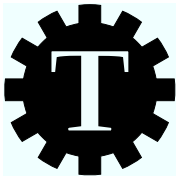


Chapter 10

Sketching and Prototyping

What's the Technique?

By now, you probably have a pretty good idea of what you want to create. It's time to flesh out the design, first by sketching the structure of your product and then by progressively zooming in on the details by designing the individual screens.

Why Is This a Good Idea?

Changes you make once you've started implementing your product can be expensive. A small user interface change can have vast implications.

Changing a sketch, on the other hand, is cheap and quick. All you need is an eraser, a pencil, and a few seconds.

Basically, you are creating simple prototypes of your product. If you're designing a remote control, you don't create the molds and start producing them. Instead, you start with simple wood or clay models of the remote to get a feel for how it should be proportioned. Then, you add more and more detail until you end up with the final design.

Sketching is the clay model prototype of your product. The more details you nail down before committing to code, the better.

Are There Any Prerequisites?

You should have a pretty good idea of what your product is going to be.

10.1 Designing the Structure

In *Rework* [FH10], Jason Fried and David Heinemeier Hansson write that "architects don't worry about which tiles go in the shower or which brand of dishwasher to install in the kitchen until *after* the floor plan is finalized."

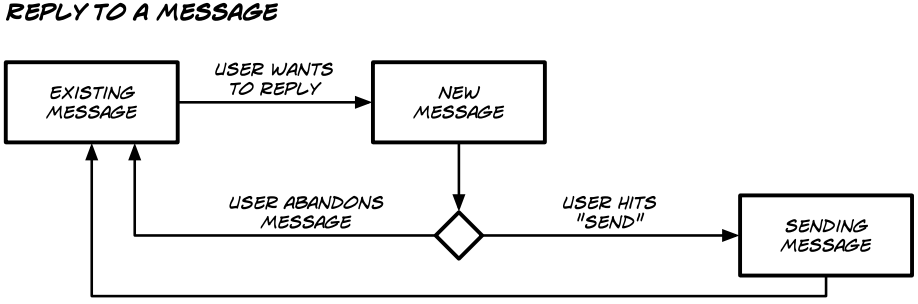
This is what flow diagrams and storyboards are: your product's floor plan. We're at the very beginning of the "design" part of our design process. Flow diagrams and storyboards are not about details. They are about the big picture: the structure.

10.2 Flow Diagrams

Flow diagrams answer the following questions: What does the user have to do to get what he wants? What steps does she have to follow to reach her goal?

Pick the most important user goals, and think about the required steps.

For our Twitter app example, a simple flow diagram for replying to a message would look a bit like this:



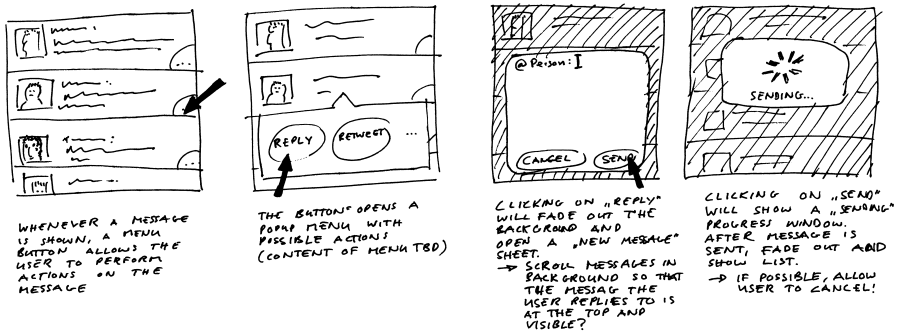
It's OK to add branching to flow diagrams, but don't make them too complex. In theory, you could have one enormous flow diagram that describes your whole product. In reality, it makes more sense to create several flow diagrams for the most important user goals. That way, you can keep the individual flow diagrams concise and clearly arranged.

The goal of this exercise is to think about what's involved in each goal. What kinds of screens do you need to show to the user? What kinds of decisions does he have to make at what point?

10.3 Storyboards

Storyboarding is a technique originally developed to plan animations for movies. Storyboards break animations down into their important frames; they turn a moving picture into a comic book. In user interface design, we use storyboards for a similar purpose. They break down the user's path into a series of snapshots.

Storyboards mostly ignore branching and focus instead on interaction details. What exactly is the user seeing on each screen? What do you want the user to do? Where should you use animations or graphics to help the user understand what he should do?



You can indicate where and how interactions happen by drawing arrows or hands (if it's a touch-screen user interface).

Making storyboards can take a lot of time, so you want to use them only for those parts of the application where the design is not obvious. Everybody who is familiar with comic books can follow a storyboard—it's a great tool for communicating design. If you need to explain to a programmer how to implement something, storyboards can be a huge help.

10.4 Sketching

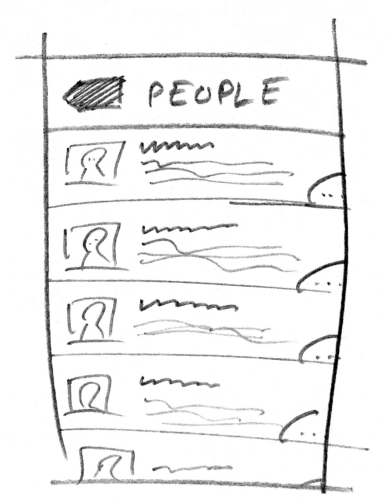
After architects design the floor plan, they design the individual rooms. After you design the structure of your product, you want to design the individual screens. By making flow diagrams and storyboards, you should have a pretty good idea of what screens your product requires and what functionality each screen should provide.

You've already done simple sketches of some of the screens while doing storyboards. In the storyboard example, some of the screens already

contained things that were unrelated to the task performed in the storyboard. For example, the pop-over menu didn't just have a "reply" button; it also had a "retweet" button. Now, we want to nail down the contents of the individual screens.

Usability consultant Bruce Tognazzini notes:[1]

> Jumping into complex, finely-tuned prototypes is perhaps the worst mistake a team can make. (...) Users (and clients) [feel more free] to express contrary views if models look less than perfected. But there's another side, too: designers and developers are more willing to listen to dissent if they haven't lavished ultimate care on what should have been a storyboard or quick-and-dirty prototype.



Experiment with using simple sketches to figure out how individual screens should look at a very basic level. Everybody can sketch, so you're free to involve other people in the process. Show them your ideas, and see whether they come up with their own.

Once you're happy with the basic design of your screens, move on to wireframes.

1. Read more at http://www.asktog.com/columns/005roughsketches.html.

Lorem Ipsum

People often use filler text like Lorem Ipsum in wireframes. If you really don't have anything better, using filler text is OK. If you can, however, include text that users might actually see. That gives you a better idea of how well the user interface will work and how big the text is going to be (if it's something that has specific lengths, like Twitter messages).

10.5 Wireframes

Wireframes represent the exact structure of a screen but without the decoration—no colors, shadows, or pictures. Wireframes are about the content. What do you want to show on each screen, and where do you want to put it? How big should things be? How close to each other should they be? Figure 10.1, on the following page shows a sample wireframe (on the left).

This is also the time when you start working on the copy. You don't need to get this exactly right yet, but you should think about text you want to include and where to put it.

The goal of wireframes is to identify exactly what you need to show on each screen and where you want to put things. Once you've done this, you can move on to decoration.

10.6 Mock-ups

Wireframes tell you about the layout of a screen. Mock-ups add the decoration, or visual details: shadows, textures, images, transparency. This is how you want the screen to look based on your current knowledge. Compare the mock-up in Figure 10.1, on the next page (on the right) with the wireframe (on the left).

Adding visual details is not just about making your product look pretty. Of course, the goal is for the end product to look good, but visual design can also give the user hints about the functionality of your product. User interface designers often call these *affordances*. I'll talk more about this in Section 9.5, *Principle 7: Affordances*, on page 90. For now, here are some examples of affordances: textures to let the user know that he can touch and drag something, shadows and bevels to show

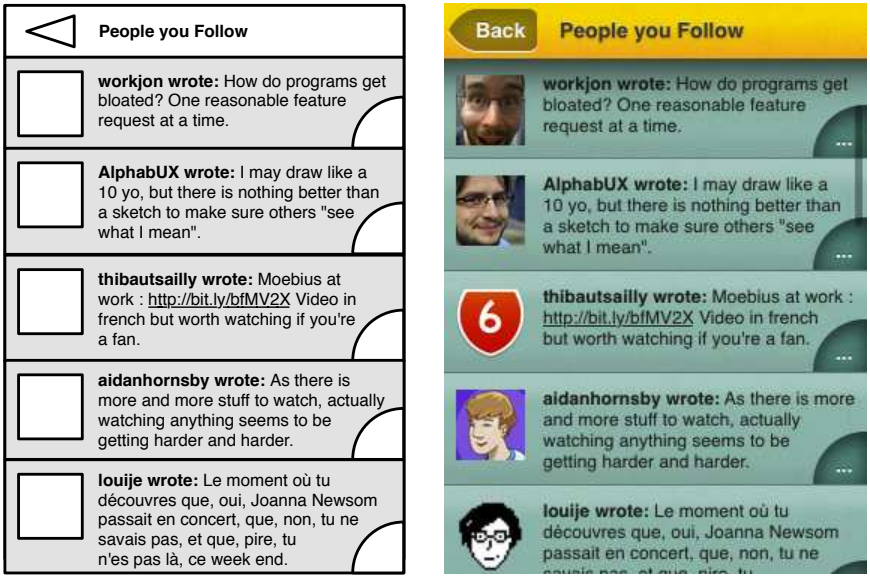


Figure 10.1: A sample wireframe (left) and mock-up (right)

that he can push something or to emphasize hierarchies, and colors to convey importance and draw the user's attention.

These are the kind of things you need to keep in mind when you do a detailed mock-up of how your screen should look.

You don't necessarily have to create mock-ups using a graphics application. If you prefer, you can do mock-ups in code. Simply use the medium that will be easiest for you when you have to make sweeping changes. But keep in mind that you are creating prototypes, not early versions of the final product. In *The Pragmatic Programmer* [HT00], Andrew Hunt and David Thomas write:

> With a prototype, you're aiming to explore specific aspects of the final system. With a true prototype, you will throw away whatever you lashed together when trying out the concept, and recode it properly using the lessons you've learned.

Even if you do mock-ups in code, the goal is still to explore ideas in a way that makes it easy to throw things away if they don't work out.

Terminology

In this book, I use the following terms:

Sketch	Any representation of a user interface as a drawing.
Wireframe	A static representation of a user interface where the individual elements are at least roughly where they are supposed to go, at their supposed sizes.
Mock-up	A (usually static) representation of a user interface where decorations like shadows and colors are introduced.
Prototype	Any representation of a product that is not the final product.

Not everybody uses these terms in that way. Sometimes, the term *prototype* denotes only interactive, high-fidelity representations. Sometimes, the term *mock-up* is used for any type of sketch of the final user interface. When you read about these things on the Internet or in other books, be aware that the author may use these words differently.

10.7 Tools

Sketching and prototyping are popular activities in the software development community—enough to have given rise to a whole ecosystem of products. The sole *raison d'être* for these applications and services is to help you sketch or prototype your product.

There are many good reasons for using such products. One of the main ones is that they make it easy to collaborate on designs even when designers live far away from each other.

Balsamiq[2] and Mockingbird[3] are online tools that allow you to create and share sketches of your user interface.

Google Docs has a drawing component called Google Drawings, which allows several people to collaborate on a design. Morten Just has a set of Google Drawings templates with user interface elements.[4]

2. At http://balsamiq.com.
3. At https://gomockingbird.com.
4. At http://mortenjust.com/2010/04/19/a-wireframe-kit-for-google-drawings/.

On the Mac, OmniGraffle[5] fills many of your user interface sketching needs. People have created user interface stencils specifically for OmniGraffle.[6] If you're an iPhone developer, check out Briefs[7] and Review.[8]

Some people also use tools like PowerPoint[9] or Keynote[10] to create mock-ups and prototypes. These tools even allow you to add simple animations and interactions. Sites like Keynotopia[11] and Keynote Kungfu[12] provide user interface templates for these applications. Templates can help you create pixel-perfect mock-ups that use a platform's standard user interface elements.

All of these tools can make your life a lot easier. On the other hand, there are good reasons for going with paper. It's faster, it's more natural, and as long as everybody is in the same room, it's easy to collaborate with other people; the only thing people need in order to contribute ideas is their own pencil. And since everybody has some kind of gadget with a built-in camera nowadays, sending the sketched user interface to somebody else is as easy as taking a picture and emailing it.

Just use what's easiest for you. If sketching on paper works best, sketch on paper. If you prefer to use an application specifically created for user interface designers, use that. If you want to use a graphics application instead, that's fine too. Or simply use all of these tools, depending on your current task.

Takeaway Points

- Start with a bird's-eye view and work your way down to the details, from flow diagrams to storyboards to simple sketches to wireframes and eventually to detailed mock-ups.
- Fix problems early. The earlier you notice a problem with your design, the cheaper it is to fix it.
- Flow diagrams help you make it as simple as possible for your users to reach their goals.

5. At http://www.omnigroup.com.
6. For example, at http://graffletopia.com/categories/user-interface.
7. At http://giveabrief.com.
8. At http://www.getreviewapp.com.
9. At http://office.microsoft.com.
10. At http://www.apple.com/iwork.
11. At http://keynotopia.com.
12. At http://keynotekungfu.com.

- Storyboards help you flesh out and communicate the interaction design of your product.
- Simple sketches help you figure out *what* to put on individual screens.
- Wireframes help you decide *where* to put things on individual screens.
- Mock-ups help you iterate quickly on the visual design.

Further Reading

Flow diagrams, storyboards, wireframes, and mock-ups are the four techniques I use most often. They allow me to start with a bird's-eye view and progressively move down to the details. Other designers, however, prefer a different combination of techniques. *Undercover User Experience Design* [BB10] by Cennydd Bowles and James Box teach you these and other techniques.

Bill Buxton's *Sketching User Experiences* [Bux07] is another great book on this topic, and Robert Hoekman also covers some of these techniques in *Designing the Obvious* [Hoe06].

Tyler Tate writes about different sketching and prototyping methods.[13]

13. At http://www.uxbooth.com/blog/concerning-fidelity-and-design/.