 B 221

Hover link
#00539F
H 209 R 0
S 100 G 83
B 62 B 159

Photo credit: Mozilla

3. **Mood Boards** – Showcases more abstract mood of the product, usually through collages, and sometimes with minor style guide elements. Great for communicating a consistent product atmosphere during the early stages.

4. **Change Log** – Somewhat of a relic from the waterfall days, but occasionally useful for settling disputes or general record-keeping.

Since the less documentation the better, ask yourself whether or not these will actually help you generate better designs before committing to any of them.

Takeaway

There is no shortage of methods for collecting this pre-design data, and each one brings something special and potentially useful. Card sorting, focus groups, participatory design – these all lend their own type of insight to the process. The methods we mentioned above, however, are just the most essential. They are the tried and true methods that any company can implement.

Now that you have all your data, what do you do with it?

In the next chapter we'll show you how to map out your user information.

Mapping Out Your Users

No matter what process or documentation you use, one guideline always remains relevant: design for the user.

Photo credit: "Anna." Stefano Montagner. Creative Commons.

Understanding – even empathizing – with the user is the key to designing a product that they will choose, even in a flooded market. But it's no coincidence that one of the most important feats of design is also one of the most difficult. That's what the documents in this chapter come in – to help you understand the people you're designing for.

User Personas

User personas are perhaps the most important document you'll create for analyzing users. They are the foundation for the rest of user documentation, which expands on personas for deeper insights. Under the right circumstances you can get away with skipping some of the other user documents, but skipping personas is not at all recommended.

1. Purpose

Personas are the most important person in the room when making design decisions. The psychology, behaviors, and demographics of your target users feed into these fictional identities.

Personas make the data more comprehensible by giving it a name and a face, instead of cold numbers and words without context. Designers use personas at every stage of the process and every important design question. Simply asking yourself "which options would this persona enjoy most" allows you to hone in on creating a product that will appeal most to a specific type of person.

This may seem lofty to designers who never used such "imaginary friends" before, but it's been proven that personas directly increase the success of the final product.

2. Best Practices

Personas can be as large or as small as you need them to be. The important concern is not how much information they contain, but that every piece of information is relevant and supported by product usage data.

Here’s the process we follow at UXPin to create more data-driven personas:

1. Examine usage data in our app, segmenting users based on overall engagement. E.g., people who started a trial but didn't buy, people who started a trial and bought, etc. Once we've defined the segments, we look at how they behave in the app based on events created in KISSMetrics.

2. To add a qualitative dimension, we interview ~30 users total from all segments to try to understand the "why" behind the data.

3. Based on quantitative data and interviews, we can start plotting out patterns that eventually form our user personas.

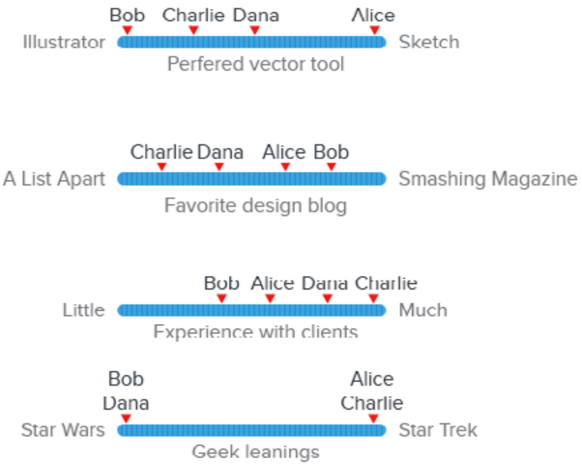
While the type of data on display can change, typically the best personas all have the following:

- **Photo & Name** – Even if just a stock image, a photo helps to think of your persona as a real person. Avoid celebrity photos and use a real name instead of “Pete the Power User”.

- **Demographics** – A good reference guide for the basic groups you’re targeting.

- **Personality** – While a detailed narrative gives the best description, at the very least include a few relevant traits like "spontaneous" or "lazy."
- **Technological Expertise** – This shows the level of complexity to make the system, or maybe how much explanation is needed for certain functions.
- **Platforms** – On which platforms your user will likely use your product.
- **Goals (Motivations)** – These can be divided into what your user hopes to accomplish with your product, as well as their life goals for understanding their character.
- **Personal Quotes/Mottos** – A personal quote is a shortcut to establishing the persona's character.

While detail is always good, don't get carried away. Your persona should only be as thorough as it is useful, and there will come a point where your time is better spent designing.



> Spectrum scales are the key to boiling down dozens of people into actionable patterns. Plot out 5-10 scales for different user characteristics based on people you've researched and interviewed. As patterns emerge across the different characteristics, you can see where multiple people can be represented by a single persona.
>
> The persona itself should be readily available to allow the team to check it (or share it) easily when the need arises. Share them with the team, print them out, and refer to them often.

User Stories

Once you have your personas set up, it's time to use them. A good first step is creating a user story. These are simple sentences or small paragraphs that elaborate on the user's motivations and goals.

User stories typically follow this format:

> *As a* [**type of user**], *I want* [**a feature**] *so that I can* [**complete a goal**].

For example, user stories for some UXPin customers would look like this:

> *As a marketer, I want to quickly offer feedback on designs so that I can return to my daily non-design work.*

As a UX designer, I want to consolidate feedback from all teams so that I don't waste time digging through email chains.

1. Purpose

User stories allow designers to hone in on the user goals and motivations, which can otherwise get lost with all the other information. As the most important element to consider during a design, your users' goals should be clearly defined. User stories relate them in a digestible snippet that's easier to comprehend than dry requirements, sometimes written from a company-centric perspective.

Tom Brinton also points out that user stories prevent "feature creep" – an unending cycle of adding extraneous features – by zeroing in on only the essentials.

2. Best Practices

User stories are quick and easy, but also insightful, all of which make them a great collaboration tool. Trying opening up participation to the entire team – perhaps in a Google spreadsheet like this – so that you document a broad range of company goals.

Don't forget that the value of user stories is their simplicity: they should be succinct and meaningful, and written in plain English. This also aids in collaboration across departments, since not everyone will know each other's jargon.

At the same time, be specific. Dive into your user research (especially interviews) to fill in the "So that…" section as accurately as possible. The hardest part about user stories isn't deciding the core tasks, but then understanding why.

User Scenarios

Now you have your personas and their goal. A user scenario explains the step-by-step process for getting there, including which pages they go to, where they click, and how long each stage takes.

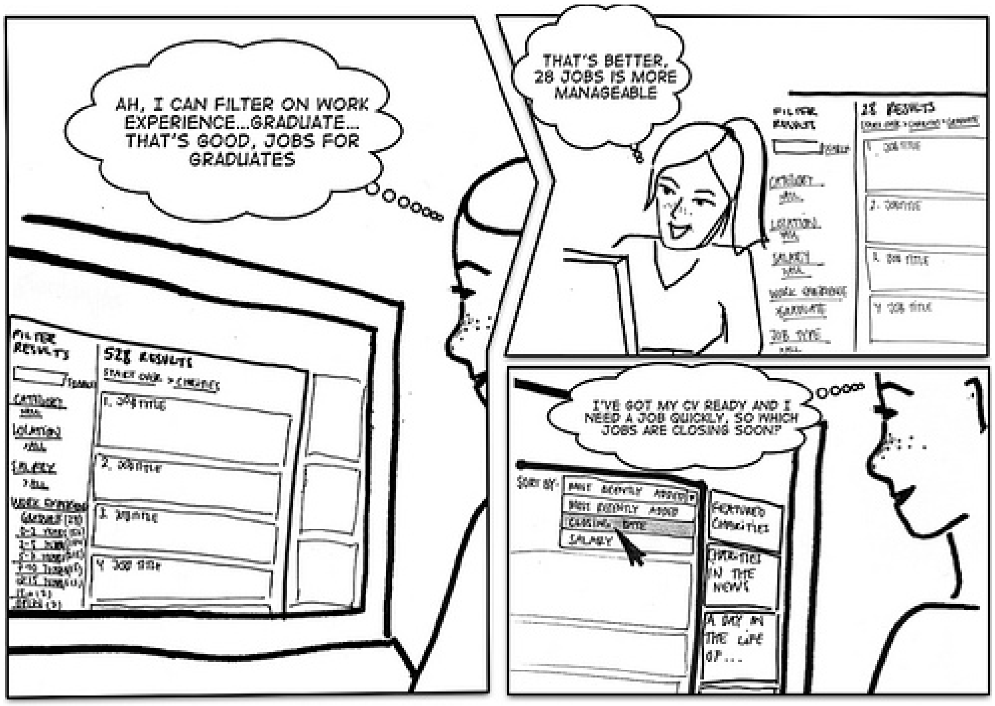
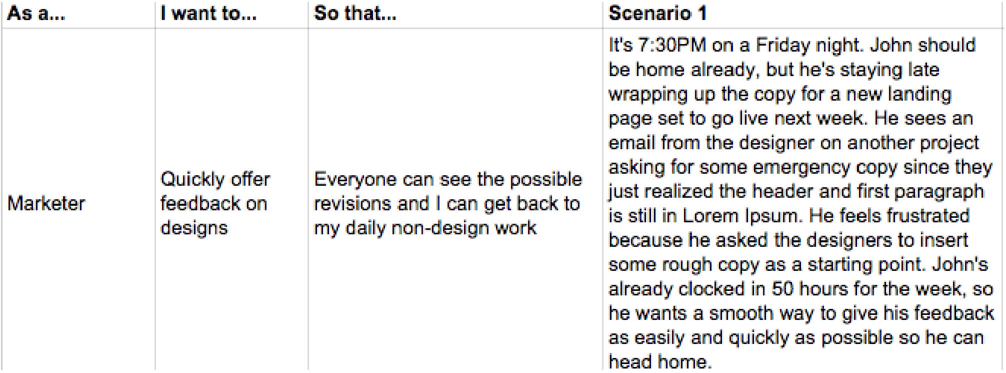

AH, I CAN FILTER ON WORK EXPERIENCE...GRADUATE... THAT'S GOOD, JOBS FOR GRADUATES
FILTER RESULTS
528 RESULTS
THAT'S BETTER, 28 JOBS IS MORE MANAGEABLE
28 RESULTS
I'VE GOT MY CV READY AND I NEED A JOB QUICKLY, SO WHICH JOBS ARE CLOSING SOON?
SORT BY:
CLOSING DATE
SALARY
FEATURED COMPANIES
COMPANIES IN THE NEWS
A DAY IN THE LIFE OF...

Photo credit: *Rosenfeld Media*. *Creative Commons*.

1. Purpose

User scenarios create the context behind the user stories.

What are the specific situations in which that user story might occur? You really start to explore the user's emotions during the process, helping to not just understand the user, but to empathize with them.



As a...	I want to...	So that...	Scenario 1
Marketer	Quickly offer feedback on designs	Everyone can see the possible revisions and I can get back to my daily non-design work	It's 7:30PM on a Friday night. John should be home already, but he's staying late wrapping up the copy for a new landing page set to go live next week. He sees an email from the designer on another project asking for some emergency copy since they just realized the header and first paragraph is still in Lorem Ipsum. He feels frustrated because he asked the designers to insert some rough copy as a starting point. John's already clocked in 50 hours for the week, so he wants a smooth way to give his feedback as easily and quickly as possible so he can head home.

In the above example, note the level of detail we add so that we understand the user's logical and emotional needs.

2. Best Practices

The format of the user scenario depends your needs: it could be a bulleted list with statistics listing the technical details ("5 seconds on the home page until he notices the log in link in the corner") or it could delve into emotional details, reading more like a narrative than a UX document ("eager to get started, his eyes frantically scan the home page until finding the log in link").

Whatever style you choose, keep these factors in mind when determining how your users act:

1. **Behavior** – Are there any existing user habits you must account for?
2. **Motivation** – How important is accomplishing their goal, and how will that impact their decision-making?
3. **Environment** – Where are they using your product? Work, home, and on the go all have different conditions and distractions. Moreover, what device are they using?

4. **External Factors** – These are miscellaneous factors that impact their usage, from their internet speed to time constraints.

It's quite normal to create multiple scenarios for each user story. If you already have a Google spreadsheet for the user stories, it's as simple as adding more columns.

As with the other documentation, details are good, but don't overdo it. There's no need to plot out the process for *every* task – handling the most common ones well will give you a good enough understanding to plan for the less common ones.

For more advice on user scenarios, including more examples, read this article by Sabina Idler, a UX researcher and consultant.

Customer Journey Maps

Personas, user stories, and user scenarios will give you enough understanding of your user within the context of your product, but you must create a customer journey map if you want to understand the entire brand experience.

Customer journey maps are the ultimate user document. They don't limit themselves to the beginning and end of a goal – they span the periods before and after the experience, so you account for all the lasting effects.

1. **Purpose**

   A good product of any type anticipates user wants before even the user knows, creating a satisfaction that can best be described as "magical" – that delightful feeling when a product you're using just intuitively "clicks." Customer journey maps help to isolate the opportunities in which you can design for such moments.

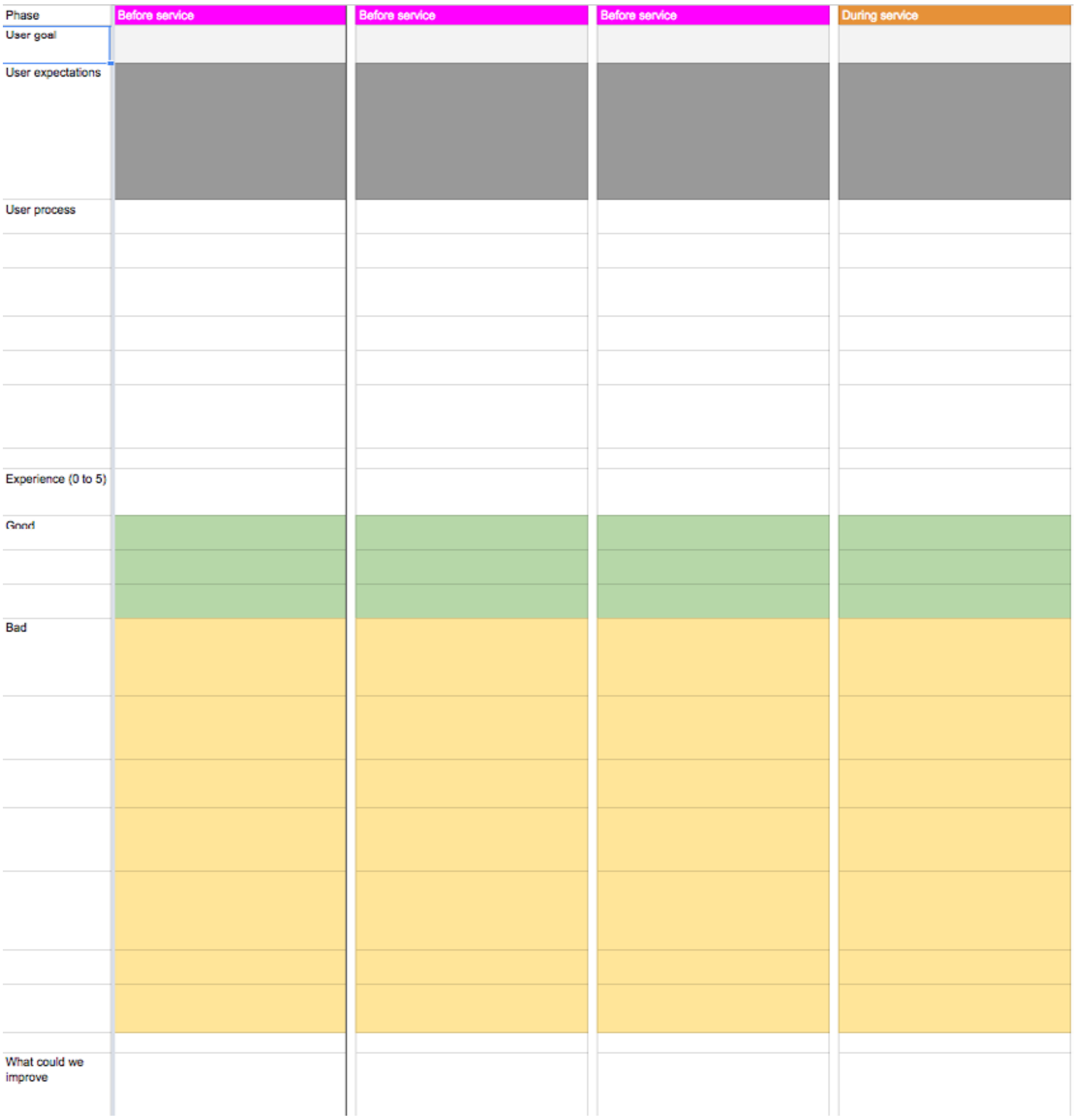
   For a real world example, Joyce Hostyn points us towards Domino's (slides 96-97). The pizza makers realized that, in the time surrounding their interaction with the product – not just interacting with the product itself – users experienced anxiety between the order and the delivery. They then implemented the Domino Tracker, which allows users to track the status of their delivery before it arrives.

   These emotional insights can make the difference between a product that's used, and one that's loved.

   Rian van der Mere praises customer journey maps for their aid in prioritization and keeping designers focused on content. As he points out, this document can be referenced with each new design idea to see if on track with the ultimate company goals, or something unnecessary.

2. Best Practices

At UXPin, we use the following customer journey map template:



Phase	Before service	Before service	Before service	During service
User goal				
User expectations				
User process				
Experience (0 to 5)				
Good				
Bad				
What could we improve				

Photo credit: UXPin

The basic setup breaks the entire process into the key steps, including the time before and after the interaction. For each step, the customer journey map addresses the users' states of mind in these areas:

- **Goal** – What do they hope to accomplish at this step?

- **Expectations** – How do they think this step will go? Outside influences and competitors' product have big impact on this.
- **Process** – How do they hope to accomplish their goal? Notice the difference between their chosen process and the best possible process.
- **Rating of Experience** – If the user had to rate this stage, what would they say?
- **The Good** – What does the user like about this stage?
- **The Bad** – What doesn't the user like about this stage?
- **Improvements** – Taking all other information into account, what can you change to improve your user's experience?

Adam Richardson suggests also addressing questions and barriers. Examining a design from a different perspective may reveal that problematic aspects of a design, such as areas of confusion and obstacles interfering with the user achieving their goals.

This article by Megan Grocki for UX Mastery further dissects the components of customer journey maps and provides more details into the theory and best practices behind them.

Takeaway

As we've mentioned before, the design process is rarely linear. In an ideal world without deadlines or other restraints, you could build all these documents at their leisure before even considering the design. But that's not always the case, and sometimes these documents can come "out of order."

We highly recommend creating and applying user documents early on, while you're still defining the product. However, they can still be revised, improved, or replaced later on in the process, especially if new data comes in. And if you're testing early and often like we suggest, this will be likely.

Design as Documentation

During the actual design process, documentation becomes more than just pieces of paper to pass around – they become the design itself.

For starters, they are concrete forms of previously abstract ideas. With every new iteration – every new document – your ideas become more and more tangible, up until the finished product. They also allow for easy backtracking. Each stage is recorded so you can revisit an earlier version, which comes in handy with constant usability testing proving what works and what doesn't.

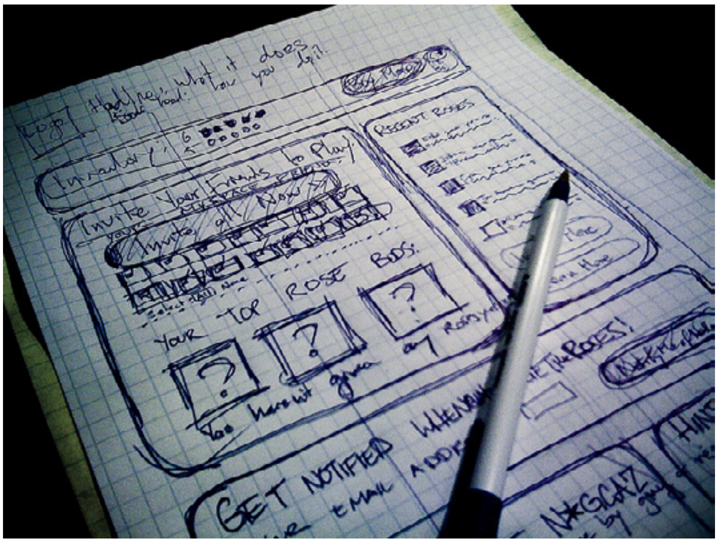


Photo credit: "war of the roses." jm3 on Flickr. Creative Commons.

Not to mention, design documents ease collaboration, both as physical representations for explaining ideas, and as sharable documents that can be edited across a team.

All of these documents bring something unique to the design process, but that doesn't mean every product design needs them all. Aside from wireframing and prototyping – which can benefit any project – the rest can be decided based on the project's individual needs. For example, a site map might not be necessary for a long-scrolling one-page site, but essential for a 100-page site with complex taxonomies.

Read our advice below to see which ones can help you most.

1. Sketches

There's no simpler way to document an idea than to simply draw it. In fact, this was how UXPin CEO Marcin Treder jump-started the (re)design process of the Yelp site as part of an exercise, which you can read in its entirety in our free *User Testing and Design: Improving Yelp's website*.

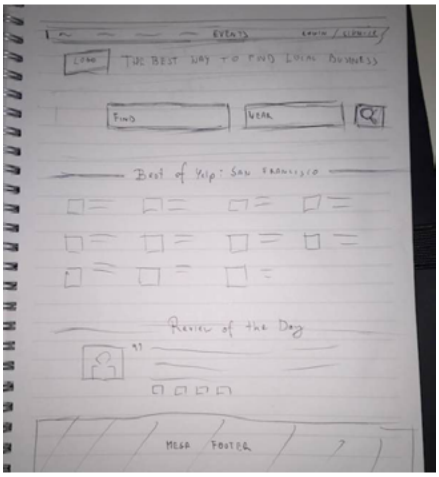

EVENTS
LOGIN / SIGNUP
LOGO
THE BEST WAY TO FIND LOCAL BUSINESS
FIND
NEAR
Best of Yelp: SAN FRANCISCO
Review of the Day
MEGA FOOTER

Photo credit: UXPin

Purpose

The basic simplicity of a pen-and-paper sketch is its strongest advantage. Sketches allow for quick and easy sharing of ideas since their rough nature invites honest feedback.

Because they're so easy and noncommittal, they're a great way to get the creative juices flowing at those crucial early stages when no other designs documents exist. Sketching is ideal for brainstorming sessions and design studios (see below), where ideas should be proposed, trashed, and recycled rapidly.

Moreover, anyone can make a sketch, so "less artistic" specialists like marketers or executives can still illustrate their suggestions with less fear of judgment.

Best practices

The important guideline for sketches is to not focus on how good they look. The point is quick and easy, so don't get bogged down on details that will likely be changed later.

In fact, during collaborative sketching sessions, rough sketches are actually *better*: they leave details open-ended, which helps generate more ideas from the team – this is known as "divergent thinking," which we describe in our *Design Collaboration in the Enterprise*.

In the same ebook, we also recommend design studio exercises as a way to generate ideas in the early phases. These are regulated design activities that facilitate both brainstorming and interdepartmental collaboration through sketches. For the specifics of the process, watch Todd Zaki Warfel describe his method.

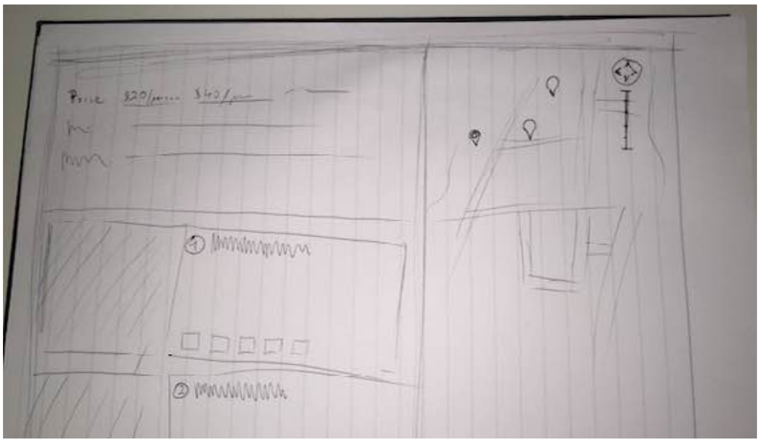


Photo credit:

UI designer Jake Rocheleau recommends what he calls "thumbnailing." This is simply sketching images as smaller than their actual size in order to speed things up and keep from getting sidetracked with the details. Focus less on the details and more on structural items like the navigation, column sizes, and column positions.

2. User Flows

User flows chart the actions a user takes with your product. While user scenarios focus on the context of use (e.g. the user's situation, mindset, etc.), user flows focus on how that user actually progresses through the design to complete tasks.

Starting with objectives and the common points of entry, you can map out how a user achieves their goals. For example, a user flow for an Amazon.com user looking to buy toasters will likely differ depending on how they enter the site:

Direct Traffic:

1. Enters through direct URL.
2. Types “toasters” in the top search bar.
3. Scrolls through results until finding “Cuisinart 2-Slice Toaster” and clicks on it.
4. Clicks on “Add to Cart.”

Organic Search Visitor:

- Searches for reviews of different toasters
- Enters Amazon.com.
- Uses search bar to find a listing of toasters
- Clicks on “Cuisinart 2-Slice Toaster” to read review.
- Returns to search listing
- Clicks on “Hamilton Beach 2-Slice Toaster” to read review.
- Returns to the Cuisinart product page
- Buys Cuisinart toaster.

We’ve greatly simplified the comparative online shopping experience here, but you start to see how just the point of entry defines great differences in user behavior. To account for that, you must invest in creating a thorough user flow.

Purpose

User flows help you improve the efficiency of your design. As you plan out the steps within a user path, you begin to understand how to resolve points of friction. They help you visualize your prototype without needing to jump into a design tool just yet.

Best practices

User flows revolve around objectives: both the users and your own.

User flows can be as detailed or simplistic as you want. Try a shorthand technique (shown below) to make things easier.

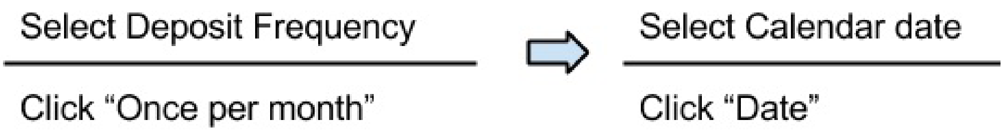


Photo credit: UXPin based on Ryan Singer's shorthand technique

You can also try the writing-first approach, which Jessica Downey writes about in her article Jumpstarting Your App Conception Without Sketching UI. This outlining method helps flesh out ideas and build a "common understanding" of each page of your app or site.

Let's create one for, say, a banking app. The scenario: someone wants to turn on auto deposit. Note in the outline below, content in [brackets] represents action buttons/links.

- **Step 1: Would you like to set up auto deposit?**
  [Set auto-deposit]

- **Step 2: Select Deposit Frequency**
  [Once per month][Twice per month]
  [Every other week][Every week]

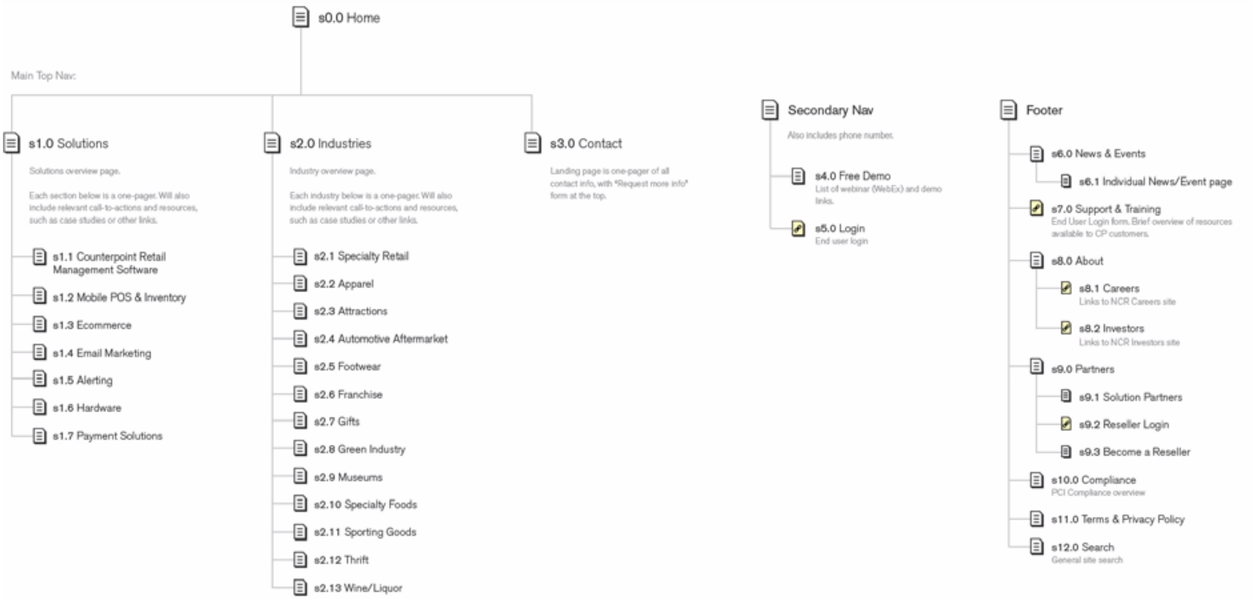
- **Step 3: Deposit Once per Month**
  [Select calendar day]

- **Step 4: Set Amount**
  Display amount field
  [Set auto-deposit]

Don’t forget to test your flows. Tree-testing is a good start for this – it tests a preset information architecture to determine if users can clearly navigate your product.

3. Site Maps

While sitemaps and user flows are related, the two are not interchangeable. As Dan Brown explains in Communicating Design, the differentiating factor is user behavior: site maps represent your content structure, while user flows show different ways of navigating through it to accomplish various tasks.

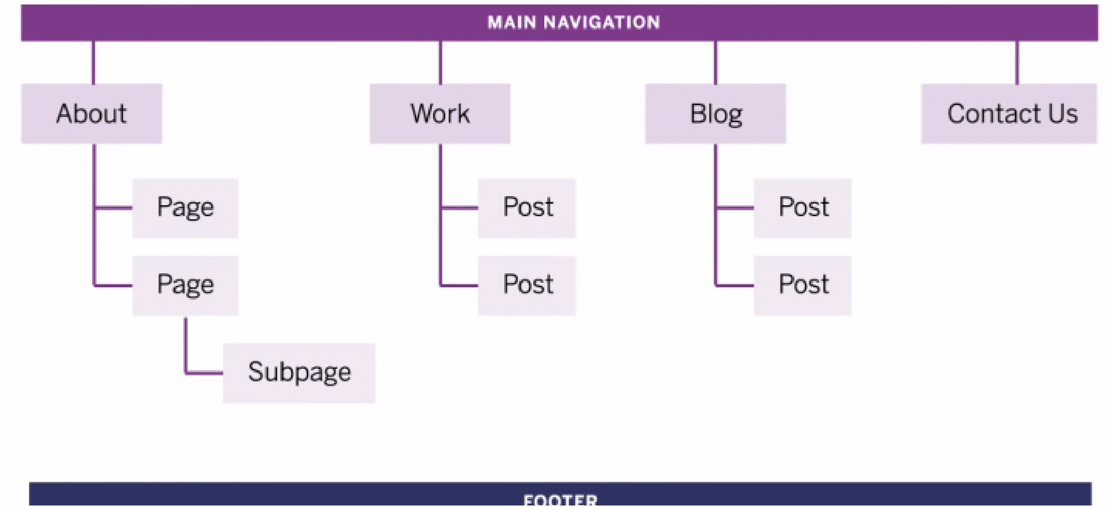

s0.0 Home
Main Top Nav:
s1.0 Solutions
Solutions overview page.
Each section below is a one-pager. Will also include relevant call-to-actions and resources, such as case studies or other links.
s1.1 Counterpoint Retail Management Software
s1.2 Mobile POS & Inventory
s1.3 Ecommerce
s1.4 Email Marketing
s1.5 Alerting
s1.6 Hardware
s1.7 Payment Solutions
s2.0 Industries
Industry overview page.
Each industry below is a one-pager. Will also include relevant call-to-actions and resources, such as case studies or other links.
s2.1 Specialty Retail
s2.2 Apparel
s2.3 Attractions
s2.4 Automotive Aftermarket
s2.5 Footwear
s2.6 Franchise
s2.7 Gifts
s2.8 Green Industry
s2.9 Museums
s2.10 Specialty Foods
s2.11 Sporting Goods
s2.12 Thrift
s2.13 Wine/Liquor
s3.0 Contact
Landing page is one-pager of all contact info, with "Request more info" form at the top.
Secondary Nav
Also includes phone number.
s4.0 Free Demo
List of webinar (WebEx) and demo links.
s5.0 Login
End user login
Footer
s6.0 News & Events
s6.1 Individual News/Event page
s7.0 Support & Training
End User Login form. Brief overview of resources available to CP customers.
s8.0 About
s8.1 Careers
Links to NCR Careers site
s8.2 Investors
Links to NCR Investors site
s9.0 Partners
s9.1 Solution Partners
s9.2 Reseller Login
s9.3 Become a Reseller
s10.0 Compliance
PCI Compliance overview
s11.0 Terms & Privacy Policy
s12.0 Search
General site search

Sitemaps live up to their name – they are the maps on which the user flows are plotted.

Purpose

The point of a sitemap is to improve a product's information architecture.

A site's IA should be as logical and self-evident as possible The larger the site, the more you need to create a sitemap so you can contain the complexity as much as possible.



Site maps in UXPin

Best practices

Traditionally, sitemaps are static. You want to show every single layer of the site (down to each subpage) with clear labels. The homepage sits near the top with the main navigation clearly illustrated. Every site is different, so there is no "magic number" of pages to include – that being said, always simplify when possible.

But because clients and stakeholders often question how people might progress through the site, we've found that clickable sitemaps are actually more helpful.

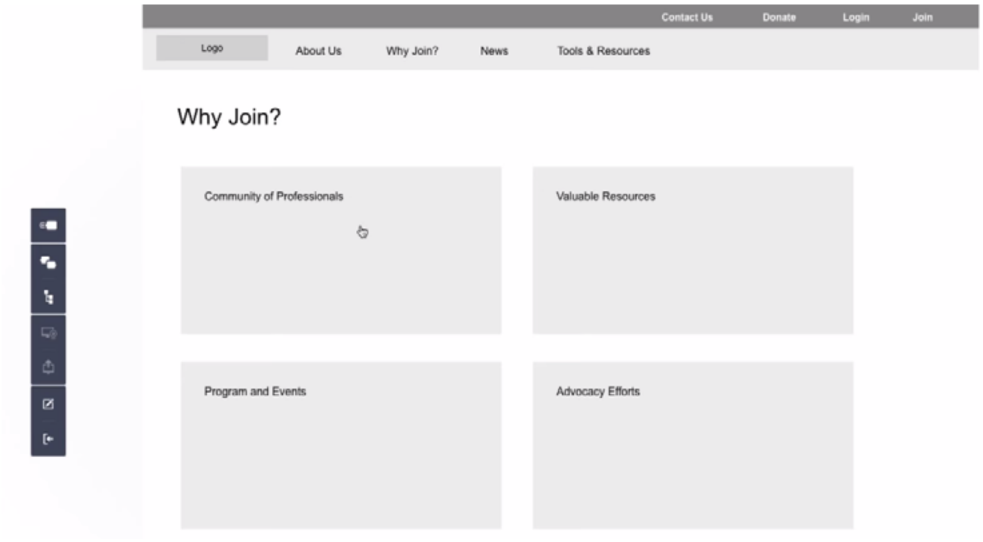


Photo credit: UXPin

The clickable sitemap can either supplement or replace the static site map entirely. Either way, you won't be doing any extra work since the clickable sitemap is essentially the "shell" of a low-fidelity prototype. Here's a quick process we found helpful when using UXPin:

- First create the primary navigation
- For each item in the primary navigation, create a new page
- On each page, draw boxes to represent all the clickable links. It's best to arrange them in a visual hierarchy that makes sense.
- For each link (like "Programs and Events" or "Advocacy Efforts" in the above screenshot), create a shell page.

Once you're done, you've created a rough outline for multiple user flows. As design agency Barrel shows in this video tutorial, clickable sitemaps are fast ways to help others understand how the design might work.

Of course, don't leave the users out of the process. Before you start sitemapping statically or interactively, make sure you conduct some card sorting. These simple test allows users to show you how they naturally categorize information so that your IA takes the path of least resistance.

4. Paper Prototyping

In the same vein as sketches, paper prototypes allow for quick validation and iteration. Paper prototyping has been around since long before digital products even existed, but their practicality makes them still useful today, though with some modern tweaks.

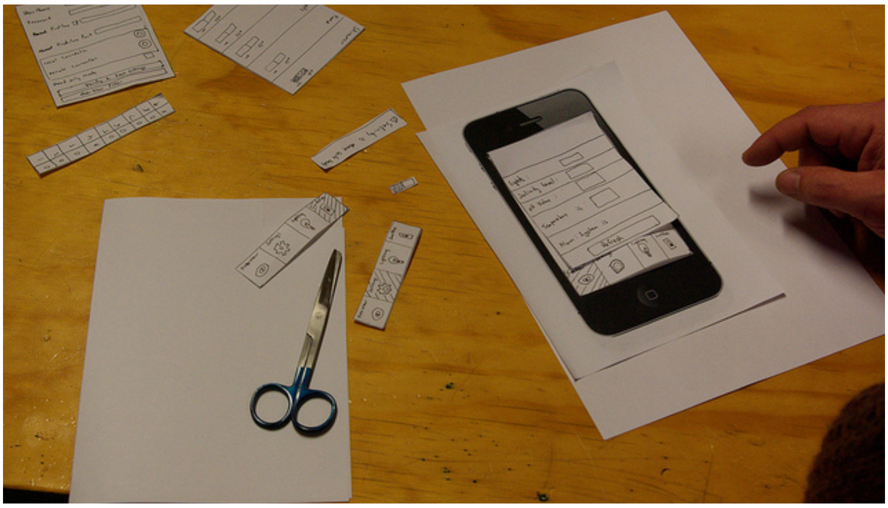


Photo credit: "Projects Paper-based Prototyping and Functional Testing Part." Samuel Mann. Creative Commons.

Purpose

Paper prototyping allows designers to test and/or share ideas without creating digital systems that might be scrapped later. Cheap to make and quick to create, paper prototypes are time-savers – it's easier to throw out a prototype that took 15 minutes than one that took 5 hours.

Paper prototypes can also be a good team-building activity. The crafts aspect can be as fun as it is useful, especially if conducted as a group. Moreover, paper prototypes fulfill the role of actual documentation well – past prototypes can be readily examined and annotated.

Paper prototypes are most applicable in the early stages of design. Their main drawbacks are their visual limitations: they're able to communicate basic ideas, but they certainly cannot accurately represent the product experience. With a paper prototype, the most helpful feedback will be on design efficiency – how clear and easy was it for users to accomplish their tasks? For later stages, more advanced prototypes are more helpful.

Of course, design is rarely a linear process. Even if you already have a hi-fi prototype, it's not uncommon to test a new idea with a paper prototype before incorporate it digitally.

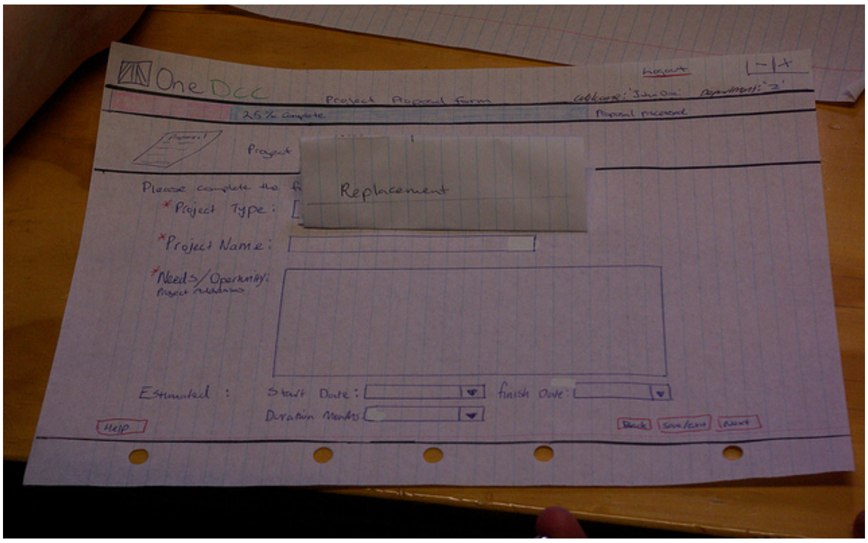


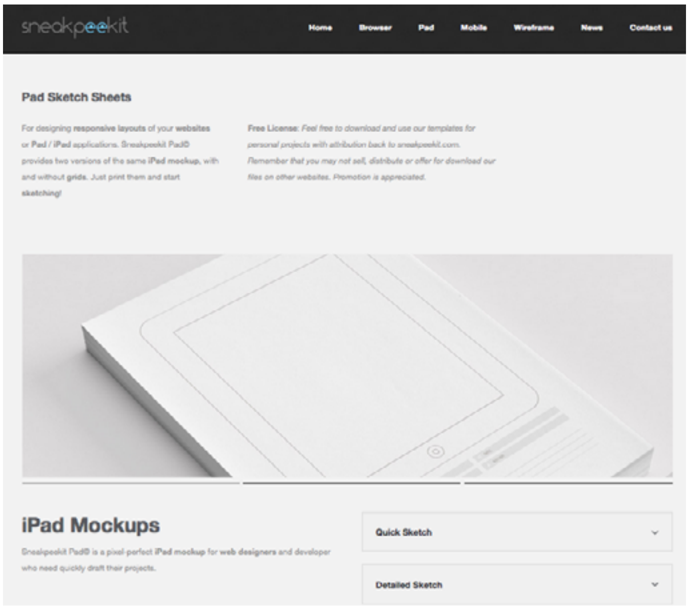
Photo credit: "Projects Paper-based Prototyping and Functional Testing Part." Samuel Mann. Creative Commons.

Best practices

Remember that, like sketches, simplicity is the main appeal. Don't get too involved in the details of the images or the mechanics of mimicking an interface.

… But don't make it too simple. Sketch out each page individually on different sheets of papers to make switching between them easier. When your prototype is ready, find a partner to act as the "human computer" who will operate the prototype. Walk them through the functionality, and remind them to not give suggestions or hints to users when you test the prototype.

Paper prototypes are great for "over-the-shoulder" sessions with stakeholders, but they might confuse people in a formal review. If you present a paper prototype, provide plenty of context by explaining which areas still need to be built, and the specific areas for which you'd like feedback.



Pad Sketch Sheets

For designing **responsive layouts** of your **websites** or **Pad / iPad** applications. Sneakpeekit Pad© provides two versions of the same **iPad mockup**, with and without **grids**. Just print them and start **sketching!**

Free License: *Feel free to download and use our templates for personal projects with attribution back to sneakpeekit.com. Remember that you may not sell, distribute or offer for download our files on other websites. Promotion is appreciated.*

iPad Mockups

Sneakpeekit Pad© is a pixel-perfect **iPad mockup for web designers** and developer who need quickly draft their projects.

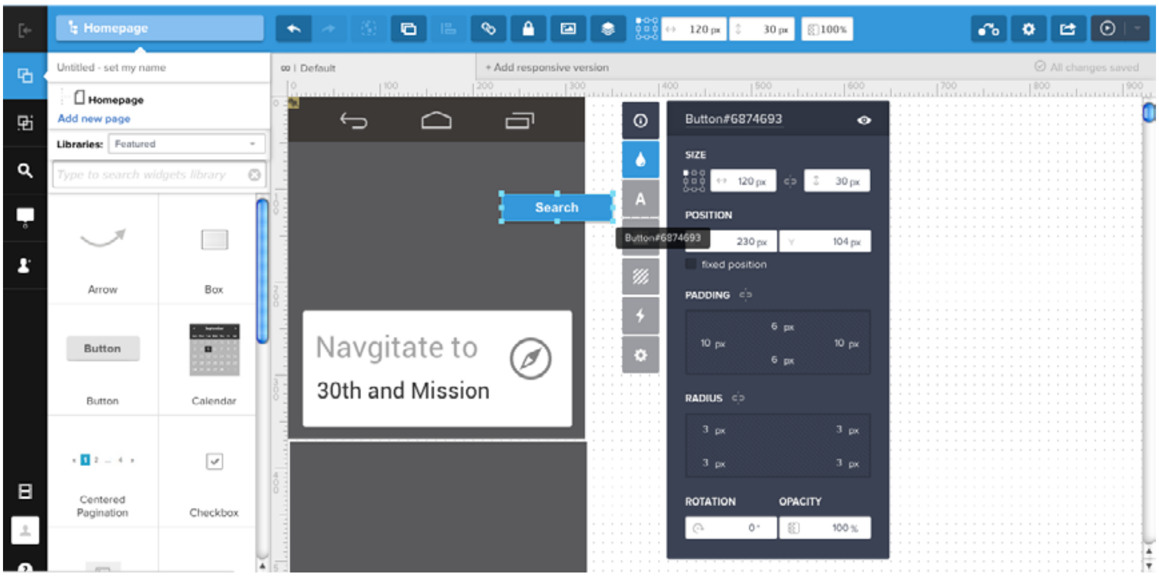
Photo credit: Sneakpeekit

Last, you don't need to build it from scratch. Sites like Sneakpeekit provide downloadable templates to get started. Sheets of paper with images of iPhones or iPads, along with extras like grids and notes, help keep everything organized.

5. Interactive Wireframe

In the past, wireframes acted as a bare-bones structure for content. They were traditionally static, with no interactivity or use beyond the basics. Some designers even treated them as specs documents, annotating details in the margins.

Unfortunately, static wireframes are dead-end deliverables since you need to completely rebuild your prototype.



Button#6874693

SIZE: 120 px, 30 px

POSITION: 230 px, Y 104 px

fixed position

PADDING: 6 px, 10 px, 10 px, 6 px

RADIUS: 3 px, 3 px, 3 px, 3 px

ROTATION: 0°

OPACITY: 100 %

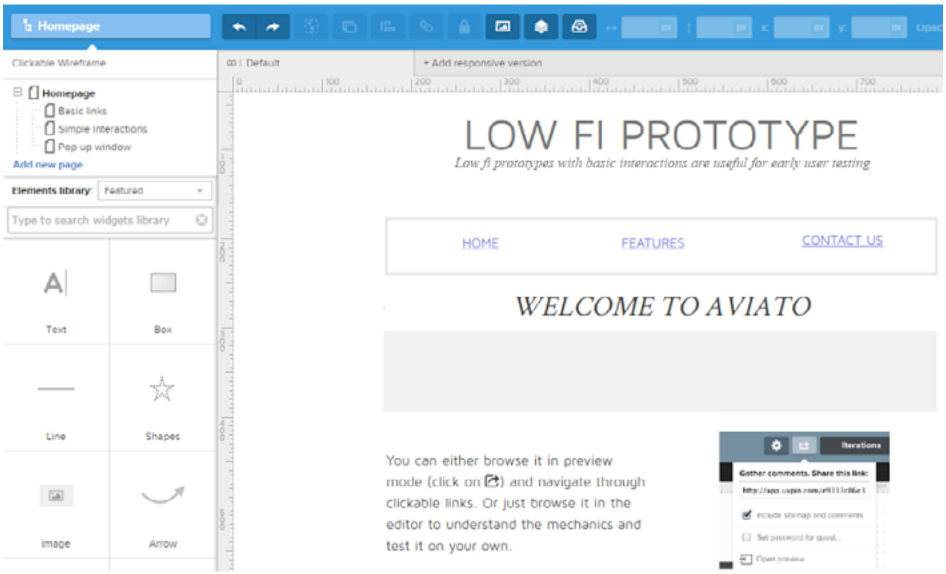
Photo credit: UXPin

However, as with the other documents, technology has enabled wireframing to evolve into something more useful. Today's wireframe allows for interactivity. This makes it less of a wireframe, and more a low-fidelity prototype.

Purpose

Interactive wireframes still retain their original purpose of solidifying the information architecture. Some designers may think wireframes are a waste of time, but we think that's a bit extreme. Interactive wireframes allow you to simultaneously test a site's content structure and page flow (the two most important ingredients of any design). They force you to focus on the foundation without being worried about the interior design.

Adding interactivity to these outlines makes them even more advantageous. For one, this permits early usability testing, which helps facilitate Lean and Agile UX processes. You always want to test your designs as early and often as possible.



LOW FI PROTOTYPE

Low fi prototypes with basic interactions are useful for early user testing

WELCOME TO AVIATO

You can either browse it in preview mode (click on ⧉) and navigate through clickable links. Or just browse it in the editor to understand the mechanics and test it on your own.

Gather comments. Share this link:

☑ Include sitemap and comments

□ Set password for guest...

Photo credit: *UXPin*

Furthermore, these lo-fi prototypes are quick and easy to make. This means that, during the early stages, designers can build and test multiple versions and keep the one that performs best. This is a great aid in rapid prototyping, which we'll explain below.

Additionally, interactive wireframes save time on building the wireframe itself. The interactivity speaks volumes about how a product's functionality works, saving you time spent annotating.

Best practices

Build your interactive wireframe with a prototyping tool that allows you to integrate it into later, more advanced designs.

Tools like UXPin allow designers to build on top of the initial wireframes instead of recreating them. Interactive wireframes evolve into higher fidelity prototypes, thus the work put into them is not lost. The documentation becomes the design, not just supplementary material.

With these advantages in mind, we can see how interactive wireframes are the natural offspring of the growing Lean UX and Agile UX methodologies, both of which favor less deliverables and more user-informed iteration.

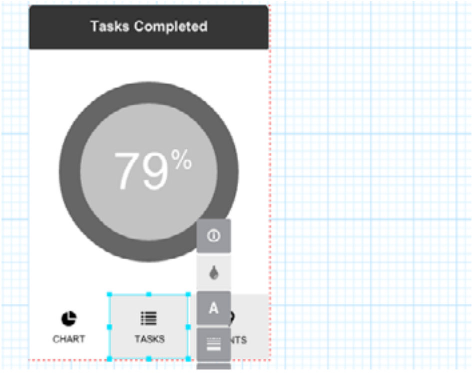


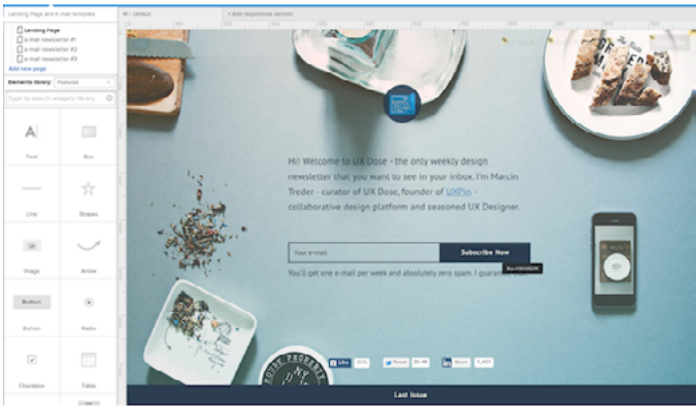
Photo credit: UXPin

- **Use common elements** – Interactive wireframes only need to be "good enough". Don't worry about unique visual design. The goal is to illustrate your ideas and flows as quickly as possible. Use element libraries as much as possible.

- **Embrace mobile-first design** – As recommended in the Guide to Interactive Wireframing, start with the smallest device first and then scale up. The alternative approach of "progressive degradation" makes mobile design an afterthought rather than a priority. To help facilitate the process, our app comes pre-loaded with multiple breakpoints.
- **Focus on light interactivity** – Interactive wireframes only need to be clickable. Leave the fancy animations for hi-fi prototyping. For now, just make sure you've linked all the important design elements to their respective pages. Focus on navigation items, calls to action, popup windows or modals, and alert or dialogue boxes.

6. Mockup

Mockups are a visual document that showcases what the product will look like. They are static documents for fine-tuning the visual design.



Hi! Welcome to UX Dose - the only weekly design newsletter that you want to see in your inbox. I'm Marcin Treder - curator of UX Dose, founder of UXPin - collaborative design platform and seasoned UX Designer.

You'll get one e-mail per week and absolutely zero spam. I guaran

Photo credit: *High-Fidelity Mockup*

Purpose

While wireframes focus on structure and prototypes focus on flow, mockups focus on fidelity.

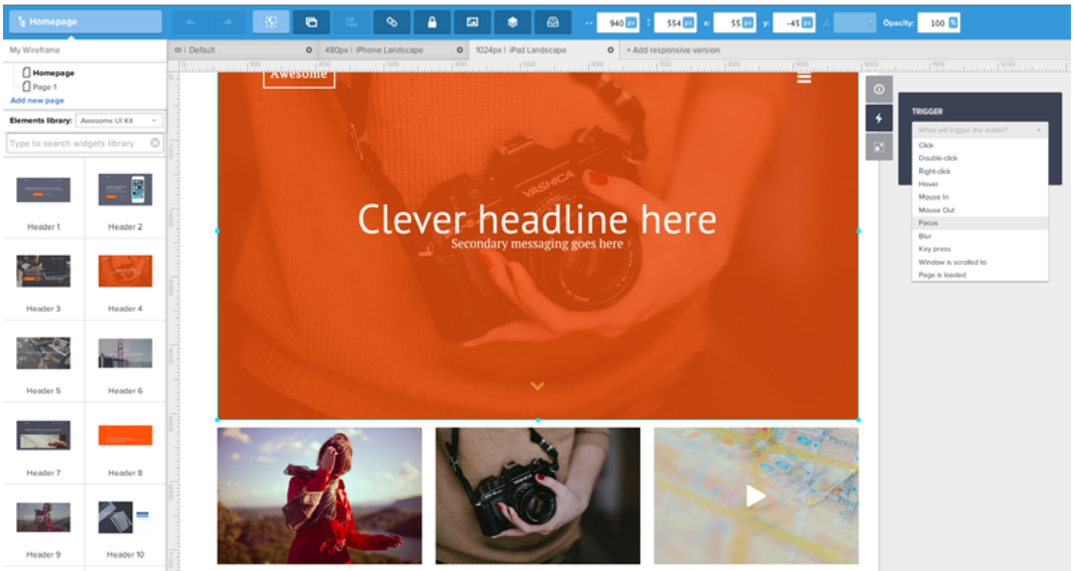


*Photo credit:*Guide to Mockups

Mockups act as visual specifications for developers The earlier the developer critiques the mockup, the better, so that designers don't waste time on plans that simply can't be done.

Best practices

As explained in *The Designer's Guide to Collaborating With Developers*, make sure you create mockups as collaborative docu.

- **Use proper naming conventions** – Use a clear organization system and straightforward names for folders, files, layers, icons, etc. Using a version control system, too, will keep things tidy and come in handy if you need to backtrack.

- **Use a grid system** – Grids maintain structure, enable pixel-perfect designs, and prevent needless tweaking.

- **Keep backup typefaces in mind** – Since typography is a large part of mockups and visuals in general, keep an eye-out for web-safe backups in case your first choice is rejected by developers.

- **Responsive design is mandatory** – Design separate mockups for different screen sizes to make everything crystal clear. We suggest a new mockup for every breakpoint.

Most designers prefer to use specialized graphic software, especially Photoshop and Sketch. Each other these programs, though, have their own best practices. Read the *Guide to Mockups* for best practices for both.

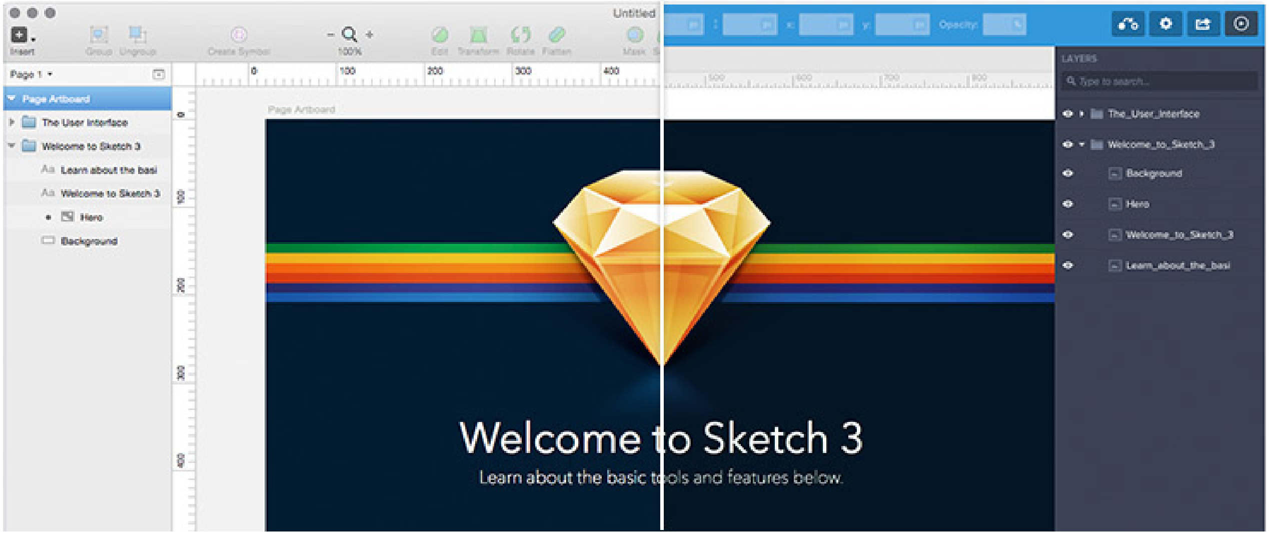
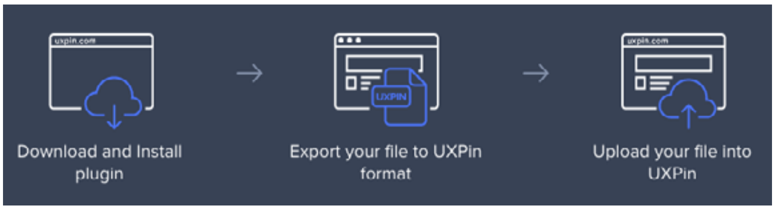


Photo credit: UXPin

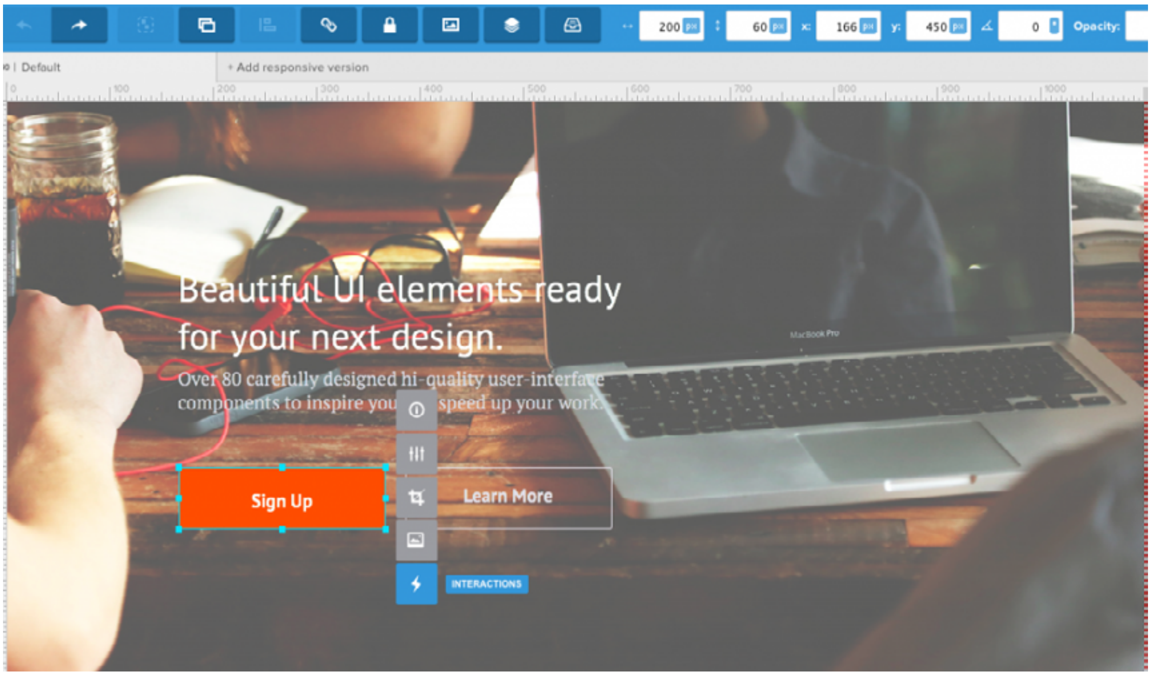
Whether you use Photoshop or Sketch, you'll still be able to import your mockup into UXPin for prototyping. As we said when discussing wireframes, the best documentation evolves with the process instead of being thrown away like a deliverable. We allow drag-and-drop prototyping from PS and Sketch, so your mockups don't become another dead-end document.


uxpin.com
Download and Install plugin
UXPIN
Export your file to UXPin format
uxpin.com
Upload your file into UXPin

7. **High-fidelity Prototype**

   The next best thing to final product itself, high-fidelity prototypes are the ultimate design document. As documents you can interact with, hi-fi prototypes (along with interactive wireframes, which are really just lo-fi prototypes) do more than just document, they allow testing and analysis on a level like no others.

   Ian Schoen even proposes having prototypes of varying fidelity replace all documentation. While we don't think it's as simple as that, the prototype certainly excels at being "living documentation". Instead of revising spec documents to keep up with the design, a prototype represents the latest iteration.


Beautiful UI elements ready for your next design.

Over 80 carefully designed hi-quality user-interface components to inspire you speed up your work.

Photo credit: UXPin

Purpose

Hi-fi prototypes are the most refined collaborative design tool. Developers understand the language of interactions so they can provide accurate feedback on the feasibility of the entire system. Non-design stakeholders can also play with your design as if it's a final product.

While a hi-fi prototype might require more design work, you end up saving time in development. Developers can misinterpret specs documents, but they're less likely to misunderstand if they can interact with a realistic representation.

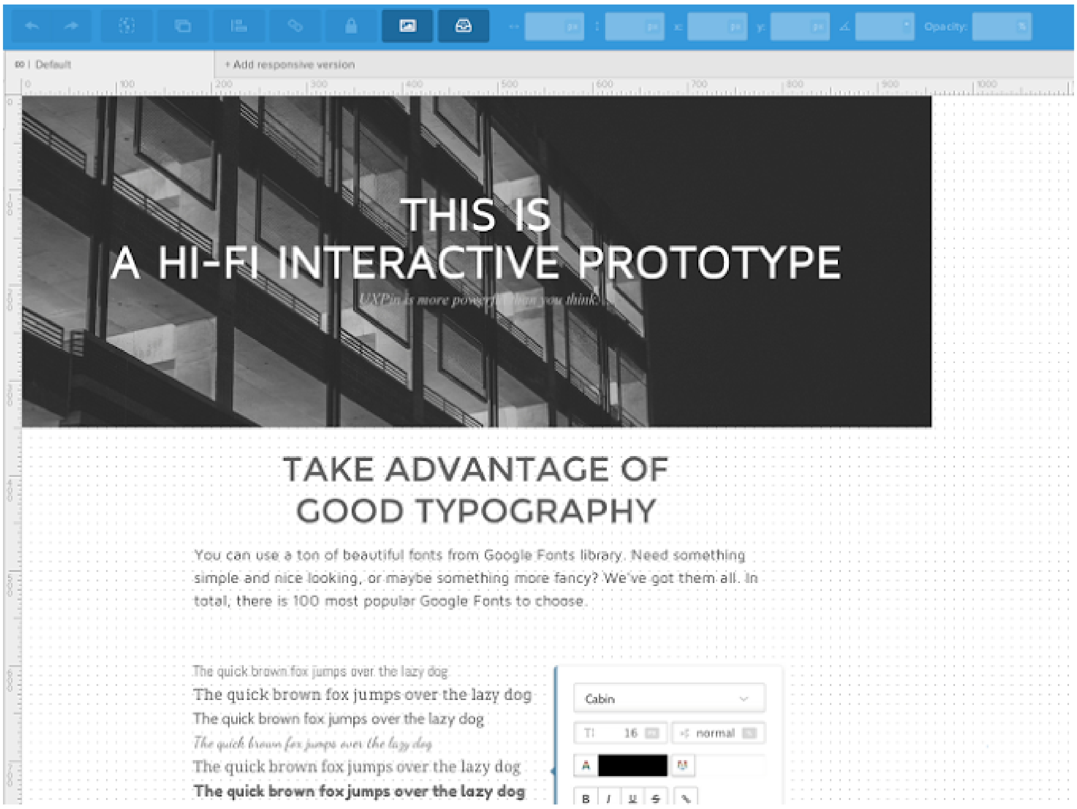
The learnings from testing hi-fi prototypes are incredibly valuable. If you find critical errors, you've just saved a ton of headaches (and money) versus fixing them in code. If you find the prototype performs well, the whole team now has a better vision of what the final product should look like.

Best practices

When building your prototype, keep usability testing in mind. That means, among other things, filling in as many gaps as possible to bring the prototype as close to the finished product as possible. Apply the same user-focused design thinking – targeting personas, interdepartmental collaboration, etc. – to your prototype for the best results.

1. **Avoid using placeholders such as lorem ipsum text or unrelated images** – While lorem ipsum text can be acceptable

for interactive wireframes, try to use as realistic content as possible. Rough is fine, but fake is not. If you are absolutely unable to use the final content for whatever reason, the next best choice is related content – such as text from an older version or competitor's product.



THIS IS
A HI-FI INTERACTIVE PROTOTYPE

UXPin is more powerful than you think.

TAKE ADVANTAGE OF
GOOD TYPOGRAPHY

You can use a ton of beautiful fonts from Google Fonts library. Need something simple and nice looking, or maybe something more fancy? We've got them all. In total, there is 100 most popular Google Fonts to choose.

The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog

Photo credit: UXPin

2. **Avoid using celebrity names, jokes, or even *X*s in place of contact data** – These are distracting and, like lorem ipsum, will destroy the immersion. The more your user has to think about external factors, the less accurate the testing results.

3. **Simplify the steps** – Fewer clicks mean less friction, which means better user flow. But don't let the 3-click myth restrict your design decisions. You could theoretically create a site with all of its pages accessible in one click from the site's front

page, but that would increase the cognitive strain due to all the information sifting. Instead, make each step feel as effortless as possible.

4. **Don't skimp on animations** – If you're using UXPin, try the custom animations editor to create interactions step-by-step without any code. When you can test your animations in a prototype, you prevent developers from sinking time in jQuery or Javascript only to discover feasibility or usability issues.

5. **Know when to stop** – Because hi-fi prototypes are so advanced, it's easy to get so caught up in the details that you forget it's not the final product. Remember that prototypes are just a means to an end, not the end itself. Even the highest fidelity prototype will probably only represent 75-80% of the final product.

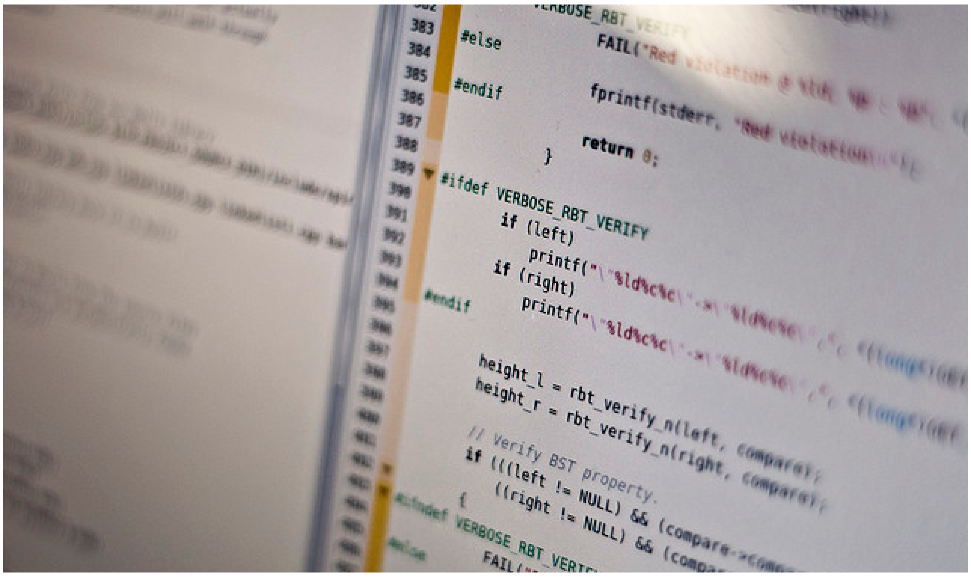
Coding debate

There is debate among the designers about whether or not it's worth it to using coding in prototypes. This technique takes "living documentation" a step further, and blurs the line between prototype and product.

Their thinking is logical: prototyping in code reveals definitively what you can and cannot do in the final product, which saves you time backtracking in the end, and hastens the iteration process. Ash Maurya even takes this to the next level by building mockups in code. Designers who use code early on are of the school of thought that any documentation not in code will ultimately just be thrown away, so *not* prototyping in code is a waste.

However, there are several counterpoints. First, a lot of designers don't understand code, or have a limited knowledge of it. Designing in code inhibits their creativity, preventing ideas that developers would know how to bring to life even if they do not.

Second, coding this early impedes rapid prototyping – the coding makes multiple prototypes take longer and require more work, and in the end creates more waste when some designs are proven ineffective by testing.



```
#else
            FAIL("Red
#endif
            fprintf(stderr,
            return 0;
        }
#ifdef VERBOSE_RBT_VERIFY
    if (left)
        printf("\"%ld%c%c\"
    if (right)
        printf("\"%ld%c%c\"
#endif

    height_l = rbt_verify_n(left, compare);
    height_r = rbt_verify_n(right, compare);

    // Verify BST property.
    if (((left != NULL) && (compare->
        ((right != NULL) && (compa
    {
#ifndef VERBOSE_RBT_VERIF
```

Photo credit: *"Code." Michael Himbeault. Creative Commons.*

The conclusion we come to is this: don't code unless you're extremely proficient technically. Some designers can speak both languages, but coding is usually the responsibility of the developers. The beauty of collaboration is combining the strengths of each member of the team to mitigate the weaknesses.

Unless the designer is well-versed in code, having them think in this language will just restrict their design thinking.

Takeaway

Each of the above documents offers a unique perspective not just for the design process, but for the designer as well. Different designers with different strategies and ways of thinking will find some documents more helpful than others. They feed on certain strengths, but come with their own drawbacks. With the exception of wireframing and prototyping, not all are necessary for every project.

Before you even think about which documentation to use and which to skip, the most important step is knowing your team's personal design process.

A Practical Approach to Usability Testing

Usability testing makes the difference between design thinking (designing for the user) and the outdated modes of thinking favoring features, businesses, or the product itself. The latter test only at the end to validate their ideas, while design thinkers test throughout the process to generate their ideas.

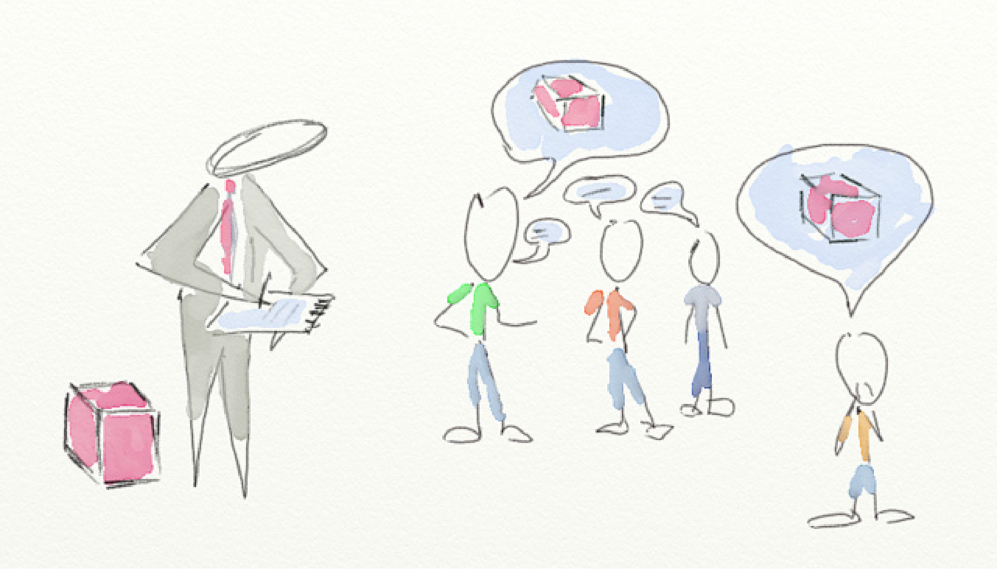


Photo credit: "Listening (the extended research dept)" Xavier Vergés. Creative Commons.

The cycle of iterating, testing, and implementing the feedback will chip away at all the imperfections until all that remains is the best possible version of your product.

In this chapter, we'll outline the four steps of usability testing (and the documentation that occurs in each):

1. **Define Goals** – Determine which questions you want the test to answer.
2. **Prepare the Test** – Determine which test will answer your questions, then plan the best way to conduct it.
3. **Conduct the Test** – Recruit participants and administer the test.
4. **Present Results** – Compile data into an easy-to-understand format and share it with your team-members.

We'll start with the planning phase.

1. Define Goals

The first step to any successful usability test is defining your goals: what questions do you want the test to answer. This could be broad, such as,

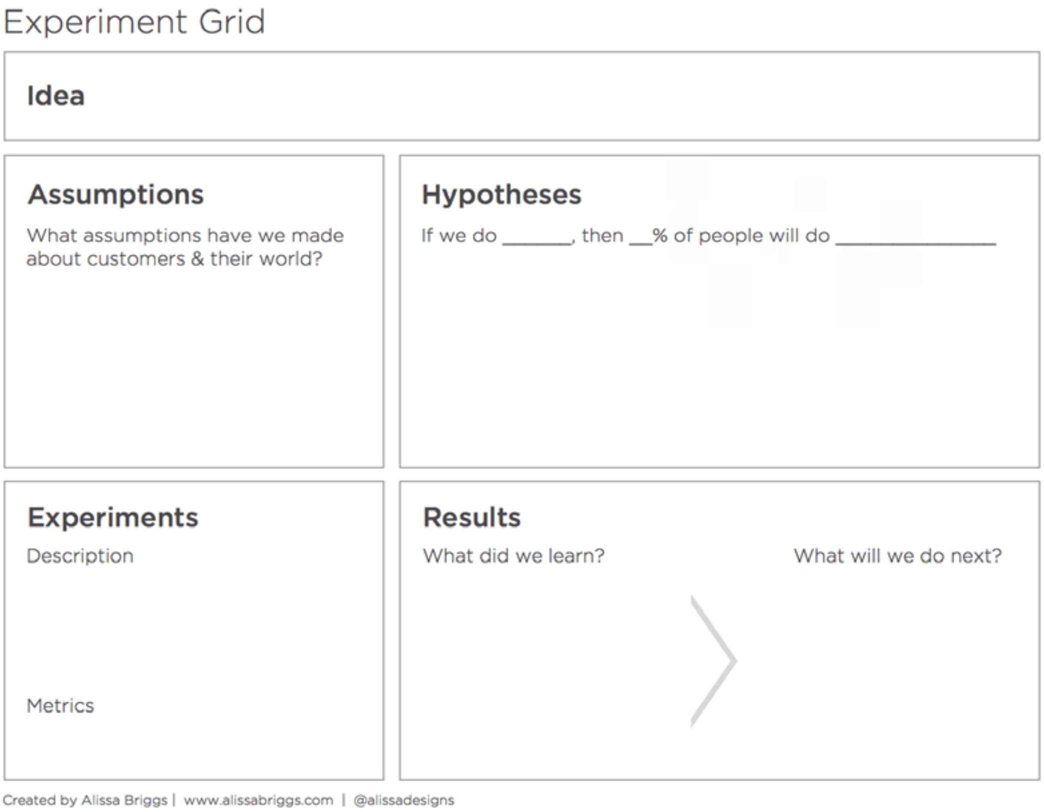
Which checkout methods are most intuitive to our users?

Or specific, such as:

Which form design works best for increasing ecommerce purchases?

The important thing is that you know *why* you're conducting the test. Knowing where you're going will let you find the best route there.

Naturally, you'll have a lot of questions about your product, and this curiosity is good. However, remember to limit each test to only the most relevant issue at the moment. Each test should have a central focus for the most accurate results – the more objectives you test at once, the more room for error.



Experiment Grid

Idea

Assumptions

What assumptions have we made about customers & their world?

Hypotheses

If we do ______, then __% of people will do ____________

Experiments

Description

Metrics

Results

What did we learn?

What will we do next?

Created by Alissa Briggs | www.alissabriggs.com | @alissadesigns

Photo credit: Alissa Briggs via Enterprise UX Conference

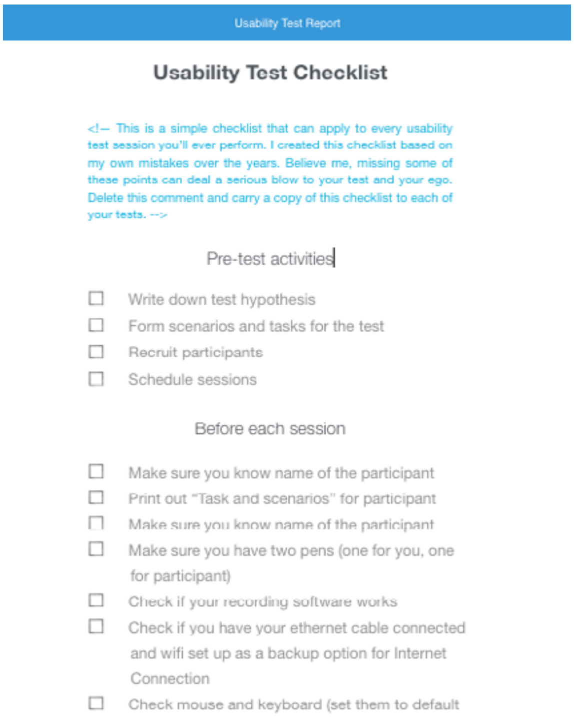
This is another advantage to testing often: you can address each issue with the attention it deserves. If you find yourself with too many questions, make a list and prioritize the questions based on the steps of the design process. You can always save this list for later tests.

Deciphering these questions will automatically set your mind up to think up answers on its own. As David Sherman mentions in his article on usability testing, these potential answers will be your test's hypothesis. By stating the hypothesis outright, you can draw attention to any potential bias in order to prevent it, and will also help you communicate the results later ("we originally thought *this*, but then discovered *this*").

You can generate hypotheses simply by setting aside time, for your and your team, to try to answer the goal questions on your own.

2. Prepare the Test

There are literally hundreds of types of usability tests to choose from, each one with their own special area of expertise and their own limitations. It's not about knowing which tests work and which don't, it's about knowing which will work for a specific need. That's why defining your goals first is crucial.



Usability Test Report

Usability Test Checklist

<!– This is a simple checklist that can apply to every usability test session you'll ever perform. I created this checklist based on my own mistakes over the years. Believe me, missing some of these points can deal a serious blow to your test and your ego. Delete this comment and carry a copy of this checklist to each of your tests. -->

Pre-test activities

- □ Write down test hypothesis
- □ Form scenarios and tasks for the test
- □ Recruit participants
- □ Schedule sessions

Before each session

- □ Make sure you know name of the participant
- □ Print out "Task and scenarios" for participant
- □ Make sure you know name of the participant
- □ Make sure you have two pens (one for you, one for participant)
- □ Check if your recording software works
- □ Check if you have your ethernet cable connected and wifi set up as a backup option for Internet Connection
- □ Check mouse and keyboard (set them to default

Photo credit: UXPin: Usability Test Kit

In *The Guide to Usability Testing*, we divide the tests into four categories based on Christian Rohrer's fantastic article:

1. **Scripted** – These tests analyze the user's interaction with the product based on set instructions, targeting more specific goals and individual elements. (tree testing, hallway usability tests, benchmark testing)

2. **Decontextualized** – Ideal for preliminary user testing and persona research, these tests don't necessarily involve the product, but analyze more generalized and theoretical topics, targeting idea generation and broad opinions. (user interviews, surveys, card sorting)

3. **Natural (or near-natural)** – By analyzing the user in their own environment, these tests examine how users behave and pinpoint their feelings with accuracy, at the cost of control. (field and diary studies, A/B testing, first click testing, beta testing)

4. **Hybrid** – These experimental tests forego traditional methods to take an unparalleled look at the user's mentality. (participatory design, quick exposure memory testing, adjective cards)

Before you test your users, you might also send out a user survey. Obviously, these aren't usability tests, but they certainly contribute valuable research to the testing process. You'll want to include a mix of the following types of questions:

- **Multiple Choice** – Whether the user selects from an assortment of prewritten answers, or simply just yes-or-no, multiple choice questions allow you to easily categorize answers and give you control for specialized data, but do not allow user insight and risk being biased. These lean towards quantitative.

- **Verbal (or Written) Responses** – Asking users open-ended questions and encouraging their elaboration may reveal some insights you had not anticipated. However, you are at the whim of how well the user can articulate themselves, and this data can be difficult to categorize and therefore analyze. These lean towards qualitative.
- **Rating Scale** – By asking users to rate their feelings on a numeric scale, you're able to capture qualitative data in a quantitative way. This allows you to analyze a user's feelings in a concrete way.

Once you determine the type of usability test(s) to run, you should send out a descriptive announcement to give your team a heads up. It's even more helpful, in fact, if you summarize your tactics with a quick planning document.

Research Plan Document

Modified from Tomer Sharon's One-Pager (fantastically helpful yet lightweight), our research plan document is a formalized announcement with all the necessary details of the testing to both explain what's happening and invite collaboration.

1. Purpose

In addition to keeping the test planners organized, the research plan lets the entire team know all the relevant details about the test, and gives them the chance to contribute their feedback or improvements before the test is conducted. It also appeases stakeholders seeking a bottom-line assessment.

2. Best Practices

Brevity is the name of the game with research plan documentation. You want to hand your team a slim document around one page to encourage them to actually read it.

While keeping things brief, you'll want to cover at least these 7 sections:

1. **Background** – Here is where brevity is important. In a single paragraph, describe the reasons and events leading to the research.

2. **Goals** – In a sentence or two (or bullets), summarize what the study hopes to accomplish. Phrase the goals objectively and concisely. Instead of "Test how users like our new checkout process," write "Test how the new checkout process affects conversions for first-time users."

3. **Questions** – Here is where you elaborate if needed. List out around 5-7 questions you'd like the study to answer.

4. **Tactics** – Where, when, and how the test will be conducted. Explain why you've chosen this particular test.

5. **Participants** – Describe the type of user you are studying, including their behavioral characteristics. You could even attach personas (or link to them) for more information.

6. **Timeline** – The dates for when recruitment starts, when the tests will be expected to take place, and when the results will be ready.

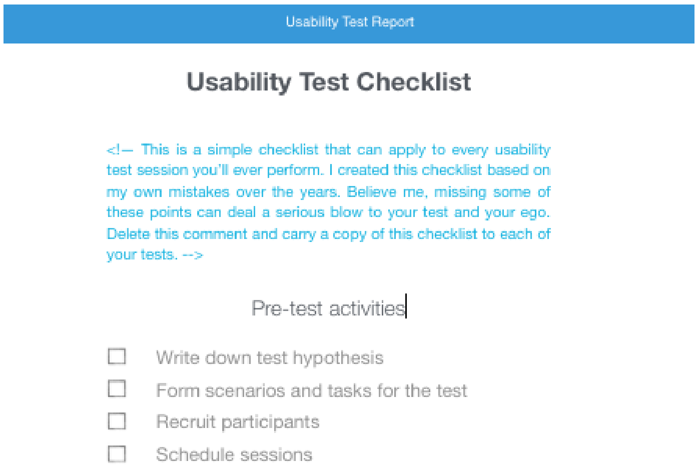
7. **Test Script** – If your script is ready, include it here.

Check out Sharon's sample One-Pager to see how it should look.

Encourage your team-members to give suggestions or advice so that the test results are helpful to everyone. Find out the questions that they want answered as well.

3. Conduct the Test

After gathering feedback from the team, you're ready to actually conduct the test. This involves recruiting the right participants, scheduling times, and writing the actual test documentation.



Usability Test Report

Usability Test Checklist

<!– This is a simple checklist that can apply to every usability test session you'll ever perform. I created this checklist based on my own mistakes over the years. Believe me, missing some of these points can deal a serious blow to your test and your ego. Delete this comment and carry a copy of this checklist to each of your tests. -->

Pre-test activities

- □ Write down test hypothesis
- □ Form scenarios and tasks for the test
- □ Recruit participants
- □ Schedule sessions

Photo credit: UXPin: Usability Test Kit

For recruiting users, stick to your target audience defined in the onset of the design process. These are the same types of people who influenced your personas. If you'd like help reaching out to these people, Jeff Sauro, founder of Measuring Usability LLC, lists 7 methods for user recruitment, including online tools. In our experience, we've found hallway testing and tools like UserTesting incredibly helpful (part of the inspiration for integrating usability testing into UXPin).

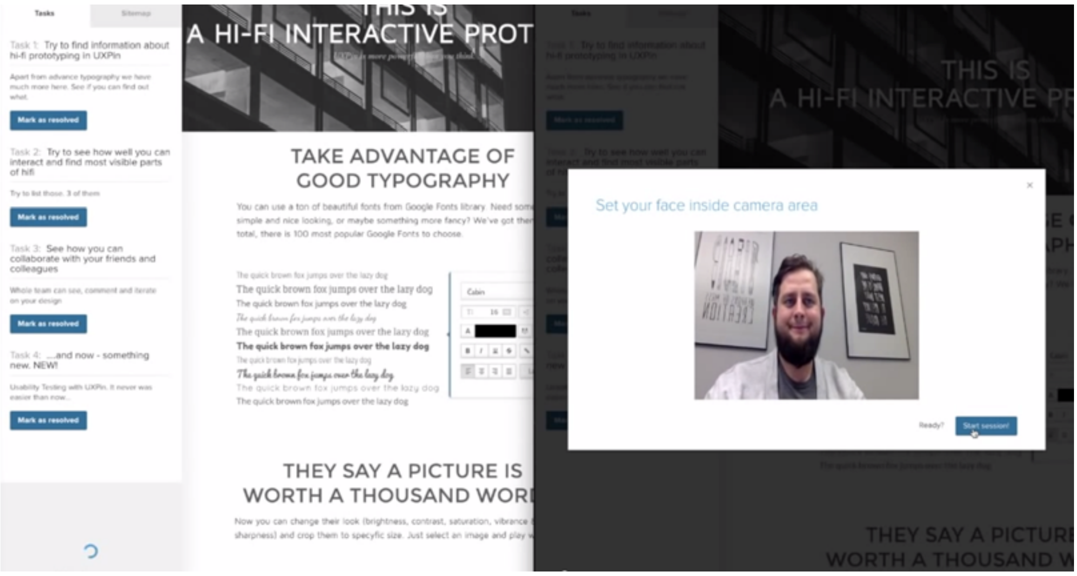
As for your role during the actual test, sometimes you must make the choice between being present (moderated) or allowing the user to work on their own (unmoderated):

- **Unmoderated** – Unmoderated tests are cheaper, faster, and generally easier to recruit and schedule. They also remove the influence of a moderator, leading to more natural and less biased results. On the downside, there is less opportunity for follow-up questions or supporting users who go astray during tests. Moreover, you put yourself at risk of finding users only interested in the compensation, thus reducing the quality of the results (unless you use tools like UserTesting who filter for this).

- **Moderated** – While costlier and requiring more effort to organize, moderated tests allow you to "lead" the user, for better or worse. Moderated tests are recommended for rougher prototypes (higher risk of bugs and usability issues) or incredibly complex prototypes (users might need some clarification).

Additionally, you can also choose to conduct your test on-location or remotely. While every test has different qualities and best practices, the following advice works across the board:

- **Make users comfortable** – Even the word “test” makes people a little nervous, so put in extra effort to put the them at ease. Remind them you are testing the product, not their capabilities. A test script helps ensure you hit upon a few reassuring points in the beginning of each test.
- **Test competitor products** – If applicable, use competitor products as a frame of reference. This distinguishes whether your user’s opinion exists independent of your product, or as a consequence of it.



Task 1: Try to find information about hi-fi prototyping in UXPin

Apart from advance typography we have much more here. See if you can find out what.

Task 2: Try to see how well you can interact and find most visible parts of hifi

Try to list those. 3 of them

Task 3: See how you can collaborate with your friends and colleagues

Whole team can see, comment and iterate on your design

Task 4: ...and now - something new. NEW!

Usability Testing with UXPin. It never was easier than now...

THIS IS A HI-FI INTERACTIVE PROT

TAKE ADVANTAGE OF GOOD TYPOGRAPHY

You can use a ton of beautiful fonts from Google Fonts library. Need som simple and nice looking, or maybe something more fancy? We've got the total, there is 100 most popular Google Fonts to choose.

The quick brown fox jumps over the lazy dog

THEY SAY A PICTURE IS WORTH A THOUSAND WORI

Now you can change their look (brightness, contrast, saturation, vibrance sharpness) and crop them to specyfic size. Just select an image and play w

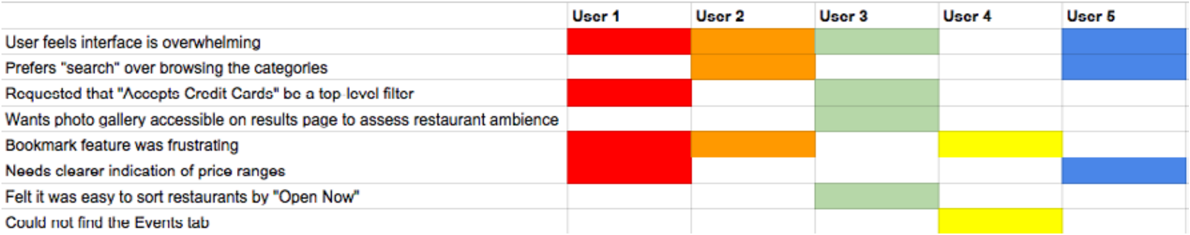
Set your face inside camera area

Photo credit: *UXPin Usability Testing*

- **Don’t interfere** – Unless the user is at a complete standstill, sit back and allow them to figure out the product on their own and make mistakes. This avoids bias, and may reveal insights into user behavior you hadn’t predicted. The best insights usually come from when a user isn't engaging with the product the way it's designed. Pay attention to workarounds and let them inspire feature improvement.

- **Record the session** – This makes a solid reference point for later, when interpreting the results. If you're running the test through UXPin, you can record data like facial reactions, clicks, and all audio.

- **Collaborate** – If you have a team observing a moderated user test, Tomer Sharon (mentioned above) suggests creating a Rainbow Spreadsheet, a shared observational sheet that allows everyone to record their own interpretations of the data for quick and easy comparisons later. We used his spreadsheet during our Yelp redesign exercise and found it was very helpful for summarizing results for designers and stakeholders.



	User 1	User 2	User 3	User 4	User 5
User feels interface is overwhelming	🟥	🟧	🟩		🟦
Prefers "search" over browsing the categories		🟧	🟩		🟦
Requested that "Accepts Credit Cards" be a top level filter	🟥		🟩		
Wants photo gallery accessible on results page to assess restaurant ambience			🟩		
Bookmark feature was frustrating	🟥	🟧		🟨	
Needs clearer indication of price ranges	🟥				🟦
Felt it was easy to sort restaurants by "Open Now"			🟩		
Could not find the Events tab				🟨	

Photo credit: UXPin for Yelp design usability testing based on exercise suggested by Tomer Sharon

Testing for mobile products, especially, require care and attention, given the technical difficulties. For advice specific to this, read Rosie Sherry's article, A Field Guide To Mobile App Testing.

Of course, the way your write the actual tasks will also influence the results.

User Tasks

As the name suggests, user tasks are what the user tries to accomplish during the test.

1. **Purpose**

   Well-defined user tasks make the difference between an organized procedure and simply "winging it." More-so than simply telling the user what to do or posing them questions, it accounts for the purpose of the test.

2. **Best Practices**

   Everything you present to your users during the test – both the content of the question/task, as well as the phrasing – impacts how they respond.

   Tasks are either open or closed, and your tests should incorporate a healthy mixture of both:

   - **Closed** – A closed task offers little room for interpretation – the user is given a question with clearly defined success or failure ("Find a venue that can seat up to 12 people."). These produce quantitative and accurate results.

   - **Open** – By contrast, an open question can be completed in several ways. These are "sandbox" style tasks ("Your friends are talking about Optimal Workshop, but you've never used it before. Find out how it works.") These produce qualitative and sometimes unexpected results.

Photo credit: "A Five-Step Process for Conducting User Research."
David Sherwin. Smashing Magazine.

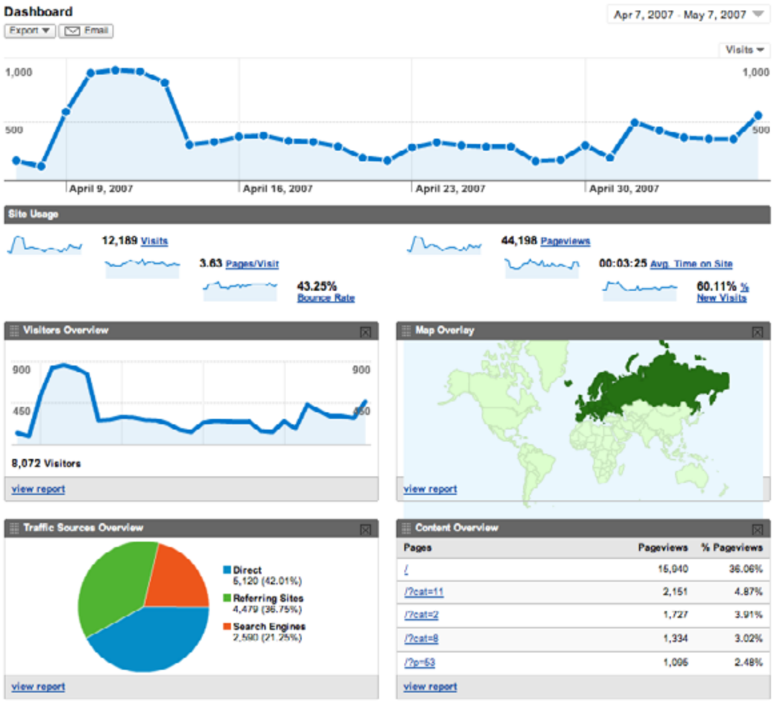
Read Tingting Zhao's piece for more advice on optimizing tasks.

As for the wording, be careful to avoid bias. Just one wrong word can skew results.

For example, if you want to find the most natural ways in which users browse your online shop, writing a task like "It's 10 days before Christmas and you need to *search* for a gift for your mother," might lead the user to use the search functional, as opposed to their normal method of window clicking.

3. **Present Results**

   The point of the testing, is, of course, to collect data to influence design. But if you don't present the results effectively, or at all, the entire testing process is a waste.



12,189 Visits

3.63 Pages/Visit

43.25% Bounce Rate

44,198 Pageviews

00:03:25 Avg. Time on Site

60.11% % New Visits

8,072 Visitors

Direct 5,120 (42.01%)

Referring Sites 4,479 (36.75%)

Search Engines 2,590 (21.25%)

Pages	Pageviews	% Pageviews
/	15,940	36.06%
/?cat=11	2,151	4.87%
/?cat=2	1,727	3.91%
/?cat=8	1,334	3.02%
/?p=63	1,096	2.48%

Photo credit: "Google Analytics 2.0." Panayotis Vryonis. Creative Commons.

The usability data from tests can and often means the difference between success and failure. For example, Venmo, a money-exchanging app, too the data from its analytics and used it to fix a critical error.

But it was not as simple as that. First, the support team brought the problem to the product team, who collaborated with the data team. The data was analyzed in a helpful way through Looker, which showed them the problem in a way everyone could understand. After that, it was a quick fix.

From the Venmo example, which you can read about here, we can identify the two criteria for success at this stage of the process:

- presenting the results in a way that's helpful
- sharing the results with the team in a uniform way

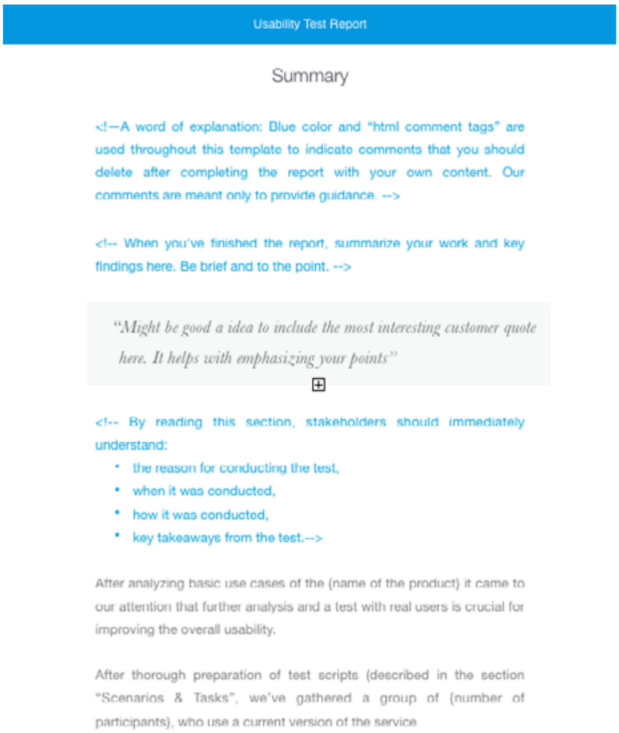
This phase is about taking raw data – sometimes even just numbers – and turning it into something useful. It's not just enough to conduct tests, you have to show the results the right way, and then share them.

Usability Report

The usability report is the way to share the results with the team, so that everyone's on the same page.

1. **Purpose**

   First, the usability report is a universal document (or, more accurately, a collection of documents) that everyone on the team can reference. It makes sharing easy, especially with a cloud folder (see below) where everyone can access the same information.



Usability Test Report

Summary

<!—A word of explanation: Blue color and "html comment tags" are used throughout this template to indicate comments that you should delete after completing the report with your own content. Our comments are meant only to provide guidance. -->

<!-- When you've finished the report, summarize your work and key findings here. Be brief and to the point. -->

> *"Might be good a idea to include the most interesting customer quote here. It helps with emphasizing your points"*

⊞

<!-- By reading this section, stakeholders should immediately understand:

- the reason for conducting the test,
- when it was conducted,
- how it was conducted,
- key takeaways from the test.-->

After analyzing basic use cases of the (name of the product) it came to our attention that further analysis and a test with real users is crucial for improving the overall usability.

After thorough preparation of test scripts (described in the section "Scenarios & Tasks", we've gathered a group of (number of participants), who use a current version of the service

Photo credit: UXPin: Usability Test Kit

Collaboration, in general, is a key aspect of this stage, if for no other reason than to remove bias. The more people to comment on the notes, the less influenced they are by personal interpretations. As Alla Kholmatova describes, a second evaluator increases problem detection by 30%–43%.

Furthermore, documentation of the test helps down the road. Later in the design process, you may want to draw on the notes of the testing, in which case you'll be glad for an easily accessible formal report.

2. Best Practices

To best organize and make the results readily available, we suggest creating a cloud folder with universal access. At UXPin, any team member can read or reference the latest results at any time.

As you write the report, keep the following tips in mind:

- **Avoid vagueness** – Mentioning that "Users couldn't buy the right product" isn't very helpful since multiple factors might be involved. Perhaps the checkout process was difficult, or the product listings were hard to browse. Explain the root of each issue from an interaction design and visual design perspective (e.g. confusing layouts, a checkout process with too many steps, etc.).
- **Prioritize issues** – Regardless of how many issues you find, people must know what's most important. We recommend categorizing the report (e.g. Navigation Issues, Layout Issues,

etc.) and then adding color tags depending on severity (Low/Medium/High). List every single issue, but don't blow any out of proportion. For example, don't say that a red CTA button lead to poor conversion if the steps of the checkout process don't make sense.

- **Include recommendations** – You're the expert and stakeholders need an action plan. Don't include any hi-fi prototypes or mockups in the usability report, but definitely suggest a few improvements. To supplement written suggestions, our own UX Researcher Ben Kim also links to lo-fi wireframes or prototypes in a UXPin project dedicated to usability testing.

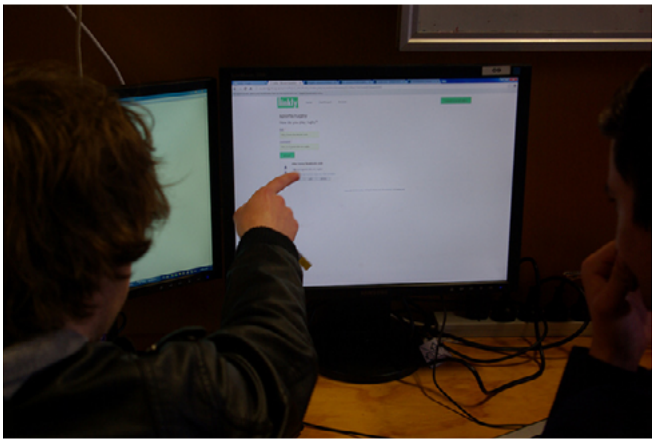


Photo credit: "Project User Experience Testing." Samuel Mann. Creative Commons.

When presenting the results, include any and all relevant materials. The usability report should be a folder, not a single file. Don't forget to include things like:

- Formal usability report
- Supporting charts, graphs, and figures
- Previous testing documentation (i.e., the list of questions the user was asked)

- Videos or audio tracks of the test (which is why it's good to record sessions)

Finally, do not treat the usability results as a folder meant to be handed off. The documentation is just the starting point. Schedule a follow-up meeting with the team to review the usability report and relevant data, discussing issues and the outlined recommendations.

Takeaway: Test Early and Test Often

User testing insights speak far louder than guesswork and conjecture. They help guide design decisions, and serve as powerful evidence to counter people's opinions.

Don't wait until the end of the project to conduct your usability testing. Once you have a lo-fi prototype, start testing. The data is less about validation and more about inspiration: test early, and test often, so you can actually put the results to use before it's too late.

Design Sprints: Condensing the UX Process

The process as we've described is meant to be cyclical. But what if you could speed up each phase for better results and more frequent feedback?

This is the thinking behind design sprints, a fast process for yielding more iterations in less time.

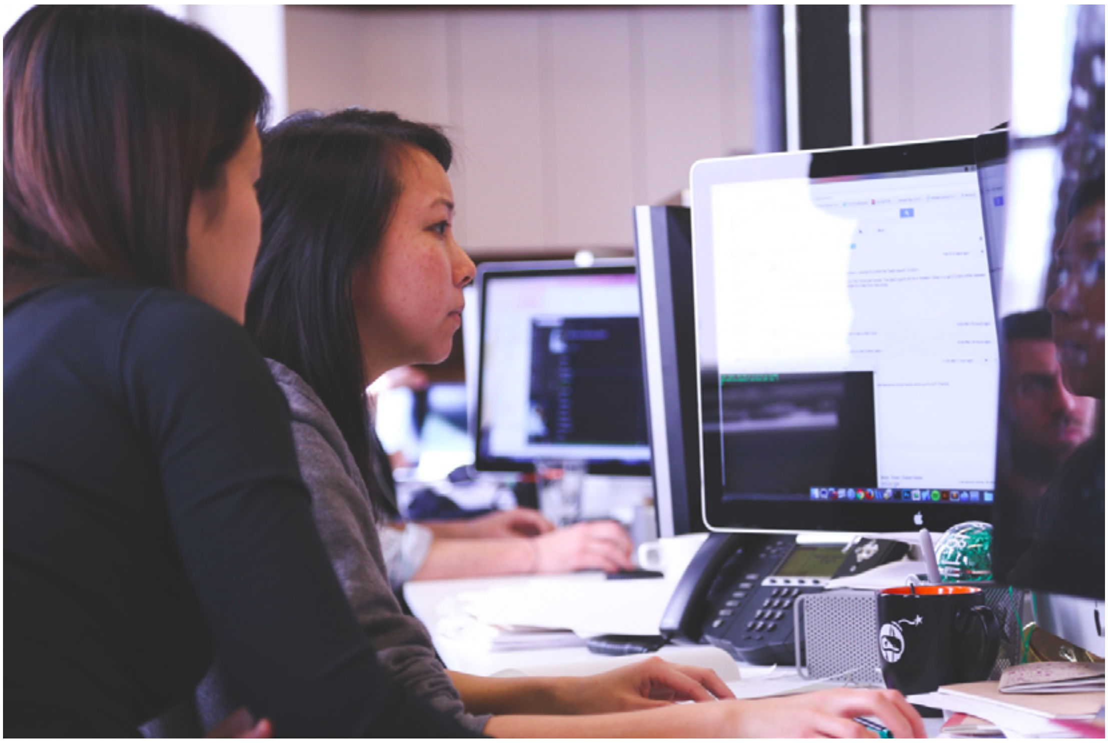
Photo credit: "UX for Good Breakout. WIAD DC. Creative Commons.

Design sprints aren't simply "rushing" everything. We'll describe a few best practices for ensuring the process is quick but productive.

First, let's explore why design sprints work.

The What and Why of Design Sprints

Popularized by Google Ventures and their design partner Jake Knapp, the design sprint aims to validate your ideas as quickly as possible through fast research and prototyping.



In general, design sprints typically last about a week, with each day dedicated to a different activity, plus a preparation period. By the end of the week, you'll have not only a completed prototype, but also substantial user data.

You can conduct sprints at virtually any time in the design process.

Design sprints are effective for four main reasons:

1. **User-focused** – Usability testing precedes iteration, ensuring you never veer away from creating what users need.
2. **Naturally collaborative** – When you work fast, you need the whole team onboard.
3. **Perfectly efficient** – With rapid prototypes, you build just enough product to test if the idea has any merit with users.

Photo credit: "A List Apart big meeting, 30 January 2015." Jeffrey Zeldman. Creative Commons.

Within the constraint of a week, the team limits itself to only the most important issues, and can't afford to overthink anything. But the pressure is taken with grain of salt – participants recognize that sprints are meant for speed, not perfection. As Knapps describes, the approach has worked out well for some highly successful companies like Blue Bottle Coffee and Nest.

Design Sprint Preparation

Everything that *can* be done beforehand, *should* be done beforehard. In our experience running our own design sprints, this frees your team up to concentrate only on designing during the actual sprint. By the morning of the first day, we try to complete the following tasks as suggested by Google Ventures:

- **Review existing user research** – While the point of the sprint is to generate new learnings about users, you shouldn't start from scratch. Collect and distribute the previous documentation to the team so everyone's on the same page, or collect new relevant data (for example, if a new market segment were introduced). Research sprints, described later, can accomplish this quickly.

- **Schedule user tests** – Yes, you'll need to book tests for a prototype that doesn't exist yet. For the sake of speed, schedule a minimum of five users for testing on the last day of the sprint. This "no turning back" also sets a clear finish line for the team.

- **Recruit the right people** – Too many participants bog down the process, so you want to make sure you have only the right people. We follow Knapp's advice when we run design sprints at UXPin by including the CEO, product manager, visual designer, developers, and UX designer in the first few days.

- **Elect a sprint leader** – Because design sprints can get hectic with all the team members involved, decide on a leader beforehand to keep everything on task and on time. This leader may need to limit their involvement in the actual designing to moderate the sprint.

Deal with these first so you can start the sprint with a few less things to worry about.

The 5-Day Plan

Sprints can be modified and customized, but the basic foundation follows this 5-day plan:

- *Monday:* Analysis and Goal Creation
- *Tuesday*: Idea Generation
- *Wednesday*: User Flows
- *Thursday*: Prototyping
- *Friday*: User Testing

We delve into the best practices for each phase below.

1. Analysis and Goal Creation

The first order of business is to analyze the situation and bring everyone up to speed.

Photo credit: "Booksprint Futurish '14." Time's Up. Creative Commons.

While the relevant documentation should have been distributed before, now is a good time to go over it together to ensure nothing important slipped through. This is also a chance for different departments to share their unique perspectives on the project.

Getting to the task at hand, the first day must also organize the project's goals. If the team works well together, this can be a simple discussion, but for more control and efficiency we recommend an affinity diagramming exercise that's worked quite well for us.

Also known as the KJ Method, an affinity diagram is an activity for organized prioritization, following four steps:

1. **Ask a question** – Take the problem you want the sprint to solve, and pose it as a question. This could be as specific as, "What feature could help the user achieve their goals?" or as vague as "What issues do we want to address during this sprint?"
2. **Collect ideas** – Thinking individually, everyone writes down their ideas on sticky notes and puts them on the wall.
3. **Sort notes in silence** – Once all the ideas are collected, the team reorganizes them into logical groups. The key component of this stage is that no one talks – this way, no one person dominates and the ideas are grouped organically. Afterwards, give each group an appropriate title (it's okay to talk here).
4. **Vote on prioritization** – Within each category, conduct votes on which ideas are more important. Then, vote on which categories are more important.

After the affinity diagram, the team has established a clear hierarchy of ideas and priorities for the next days' designing. In our experience, it's best that the core design team still has final say over which categories are most important. The votes help inform their decisions, but don't act as design by committee.

Photo credit: "Affinity Diagramming 2." d_jan. Creative Commons.

To learn more, we highly recommend reading usability expert Jared Spool's guide to the KJ method.

2. Idea Generation

The second day is the most important for brainstorming. Here is where the team creates most of its ideas and the product begins to form. Based on the goals established the day before, the team works with coming up with the best possible solutions.

A design studio is a great way to get people working together to sketch viable ideas. In fact a design studio is like a next-tier affinity diagram – using similar methods, it fleshes out potential solutions from the highest priority goals that we already determined.

Basically, everyone sketches potential solutions to the problem, the work is critiqued, and the design team can further iterate on useful ideas. We suggest the following method:

- **State the task** – Describe clearly the problem for which everyone is sketching solutions, such as “Redesign the checkout process for more sales”.
- **Create your teams** – Divide your participants into at least 2 teams, with no more than 4 people per team. Ideally, you want at least one designer per team.
- **Sketch quickly (15 minutes)** – Encourage people to quickly sketch at least 5 ideas, focusing on the concepts and using annotations where necessary.
- **Critique ideas within the team (15 minutes)** – Within each team, every member explains their sketches for feedback
- **Iterate (20 minutes)** – With feedback gathered, every member now distills their many concepts into one unified idea.
- **Vote & Polish (20 minutes)** – Each team now reviews everyone’s iterated sketch as they try to hone in on one concept. Once that concept is decided, each team can add any finishing touches based on all the feedback gathered.

- **Group Discussion & Review (40 minutes)** – Each team now has one unified concept. At this point, both teams can merge together to review and iterate the main concepts. When the session is finished, one concept will emerge for the design team (along with other alternatives created in the process).

There will also be people resistant to sketching for concerns about their artistic skill. Reassure everyone that this isn't about the art, it's about the ideas (hence the time limits).

Photo credit: "Design Studio." LindsayT.... Creative Commons.

The goal of day two is to come up with many viable solutions so you can zero in on the best idea. The sketches make a great jumping off point for the next day, when the structural designing begins.

3. User Flows

In our process, this step is when collaboration becomes more selective. Like we described in our 99U article The Right Way to

Collaborative Design, involving everybody at all times just invites a “swoop and poop” mentality. Instead, we’ve found it most helpful if the designers take over from here and involve others for feedback.

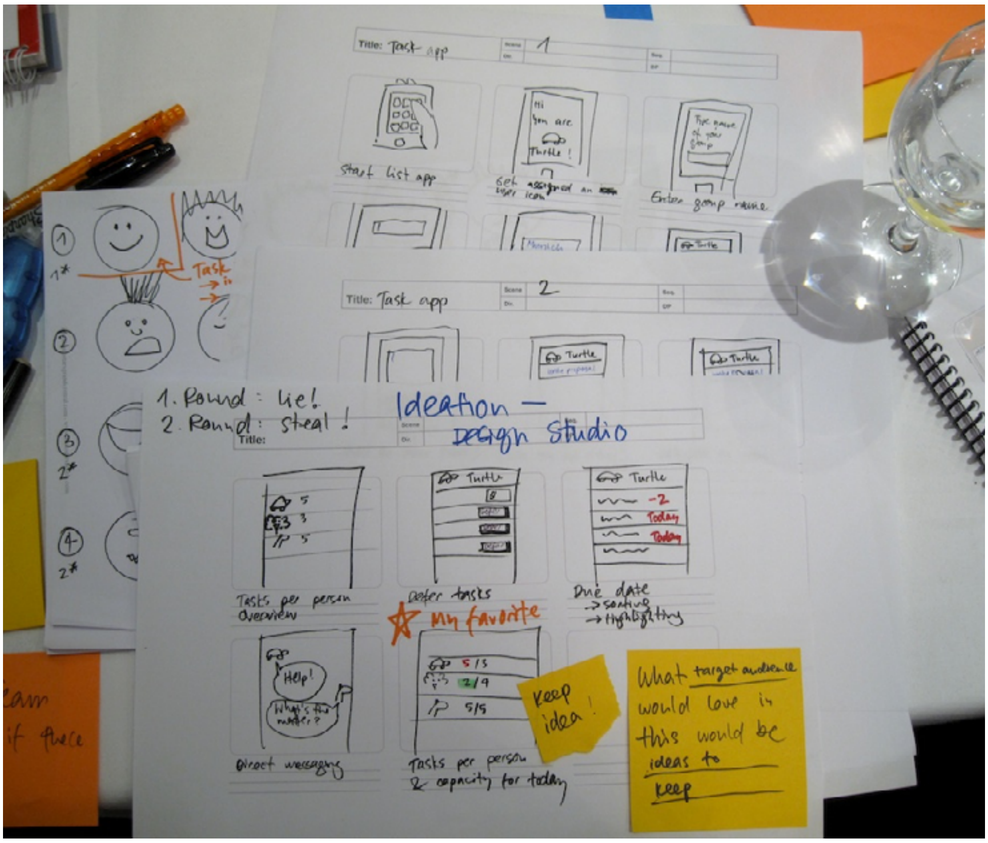


Photo credit: “Ideation sketching.” visualpun.ch. Creative Commons.

We will now create user flows will help us optimize all the steps required for completing user tasks. Here are two of the most useful techniques we’ve tried.

Writing-first approach

In her article Jumpstarting Your App Conception Without Sketching UI, Jessica Downey explains how the writing-first approach fleshes out the details, while creating a common language that’s recognizable on every page of your app or site.

For example, in the below scenario, the user wants to activate auto deposit. Content in brackets represents buttons or links.

Step 1: Would you like to set up auto deposit?
[Set auto-deposit]
Step 2: Select Deposit Frequency
[Once per month][Twice per month]
[Every other week][Every week]

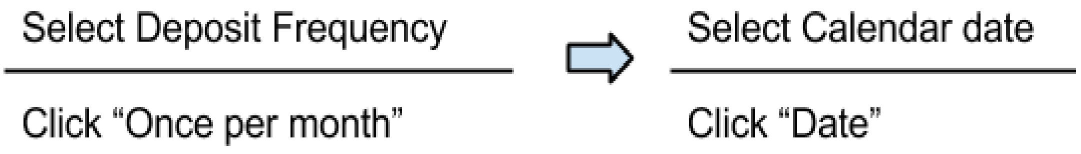
Step 3: Deposit Once per Month
[Select calendar day]

Step 4: Set Amount
Display amount field
[Set auto-deposit]

Shorthand approach

As an alternative, the shorthand approach, used by Ryan Singer at Basecamp, treats flows as ongoing conversations.

Using the same scenario above, this method abbreviates Steps 2 and 3 with shorthand:



Singer can illustrate even complex flows with this process, as you can see in his article A Shorthand for Designing UI Flows.

4. Prototyping

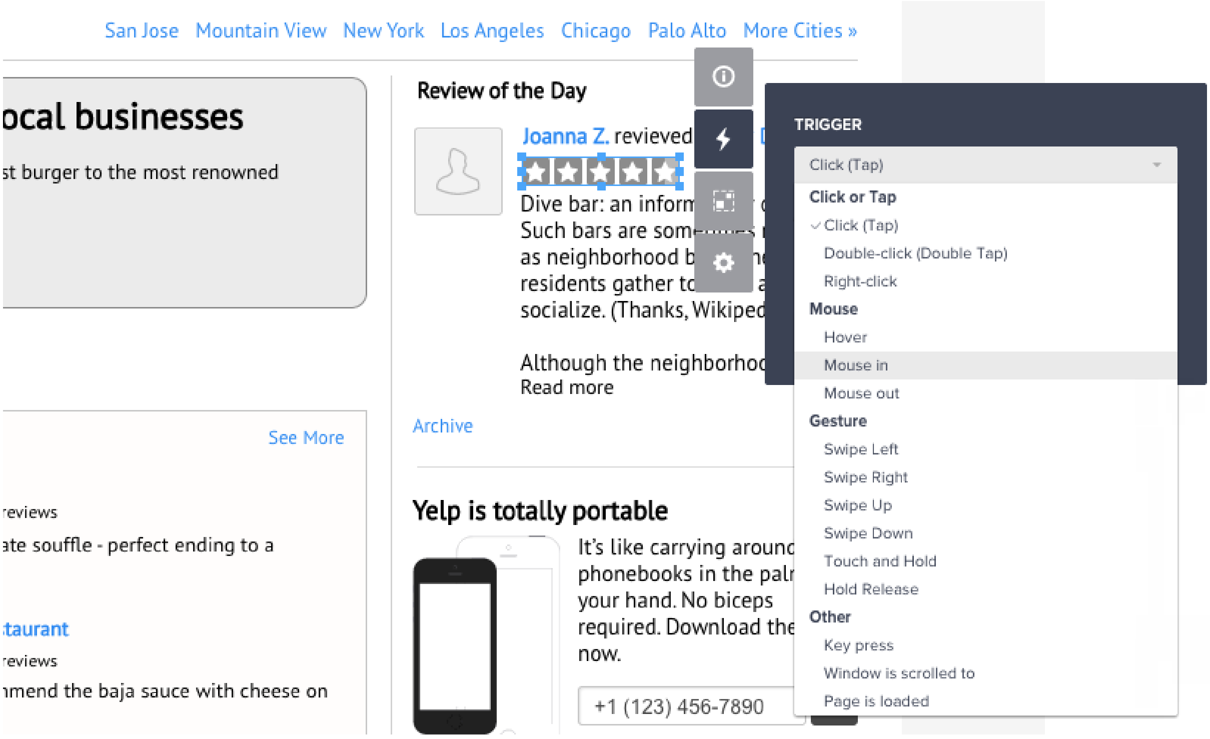
This is the big day, in which the team builds the actual prototype. Given the fast pace of the design sprint, a prototyping tool will be much faster than prototyping with HTML or Javascript.

No one said it'd be easy to build a prototype this fast, so here is some advice to help:

- **Know it's not perfect** – Nobody's expecting it to be perfect; the design sprint prototype only needs to be functional enough to test. Tackle only the most important aspects, then handle the details if there's time later.
- **Avoid placeholders** – It's tempting to use placeholders like lorem ipsum under this time constraint, but you may confuse users. Low fidelity does not equate to low quality, so use real content (it's fine if it's rough).
- **Follow good visual hygiene** – Follow strict consistency standards. Use a grid for alignment. Ensure that interface labels are consistent across each page. Make sure the design matches user expectations (external consistency) and doesn't contradict itself (internal consistency). We use customizable libraries and smart elements in our own app to create consistent UI patterns.
- **Delegate & collaborate** – When we use UXPin, multiple teams can co-create and comment on wireframes and prototypes. While the design team should focus on the "core" prototype, it's totally fine if developers or PMs want to create a simple prototype for specific interactions (e.g. a check-out process, or

sign-up form). Don't turn them away. Just remember to filter their ideas before adding them into the main prototype.

- **Embrace over-the-shoulder critiques** – Get at least one round of feedback from people who didn't create the design. Developers and marketers are always great candidates. The feedback sessions can be as brief as 15-20 minutes.



st burger to the most renowned

eviews
ate souffle - perfect ending to a

taurant
eviews
mend the baja sauce with cheese on

Dive bar: an inform
Such bars are som
as neighborhood b
residents gather to
socialize. (Thanks, Wikiped

Although the neighborho
Read more

It's like carrying around
phonebooks in the palr
your hand. No biceps
required. Download the
now.

Photo credit: UXPin (inspired by Yelp)

5. User Testing

Now it's the moment of truth. Since the prototypes are low fidelity, we've found moderated usability tests to be most helpful since someone can help guide users if they go way off track or are on the verge of giving up.

To save some time, you can limit your documentation to just the test script, questions, user tasks, and usability report. The usability report can be as simple as a list of prioritized problems (based on patterns you saw during the test). Discuss the problems with stakeholders, then start tackling them right away.

Now you have even more user data to draw from and a even better prototype. Implement what you've learned into the next iteration, and start the whole process again. In fact, Lyft has managed to condense the whole process into 4 days.

Takeaway

Design sprints aren't just for designers strapped for time. Sometimes deadlines can be the greatest incentive, and in situations when they don't occur naturally, creating one with a design sprint can get the team more focused and motivated.

So many of the traditional design methods are starting to show their age, and more modern tactics like sprints can help shake things up, in addition to their practical benefits. At times they can be difficult, but maybe that kind of constraint is exactly what good design requires.