SMASHING
MAGAZINE

UX Design Process

Imprint

ISBN: 978-3-943075-58-8 (Version 1: January 2013)
Cover Design: Ricardo Gimenes
PR & Press: Stephan Poppe
eBook Strategy: Andrew Rogerson, Talita Telma Stockle
Technical Editing: Talita Telma Stockle
Idea & Concept: Smashing Media GmbH

About This Book

UX design isn't a new field. But sometimes new approaches lead to new perspectives. In this book, you'll learn how lean UX has made the whole discipline more approachable and attractive to startups, and you'll see that UX issues aren't just a quick fix, but should also address big-picture issues. Sometimes, the solution to a problem is just to fix a broken UX. At other times, you need to constantly fine-tune in order to keep up with changing demands. You'll also explore wireframing techniques, research planning and design bias.

TABLE OF CONTENTS

Interaction Design In The Cloud

BY ERIK PEROTTI ❧

Interaction designers create wireframes in tools such as Adobe Illustrator, OmniGraffle and Microsoft Visio. Originally, these wireframes were primitive shapes drawn to represent various UI elements. Many of us cannot imagine life without them.

There are, however, reasons to consider moving to the cloud to do interaction design. In short, today's cloud-based tools are:

- Optimized for collaboration,
- Editable anywhere,
- Interactive,
- Published in real time,
- Self-maintaing (the user doesn't need to update software),
- Payable monthly,

Emailing your old static designs will feel old fashioned once you see what these tools can do. Going a step further, there are tools for the user review process, too. Just upload your ideas, from simple mockups to final layouts, link them together, and share them for comment.

(Image credit: baldiri[1]*)*

This article walks you through the current selection of **cloud-based tools and provides some recommendations**. The number of offerings and amount of functionality are pretty vast. For the sake of brevity, we'll address two functions: prototyping and wireframing. But if you're intrigued, you might want to explore cloud-based image editing, mind-mapping tools and other UX activities. These tools are already out there, and surprisingly good.

Prototyping

For our purposes, prototyping involves uploading images (screens) and linking them together via hotspots. Once these are set up, the prototype is published and available to reviewers for usability testing, commenting or both.

Review criteria

Here are the **fundamentals that a product should support** in order to compete in this space:

- Quick uploading process,
- Support for several source image file formats,
- Easy linking between pages,

---

[1] http://www.flickr.com/photos/baldiri/5735003580/sizes/l/in/photostream/

- Support for feedback from end users.

  Some items aren't available as of this writing:

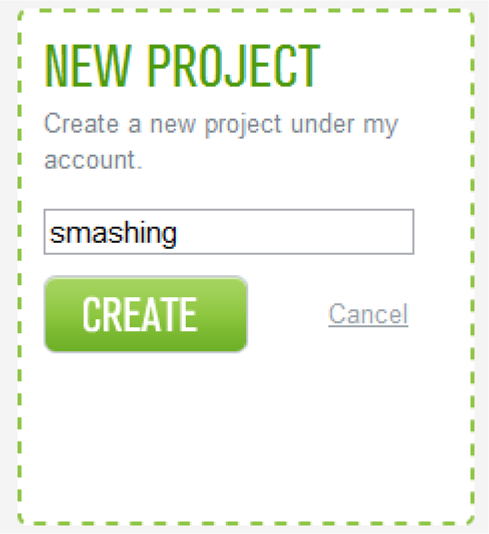
- The ability to nudge images in line without having to recreate them;
- The ability to create interactive objects and layer them (such as a menu bar that appears on every page).

INVISION

What it does

Create your screens in your favorite tool and upload them to InVision[2]. Then add hotspots; a hotspot links to another page. This is great if you live and die by the comp (Photoshop file). For example:

1. Create a new project. Think of a project as a collection of previously generated comps that you are going to tie together as a prototype via InVision.



NEW PROJECT

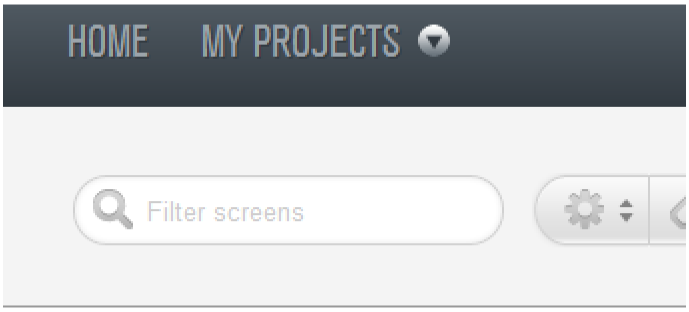
Create a new project under my account.

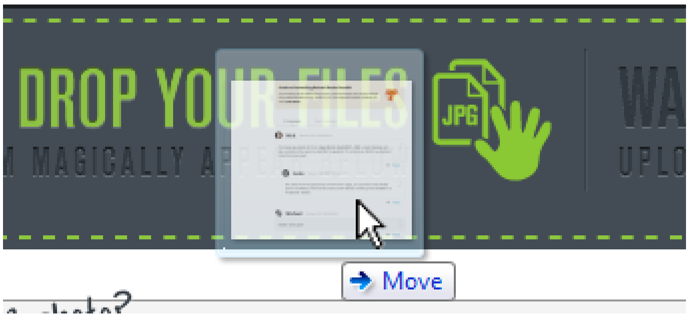
smashing

2. Upload your files to this new project (the images in this article are PNGs).

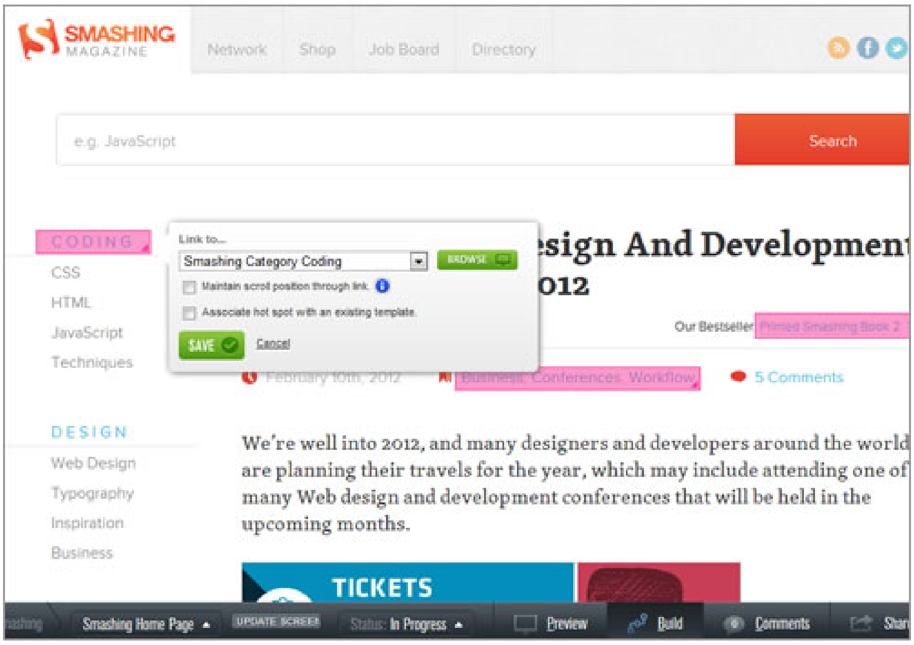
---

[2] http://www.invisionapp.com





3. In “Build” mode, create the hotspots. Basically, you are linking together the prototype. If you have all of the collateral that you need, this will go quite fast—exactly as you’d want it to work.



Link to...
Smashing Category Coding
BROWSE
☐ Maintain scroll position through link.
☐ Associate hot spot with an existing template.
SAVE
Cancel

Status: In Progress

esign And Development
012

We’re well into 2012, and many designers and developers around the world are planning their travels for the year, which may include attending one of many Web design and development conferences that will be held in the upcoming months.

Observations

The application **works as advertised**. It enables the user to quickly wire up prebuilt comps, wireframes and sketches. The tutorials also explain useful actions, such as creating hotspots that will be the same on multiple pages (these are called “templates” in InVision).

Speaking of templates, they expose both a major advantage and a major disadvantage of this tool: if the uploaded images are not placed perfectly, then the templates will not line up properly. One would want the ability to adjust the x and y coordinates of any image so that they line up perfectly without having to change the source files. On the upside, if you've done the prep work right or you've made your hotspots large enough, you can fudge this a bit, and the templates really accelerate the build process.

A number of usability issues have made me scratch my head. For example, the first time I tried adding a hotspot to the search input field, the "Link to…" modal dialog was off to the left side of the browser, which made it impossible to save or cancel the dialog. I then tapped the "Update screen" at the bottom of app to refresh the screen. It turns out that in InVision speak, "Update" = "Replace." I was afraid to refresh the browser because there is no indication of whether the application saves automatically. So, in the end, I switched to "Preview" and then back to "Build."

Once you're familiar with the quirks, however, the application is useful. If you're a designer or want to work offline to generate wireframes, then give this app a hard look.

- **Upload process**
  Drag and drop, or browse the file system
- **Supported file formats**
  JPG, PNG and GIF
- **Linking pages**
  Easier than the others tested because of templates
- **User feedback**
  Easy, nested
- **Marquee clients**
  eBay, Google, Intuit, Whole Foods and many others. Very impressive.

You can taste the fruits of my labor[3]. Remember, being online means it's delivered to whoever, wherever, whenever it's done. Feel free to add comments.

FIELDTEST

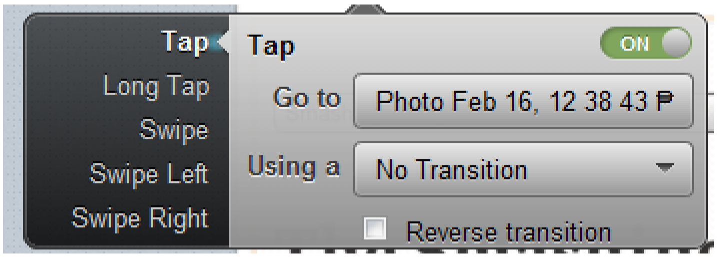
In spirit, FieldTest[4] (in private beta) serves the same space as InVision. The designer uploads prebuilt comps, wireframes and the like to FieldTest, ties them together, and then publishes them for review. One advantage is that FieldTest **leverages device gestures**. In short, you can

---

3. http://invis.io/3R1UL1JG
4. http://www.fieldtestapp.com

"play" FieldTest prototypes on your iOS, Android or Windows Phone 7 device and have it respond to gestures. Combined with the built-in screen transitions, this is a powerful function for mobile app designers.

As with InVision, screens are grouped into "prototypes" (projects). Including them in a project means that they can be linked to and from other screens. The process is the same, too: create the prototype collateral, link it together via hotspots, and publish it for review. For comparison's sake, here are the hotspot configurations for the two apps.

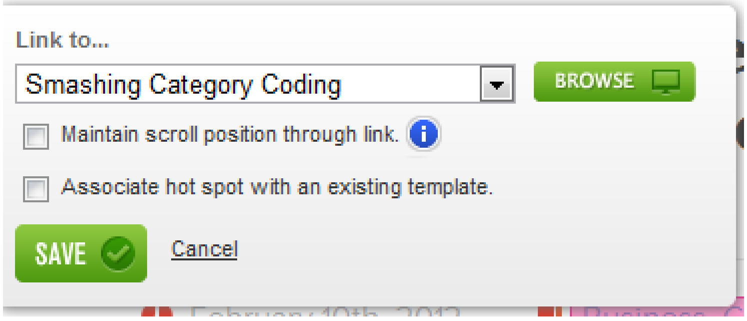


Tap ON

Go to: Photo Feb 16, 12 38 43 P

Using a: No Transition

☐ Reverse transition


Link to...

Smashing Category Coding

☐ Maintain scroll position through link.

☐ Associate hot spot with an existing template.

This demonstrates the differences in approach. On the top is FieldTest. It allows a user to choose between gestures (the prototype I built was an iPhone app). The gestures are tap, long tap, swipe, swipe left and swipe right. Multiple gestures can be active for the same hotspot, which is particularly cool and gives a **realistic experience** of various actions. On the bottom is InVision, whose ace is templates. The author can create a template for several controls that appear together, and they can reuse that template on several screens.

Observations

If I were to choose between these prototyping tools, FieldTest would be my choice, largely because I build mobile applications. Having listeners for multiple gesture types makes for a more realistic prototype. If the app were Web-based, then InVision is more mature.

FieldTest still has work to do, though. In the beta, gestures such as up and down are missing. Templating as InVision does is really useful. It streamlines the addition of hotspots. Another area for improvement is in comments, and allowing a prototype's end user to provide feedback.

There are other usability nits. For example, FieldTest includes a status bar at the top of each screen. I have yet to figure out why someone would want this, and it's not optional. So, if you take a screenshot on an iPhone, you'll have to edit it to remove this status bar, only for FieldTest to put it back.

Try it out for yourself on the prototype built for this review[5]. Please note, there is no down gesture, so if you want to try that, gesture from right to left (like when advancing through pictures in iPhoto).

- **Upload process**
  Browse the file system
- **Supported file formats**
  JPG, PNG and GIF
- **Linking pages**
  Fairly easy
- **User feedback**
  Enables gestures on the device, which is great.
- **Marquee clients**
  In private beta

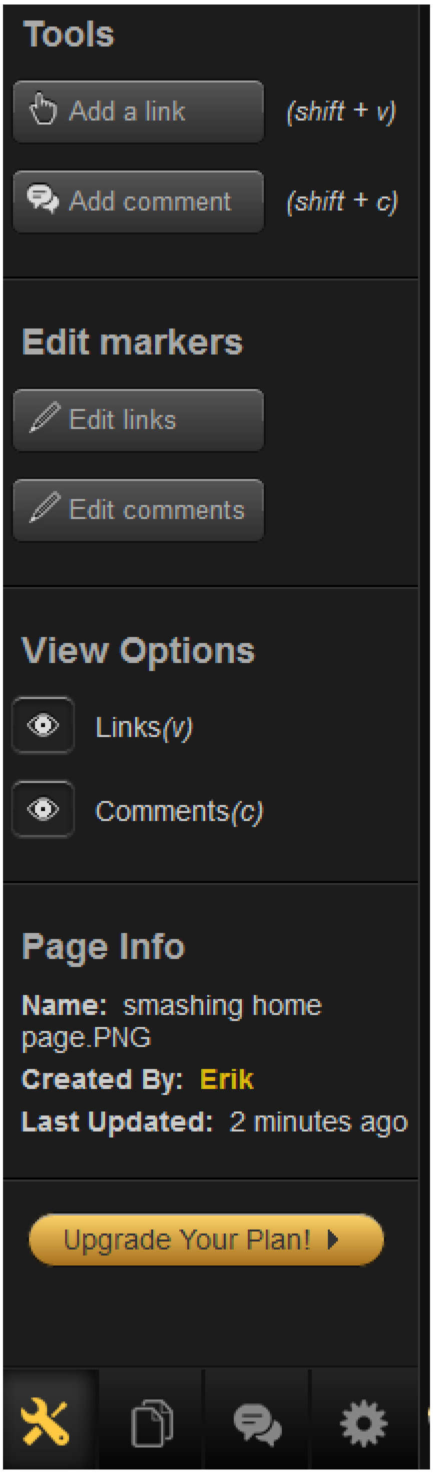
CLICKDUMMY

ClickDummy[6] is another competitor in this space and has the **same process as the others**. The user uploads materials and then links them together through hotspots. The link function is a "tool" contained in a drawer (i.e. a UI element that slides in and out from one side of the screen).

---

[5] https://go.fieldtestapp.com/13d3izk

[6] http://www.clickdummy.com



Name: smashing home page.PNG
Created By: Erik
Last Updated: 2 minutes ago

Observations
This drawer seems innocent enough, but it creates unnecessary hurdles for the user. In an attempt to simplify the problem, it has added confusion and multiple steps to an easy process. How? The user has to toggle between this tool drawer and the page-picker drawer a lot. The page picker also has to be overloaded in order to provide both functions (selecting a page, as in navigation, and selecting a page, as in a hotspot target).

A second issue: the website says that the user can drag and drop images onto the pages drawer. This doesn't work in my (Chrome) browser. It instead replaces the current page with the image. After a panicked "Backspace," the user is restored to their project but has lost their location and has to start over.

Another point: this all-important drawer is closed when the app launches. It took about five minutes to determine that the app was working, and this after weeks of looking at apps in this space.

Lastly, unlike both of the apps reviewed above, this one has no compelling feature that makes the additional effort worth the time. In future, hopefully, the addition of some product differentiation, combined with a rework of the primary use case, would make this application worth another look.

You can see the output from this exploration[7] for yourself.

- **Upload process**
  Drag and drop, or browse the file system.
- **Supported file formats**
  JPG, PNG and GIF
- **Linking pages**
  Most difficult of those tested
- **User feedback**
  Easy to test, but comments require registration
- **Marquee clients**
  Not provided

Wireframes

Think of a wireframe as a black and white low-fidelity screen mockup. The mockups I create also include call-outs to give the development team additional context.

In the process, the user will create an account, create a project, and then land on a blank screen. The user then drag and drops UI controls (radio buttons, text input fields and so forth) onto the page.

Once the project is saved, a permalink can be given out for people to see your work. If you change a screen, it will auto-magically show your updates the next time that URL is opened (or refreshed) by a team member. This last point **is what the cloud is all about**: everyone al-

---

[7] http://clickdummy.com/projects/nayres

ways has the same (i.e. current) version of your work. Changes are instantaneously available whenever the wireframe is saved.

Compared to most offline tools, the library of available objects is focused on low-fidelity UX. Don't expect to create gradients or to use a pencil tool.

Review criteria
Here are some basics that are fairly universal in my experience:

- **Robust set of standard UI controls**
  If the tool doesn't have off-the-shelf drop-downs, toggles, cover flows and the rest, then creating those controls will require additional work.
- **Good as a documentation medium**
  Plan on your development team using your wireframes to dictate the logic and layout of the application.
- **Good for making wireframe clones, templates or whatever you want to call them**
  Not surprisingly, all of the iPhone wireframes I create have the app's name at the top. I want to do this on the first wireframe and not have to do it again.
- **Responsive**
  It all takes place in a Web browser. If the application is slower than a locally running application, then your productivity will suffer. Case in point: a year or two ago, I created about 50 wireframes for a project. Each page took a minute to load. To see my changes reflected, another minute. Trust me, this gets old quickly.
- **Not written in Flash**
  "Dear development teams who write these tools: I love Flash, Flex and the rest. Not as much as I love my iPad, however. I want to edit my work across form factors. It's all drag and drop, right?"

Here's what you won't see right away from the tools out there:

- **An extensive stencil library or the ability to easily create and reuse stencils**
  OmniGraffle excels at this. Don't expect Yahoo to create a stencil library for Mockingbird anytime soon.
- **A wide user base**
  Momentum is building, and there are believers. This is still a minority position and will be for some time. I would say customer support is

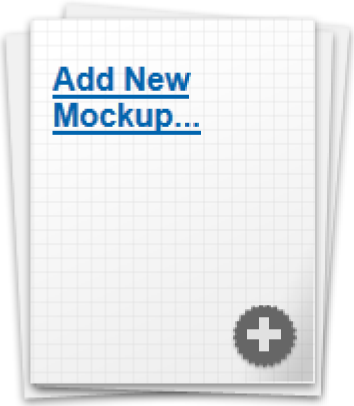
great, but the more people use these tools, the better the tools will become.

- **Font selection**
  I won't dwell on this, but you can tell there is still some lively debate about what a wireframe should and should not communicate just by looking at what features are included in any given product.

BALSAMIQ

As with the prototyping tools, wireframes—or "mockups" in Balsamiq[8]-speak—are organized into projects. From there, things change. Tools like InVision and FieldTest assume that you have created your pages or screens in another tool. In Balsamiq (and Mockingbird, discussed next), the tool is designed for wireframe creation, with extremely limited functionality for prototyping.

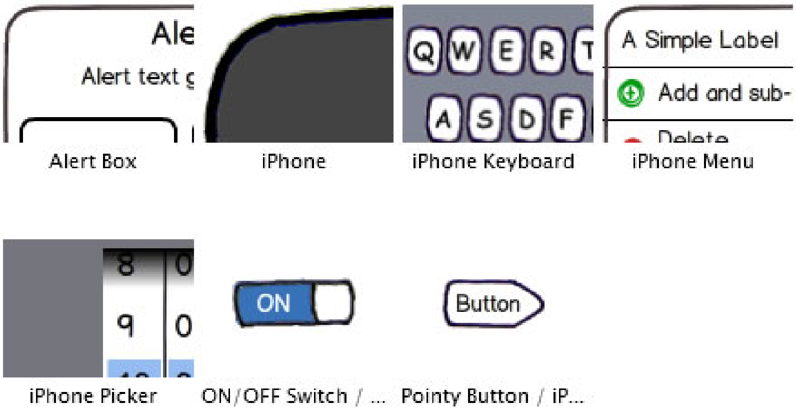
1. Create a new mockup.



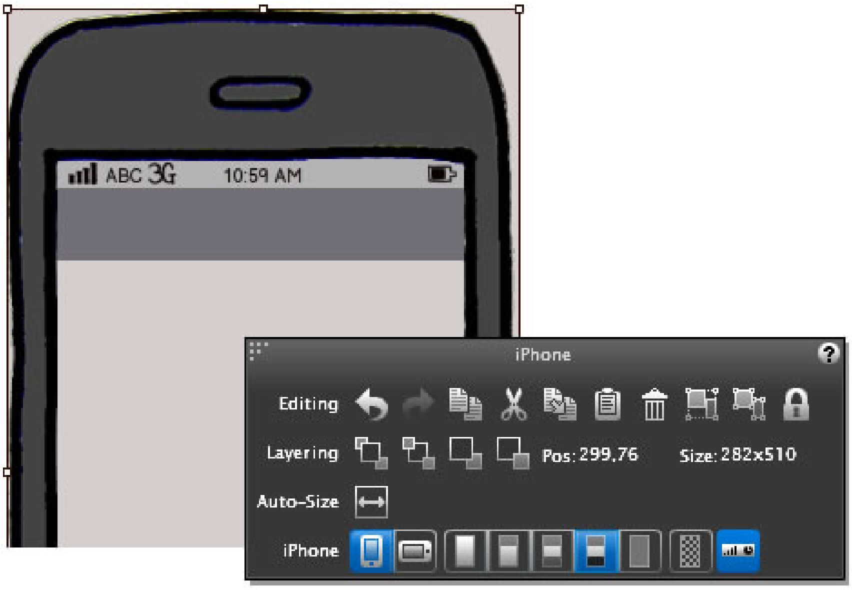
2. Drag and drop an off-the-shelf UI control from the ones available.

---

[8] http://www.mybalsamiq.com



3. Configure the control to your needs. This is noteworthy, because Balsamiq **provides a number of important options**. For example, there is one toggle to put the iPhone in landscape orientation instead of portrait.



4. Add the rest of your UI controls; document for the development team; and publish.

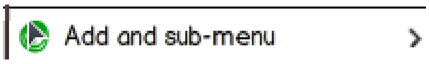
Observations

Having worked with some other tools, I've become a fan of Balsamiq. A great UI control library and easy object configuration are two areas where this tool excels. There are some areas for improvement, though. First, and I'm sure the development team is tired of hearing it, the sketching style is fine for those who understand low-fidelity mockups, but you probably wouldn't want to show the mockups to your CEO.

A second gripe is that the editing tool is built in Flash, so work is limited to platforms that support it.

On the upside, a few non-obvious pros:

- The icon set is great. I've noticed that only one icon is not in the box: Bluetooth. Anything else I've needed has been available.
- In addition to drag and drop, there's a great quick-add feature. After typing in a few characters of the name of a UI control, a filtered list appears, allowing you to add controls quickly.
- Balsamiq has an odd markup language that, once mastered, allows the user to add common things. For example, `+ Add and sub-menu,` `>` translates to:



And here's the rundown:

- **UI controls**
  More than 70, including iPhone-specific
- **Good for documentation?**
  Call-outs are one of the controls; drag and drop them onto the canvas.
- **Good at duplicating screens?**
  Yes.
- **Responsive?**
  Yes. You will forget you are working in the cloud.
- **Written in Flash?**
  Yes.

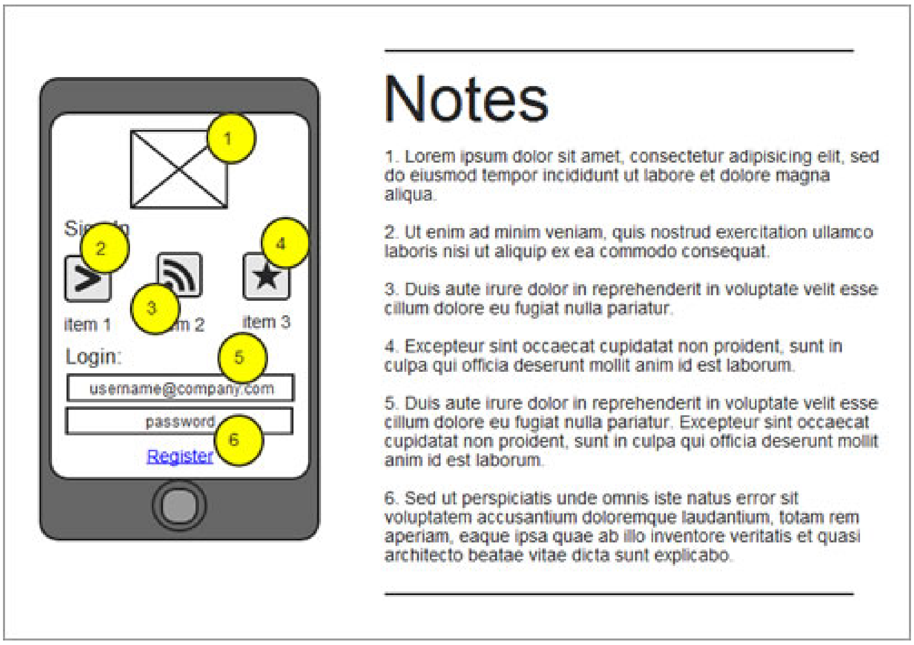
MOCKINGBIRD

Mockingbird[9] is also a wireframing tool, and **a good one** at that. In some ways, it compares favorably to Balsamiq: Mockingbird's editor isn't Flash-based; it uses an unobtrusive font; and adding UI controls is (almost) comparable to Balsamiq.

The process is similar, too. Here's the outcome:

---

[9] http://gomockingbird.com



Notes

1. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

2. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

3. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

4. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

5. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

6. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Observations

More professional, right? On the surface, it is more polished, but there are some subtle shortcomings. For example, one cannot left-justify text in an input field. Also, I couldn't get the icons to all be exactly the same size (36 pixels). And so forth.

There are some logistical hurdles as well. Many of the controls are primitive. To add a call-out, like ones in yellow above, you actually have to add two objects: the yellow circle and the black text. And when a control is added via the quick-add function, the filtering text is not cleared, so after every addition, one has to clear the previous query. Put practically, this mockup took about four times as long to create as the Balsamiq version.

- **UI controls**
  Fewer than Balsamiq, and no mobile-specific controls.

- **Good for documentation?**
  Call-outs are created with circles and overlaid text.

- **Good at duplicating screens?**
  Yes.

- **Responsive?**
  Mostly—don't use Chrome.

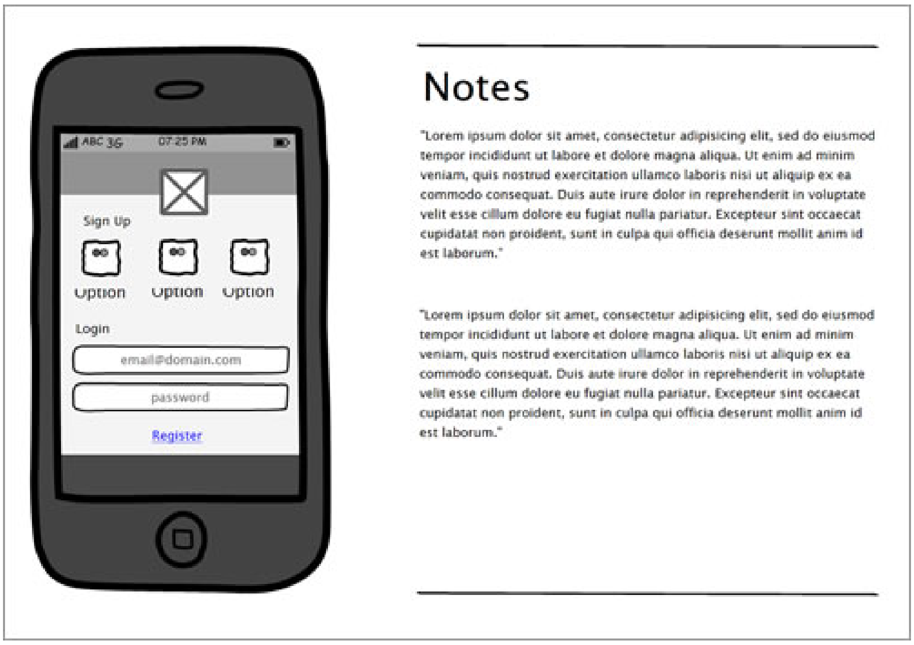
- **Written in Flash?**
  No.

MOCKUP BUILDER

Another entry in the wireframing space is Mockup Builder[10]. Functionally, it lies somewhere between Mockingbird and Balsamiq. It has a fairly good library of controls—in fact, it's the only cloud-based solution with native Android controls (Ha!). Moreover, I find its aesthetic better than that of competitors.

Like the others, Mockup Builder starts with a blank canvas, and the user drag and drops controls onto the canvas for configuration.

Here's the mockup created for this review:



Notes

"Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

"Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Again, the mockup is fairly clean, but there are some issues: the icons use some funny clipping, and they do not scale properly. The user cannot toggle the various defaults for the iPhone, such as the gray bars at the top and bottom of the screen.

Observations

This tool is a little too buggy for everyday use. For example, the notes to accompany illustrations are in Lorem Ipsum text. Also, when copying text from the Web and pasting into a multi-line text area, the text does not wrap to the control's width—meaning that the text shows exactly one line, and the user has to use the control's handles to wrap it. I also wanted to show two paragraphs of text but could not figure out how to insert a "Return" in the text.

Another grievance: the tool could use more polish. For example, the screen surface on the iPhone control is narrower than the keyboard, so

---

[10] http://www.mockupbuilder.com

the user has to resize the keyboard by hand. When that's done, the "e" is missing in the button. I understand that these are minor, but one would expect these t's to be crossed off before moving away from a beloved tool like OmniGraffle.

- **UI controls**
  More than the others, including iPhone- and Android-specific ones
- **Good for documentation?**
  Call-outs are one of the controls; drag and drop them onto the canvas.
- **Good at duplicating screens?**
  Yes.
- **Responsive?**
  Yes.
- **Written in Flash?**
  No.

Conclusion

Cloud-based tools are now available and **well designed for UX work**. Many of the features in the offerings above are not available in software running locally on your machine. While this space is still growing, I've been working in the cloud for the past two years and cannot imagine going back.

Collaboration is instantaneous, and the tools are optimized for the right activities: wireframing and testing with users. In fact, these apps have several unexpected and delightful features, and you might find yourself walking away from your favorites, too.

Of course, there are valid reasons to avoid working in the cloud. Stay with your old standbys if any of the following apply:

- **Your IT department disapproves.**
  This is a hot-button issue. All of these tools protect your information, but this is still worth considering.
- **You expect the polish of offline tools.**
  These tools are for early adopters. Still, they are Web applications, so they will evolve. That's what happens on the Web. You'll just wake up one morning to find some annoyance removed or a key feature added.
- **Your project is *big*.**
  If you plan on a 200-screen flow, you will feel a steady degradation in

performance. That said, I’ve just completed a 70-pager with one of these tools and was just starting to notice some minor falloff.

- **You think in terms of “deliverables,” with a complete set of create-once mockups.**
  Many of these tools have coauthoring functionality (if the roles are set up that way). In my experience, however, no one has actually changed anything, even if I wanted them to.
- **Your Internet connection is a problem.**
  But that’s not to say that you’ll lose data whenever the network is interrupted.

These could be a deal-breaker for some. But these tools **are free to try**, and some are so simple that you might get hooked in five minutes (as I did a few years ago). Almost all of the research for this article was done with free trials. Given the ease with which you can try these out, you have every reason to go out and see whether one or more is right for you. ❧

Lean Startup Is Great UX Packaging

BY TOMER SHARON ❧

When Albert Einstein was a professor[11] at Princeton University in the 1940s, there came the time for the final exam of his physics class. His assistants passed the exam forms to the hundreds of students, and the hall was dead silent. One of the assistants suddenly noticed something was wrong.

She approached Einstein and told him that a mistake had been made with the exam form and that the questions were the same as those in the previous year's exam. Einstein glanced over the exam form and said that it was OK. He explained that physics had changed so much in the last year that the answers to the questions were now different.

The **lean startup movement**, like Einstein's physics exam, talks about the same things that UX people have talked about for decades. The difference is that **people are now listening**. Lean UX is an approach that quickly followed the lean startup movement. It is not a new thing. It's just a new name for things that were always around. The difference is in the packaging of these ideas.

One other factor that has changed dramatically is the audience. Entrepreneurs and startup founders have always been asking themselves how to develop great products. The answer that UX practitioners, usability professionals and UX researchers have been giving them was too complicated. UX people (me included) have been using disastrous jargon that only we understand. We have been talking about usability tests, personas, field studies and areas of interest in eye-tracking studies.

The lean startup answer to the same question uses plain language that people understand. When I say, "We need to conduct a contextual inquiry," I usually get a deer-in-the-headlights reaction. When a lean startup person says they are "getting out of the building," it is a whole different story. We mean the same thing; we use different words.

Does it matter? I think it does. Who would have thought that startup companies would be looking for UX people and UX founders, and would become interested in doing usability testing, iterative design and customer interviews?

---

[11] http://oaks.nvg.org/sa5ra17.html#einstein-anecdotes

This article takes the **principles of the lean startup and suggests their UX research equivalents**. Hopefully, it sheds some light on why the lean startup concept is so very well accepted in the entrepreneurial world and why startups suddenly want to do UX research and design.

Validated Learning And Usability Testing

The lean startup movement claims that startups exist not just to make stuff, but to learn how to build sustainable businesses. This learning can be validated scientifically by running frequent experiments that enable entrepreneurs to test each element of their vision, as outlined by Eric Ries in his book *The Lean Startup*[12]. In my interview with Ries (embedded below), the most familiar voice of the lean startup movement, for my book *It's Our Research*[13], he calls for entrepreneurs to double-check their assumptions to verify that they are right. He determines that validated learning exists to help entrepreneurs test which elements of their vision are brilliant and which are crazy.

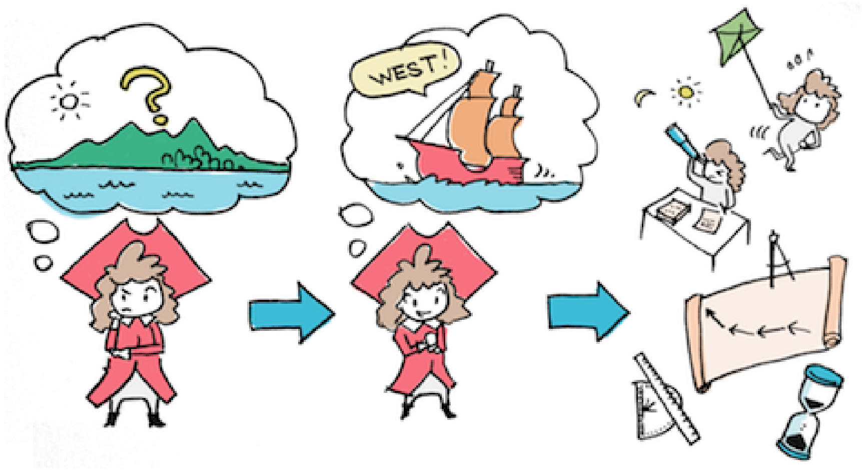
In the UX world, we call in the product development people to evaluate their design assumptions in usability tests. We urge them to ask users to complete tasks while using the think-aloud protocol and to identify usability problems.

An interview with Eric Ries[14] *about getting stakeholder buy-in for UX research and how it relates to the Lean Startup ideas.*

---

[12] http://www.amazon.com/The-Lean-Startup-Entrepreneurs-Continuous/dp/0307887898/
[13] http://itsourresear.ch
[14] http://player.vimeo.com/video/36485988?title=0&byline=0&portrait=0

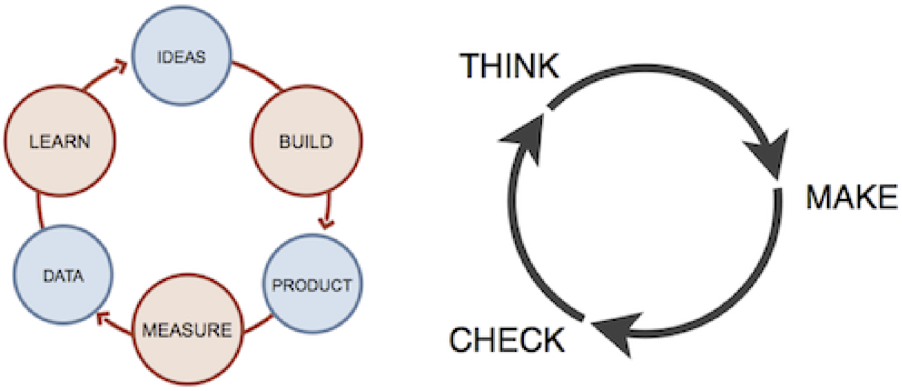
When entrepreneurs hear “validated learning,” they can see the benefit. They understand that this concept refers to proving or disproving their assumptions. When they hear “usability testing,” they associate it with a time-consuming, money-eating, academically oriented project.



Validated learning: You believe you'll find a new continent if you keep sailing west. So, you test your idea and verify the route using scientific methods and measurements.

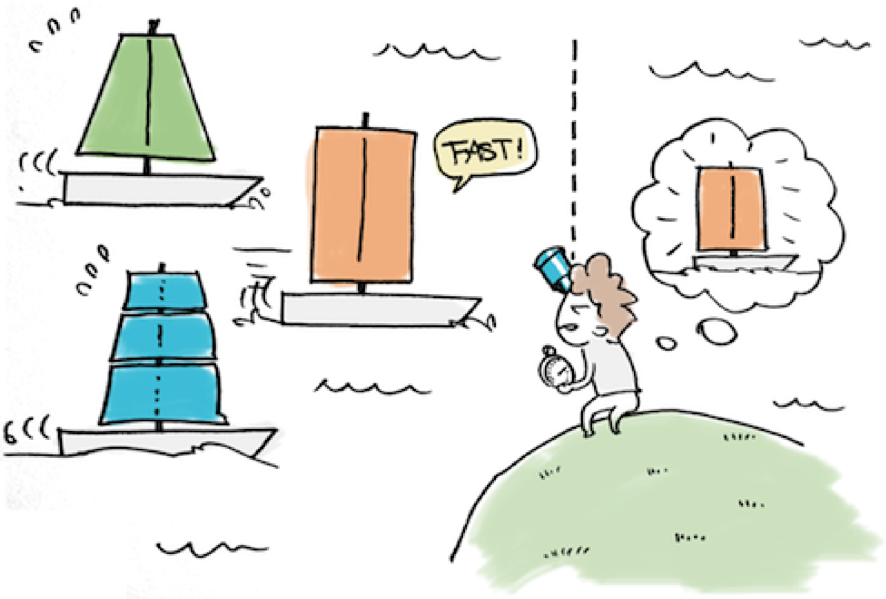
Build-Measure-Learn And Think-Make-Check

The fundamental activity of a startup is to turn ideas into products, to measure how customers respond and then to learn whether to pivot or persevere. All successful startup processes should be geared to accelerate that feedback loop. As Ries explains, the feedback loop includes three primary activities: build (the product), measure (data) and learn (new ideas).



Eric Ries's Build-Measure-Learn feedback loop and the Think-Make-Check UX cycle.

The lean UX approach calls for a slightly different cycle: Think-Make-Check. The difference, according to Janice Fraser (cofounder and first CEO of Adaptive Path), is that this latter feedback loop incorporates your own thoughts as a designer, not just ideas learned through measurement. Janice, who now leads LUXr[15], indicates that the pattern of a lean startup is an endless loop consisting of two steps: Prove-Improve, Prove-Improve, Prove-Improve. This means that you design something, learn about it, make it better, learn again and so on. There is no room for people who are afraid to put their creations on the line for testing. These two feedback loops are very similar and are making a lot of sense to people in both the entrepreneurial and the UX worlds.



Build-Measure-Learn: How do you build the fastest ship? You try to build and test your hypothesis; you measure the result; and then you learn new knowledge that you can bring to your next ship design.

MVP, And "Test Early And Often"

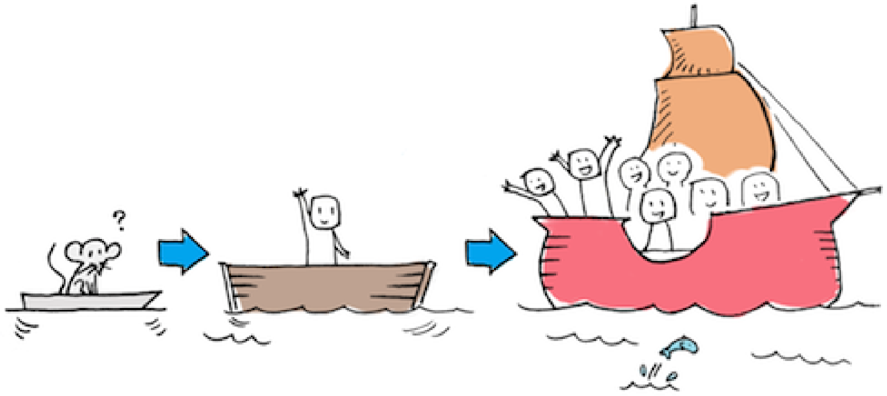
The minimum viable product (MVP), as Ries explains it, is a version of the product that enables a full turn of the Build-Measure-Learn loop with a minimum amount of effort and the least amount of development time. How many times have UX people told their stakeholders that for every dollar spent on solving a problem during product design, $10 would be spent on the same problem during development, and $100 if the problem had to be solved after the product is released?

We've known for years that product prototypes are to be evaluated early in the development process (not just prior to launch). We've also known that these evaluations are most valuable if they are repeated

---

[15] http://luxr.co

throughout the process. The MVP is, in fact, an early prototype that serves as a tool to learn and test the team's assumptions.



MVP: You want to build a huge ship, but instead of building the ship right from the beginning, you start by testing your idea with minimal design to see if it floats.

Pivot And Iterate

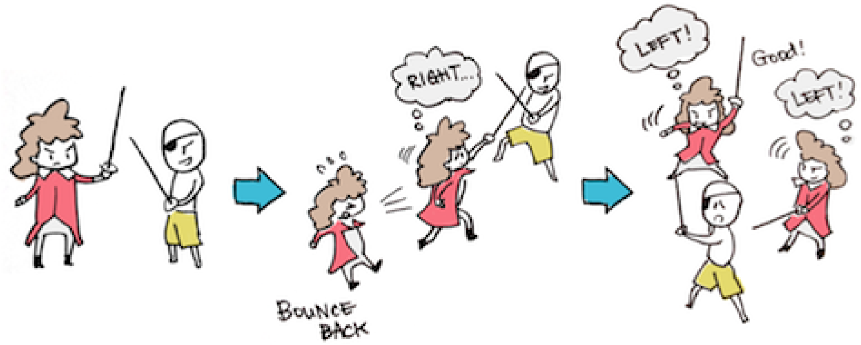
To use the analogy of a basketball "pivot," one foot of a business is always firmly rooted in what the team has learned so far, while the other foot is moving and exploring new ideas for the business. Instead of taking the big risks of developing something huge, **lean startups take small steps forward**, developing things and pivoting to better directions. This way, if they fail, the fall will be less painful and will allow them to bounce back and continue. On the other hand, if they had climbed a big cliff, the potential fall would be deadly.

This reminds me of why we pitch for an iterative design process or for using the RITE methodology (rapid iterative testing and evaluation). Many product development decision-makers feel that the best time to conduct a usability test is near launch time, when things look good and are "ready" for users to play with. Many UX research practitioners know that when they **agree to conduct a usability test right before a product is launched**, especially if this is the first usability test for the product, the following is most likely to happen:

1. The study will result in a long list of defects (i.e. usability problems);
2. The product team will be presented with a long list of issues exactly when they are trying to shorten the list of issues;
3. Only the easiest problems to fix will be taken care of;
4. The most important problems will be ignored and the product will be launched;

5. By the time the team is ready to start working on the next version, there's already a long list of new features to be developed, leaving the usability issues low down on (or off) the priority list.

The solution to all of this is to adopt an iterative design process that involves fast rounds of small-scale usability tests. Jakob Nielsen has been preaching this[16] for years now. And then along comes Eric Ries, who talks in the most natural way about pivoting companies, directions, customer segments and design. People don't iterate, they pivot.



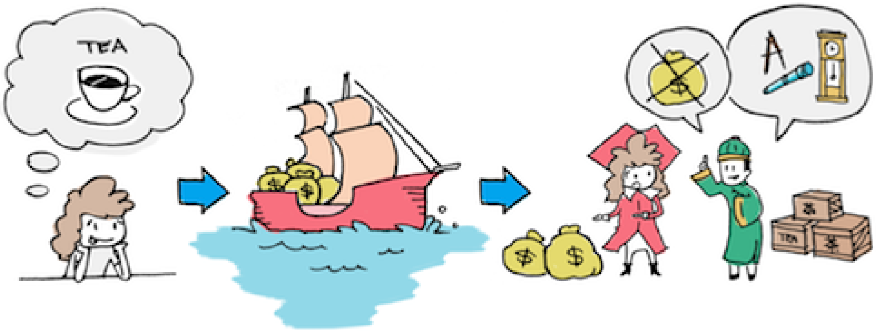
Pivot: You want to defeat your opponent, but it is difficult to win instantly by launching a full-scale attack in one shot. The proper way would be to advance and attack step by step, always keeping one foot on the ground and ever ready to bounce back in case an attack is not successful.

Customer Development And Fieldwork

The term "customer development" was coined by Stanford University professor Steve Blank, one of the fathers of the lean startup movement. Customer development means developing your own understanding of who your customers are, what they are like and what their needs are. This is done through an approach guided by the mantra "Get out of the building." This mantra urges entrepreneurs to interview potential customers, to observe them in their own environment and to try to make sense of it. What a revelation to our UX research ears, huh? We UX people have been getting out of the building for a living for decades now. We call it by different names: ethnography, fieldwork, generative research, exploratory research, discovery research, user research, design research. Phew!

---

[16] http://www.useit.com/alertbox/20000319.html



Customer development: You want to trade with a country in the Far East. However, when you finally get to talking with the people of the country, you realize that they prefer to trade with your scientific equipment rather than your gold coins.

The Bottom Line

The lean startup movement, like the story of Einstein's physics exam, talks about the same things that UX people have talked about for decades. The difference is that people are now listening. The lean startup movement, followed by the lean UX approach, did not reveal any new UX concept. Lean startup thought-leaders do a terrific job and do an awesome service to UX people who struggle to get buy-in for design thinking and UX research.

The secret sauce of lean startup people is that they advocate for user experience research and design as one of the primary solutions to their business problems, and they do it using plain language. I highly encourage UX practitioners to closely monitor the developments and thought-leadership in the lean startup world to see how they can use what they learn in their own organizations, "lean" or not. ❧

Further Reading

BOOKS

- The Lean Startup[17], Eric Ries (Crown Business: 2011)
- The Startup Owner's Manual[18], Steve Blank and Bob Dorf (K&S Ranch: 2012)
- The Entrepreneur's Guide to Customer Development[19], Brant Cooper and Patrick Vlaskovits (2010)

---

[17] http://www.amazon.com/The-Lean-Startup-Entrepreneurs-Continuous/dp/0307887898/
[18] http://www.amazon.com/The-Startup-Owners-Manual-Step-By-Step/dp/0984999302/

- Running Lean[20], Ash Maurya (O'Reilly Media: 2012)

VIDEOS

- "An Interview With Eric Ries[21]," Tomer Sharon (18:06 minutes) About UX research and the lean startup movement.
- "An Interview With Steve Blank[22]," Tomer Sharon (18:01 minutes) About UX research and customer development.
- "A Conversation With Lane Halley[23]," Tomer Sharon (19:05 minutes) About lean UX.
- "An Interview With Jeff Gothelf[24]," Tomer Sharon (8:21 minutes) About lean UX.
- "Crushing the Boulder: User Experience and the Lean Startup[25]," Janice Fraser, MX 2011 conference (40:37 minutes).
- "The Lean Startup[26]," Eric Ries, Authors@Google (58:09 minutes)

Illustrations by Calvin C. Chan[27], (@calvincchan[28]), UX designer, Hong Kong.

---

19. http://www.amazon.com/Entrepreneurs-Guide-Customer-Development-Epiphany/dp/0982743602/
20. http://www.amazon.com/Running-Lean-Iterate-Works-Series/dp/1449305172
21. http://itsourresear.ch/ries.html
22. http://itsourresear.ch/blank.html
23. http://itsourresear.ch/halley.html
24. http://itsourresear.ch/gothelf.html
25. http://mxconference.com/2011/videos/janice-fraser-2
26. http://www.youtube.com/watch?v=fEvKo90qBns
27. http://www.calvin-c.com
28. http://www.twitter.com/calvincchan

Fitting Big-Picture UX Into Agile Development

BY DAMON DIMMICK ❧

The rapid pace of UX design in the agile world can lead to shortsighted design decisions. Focusing on addressing the immediate needs of particular user stories within the limits of a sprint can lead to neglect of larger design questions, which can come back to haunt UX designers later.

Sometimes, UX practitioners just need some time to work through big design issues that don't fit neatly into an existing user story or an individual sprint. This article will explore one answer to these problems—namely, design spikes, an agile approach that I have developed for large projects. Design spikes, which are bubbles of time that allow designers to focus on complex UX issues, can fit comfortably within the scrum framework and can be an effective tool for designers who have holistic design questions whose answers could potentially invalidate the work being tackled by the team.

Tunnel Vision And The Problem With Sprint Zero

Since the wide adoption of scrum as a development methodology, designers have struggled to fit design practices into its framework. Scrum, with its iterative work ethic focused on a releasable product at the end of each sprint, tends to lead designers to focus on very small design questions that can feel divorced from a cohesive design vision or interaction concept. The result is design tunnel vision.

Designers have struggled with the need to step back and work on holistic concepts while at the same time supporting development teams that are actively moving forward on bits of functionality that, from a design point of view, cannot be easily divorced from the whole. For many designers, the process often feels like an effort to design an engine, wheel or windshield without really knowing in what kind of car these parts will need to be integrated.

To deal with this problem, agile organizations have come to rely on the concept of sprint zero, a preliminary sprint dedicated exclusively to preparing for the first sprint, during which design teams are generally free to address initial design concerns. For complex projects, however, **sprint zero may simply not be enough to create concepts**, perform research and imagineer complex systems. More importantly, as a one-time event that exists only at the start of the process, sprint zero assumes that the need for a design-centered thought bubble is discrete, as opposed to an ongoing concern during product development.

Although any sprint can theoretically aim to address design issues, the pressure on designers to not hold up the rest of the development team once normal sprinting has begun is enormous. However, the time needed to explore very complex design decisions could drastically slow down team momentum. Making short-term design decisions based on limited understanding could end up invalidating valuable chunks of team output, an outcome that should be avoided if at all possible.

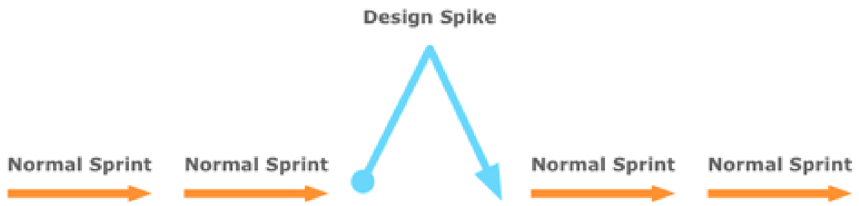
Development teams have long had a reliable solution for taking extra time to work on thorny or ill-defined obstacles: the spike. Spikes, in their most basic forms, are time-boxed periods that are used for exploration, research and sometimes prototyping. The purpose of a spike is to explore information necessary to understand a development approach or a requirement or to make a story estimate more reliable.

With a few modifications, the spike construct can be adjusted to serve the needs of design teams while maintaining the overall structure of scrum and preserving momentum. I call it a design spike—a method by which design teams can address complex UX issues within the confines of the scrum process.

What Is A Design Spike, And What Makes It Different?

A design spike is a bubble of time during which the designers and potentially other team members focus primarily on design questions. Design spikes can take place at the start of a project or anytime during the normal scrum process, but the introduction of a design spike changes the nature of the working scrum team temporarily. A single project may have multiple design spikes, as many as deemed necessary by the team.

When a design spike is called, any development work that could be affected by the spike temporarily ceases. This is because design spikes *inherit and create* obvious dependencies in the agile process.



Design spikes drop into the normal scrum process and temporarily take the place of regular sprints.

Members of the development team who can move forward with work that would not likely be affected by design decisions may do so, pursuing non-dependant stories from the backlog during the spike. Remaining members can participate in research and design. Members also will contribute ideas to the design process, vet design ideas for feasibility, suggest alternative solutions and provide general estimates of work based on the designs being produced. Development insight is crucial to keeping the design process rooted in the reality of production. Again, **the key point is that the team does not move forward with development work that could become wasted effort** based on design spike decisions.

In organizations with additional UX and design resources, the team may also request to be temporarily augmented with new members. This is useful in situations where additional researchers or designers are needed for short turnaround testing and prototyping, but it is not required. These temporary team members would exit the team once the spike has ended.

So, What Happens To The Backlog, The Product Owner And The Traditional Sprint Artifacts?

The design spike inherits the product owner of the project but also adds another decision-maker: the design owner. The design owner will usually be a senior member or manager of the organization's design staff brought in from outside the scrum team, such as a UX director or VP, although the spot could potentially be filled by a decision-maker from marketing or product, depending on the organization's structure. The key is that the person in this role needs to have a design vision for what is to be built and the capacity to make decisions about a design's viability relative to broader design considerations in the organization (e.g. other products, design patterns, style prerequisites, etc.).

The design owner is also partially responsible for approving the done-ness of a design sprint's results. Both the **product owner and design owner must agree on the done-ness of backlog items** before the item may be considered done. The definition of "done" is decided during the design spike's initial sprint planning meeting and is an agreement between the team, the design owner and the product owner. For most spikes, the definition of done will probably be tied to the definition and completion of design artifacts, research results and prototyping endeavors.

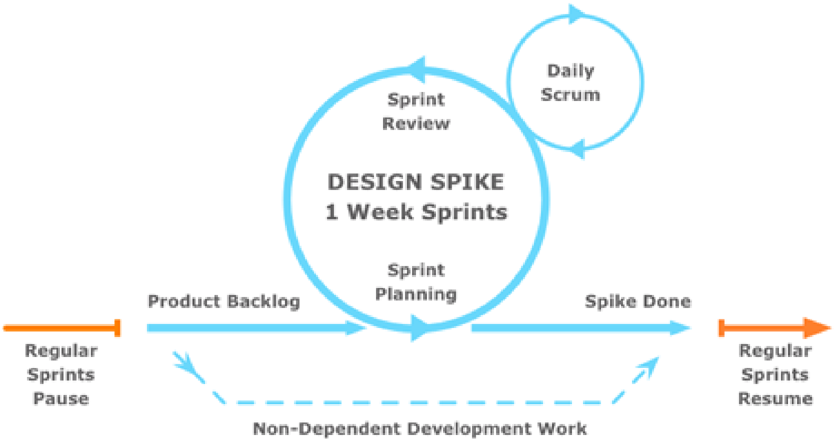
Why bother adding a design owner? In mature organizations with multiple product offerings that share a unified visual, interactive or brand language, design owners may be responsible for consistency issues that extend beyond a given project. Design decisions and product decisions might be at odds, and making transparent the process of these two constituents discussing and exposing the acceptability of solutions is key to the transparency of the design sprint. Formalizing the involvement of a design owner helps to ensure that the work being done in a design spike is being vetted and approved by senior design stakeholders and decreases the likelihood of later contradictions.

The design spikes also inherit the scrum product backlog, and the design team selects stories from the same backlog and accepts them into the spike, trying to complete as many as possible during a design sprint.

Design sprints work just like normal sprints but focus almost exclusively on design decisions. Design spikes should **prioritize work to resolve the biggest unknowns** in the current design first, and they do not need to follow the backlog's priority order. For instance, if a high-priority item calls for displaying data in a certain widget, but a lower-priority item calls for including more data than the widget can currently accommodate, then the design team may select the lower-priority problem in order to think about how this change could affect the widget's overall design downstream. Again, design prioritization is a different beast than product prioritization.

Importantly, because the spike focuses only on the design portion of backlog items and not the development portion, backlog stories are completed only from a design point of view. When the normal scrum cycle resumes, any stories calling for development work will still be alive on the backlog list, ready to be selected for a normal sprint.

The duration of the design spike is not generally time-boxed, but the team should aim to end the spike as soon as possible so that team resources are not left idle. The product owner may elect to time-box the design spike if necessary, although anticipating how long it will take to work through complex design issues can be difficult. The goal of each spike is not to produce potentially releasable code, but rather to provide actionable design decisions in the form of wireframes, mockups, prototypes or research. **Design spikes work best when they are of shorter duration**, providing a greater number of sprint reviews in which feedback from stakeholders can be gathered. The recommended time length for a design spike is one to two weeks. The spike would continue in the form of design sprints until the product owner and design owner decide that the spike has reached its goal or is no longer of use. All effort would then be made to return the team to its normal sprint schedule as soon as possible.



Each design spike may contain one or more design sprints, which work very much like normal scrum sprints but are focused simply on design.

As in a typical sprint, the design spike process retains all of the milestones that you've come to know and love: product backlog grooming, effort estimates, sprint planning, sprint reviews, sprint retrospectives. Those all stay the same, but the focus here is on design, not building, and all of those artifacts now track the design work that the team is aiming to finish.

When Do Design Spikes End, And What Happens Next?

When the product owner and design owner agree that the design spike has reached the point at which the normal scrum process should begin or resume, the spike ends. The design artifacts created during the spike are carried over into the normal product scrum to serve as starting points for the detailed design process.

The designers who were part of the design spike (some of whom might transition to be part of the scrum development team) serve as a resource to the development team on any open questions and design issues, and the scrum process starts anew. Importantly, these design artifacts are *not* set in stone. The development team still functions as an autonomous decision-making unit. They simply now have the benefit of a holistically designed foundation on which to build.

Design Spikes Let You Zoom Out, And Then Zoom Right Back In

In conclusion, design spikes give UX teams a framework to conduct big-picture design within the scrum process. They allow for comprehensive design bubbles that focus on holistic issues, rather than the granular design concerns that scrum sprints sometimes emphasize.

The design spike gives the team the opportunity to explore systematically rich UX questions from a macro level, and allows the team to break out into design-centric thinking at any time for the purpose of solving larger UX challenges. With the addition of the design owner as a decision-maker in the process, design spikes also offer large, mature organizations the opportunity to incorporate senior design stakeholders who can lend expertise and guidance on important design solutions.

With the adoption of design spikes, UX teams can leverage some of the flexibility that development teams employ in the agile process, while gaining the immensely important time needed to address design concerns that simply might not fit neatly into the hyper-focused needs of standard scrum sprints. ❧

You Already Know How To Use It

BY CHARLES HANNON ❧

In the first television advertisement for the iPad, the narrator intoned, "It's crazy powerful. It's magical. You already know how to use it." This was an astonishing claim. Here was a new, market-defining, revolutionary device, unlike anything we had seen before, and we already knew how to use it. And yet, for the most part, the claim was true. How does a company like Apple make such great new things that people already know how to use?

What is iPad? Intro video[29] on YouTube.

One answer lies in the ability of Apple designers to draw upon patterns that people are familiar with. The interaction medium might be completely new: before the iPhone, few people had used a multitouch screen. But everyone knew how to pinch or stretch something, and this interaction pattern was easily transferrable to the small screen after seeing it done just once. As Alan Cooper writes in *About Face*, "All idioms must be learned; good idioms need to be learned only once."

---

[29] http://www.youtube.com/embed/D2BvVcSkNkA

The Role Of Dopamine In Pattern Recognition

Our brains like to find such patterns. We are wired to search for patterns that our past experiences have shown will lead to successful interactions (in love, war, gambling, investing, etc.). Jonah Lehrer, in *How We Decide*, writes that our brain produces a pleasure—inducing neurochemical, dopamine, when we recognize familiar patterns in the world around us. When we act on these patterns and are successful in whatever we are trying to do, we get an additional burst of this pleasing chemical.

If we think we recognize a pattern but are mistaken, or if the pattern doesn't behave in the way we expect it to, then we do not get that second infusion of the neurochemical, and we readjust our expectations. Many neuroscientists believe this reward system is one way in which learning takes place. The process creates a self-reinforcing, pleasure-based cycle that encourages us to learn from our mistakes and to become better interpreters of the world around us.

The **dopamine reward system** produces positive or negative emotions based on our experiences in the world. Lehrer argues that this reverses our age-old understanding of the role of emotions in the decision-making process. Since Plato, the rational mind has been depicted as the charioteer holding the reins on our unruly emotions. What makes humans unique, according to this metaphor, is our ability to use logic and rationality to control our emotions and make rational decisions. Lehrer's book details recent research in neuroscience that upends this reason-based model of decision-making. Emotions, some of them caused by the dopamine-based reward system, play a central role in our decision-making processes.

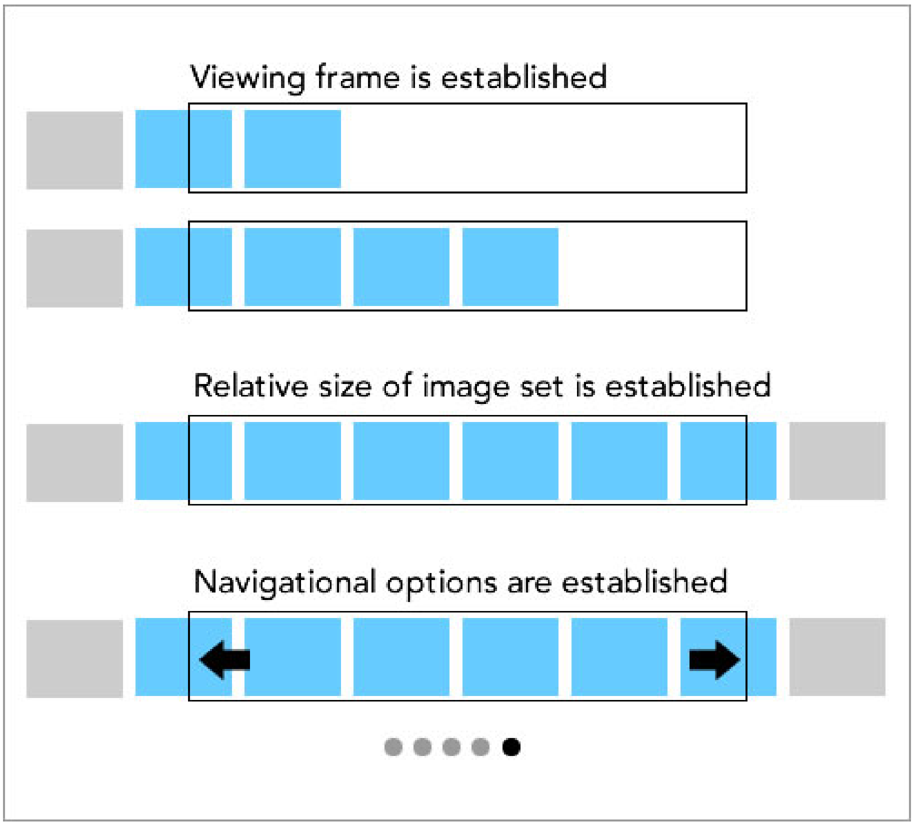
(Image: Sue Clark[30]*)*

These discoveries in neuroscience provide a strong argument for using design patterns in interaction design. Take the **carousel pattern**, which is prevalent in desktop, tablet and handheld devices. The Yahoo Design Library has a useful illustration of this design pattern. Content appears to slide in from one side of the panel; items at each end are partially obscured to indicate that more virtual space, and more content, lies outside the carousel pane; arrows appear, when appropriate, to indicate how to get to that additional content. This is a very simple pattern that people can learn after using the feature just once.

New users of Pandora will encounter this carousel pattern almost to the letter, and even if they are encountering it for the first time, they will learn it almost immediately. Then, when they encounter versions of the carousel pattern in other designs, they will recognize it before they even begin to interact with it. Their recognition of the pattern will produce pleasure as the dopamine neurons begin firing. When the user then interacts with the pattern—by clicking the arrow on either end to reveal additional content, for instance—and is successful, then more dopamine is produced, leading to additional feelings of pleasure.

---

[30] http://www.flickr.com/photos/perpetualplum/3995210073/sizes/o/


Viewing frame is established
Relative size of image set is established
Navigational options are established

Carousel design pattern, via Yahoo Developer Network[31]

Admittedly, neuroscientists have not yet attached functional magnetic resonance imaging machines to users in order to measure their brain's dopamine production as they experience the carousel (or any other) interaction design pattern. To date, our insight into the brain's responses to the patterns we encounter in the world is limited to what we can extrapolate to humans from experiments that have been conducted on monkeys and to inferences we can draw from the work of psychologists.

LEHRER'S RADAR TECHNICIAN'S STORY

Lehrer tells the story of a radar technician during the first Gulf War who spent several days watching blips that represented fighter aircraft returning to ship from a certain point on the coast of Kuwait. One set of blips in the early morning made the technician feel nervous, and he couldn't explain why. They looked to him to be just the same as those he had observed hour after hour in days past, but his emotional response to this particular set of early morning blips told him that something was wrong. Acting on little more than this emotional response, he ordered the blips to be shot down—thus saving countless lives: the blips

---

[31] http://developer.yahoo.com/

turned out to be enemy missiles en route to destroy Allied ships in the Gulf.

The technician could not explain how he knew they were not just another pair of fighter jets. It was only after much review and the discoveries of a cognitive psychologist who was brought in to review the case that investigators determined that what was different about those blips was *where* they first appeared on his screen: a little farther from shore than all of the other blips. He couldn't tell at the time that this is what made them different, but subconsciously his brain detected a change in the pattern that he had been observing for hours. The change in pattern caused an emotional and somatic response of panic and anxiety and caused him, despite his reason, to order the blips to be destroyed.

The radar technician's story (and many others recounted in Lehrer's book) suggests that our brains observe and act on patterns without our being conscious of it. Recognizable patterns appear, our dopamine neurons fire, our learning is reinforced, and we remain in a state of "flow." But when a pattern is broken or behaves unexpectedly, all hell breaks loose. Our brain sends out a **"prediction error signal"**. An area of the brain called the anterior cingulate cortex (ACC) monitors the activity of the prefrontal cortex, and when the ACC detects the absence of activity among dopamine neurons that results from the predicted event not occurring, it sends out this error signal. This results in other chemicals being produced, by the amygdala and the hypothalamus, among other areas, which causes these feelings of panic and anxiety: the heart races, the muscles tense, we become short of breath.

Broken Patterns Cause Panic And Anxiety

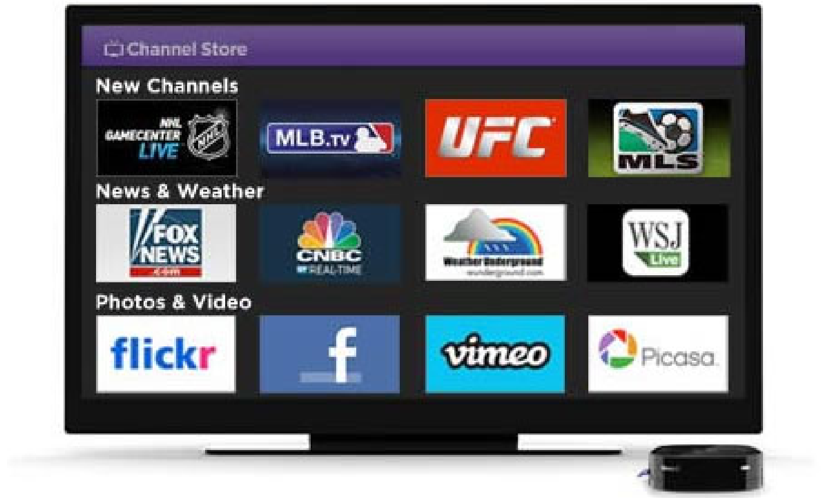
Ordinarily, we do not want our users to experience these feelings of panic and anxiety when they use our systems. Yet we know it happens frequently. One reason is that we often present users with interfaces that lack the visual cues to indicate what patterns are being employed. Consider Roku[32]'s Channel Store. When users visit the interface to add a channel to their system, they are confronted with what appears to be a static table of contents. Without prior experience of the carousel pattern, users might interpret this 3 × 4 tabular interface as offering only 12 channels.

In fact, this table *does* behave according to the carousel pattern. Additional content does lie to the right and left of each row. The content even scrolls vertically as well, but users would never know this from

---

[32] http://roku.com

the visual display of the information. Even worse, a new user will learn little about the carousel pattern to apply to their next encounter with it. Ironically, Roku is best known as a Netflix streamer, and Netflix itself applies the carousel pattern expertly to its similar table of contents in its streaming interfaces on game devices such as the Wii. Way back in *Design of Everyday Things*, Donald Norman defined “visibility” as meaning that “the correct controls are visible, and they convey the correct information.” Neither is the case in the design of Roku’s Channel Store, so users have no way of knowing, without extensive exploration, that the carousel pattern is being employed.



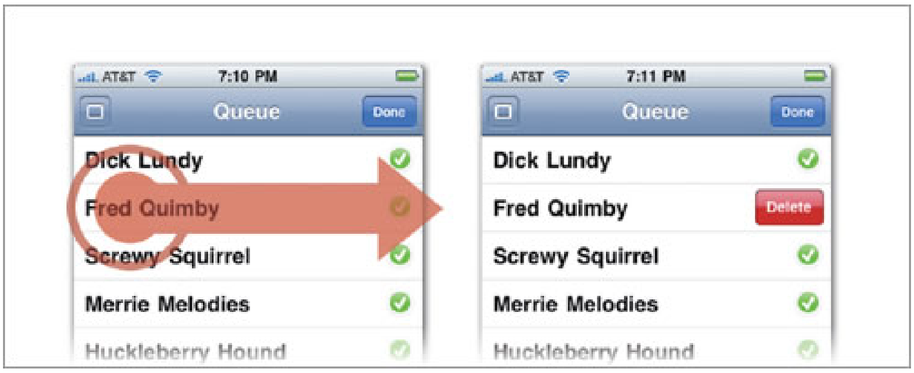
Roku’s Channel Store[33].

Sometimes, the problem is the reverse: **users will think a design pattern is being used when it isn’t**. What we recognize as a pattern doesn’t function as we expect; our brains think that something has gone wrong, and the result, again, is anxiety and panic. Take the basic design of a list of items on a smartphone. Users of iOS know this pattern well; it is famously illustrated in Josh Clark’s *Tapworthy: Designing Great iPhone Apps*. A left-to-right swipe gesture opens a control for deletion, prompting the user to confirm the delete action. This design pattern is easy to learn, but its implementation in other smartphone applications is sporadic and unpredictable. Palm’s webOS email system, for example, uses the swipe gesture for deletion but offers no “Delete” button to confirm the gesture. The email item simply vanishes off the

---

[33] http://roku.com

screen. In the messaging application on Palm's OS, on the other hand, the system does present a deletion control.



Swiping left to right to delete in iOS. (Image: Josh Clark[34]*)*

Early versions of the Android OS didn't acception the swipe gesture for deletion at all, and it usually interpreted the gesture as a tap by opening the "Edit Item" page. The Gingerbread update introduced even more inconsistency to the user experience: a right-to-left swipe over a contact, for instance, opens the instant messaging app, and a left-to-right swipe opens the phone app—and initiates a phone call! A user who would naturally expect this gesture to trigger a prompt to delete the contact suddenly finds themselves calling that contact. Talk about panic!

Pattern-Matching Is Harder Than It Sounds

All of us have experienced this feeling of panic to one degree or another. I still feel it when I instinctively move my mouse (in Windows) to the task bar to return to a Web page that I thought I had minimized, when in fact (and for at least three years now) the page I am looking for is open in a different *tab*, rather than in a minimized, separate window. **Interaction habits of mind do not change quickly.** And because I use three different Web browsers on at least four different computers, I am constantly unsettled in my search for the "Home" button, which used to be to the left of the URL window in most browsers, but now is all the way on the right in the standard installation of Firefox 12 on both Windows and Mac and doesn't exist at all in a standard installation of Safari. There is no longer a reliable pattern to determine where I will find the "Home" button on a Web browser. But my brain wants one, feels good when it finds one and rebels (chemically) when it doesn't.

---

[34] http://answers.oreilly.com/topic/2129-iphone-app-design-when-an-awkward-interface-makes-sense/

To be sure, inconsistencies across platforms, browsers and software can have many causes, from patent issues to design legacies. And it is inevitable that interaction designs will change and improve over time. We should not be held to existing patterns just because the human brain prefers it. But we can design according to our developing understanding of how the brain functions. We can employ idioms, such as "pinch," that are not obvious but are quickly learned. We can progress gradually, building on fundamental elements of existing designs so that new interaction designs retain enough of the old that our brains still recognize them. We can also cautiously introduce new schemes as redundant elements: one doesn't *have* to use three- and four-finger swipe gestures on the MacBook Pro's mousepad, but once one discovers these gestures, they are easy to adopt as natural improvements to the pointer controls and buttons in application interfaces.

In fact, this last approach takes advantage of the brain's chemistry. The prediction error signal is sent when an expected event does not occur and the result is disappointment or failure. **But sometimes, the result of a prediction error is delight, not panic.** The expected result did not occur, but something *better* did. David Rock, in *Your Brain at Work*, observes that this experience of delight or novelty also produces dopamine and feelings of pleasure. The experience is similar to that of humor: jokes often work because the punchline presents an unexpected twist, a novel outcome. More importantly, jokes work best when the stakes are minimal: no one really gets hurt in a pratfall. When jokes cut too close to the bone, they are painful. We cringe.

New interaction designs can be introduced according to the same principle: if they cause unexpected delight, and no one (and no one's data) gets hurt, they will induce unexpected pleasure and will be quickly adopted over the legacy designs they are meant to replace. ❧

Further Resources

- *About Face 3: The Essentials of Interaction Design*[35], Third Edition, by Alan Cooper, Robert Reimann, David Cronin
- *How We Decide*[36], Jonah Lehrer
- *Your Brain At Work: Strategies for Overcoming Distraction, Regaining Focus, and Working Smarter All Day Long*[37], David Rock

---

[35] http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470084111.html
[36] http://www.jonahlehrer.com/books/
[37] http://www.your-brain-at-work.com/

Fixing A Broken User Experience

BY STEFAN KLOCEK ❧

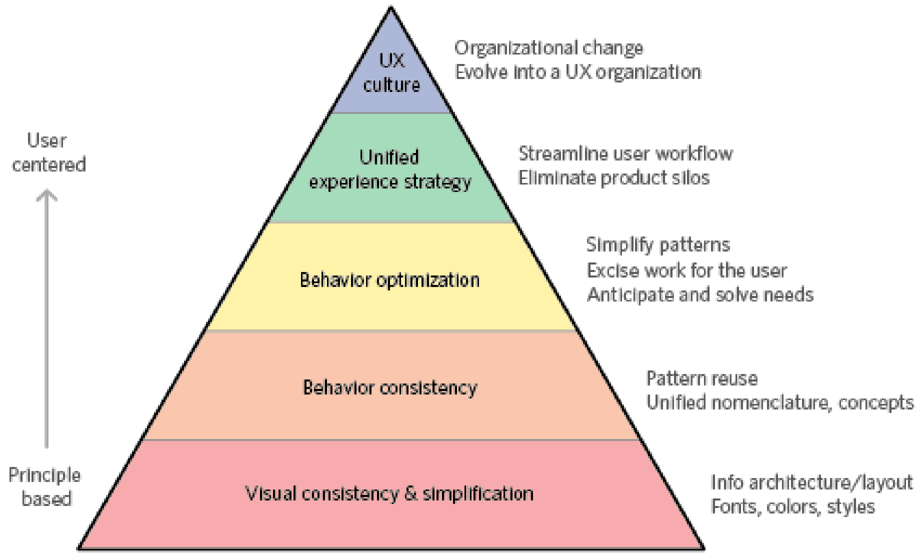
Unless you're developing completely new products at a startup, you likely work in an organization that has accumulated years of legacy design and development in its products. Even if the product you're working on is brand spanking new, your organization will eventually need to figure out how to **unify the whole product experience**, either by bringing the old products up to par with the new or by bringing your new efforts in line with existing ones. A fragmented product portfolio sometimes leads to an overall broken user experience.

Understanding an organization and its users and designing the right interaction and visual system take exceptional effort. You also need to communicate that system to teams that have already produced work that doesn't align with it. This isn't easy work. In this article, we'll introduce you to a **strategy for fixing the broken experience** that starts with surface improvements, goes progressively deeper into structural issues and ends with a big organizational shift.

The Hierarchy Of Effort

Many large successful companies end up in a situation where they must maintain dozens, if not hundreds, of applications in their product portfolios. These huge suites are the result of mergers, acquisitions, different sets of user needs, legacy services and contracts, and the inefficiencies that naturally develop in huge organizations. Sometimes the reasons for so many different product lines are legitimate; other times, the wide set of offerings doesn't serve anyone's needs particularly well. Users will often struggle to learn a suite of related products because of major differences in how they look and operate.

The initiatives to fix these broken experiences are referred to in ambitious and somewhat generic terms, such as "common look and feel," "unified online experience" and "unified look and feel." Regardless of the term, the common elements represent a drive to bring consistency to a large set of products in multiple stages of development and spearheaded by a centralized internal group. There's a sense of urgency; we often meet with some internal resistance; and frequently we're charged with fixing a previous agency's failed attempt to deliver design and guidelines that can be metabolized by the client.



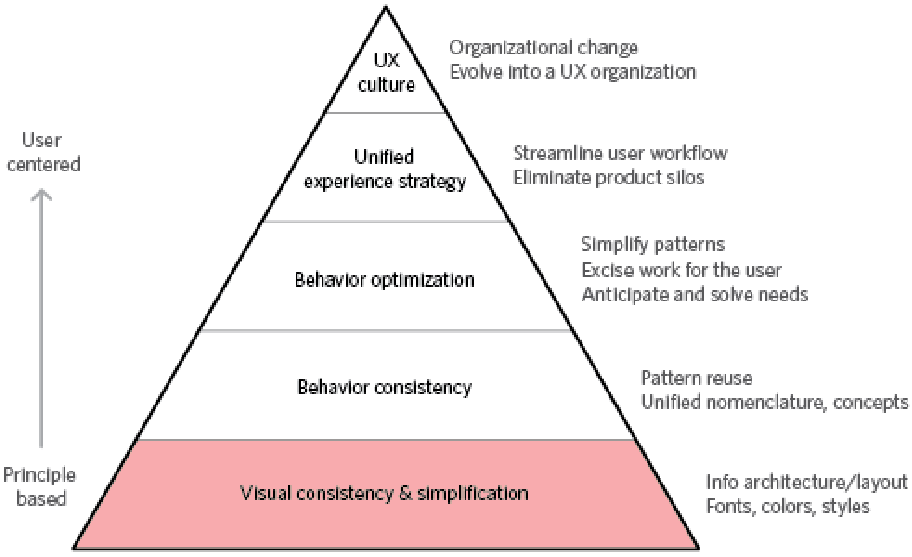
The hierarchy of effort to fix a broken user experience

One effective approach **begins with surface improvements**, goes progressively deeper into structural issues and ends with big strategic organizational shifts. We start with the low-hanging fruit and at each step reach higher to develop products that will ultimately deliver great experiences. It's worth noting that this approach was developed to make it possible for a team to make incremental improvements to products already under development, but also to look ahead to future releases, when rewriting code or rethinking interactions won't be so disruptive.

If your organization is working on its first product, then this approach would be totally backward. But in a large organization with a lot of history and many products, this approach will help you articulate both a short-term and long-term strategy for building a product portfolio that delivers a user experience that is learnable and builds confidence and a portfolio that makes your work easier and more effective.

VISUAL CONSISTENCY AND SIMPLIFICATION

The lowest amount of effort required is at the bottom of the pyramid, so we suggest starting there. Sure, it's lipstick on a pig, but simply taking a consistent visual approach will help to bring many different products under a shared brand experience.

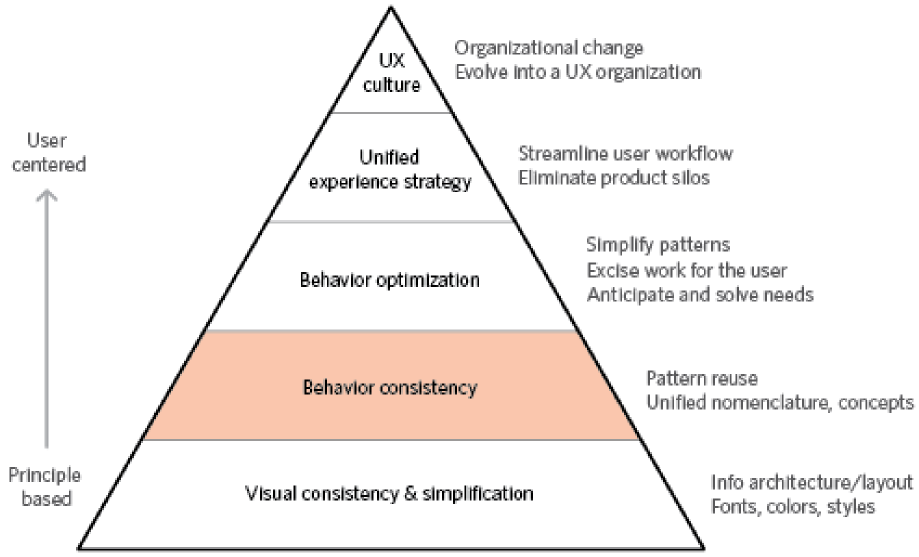


Assuming you've done the groundwork to articulate the design of an ideal experience, the simplest and arguably easiest way to start implementing it is to reskin the products currently under development. Finding ways to simplify and excise unnecessary information, unifying the information architecture, and adopting standard fonts, colors and controls are all relatively low-effort ways to improve existing products.

This is the foundation. It won't improve a poorly designed interaction, but it could dramatically increase the appearance of unity to the end user. Products that have a consistent visual language will clearly convey their membership in a single portfolio. The benefit of improving the visual system first is that changing or adjusting the skin of an application is much easier than changing things such as behavior, which will require rethinking and recoding fundamental aspects of the application.

BEHAVIORAL CONSISTENCY

If your organization has simplified and unified the visual language, the next step is to make the behavior consistent. This is basic stuff: disciplined reuse of patterns instead of applying patterns ad hoc from a grab bag of widgets, and unifying the nomenclature and conceptual frameworks. Hopefully, any individual product will have internally consistent patterns; it's when you look at sets of applications that were developed by different groups or obtained through acquisitions that you usually see wide discrepancies.

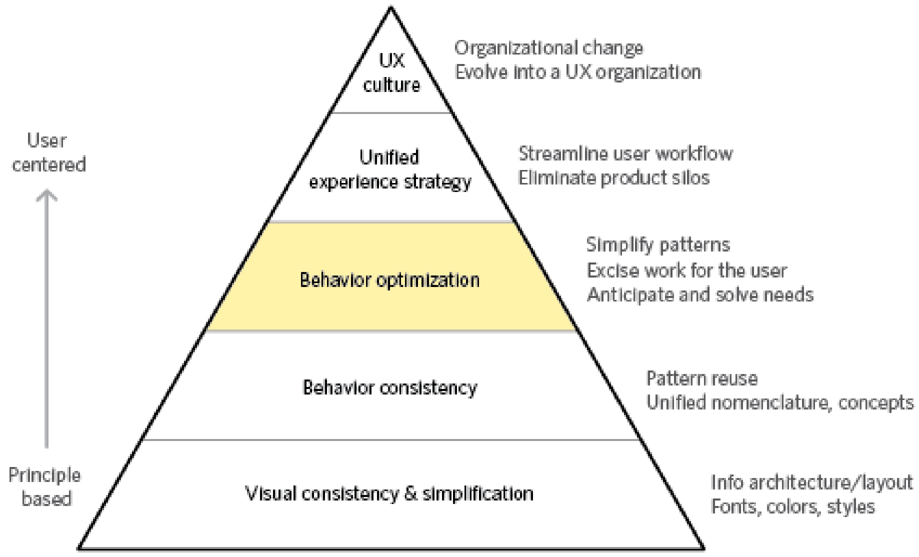


Assuming that the given design expresses high-level principles and provides a basic set of pattern libraries, the goal at this stage is to evaluate individual products and figure out how much work is required to align them. This work entails at the very least replacing widgets in some applications. It usually also entails a decent amount of coding and testing to ensure that the revisions contribute to a consistent experience. Maintaining a shared approach and understanding will require more coordination between development groups.

Behavioral consistency makes it **easier for the end user** to learn a tool and then to transfer those skills when picking up related tools. The user has to build only a single mental model of how the applications work. This gives them confidence and enables them to pick up new products without facing a steep learning curve and without being confused about how things are done.

BEHAVIORAL OPTIMIZATION

The prior step was done solely to align the behavior of the various products. A deeper level of work is required to optimize the behavior and to make the applications more powerful and easier to use.



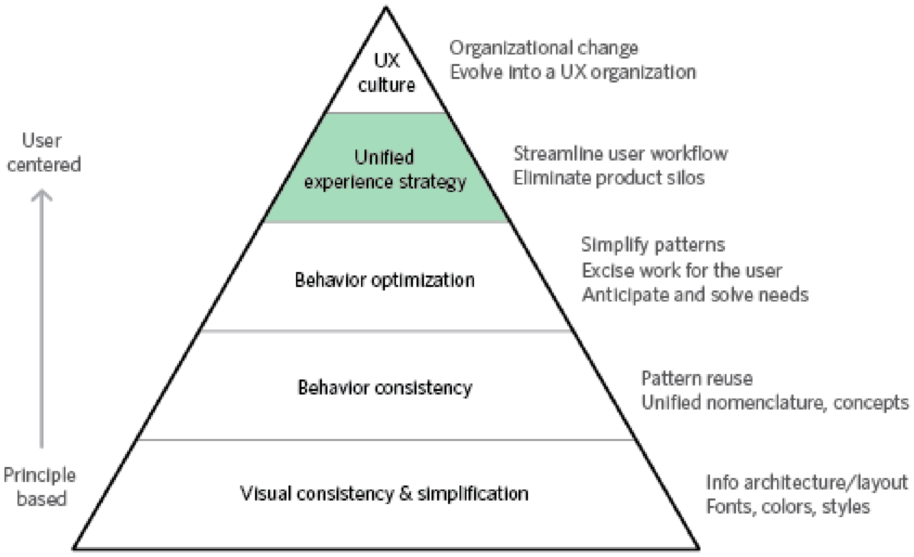
This step reworks the products even further. It means evaluating the current products against the user's needs and goals and looking for ways to eliminate work and to simplify the patterns. This assumes some measure of design effort beforehand to identify the areas where this will make the most difference. It assumes a commitment to user-centered product design, some research, as well as personas and scenarios. Without these, you've got no way to decide what patterns to simplify, which work to excise, and what user needs to anticipate and solve for.

An optimized experience enables users to perform their tasks with less or more effective work. Any work that's performed is captured in such a way that users aren't asked to perform the same task twice. Smart defaults are captured and leveraged to make tasks flow more quickly. Where possible, shift computing work to computers, and judgements to humans. Mine data to see broader patterns and opportunities that allow the system to anticipate and meet needs before they become issues.

This is where you do everything you can to **make each application the best it can be**. It takes a lot of work, with new interactions introduced and much code rewritten. A considerable investment of time and effort is required.

UNIFIED EXPERIENCE STRATEGY

The result of the last iteration is a set of products that do what they do best. The point of this iteration is to rethink how the suite works together. This often means rethinking product strategy.

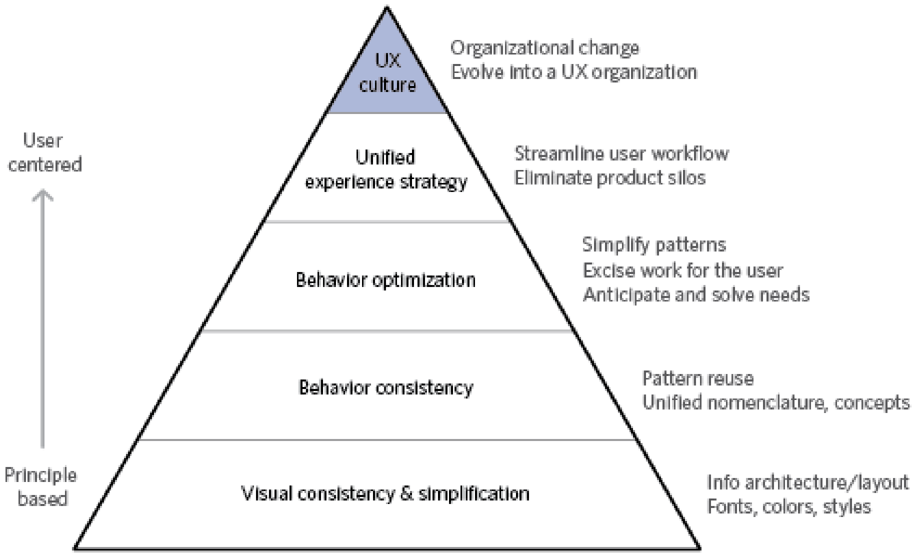


Designing a unified experience requires looking at the big picture, reevaluating the internal product silos in the organization, and reconsidering the ideal workflow for individuals and between roles. It could lead to collapsing multiple products into one, bridging gaps with new products, eliminating redundancies in capabilities or refocusing the service. This kind of work takes deep organizational commitment and a strong mandate. It takes long-range, instead of short-term, planning. It can't be done quickly, and doing it well takes organizational honesty and courage.

The real beneficiary of this kind of effort is the end user, because this product strategy is user-centered. The company recognizes that the product exists to help people perform their work and that they might use other tools and services to accomplish their goals. Users don't exist in isolation; they share work with others. Success isn't measured by how well they perform a task, but in **how competently they traverse a complex and dynamic ecosystem of people, data, devices and services**. When a company brings their product line to this stage, both the organization and the product line have been transformed.

UX CULTURE

All of the prior steps were aimed at fixing a broken user experience. By following them as an iterative path, it becomes possible to greatly improve a severely broken user experience. The way to avoid having to repeat this cycle in a few years is to transform the organization itself. Software and services are conceived and developed in a particular organizational culture, and this has a profound effect on the products. Products coming out of an engineering-oriented organization bear the unmistakable focus on technology; services with a focus on sales deeply communicate this; and products that come out of organizations with a UX bent cannot avoid their focus on a good user experience.



If you want to repeatedly deliver a great user experience, you need to go deeper than applying design to the surface. Your organization needs to understand and commit to making user experience a core priority. Executives have to support or advocate for the unique perspective that design brings; capable designers have to work for a user-centered approach; and a user-centered way of building things has to be integrated into the organization.

A great user experience almost never just happens. Understanding the user and **keeping their needs as your priority** throughout the design and development stages take deliberate effort. Products and services are created by teams of people who collaborate to bring an idea to life. The output is ultimately shaped by the agreements about what is important, the methods of performing the work and decisions on how to measure things. A shift in organizational culture takes the most effort and the longest time, but it results in the largest, most pervasive and most coherent shift—not just for the organization and its products, but for those who use them.

Isn't This All Backwards?

"But wait," you're thinking. "Isn't this all backwards? Shouldn't you design the whole system around the right workflow, optimize the behavior within it, make sure it's consistent with other products, and finally make sure it's visually simple and clear?" Yes. Yes, you should, especially if you're making a brand new product.

But we see again and again that few large companies really have the ability to clear the table, start with a clean slate and build something utterly new and great. Most start with a line of products that cannot be abandoned. They have applications that are supported by various teams around the world, perhaps owned by different subsidiaries and in various states of compliance. While **you can design the ideal experience, you can't just build it**. Moving toward something whose design really delivers will take many iterations. This situation isn't great, but it's the reality. When you find yourself here, you can't boil the ocean. You have to start somewhere. In our experience, starting at the bottom is a very practical way to move forward. ❧

Beyond Wireframing: The Real-Life UX Design Process

BY MARCIN TREDER ❧

We all know basic tenets of user-centered design. We recognize different research methods, the prototyping stage, as well as the process of documenting techniques in our rich methodological environment. The question you probably often ask yourself, though, is *how* it all works in practice?

What do real-life UX design processes actually look like? Do we have time for every step in the process that we claim to be ideal? In this article, I'll share a couple of insights about the real-life UX design process and speak from my own experience and research.

User-Centred Design: Truth Vs. Fiction

A few years ago, I joined one of the biggest e-commerce companies in Eastern Europe. When I entered my new office, I immediately spotted a huge user-centered design (UCD) poster on the wall. The whole process was described in detail that left hardly any doubts about the step-by-step approach to design. Exciting interior design for an aspiring UX designer, right? I stared at the poster with great hope and imagined how exciting following the ideal UCD process would actually be. Guess what? They didn't apply a single step from the poster to the actual process. They never did any research, nor any serious analysis of user behavior. Yikes, they didn't even prototype! This fancy poster simply hung shamefully on the wall.

For the next three years, we worked hard to put user experience design at the heart of a developer-driven culture. We forgot about the poster and structured our own process, which fitted well with the company's capabilities and allowed us to constantly optimize our main service. Why didn't we use the crystal-clear theoretical approach? Because we couldn't afford to go step by step through a classic UCD process with a lot of different activities. It would have taken too much time, and therefore it was economically invalid—the budgets for our projects were way too tight.

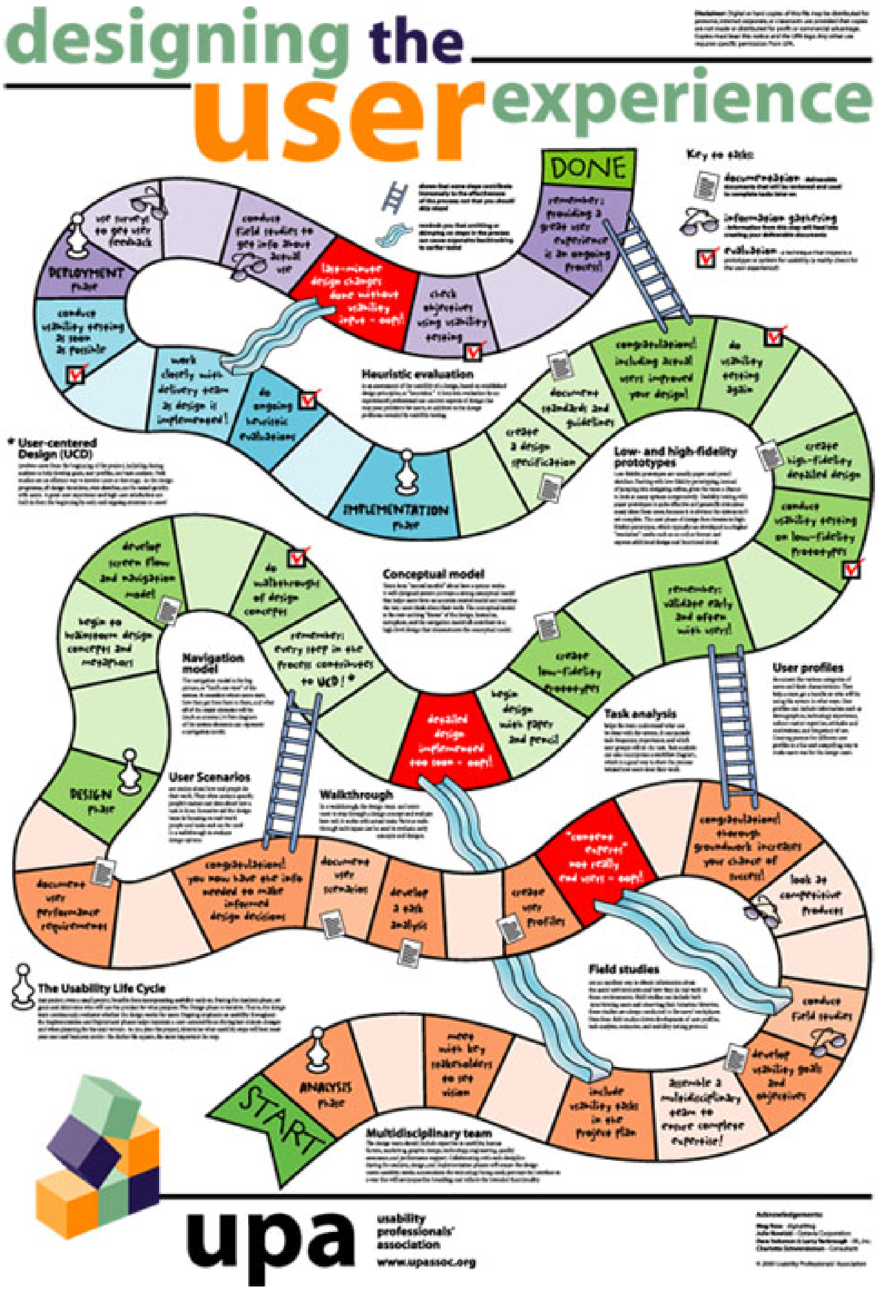
To deliver a user interface on time, we were forced to get really lean. We used a classic UCD process as inspiration and created a process that was simple but actionable for the company. We defined the problem, defined the scope of the project, iterated through paper prototyping and wireframing, pushed code to production as fast as we could, and always used multivariate split-testing and detailed Google Analytics event tracking.

Post-launch was the time to measure and plan optimization, which we executed immediately. Unfortunately, only huge projects had budgets for qualitative testing. Huge projects were also full of preliminary diagrams (site maps, flow charts, conceptual diagraming)—a enormously recommended activity to find order in a complex mess of information.

All in all, our process was simple but efficient. Of course, in general terms, it was a UCD process, but compared to any popular approach[38] and a famous UPA poster[39], we used about 20% of the recommended tools and studies. We assumed that users don't benefit from poster unicorn processes. Users benefit from the hard work of a product team; therefore, a simplified process is better than a robust unactionable theory.

---

38. http://www.usabilityprofessionals.org/usability_resources/about_usability/what_is_ucd.html
39. http://www.usabilityprofessionals.org/upa_store/books_and_posters/index.html#poster


designing the user experience
DONE
Key to tasks:
documentation
information gathering
evaluation
remember: providing a great user experience is an ongoing process!
use surveys to get user feedback
conduct field studies to get info about actual use
DEPLOYMENT phase
last-minute design changes done without usability input - oops!
check objectives using usability testing
conduct usability testing as soon as possible
work closely with delivery team as design is implemented!
do ongoing heuristic evaluations
IMPLEMENTATION phase
Heuristic evaluation
congratulations! including actual users improved your design!
do usability testing again
document standards and guidelines
create a design specification
create high-fidelity detailed design
Low- and high-fidelity prototypes
conduct usability testing on low-fidelity prototypes
User-centered Design (UCD)
develop screen flow and navigation model
do walkthrough of design concepts
begin to brainstorm design concepts and metaphors
Conceptual model
remember: validate early and often with users!
Navigation model
remember: every step in the process contributes to UCD!
create low-fidelity prototypes
begin design with paper and pencil
detailed design implemented too soon - oops!
Task analysis
User profiles
DESIGN phase
User Scenarios
Walkthrough
"content experts" not really end users - oops!
congratulations! thorough groundwork increases your chance of success!
document user performance requirements
congratulations! you now have the info needed to make informed design decisions
document user scenarios
develop a task analysis
create user profiles
look at competitive products
Field studies
conduct field studies
The Usability Life Cycle
develop usability goals and objectives
meet with key stakeholders to set vision
include usability tasks in the project plan
assemble a multidisciplinary team to ensure complete expertise!
ANALYSIS phase
START
Multidisciplinary team
upa
usability professionals' association
www.upassoc.org

Designing the user experience. (Large version[40]*)*

Suddenly, I started to wonder how others managed to apply UCD. There's a lot of talk about wireframes, but what does our work look like beyond wireframes? Was I the only one with a simplified approach? What can we do to create successful designs? What does the process beyond "the poster" look like? Is there a pattern that works well for the majority of designers?

---

[40] http://www.mprove.de/script/00/upa/_media/upaposter_85x11.pdf

The Reason For Research

Luckily enough, I was about to find some answers to my questions about the design process. I was forced to perform a worldwide reality check on my opinion about the classic UCD approach and design processes. Sharing this reality check is the *raison d'être* of this article.

- If you're fresh in the UX design world, learning how more experienced designers work might be useful.
- If you're a seasoned designer, treat this article as an incentive to reconsider your approach to design. We're all rushing our designs every day. This is the time to take a breath, see what others are doing and think about what works and what doesn't work in our real-life approach—beyond a UCD poster.

You may wonder what force persuaded me to revise my approach to the design process. The answer is simple: my own startup. Together with my friends, we created paper prototyping notepads to make our process more efficient, and then we created our own collaborative wireframing application. We suddenly became quite popular, took VC investment and decided to face the challenge: to create a user experience design toolset to support teamwork in the design process.

We felt that we were trying to fight Godzilla (or Tywin Lannister, if you prefer Game of Thrones to old Japanese movies). If my UX teams couldn't apply a classic UCD approach, how could I be sure that using any theoretical framework would enable me to design a toolset that fits anyone's real-life process? I couldn't. Is there any pattern in design processes that we actually apply in our companies? I had no idea.

We felt that we needed to find out the truth about real-life design processes and we needed it now. It appeared to us that our research might be of vast importance to the community and even beyond. A simple equation: a great tool for the design process equals less work for designers on the tools side, equals more time for creative work, equals better designs for all of us.

The stakes were great, and there was just one right thing to do: get out of the building, get our hands dirty with research, find out and learn about the real-life design process (if it exists), and literally hunt out pain points in it to make the work of our team much easier and more pleasurable. We packed our stuff and crossed the great pond, so to speak, to do some serious research in San Francisco and Silicon Valley. Read on if you want to know what we found out about the design process!

The Customer Development Process And Tons of Individual In-Depth Interviews

The life of a modern startup is full of UX design work, even if the founders don't realize it. Drake Martinet (Wall Street Journal, Stanford University) considers the whole lean startup movement to be a mere application of design principles to the business environment. I couldn't agree more.

When starting a new project, you actually need to talk to people from your target group. Here comes what are well known as IDIs (individual in-depth interviews): moderated, individual interviews in which you try to learn as much as you can about the problems of your interlocutor in a particular area of their life.

Our target group was user experience designers, so we scheduled above 50 interviews (personally and via Skype). Each focused on the same theme: the real-life UX design process. We asked designers to tell us stories of their usual process based on one of their projects. During the interviews, we asked a ton of in-depth questions to learn as much as we could about the process.

We hardly asked about problems in the design process, though—we tried to spot them in the stories on our own and then confirm our judgment by asking questions (for example, "I understand that X was troublesome in this particular project?"). We tried as hard as possible not to push any views onto our interlocutors. Letting them speak was important.

We interviewed UX heroes Mike Kuniavsky, Indi Young, Luke Wroblewski, Peter Merholz, Brandon Schauer, Jeffrey Kalmikoff and John Zeratsky and some lesser-known but excellent UX designers. Among our interlocutors were in-house UX designers, designers from consultancies and freelancers. Surprisingly enough, the problems that usually trouble UX designers were similar in all three groups.

It was an intense learning experience, and I highly recommend considering such preliminary research in every project. It will give you a ton of ready-to-use knowledge—a kind of canvas to work from.

The Process That Emerged From Designers' Stories

First of all, we didn't find any unicorns, but we did find racehorses in excellent condition. While all of the processes that emerged from the stories were somehow simplified UCD processes, they were tailored to the specialities of the designers. Flexibility is what helps us survive in the diverse jungle of projects. Processes morph to fit projects.

The approach to an e-commerce website differs from the way we design mobile apps in the healthcare industry (guess where context analysis matters most?), and government clients differ from corporate stakeholders and startup entrepreneurs, and so on. With few exceptions, though, the process looks surprisingly similar. There is a visible pattern that we all use to design interfaces in different environments:

1. COLLECTING INFORMATION ABOUT THE PROBLEM

Every UX designer needs to be a kind of detective in the early stage of a project. We need to find out as much as we can about the three Ps (people, problem, project). Activities in this stage, in contrast with the classic UCD approach, are vastly simplified:

- Meeting with the client (no matter whether externally or internally) and identifying the product's requirements (often in the form of a standardized product requirement document);
- Benchmarking and trend analysis (oh yes, most of the designers we interviewed do that).

We seldom perform user interviews, but writing user stories is one of the commonly accepted attachments to the product requirement document. Our user stories are sometimes created based on personas, which are hardly ever backed up with data. Field studies and task analysis are hardly used by any of the designers we interviewed.

2. GETTING READY TO DESIGN

This is clearly the ideation part of the process. It's completely conquered by analog tools. I haven't met a single designer who doesn't use quick messy sketching or some other paper prototyping form at the early stage of a design process!

Designers try to act on the material gathered in the first step of the process and find a design worth refining. This stage is not about documenting; it's about artistic fury and creative explosion. Many of us use Adaptive Path's multipage templates[41] to quickly create very generic sketches.

Unfortunately, testing lo-fi prototypes is not popular. We prefer to take the risk of choosing one option with a stakeholder and begin the refinement process. Not very UCD-like, but that is the reality.

---

[41] http://www.adaptivepath.com/ideas/sketchboards-discover-better-faster-ux-solutions

3. DESIGN

In contrast to the anti-documentation agile approach, most of the interviewed designers create wireframes and prototypes to document the experience and then hand them to the developers.

Refined sketches from the previous stage are still rather lo-fi and are usually not tested. Hi-fi design is left for visual designers. In Aristotelian terms, we create the form, while developers and visual designers fight to create the matter. Heuristic evaluation is definitely out of fashion, while expert review backed up with a cognitive walkthrough is quite popular.

4. APPROVAL

This is surprisingly an important part of the design process. Research documents and deliverables usually also serve as persuading factors in the “buy-in” process. This does not differ between in-house UX designers, freelancers and folks from consultancies.

Buy-in is the unfortunate peak of our process. None of us want to see our work go directly to the trash, and I’ve seen some great projects rejected just because the story behind the design process wasn’t particularly persuasive.

And guess what? A lot of the interviewed designers actually create a special presentation to tell stakeholders the design story. The presentations show stages of the process, deliverables and interactions, and they aim to give stakeholders lazy access to all of the information.

The four points mentioned above form a pattern visible in the majority of design processes that we went through with our interlocutors. You might have noticed that not a lot of iterative research is done in these processes. Sadly, the classic usability study is not a permanent part of the process. Why? The answer is simple: budgets are tight. Problems that appeared in the company that I used to work for appeared to be common. Tight budgets are forcing UX designers to tailor their processes and skip costly research.

I believe the best answer to this problem is guerrilla research methods. Startups do adapt guerrilla research as a part of the customer development process, but more “mature” companies, in my opinion, are strangely afraid of spontaneous and methodologically questionable yet efficient and cheap research methods. One of the challenges of the UX design community in the coming years will be the popularization of guerrilla research methods and bringing them into our real-life design processes.

Houston, We Have Several Recurring Problems

During our research, we tried to spot recurring problems in the design processes of our interlocutors—a so-called pattern of pain. Surprisingly enough, similar problems appeared in almost all individual interviews. Apparently, a lot of us live arm in arm with three tough unresolved problems that tend to slow us down:

1. **Spreading an understanding of the design process**
   How to engage the whole team in the process and show them that UX designers are not people who lack talent in visual design yet still insist on drawing something? How to teach that there's user experience beyond wireframes?
2. **Communication within the team**
   How to communicate with a team throughout the process and actually use different perspectives of teammates to evaluate design deliverables?
3. **Demonstrating the process to get buy-in**
   How to present the design process to stakeholders and developers to actually get buy-in, both formally and psychologically?

One of the UX designers we interviewed said the following:

> *Do you know what the most painful thing is in my job? Bureaucracy. Having to go to meetings. I would rather design than fight over the picky details. We should make at least part of the workflow online instead of in person. Have the approval process online, instead of in a meeting.*

Another said this:

> *It's really hard to show the process to clients and spread some understanding of the importance of design.*

We have probably all tried to solve these problems countless times, but we still lack efficient and fast methods. This results in less time for creative work and research.

My hypothesis is this. We as UX designers need to resolve the three painful problems identified above to have more time for creative work and research. We need to demonstrate our work beyond wireframes, spread understanding of UX design and, in fact, sell ourselves both internally (within the product team) and externally (outside the product team, in front of clients and stakeholders). This is the recipe to increase our effectiveness.

Our real-life UX processes need adjustment, and since we share the pattern of the process and the pain points, we can solve them together. This is most likely the most positive outcome of this research.

Outcome Of The Research

The research shows that UX designers are constantly modifying the classic and complex UCD approach. Less emphasis on iterative usability studies and a narrower range of design activities (compared to classic UCD) are the main traits of the current real-life design process that have emerged from our research.

A process tailored to the capabilities of our companies and our clients proved to be generally effective, but it still causes some recurring troubles that should be eliminated.

This is, generally speaking, the state of our field. Don't get me wrong: I don't mean to criticize classic UCD—it still serves as an inspiration for our work. After all, I'm happy that I worked in that office with "shame" hanging above my head (yes, I mean the UCD poster), which constantly reminds me of the need for adjustment in the process. I've learned that what matters, though, is an actionable process—possible to use, adapted to the company's culture and financially effective.

After talking with dozens of UX designers, I've started to wonder, however, whether we should actually create a poster that shows this version of the process. It could help a lot of aspiring UX designers take their first steps in the field and could be effective as an educational tool for our internal and external clients.

After all, our work is not nearly as expensive and time-consuming as the old poster says.

P.S. A study of the process and the problems spotted in it inspired us to create "The UX Design System[42]"—it's a work in progress, and I'd love to hear your feedback. ❧

FURTHER RESOURCES

- "35 Excellent Wireframing Resources[43]," Cameron Chapman
- "Free Wireframing Kits, UI Design Kits, PDFs and Resources[44]," Aquil Akhter

---

[42] http://www.youtube.com/watch?v=A6aGep7eFDs

[43] http://www.smashingmagazine.com/2009/09/01/35-excellent-wireframing-resources/

[44] http://www.smashingmagazine.com/2010/08/27/free-wireframing-kits-ui-design-kits-pdfs-and-resources/

- “Design Better and Faster With Rapid Prototyping[45],” Lyndon Cerejo
- “Free Printable Sketching, Wireframing and Note-Taking PDF Templates[46],” Paul Andrew

---

[45] http://www.smashingmagazine.com/2010/06/16/design-better-faster-with-rapid-prototyping/

[46] http://www.smashingmagazine.com/2010/03/29/free-printable-sketching-wireframing-and-note-taking-pdf-templates/

Stop Redesigning And Start Tuning Your Site Instead

BY LOUIS ROSENFELD ❧

In my nearly two decades as an information architect, I've seen my clients flush away millions upon millions of dollars on worthless, pointless, "fix it once and for all" website redesigns. All types of organizations are guilty: large government agencies, Fortune 500s, not-for-profits and (especially) institutions of higher education.

Worst of all, these offending organizations are prone to **repeating the redesign process** every few years like spendthrift amnesiacs. Remember what Einstein said about insanity? (It's this[47], if you don't know.) It's as if they enjoy the sensation of failing spectacularly, publicly and expensively. Sadly, redesigns rarely solve actual problems faced by end users.

I'm frustrated because it really doesn't have to be this way. Let's look at why redesigns happen, and some straightforward and inexpensive ways we might avoid them.

The Diagnostic Void

Your users complain about your website's confounding navigation, stale content, poor usability and other user experience failures. You bring up their gripes with the website's owners. They listen and decide to take action. Their hearts are in the right place. But the wheels quickly come off.

---

[47] http://www.brainyquote.com/quotes/quotes/a/alberteins133991.html

Most website owners don't know how to diagnose the problems of a large complex website. It's just not something they were ever taught to do. So, they're put in the unfortunate, uncomfortable position of operating like country doctors who've suddenly been tasked to save their patients from a virulent new pandemic. It is their responsibility, but they're simply unprepared.

Sadly, many website owners fill this diagnostic void—or, more typically, allow it to be filled—with whatever solution sounds best. Naturally, many less-than-ethical vendors are glad to dress up their offerings as solutions to anyone with a problem—and a budget. The tools themselves (search engines, CMS', social apps) are wonderful, but they're still **just tools**—very expensive ones, at that—and not solutions to the very specific problems that an organization faces. Without proper diagnostics to guide the configuration of tools, any resulting improvements to the user experience will be almost accidental.

Sometimes design agencies are brought in to fill the diagnostic void. And while not all agencies are evil, a great many follow a business model that depends on getting their teams to bill as many hours as they can and as soon as possible. Diagnostics can slow the work down (which is why clients rarely include a diagnostic phase in their RFPs). So, many agencies move to make a quick, tangible impression (and make their clients happy) by delivering redesigns that are **mostly cosmetic**.

A pretty face can last only a few years, but by then the agency is long gone. Invariably, the new owner wishes to make their mark by freshening or updating the website's look. And another agency will be more than happy to oblige. Repeat ad nauseam, and then some.

Oh, and sometimes these redesigns can be pricey. Like $18 million pricey[48].

See why I'm so grouchy?

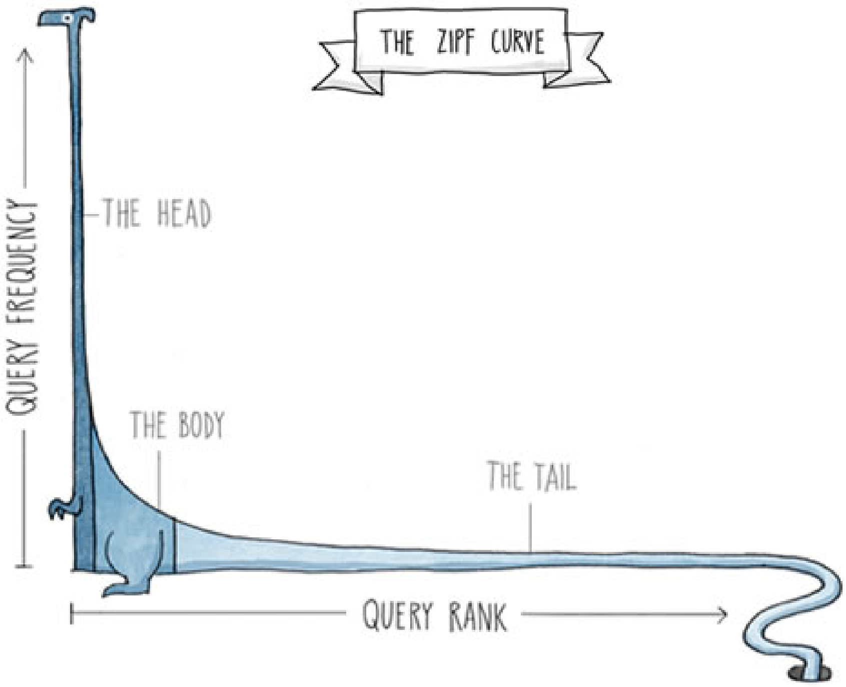
Forget the Long Tail: The Short Head Is Where It's At

Whether you're a designer, researcher or website owner, I've got some good news for you: diagnostics aren't necessarily difficult or expensive. Better yet, you'll often find that addressing the problems you've diagnosed isn't that hard.

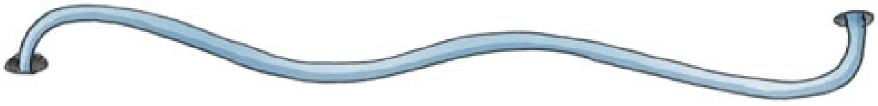
And the best news? Small simple fixes can accomplish far more than expensive redesigns. The reason? People just care about some stuff more than they care about other stuff. A lot more. Check this out and you'll see:

---

[48] http://usability.com/2012/01/31/four-seasons-18m-redesign-is-taking-a-lot-of-heat/



This hockey-stick-shaped curve is called a Zipf curve. (It comes from linguistics[49]: Zipf was a linguist who liked to count words… but don't worry about that.) Here it is in dragon form, displaying the frequency of search queries on a website. The most frequently searched queries (starting on the left) are very, very frequent. They make up the "short head." As you move to the right (to the esoteric one-off queries in the "long tail"), query frequency drops off. A lot. And it's a really long tail.



This is absolutely the most important thing in the universe. So, to make sure it's absolutely clear, let's make the same point using text:

Query's rank	Cumulative %	Query's frequency	Query
1	1.40%	7,218	campus map
14	10.53%	2,464	housing
42	20.18%	1,351	web enroll
98	30.01%	650	computer center

---

[49] http://en.wikipedia.org/wiki/Zipf%27s_law

221	40.05%	295	msu union
500	50.02%	124	hotels
7,877	80.00%	7	department of surgery

In this case, tens of thousands of unique queries are being searched for on this university website[50], but the first one accounts for 1.4% of all search traffic. That's massive, considering that it's just one query out of tens of thousands. How many short-head queries would it take to get to 10% of all search traffic? Only 14—out of tens of thousands. The 42 most frequent queries cover over 20% of the website's entire search traffic. About a hundred gets us to 30%. And so on.

It's Zipf's World; We Just Live in It

This is very good news. Want to improve your website's search performance? Don't rip out the search engine and buy a new one! Start by testing and **improving the performance** of the 100 most frequent queries. Or, if you don't have the time, just the top 50. Or 10. Or 1—test out "campus map" by actually searching for it[51]. Does something useful and relevant come up? No? Why not? Is the content missing or mistitled or mistagged or jargony or broken? Is there some other problem? That, folks, is diagnostics. And when you do that with your website's short head, your diagnostic efforts will go a very long way.

The news gets better: Zipf is a rule. The search queries for all websites follow a Zipf distribution.

And the news gets even jump-up-and-down-and-scream-your-head-off better: Zipf is true not only for your website's search queries. Your content works the same way! A small subset of your website's content does the heavy lifting. Much of the rest has little or no practical value at all. (In fact, I've heard a rumor that 90% of Microsoft.com's content has never, ever been accessed. Not once. But it's a just a rumor. And you didn't hear it here.) Bottom line: don't redesign *all* of your content—focus on the stuff that people actually need.

You'll also see a short head when it comes to your website's features. People need just a few of them; the rest are gravy.

And there's more. Of all the *audience types* that your website serves, one or two matter far more than the others. What tasks do those audi-

---

[50] http://www.msu.edu/

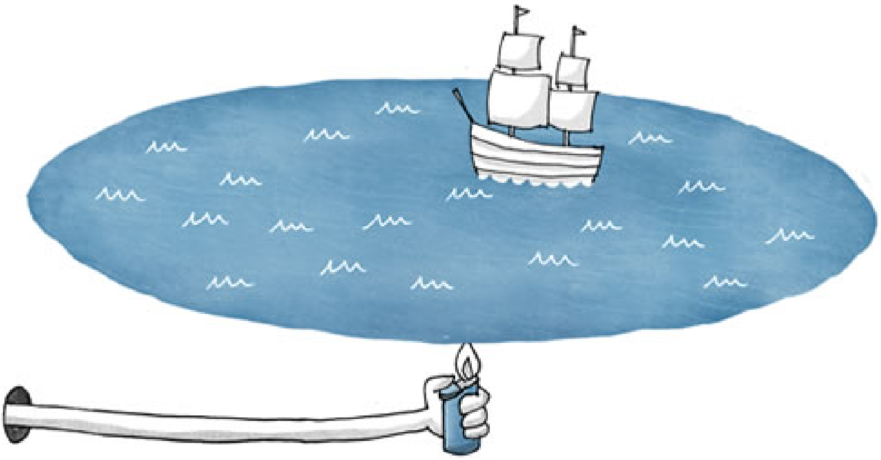
[51] http://search.msu.edu/index.php?q=campus+map

ence types wish to accomplish on your website? A few are short-head tasks; the rest just aren't that important.

As you can see, the Zipf curve is everywhere. And fortunately, the phenomenon is helpful: you can use it to prioritize your efforts to tweak and tune your website's content, functionality, searchability, navigation and overall performance.

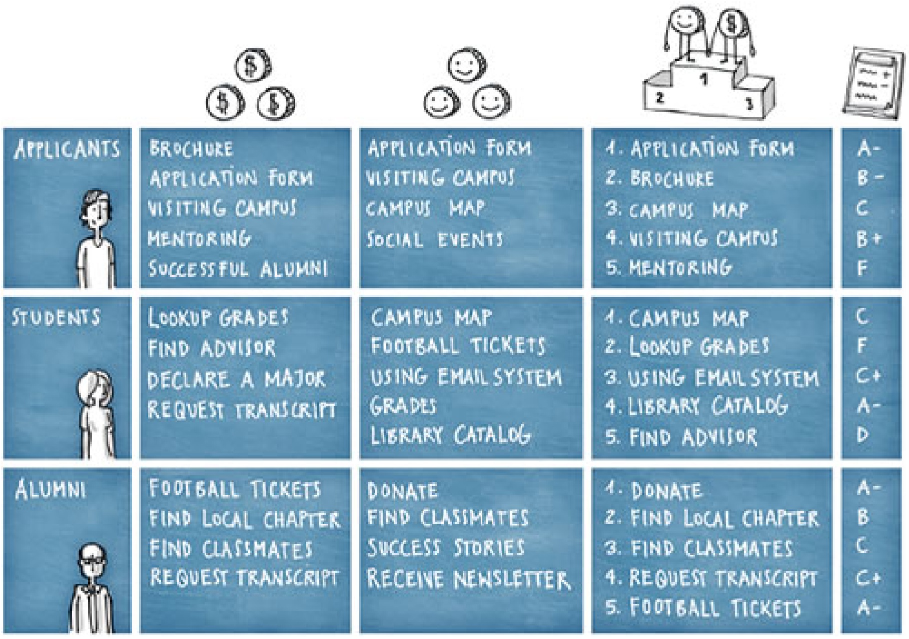
Your Website Is Not A Democracy

When you examine the short head—of your documents, your users' tasks, their search behavior and so forth—you'll know where to find the most important problems to solve. In effect, you can stop boiling the ocean…



… and start prioritizing your efforts to diagnose and truly solve your website's problems.

Now, let's put these short-head ideas together. Below is a report card for an academic website that starts with the short head of its audience:



APPLICANTS	BROCHURE APPLICATION FORM VISITING CAMPUS MENTORING SUCCESSFUL ALUMNI	APPLICATION FORM VISITING CAMPUS CAMPUS MAP SOCIAL EVENTS	1. APPLICATION FORM 2. BROCHURE 3. CAMPUS MAP 4. VISITING CAMPUS 5. MENTORING	A- B- C B+ F
STUDENTS	LOOKUP GRADES FIND ADVISOR DECLARE A MAJOR REQUEST TRANSCRIPT	CAMPUS MAP FOOTBALL TICKETS USING EMAIL SYSTEM GRADES LIBRARY CATALOG	1. CAMPUS MAP 2. LOOKUP GRADES 3. USING EMAIL SYSTEM 4. LIBRARY CATALOG 5. FIND ADVISOR	C F C+ A- D
ALUMNI	FOOTBALL TICKETS FIND LOCAL CHAPTER FIND CLASSMATES REQUEST TRANSCRIPT	DONATE FIND CLASSMATES SUCCESS STORIES RECEIVE NEWSLETTER	1. DONATE 2. FIND LOCAL CHAPTER 3. FIND CLASSMATES 4. REQUEST TRANSCRIPT 5. FOOTBALL TICKETS	A- B C C+ A-

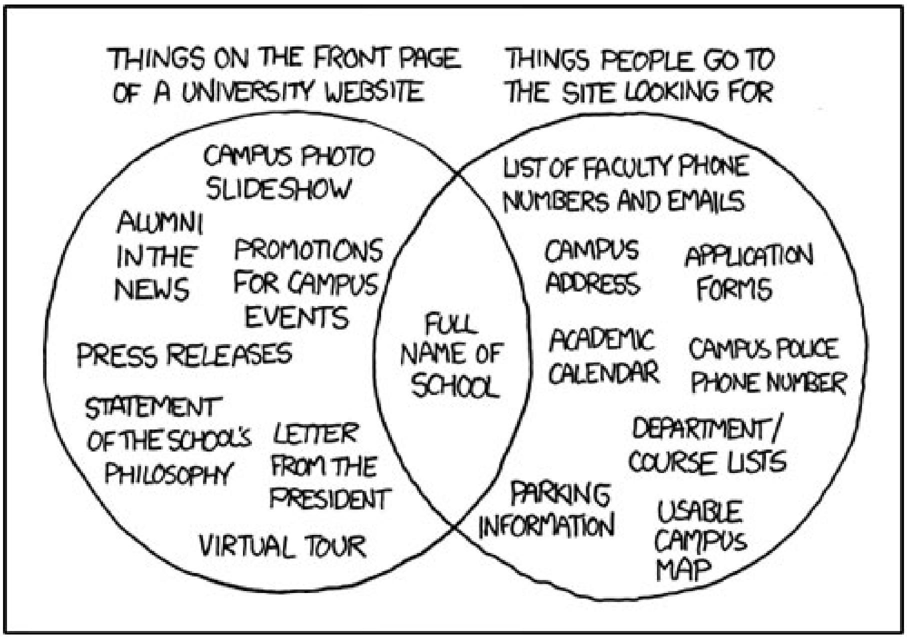
In other words, of all the audience types this university website has, the three most important are people who might pay money to the university (applicants,) people who are paying money now (students) and people who will hopefully pay money for the rest of their lives (alumni). How do we know they're the **most important audiences**? We could go by user research; for example, the analytics might suggest that these audiences generate more traffic than anyone else. Or perhaps the university's stakeholders believe that these are the most important ones in their influence and revenue. Or some combination of both. Whatever the case, these three audiences likely swamp all other segments in importance.

Then, we would want to know the short-head tasks and information needs of each audience type. We might interview stakeholders to see what they think (column 2). And we might perform research—user interviews and search analytics, for example—to find out what users say is most important to them (column 3).

Of course, as the good folks at xkcd demonstrate[52], stakeholders and users don't always see things the same way:

---

[52] http://xkcd.com/773/


THINGS ON THE FRONT PAGE OF A UNIVERSITY WEBSITE
THINGS PEOPLE GO TO THE SITE LOOKING FOR
CAMPUS PHOTO SLIDESHOW
ALUMNI IN THE NEWS
PROMOTIONS FOR CAMPUS EVENTS
PRESS RELEASES
STATEMENT OF THE SCHOOL'S PHILOSOPHY
LETTER FROM THE PRESIDENT
VIRTUAL TOUR
FULL NAME OF SCHOOL
LIST OF FACULTY PHONE NUMBERS AND EMAILS
CAMPUS ADDRESS
APPLICATION FORMS
ACADEMIC CALENDAR
CAMPUS POLICE PHONE NUMBER
DEPARTMENT/ COURSE LISTS
PARKING INFORMATION
USABLE CAMPUS MAP

That's why talking to both stakeholders and users is important. And once you've figured out the short head for each, you'll need to earn your salary and, through some careful negotiation, combine your takes on each audience type's needs. That's what we've done in column 4.

Finally, in column 5, we've tested each task or need and evaluated how well it works. (Because it's a university-related example, letter grades seemed appropriate.) You can do this evaluation in an expensive, statistically significant way; but really, enough research[53] is out there[54] to suggest that you don't need to spend a lot of time and money on such testing. More importantly, these needs and tasks are often fairly narrow and, therefore, easy to test.

So, after testing, we can see what's not going well. Finding information on "mentoring" is hard for applicants. And current students have a devil of a time when they "look up grades."

Now we're done diagnosing the problems and can begin making fixes. We can change the title of the "Paired Guidance Program" page to "Mentoring." We can create a better landing page for the transcript application. The hard part, diagnostics, is out of the way, and we can **now fix and tune** our website's performance as much as our resources allow.

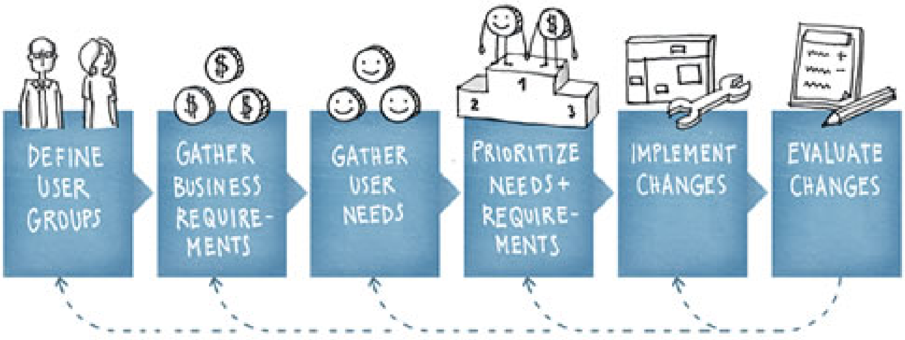
---

[53] http://www.useit.com/alertbox/discount-usability.html
[54] http://uxmag.com/articles/getting-guerrilla-with-it

From Project To Process To Payoff

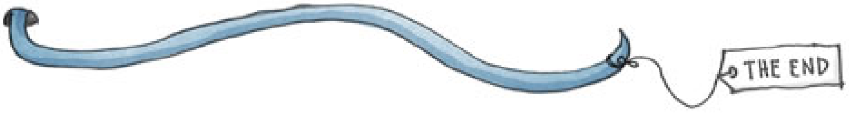
These fixes are typically and wonderfully small and concrete, but because they live in the short head, they make a huge and lovely impact on the user experience—at a fraction of the cost of a typical redesign.

The **tuning process** itself is quite simple. It's what we used to arrive at the report card below:



If you repeat this simple process on a regular basis—say, every month or quarter—then you can head off the entropy that causes fresh designs and fresher content to go rotten. Thus, the redesign that your organization has scheduled for two years from now can officially be canceled.

Your website's owners ought to be happy about all this. And you should be, too: rather than tackling the project of getting your website "right"—which is impossible—you can now focus on tweaking and tuning it from here on out. So, forget redesigns, and start owning and benefiting from a process of continual improvement.



SPECIAL THANKS – ILLUSTRATIONS

Eva-Lotta[55] is a UX Designer and Illustrator based in London, UK where she currently works as an interaction designer at Google. Besides her daytime mission of making the web a more understandable, usable and delightful place, she regularly takes sketchnotes[56] at all sorts of talks and conferences and recently self-published her second book[57]. Eva-Lotta also teaches sketching workshops and is interested in (something she calls) visual improvisation. Exploring the parallels between sketch-

---

[55] http://www.evalotta.net/
[56] http://www.flickr.com/photos/evalottchen/sets/72157607235674386/with/6848813489/
[57] http://www.sketchnotesbook.com/

ing and improvisation, she experiments with the principles from her theater improvisation practice to inspire visual work. ❧

Designer Myopia: How To Stop Designing For Ourselves

BY RIAN VAN DER MERWE ❧

> *Have you ever looked at a bizarre building design and wondered, "What were the architects thinking?" Or have you simply felt frustrated by a building that made you uncomfortable, or felt anger when a beautiful old building was razed and replaced with a contemporary eyesore? You might be forgiven for thinking "these architects must be blind!" New research shows that in a real sense, you might actually be right.*

That's Michael Mehaffy and Nikos A. Salingaros describing a phenomenon we're all familiar with, in their article "Architectural Myopia: Designing for Industry, Not People[58]." As I read the article, I became increasingly uncomfortable as I realized that the whole thing might as well have been written about Web design (and about our responses to the designs of our peers). How often do we look at a website or app and remark to ourselves (and on Twitter) that "these designers must have been blind!" Sometimes we're just being whiney about minute details (as we should be), but other times we do have a point: "What were they thinking?"

---

[58] http://shareable.net/blog/architectural-myopia-designing-for-industry-not-people

Longaberger Home Office[59]*, Newark, Ohio. Image source*[60].

In this article, we'll discuss "designer myopia": the all-too-common phenomenon whereby, despite our best intentions, we sometimes design with a nearsightedness that results in websites and applications that please ourselves and impress our peers but don't meet user and business goals. With Mehaffy and Salingaros's article as our guide, we'll investigate the causes of designer myopia, and then explore some solutions to help us take the focus off ourselves and back on the people we're designing for.

The Causes Of Designer Myopia

If the language in the opening paragraph sounds familiar, it's because most of us privately and publicly mutter "What were they thinking?" almost every day as we move across the Web. We analyze the new Twitter app[61]; we take it upon ourselves to redesign popular websites[62]—and then we wonder if we should even be doing that[63]. One thing is clear, though: we're good at pointing out designer myopia in our peers.

---

[59] http://www.travelandleisure.com/articles/the-worlds-ugliest-buildings/6
[60] http://www.flickr.com/photos/ellenm1/3603328941/
[61] http://flyosity.com/design/twitter-for-iphone-takes-a-step-back.php
[62] http://www.dustincurtis.com/dear_american_airlines.html
[63] http://ignorethecode.net/blog/2011/05/15/unsolicited_redesigns/

But what are the causes of this lack of imagination and foresight in our work? Shouldn't we be smart enough to avoid the obvious traps of designing too much from our own viewpoints and not taking the wider user context in mind? Well, it turns out that we quite literally see the world very differently than others. Again, from "Architectural Myopia[64]":

> *Instead of a contextual world of harmonious geometric relationships and connectedness, architects tend to see a world of objects set apart from their contexts, with distinctive, attention-getting qualities.*

In other words, we see typography and rounded corners where normal people just see websites to get stuff done on. We see individual shapes and colors and layout where our users just see a page on the Internet. Put another way, we're unable to see the forest for the trees.

How did we get here? Notice the striking resemblance to Web design as Mehaffy and Salingaros describe the slippery slope that has led to this state in architecture:

> *With the coming of the industrial revolution, and its emphasis on interchangeable parts, the traditional conception of architecture that was adaptive to context began to change. A building became an interchangeable industrial design product, conveying an image, and it mattered a great deal how attention-getting that image was. The building itself became a kind of advertisement for the client company and for the architect (and in the case of residences, for the homeowner seeking a status symbol). The context was at best a side issue, and at worst a distraction, from the visual excitement generated by the object.*

This is why we often see designs that seem to be built for Dribbble, portfolios and "7 Jaw-Dropping Minimalist Designs" blog posts, instead of being "adaptive to context" based on user needs. We have gained much from the "industrialization" of design through UI component libraries and established patterns, but we've also lost some of the unique context-based thinking that should go into solving every design problem.

Jon Tan touches on this in "Taxidermista," his excellent essay on design galleries in the first issue of The Manual[65]:

---

[64] http://shareable.net/blog/architectural-myopia-designing-for-industry-not-people

[65] http://alwaysreadthemanual.com/

> *Galleries do not bear sole responsibility for how design is commissioned. However, they do encourage clients and designers to value style more than process. They do promote transient fashion over fit and make trends of movements such as minimalism or styles like grunge or the ubiquitous Apple-inspired aesthetic.*

The result of all of this is that we sometimes end up designing primarily for ourselves and our close-knit community. Jeffrey Goldberg reminds us that this is true for much of the technology industry in “Convenience Is Security[66]”:

> *Security systems (well, the good ones anyway) are designed by people who fully understand the reasons behind the rules. The problem is that they try to design things for people like themselves—people who thoroughly understand the reasons. Thus we are left with products that only work well for people who have a deep understanding of the system and its components.*

And so we end up with a proliferation of beautiful websites and applications that only we find usable.

MORDAC, THE PREVENTER OF INFORMATION SERVICES.
SECURITY IS MORE IMPORTANT THAN USABILITY.
www.dilbert.com
scottadams@aol.com
IN A PERFECT WORLD, NO ONE WOULD BE ABLE TO USE ANYTHING.
© 2007 Scott Adams, Inc./Dist. by UFS, Inc.
To complete the log-in procedure, stare directly at the sun.

We all follow some rules of thumb without understanding the reasons behind them.

We can’t talk about designing primarily for the community without bringing up the awkward point that we often do it *deliberately*. We thrive on the social validation that comes from positive Twitter comments, being featured in design galleries and getting a gazillion Dribbble likes. And let’s be honest: that validation also helps us get more clients. This is just part of human nature, and not necessarily a bad thing. But it *can* be a bad thing; so at the very least, we need to call it out as another possible cause for designer myopia so that we can be conscious of it.

---

[66] http://blog.agilebits.com/2011/08/22/convenience-is-security/

The Manual[67] *brings clarity to the 'why' of Web design, and much more.*

Oh, and while we're at it, let's ask the obvious next question. Why are we so good at noticing when others fall into the myopia trap but fail to catch ourselves when we do it? In "Why We're Better at Predicting Other People's Behaviour Than Our Own[68]," Christian Jarrett reports on some recent research that might provide the answer:

> *[When] predicting our own behaviour, we fail to take the influence of the situation into account. By contrast, when predicting the behaviour of others, we correctly factor in the influence of the circumstances. This means that we're instinctually good social psychologists but at the same time we're poor self-psychologists.*

In other words, we're much better at taking the entire context into consideration when looking at other people's designs than when we are creating our own. Scary stuff.

So, if designer myopia is indeed a pervasive problem (and if we are not good at recognizing it in ourselves), what do we do to fix it? I'd like to propose some established but often-ignored techniques to get us out of this dilemma.

---

[67] http://alwaysreadthemanual.com/

[68] http://www.bps-research-digest.blogspot.com/2012/01/why-were-better-at-predicting-other.html

1. Conduct Observational User Research In Context

The first thing that Mehaffy and Salingaros suggest in their article to overcome myopia is this:

> *First of all, re-integrate the needs of human beings, their sensory experience of the world, and their participation into the process of designing buildings. Leading design theory today advocates "co-design," in which the users become part of the design team, and guide it through the evolutionary adaptations to make a more successful, optimal kind of design. Architects spend more time talking to their users, sharing their perception and understanding their needs: not just the architect's selfish need for artistic self-expression, or worse, his/her need to impress other architects and elite connoisseur-critics.*

Note that this is not just about asking users what they think. It's about making users part of the design process in a helpful, methodologically sound manner. To accomplish this, we can look to anthropology to play a substantial role in the design of products and experiences. Ethnography (often called contextual inquiry[69] in the user-centered design world) is the single best way to uncover unmet needs and make sure we are solving the right problems for our users.

In "Using Ethnography to Improve User Experience[70]," Bonny Colville Hyde describes ethnography as follows:

> *Ethnographers observe, participate and interview groups of people in their natural environments and devise theories based on analysis of their observations and experiences. This contrasts with other forms of research that generally set out to prove or disprove a theory.*

That's the core of it: we do ethnography to learn, not to confirm our beliefs. By using this method to understand the culture and real needs of our users, we're able to design better user-centered solutions than would be possible if we relied *only* on existing UI patterns and some usability testing.

Leaving the office and spending time observing users in their own environments is the best way to understand how a product is really being used in the wild. It's the most efficient way to get out of your own head.

---

[69] http://en.wikipedia.org/wiki/Contextual_inquiry

[70] http://www.cxpartners.co.uk/cxblog/using_ethnography_to_improve_user_experience/

2. Design To Blend In

Let's stick with the architecture theme for a moment. The concept of "paving the cowpaths" is another effective way to look beyond ourselves and to design websites and applications that form part of our users' landscapes (rather than break their mental models). In "Architecture, Urbanism, Design and Behaviour: A Brief Review[71]," Dan Lockton writes:

> *One emergent behavior-related concept arising from architecture and planning which has also found application in human-computer interaction is the idea of desire lines, desire paths or cowpaths. The usual current use of the term […] is to describe paths worn by pedestrians across spaces such as parks, between buildings or to avoid obstacles […] and which become self-reinforcing as subsequent generations of pedestrians follow what becomes an obvious path. […]*
>
> *[T]here is potential for observing the formation of desire lines and then "codifying" them in order to provide paths that users actually need, rather than what is assumed they will need. In human-computer interaction, this principle has become known as "Pave the cowpaths".*

This is such an interesting perspective on user-centered design. By starting a design project with an explicit goal to "pave the cowpaths," we will always be pulled back into a frame of mind that asks how the design can better blend in with our users' lives and with what they already do online. The same questions will keep haunting us, and rightly so:

- Do we have analytics to back up this behavior?
- Are we sure this is what users naturally do on the website?
- We know that most users click on this navigation element to get things done. How do we make that behavior easier for them?

In the same paragraph in "Taxidermista," Jon Tan also calls for us to step back and ask questions like these before starting to design:

> *The answers to a project's questions may have something to do with fashion, but not often. Good design does not have a shelf life. The best web designers gently disregard issues of style at the start. They rewind*

---

[71] http://architectures.danlockton.co.uk/2011/09/12/architecture-urbanism-design-and-behaviour-a-brief-review/

> *their clients back to asking the right questions, so they can rewrite the brief and understand the objectives before they propose solutions.*

By asking the right questions, we focus our effort on fitting into the ways that users move on the Web, as opposed to bending them to our will.

3. Triangulate Results

The two recommendations above are very specific, so I'd also like to make a more general point. There are, of course, several other user-research methodologies to help us get into the minds of users and bring them into the design process in a helpful, meaningful way. Methods such as concept testing, participatory design and, of course, usability testing all have their place. But the real power lies in using not just one or two of these methods, but three or more. This is where triangulation[72] comes in:

> *Triangulation is a powerful technique that facilitates validation of data through cross verification from more than two sources. In particular, it refers to the application and combination of several research methodologies in the study of the same phenomenon.*

Using multiple data sources—both qualitative and quantitative—is a great way to avoid any myopia traps along the way. In addition to (or instead of, depending on the project) the two methodologies covered above, you should use as many appropriate techniques as possible to help confirm your intuition and direction.

As Catriona Cornett points out in "Using Multiple Data Sources and Insights to Aid Design[73]":

> *When used correctly, data from multiple sources can allow us to better identify the context in which our designs live. It can help us validate our assumptions and approach design with confidence and not subjective opinion. This not only helps to create better design, but also helps us achieve that all-important buy-in from stakeholders. It's easier to defend a design when you have deep, rich insights to back it up.*

The first response I get when proposing triangulation (or sometimes even just one research method) is usually, "We don't have time!" The

---

[72] http://en.wikipedia.org/wiki/Triangulation_(social_science)

[73] http://www.inspireux.com/2010/08/16/using-multiple-data-sources-insights-feed-design/

good news is that this doesn't have to slow you down—even an hour at a coffee shop observing real users with your product will shock you out of your myopia. The only thing that's *not* an option is skipping research completely.

Summary

User research and the techniques discussed in this article aren't new, but they're usually left to specialist researchers to champion, or they're swept under the rug because "We're using established UI patterns on this one." Hopefully, this article has shown that designer myopia is too common and too dangerous to ignore or to be left to specialist researchers to fix. Sure, user researchers are critical to ensuring that a proper methodology is followed, but we can all get out there and use the data and information available to us to make sure we don't put too much of our own viewpoints into our designs.

Web design is personal—deeply personal. As Alex Charchar puts it in his gut-wrenching essay for The Manual[74]:

> *I now know that it is through love and passion and happiness that anything of worth is brought into being. A fulfilled and accomplished life of good relationships and craftsmanship is how I will earn my keep.*

I doubt that any of us would disagree with those words. Our best work happens when we throw ourselves wholeheartedly into it. But this outlook on life and design comes with its own dangers that we need to watch out for. And the biggest danger is in being unable to see beyond our own passion and taste and, with the best intentions, in failing to make the necessary connections with our users.

My hope for all of us is that the three simple guidelines discussed here—contextual user research, designing to blend in, and research triangulation—will enable us to keep the perspective we need as we throw everything we've got at the design problems that we have to solve every day. ❧

---

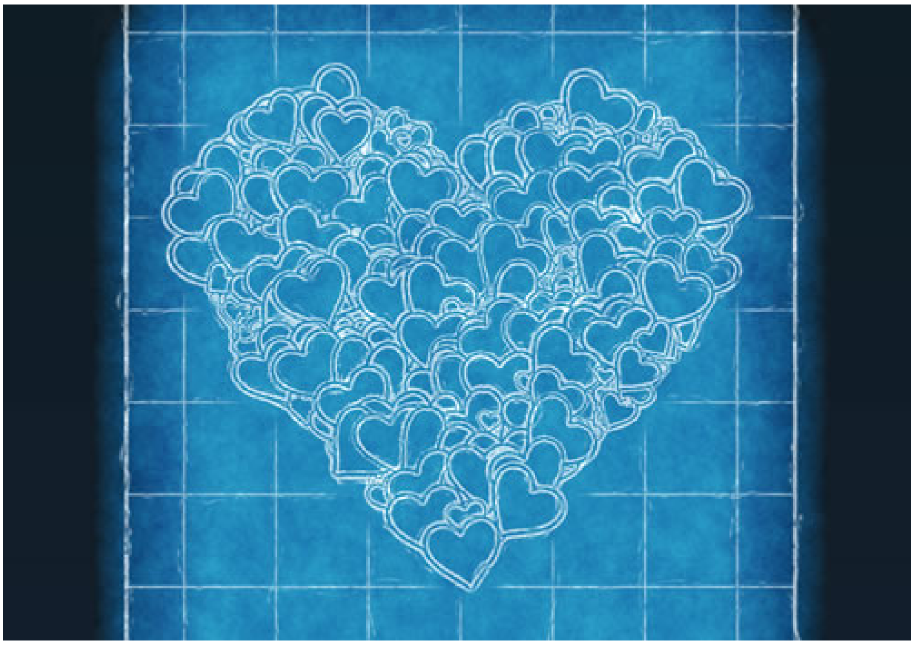
[74] http://alwaysreadthemanual.com/

The UX Research Plan That Stakeholders Love

BY TOMER SHARON ❧

UX practitioners, both consultants and in house, sometimes conduct research. Be it usability testing or user research with a generative goal, research requires planning. To make sure product managers, developers, marketers and executives (let's call them stakeholders) act on UX research results, planning must be crystal clear, collaborative, fast and digestible. **Long plans or no plans don't work for people.** You must be able to boil a UX research plan down to one page. If you can't or won't, then you won't get buy-in for the research and its results.

This article addresses one key aspect of planning UX research: the one-page plan document. Before we get to that, we'll briefly discuss the benefits of research planning and identify the audience of a research planning document.



(Image: Patrick Hoesly[75]*)*

A word about stakeholders. A stakeholder in the UX world is a code name for the people who UX practitioners work with. These are our clients, whether internal or external to our organization. These are people who need to believe in what we do, act on research results, and fund

---

[75] http://www.flickr.com/photos/zooboing/5442901351/sizes/l/in/photostream/

and sponsor future research. We all have a stake in product development. They have a stake in UX research.

The Benefits Of Research Planning

Very generally speaking, UX research can answer two types of questions:

1. **What's useful?**
   What do people need? Who is the target audience?
2. **What's usable?**
   Does the design work for people, and how it can be improved?

Dozens of research methodologies could be implemented to answer these and more specific questions, and it is up to designers, researchers and their teams to decide what works best for them and when is the right time to answer their questions.

Here are the benefits of planning UX research:

- **Get a better feel of stakeholders.**
  A written plan helps you identify what works and doesn't work for people, and what questions they are trying to answer.
- **Engage stakeholders.**
  A study plan ensures they are properly involved with the study and its results. If there's no written plan, then there's a greater chance that stakeholders won't feel engaged.
- **Writing things down helps you.**
  When you put things in writing, they look very different than how you imagined them when they were just thoughts in your head. Always have a written study plan, even if you don't share it with anyone else.

Now, let's quickly identify the target audience for the research planning document.

Who Are You Planning For? Who Are The Stakeholders?

As with every product or service, the best offering comes from carefully identifying the target audience, their needs and their wants. Different UX research stakeholders are interested in different aspects of a research plan:

- **Product managers and software engineers** are mostly interested in the study's goal, research questions and schedule. In some cases, they are also interested in the criteria for participants. These stakeholders are usually interested in goals and questions because these determine the content of the study and its focus. They are interested in the schedule to make sure it enables them to make timely design, business and development decisions. Criteria for participants interest them when the product targets a very specific demographic and they want to make sure participants are representative of that demographic.
- **Managers and executives** are probably interested in the study's goal and the overall cost of the study, because they are likely sponsoring the study. Usually, their bandwidth does not allow them more than that.
- **You!** The plan is mostly for you. As soon as you put your thoughts in writing, something happens, and you find holes in them. These holes help you improve the plan. A written plan also helps you focus and better prepare for the study. The fact of the matter is that if you can't boil your plan down to a page, you probably don't really understand it.

Now that we've discussed why a planning document is important and who it is for, let's get to the nitty gritty of the document.

The Plan That Stakeholders Love: The One-Pager

The users of a research plan love brevity and appreciate succinct definitions of what will happen, why, when and with whom. Here are the sections that go in a one-page research plan:

- **Title**
  The title should combine the thing you're studying and the methodology; for example, "Monster.com field study" or "XYZ Phone data-entry usability test." Sometimes mentioning the target audience of the study is also appropriate; for example, "Whitehouse.com news page interviews with senior citizens."
- **Author and stakeholders**
  State your full name, title and email address on one line. After you get the stakeholders' buy-in for the plan, add their details as well—the research belongs to everyone now.
- **Date**
  Update it whenever the plan is updated.

- **Background**
  Describe what led to this study. Discuss the recent history of the project. Be brief, no more than five lines.

- **Goals**
  Briefly state the high-level reason (or reasons) for conducting this study. Try to phrase it in one sentence. If that wouldn't make sense, create a numbered list of very short goal statements. If you have more than three to four goals, you are either aiming too high (meaning you have too many goals) or repeating yourself.

- **Research questions**
  These are the specifics, the core of your plan. Provide a numbered list of questions that you plan to answer during the study. It is extremely important that your stakeholders understand that you will not necessarily be asking the study participants these questions. As a rule of thumb, have no more than seven to ten questions, preferably around five. Later on, you will construct your study script to answer these questions. An effective way to think about research questions is to imagine that they are the headings in the study's summary.

- **Methodology**
  In an academic environment, this section has one primary goal: to provide as many details as other researchers need in order to repeat the exact same study. In practice, the goal of the methodology section is to briefly inform the stakeholders of what will happen, for how long and where.

- **Participants**
  Provide a list of the primary characteristics of the people you will be recruiting to participate in the study. Have a good reason for each and every characteristic. If you have two participant groups, describe both groups' characteristics in lists or in a table. Append a draft form that you'll use to screen participants.

- **Schedule**
  Inform stakeholders of at least three important dates: when recruiting starts, when the study will take place, and when they can expect results. Large research projects require more scheduling details. For example, if the study involves travel to another city or country, more dates might be required for on-site preparation and meetings or for analysis workshops.

- **Script placeholder**
  When a full study script is ready, it will appear under this title. Until then, all you need is a heading with a "TBD" indication.

A Sample UX Research Plan:

XYZ Phone Data-Entry Usability Test

by John Smith-Kline, Usability Researcher, jskline@example.com
Stakeholders: Wanda Verdi (PM), Sam Crouch (Lead Engineer)
Last updated: 13 January 2012

Background

Since January 2009, when the XYZ Phone was introduced to the world, particularly after its market release, journalists, bloggers, industry experts, other stakeholders and customers have privately and publicly expressed negative opinions about the XYZ Phone's keyboard. These views suggest that the keyboard is hard to use and that it imposes a poor experience on customers. Some have claimed this as the main reason why the XYZ Phone will not succeed among business users. Over the years, several improvements have been made to data entry (such as using horizontal keyboards for most features), to no avail.

Goals

Identify the strengths and weaknesses of data entry on the XYZ Phone, and provide opportunities for improvement.

Research Questions

1. How do people enter data on the XYZ Phone?
2. What is the learning curve of new XYZ Phone users when they enter data?
3. What are the most common errors users make when entering data?

Methodology

A usability study will be held in our lab with 20 participants. Each participant session will last 60 minutes and will include a short briefing, an interview, a task performance with an XYZ Phone and a debriefing. Among the tasks: enter an email subject heading, compose a long email, check news updates on CNN's website, create a calendar event and more.

Participants

These are the primary characteristics of the study's participants:

- Business user,
- Age 22 to 55,
- Never used an XYZ Phone,

- Expressed interest in learning more about or purchasing an XYZ Phone,
- Uses the Web at least 10 hours a week.

[Link to a draft screener]

Schedule

- Recruiting: begins on November 12
- Study day: November 22
- Results delivery: December 2

Script
TBD

Recap

A short plan that you and your stakeholders prepare together is key to a **successful start** of a UX research project.

- Boil down your collective knowledge, agreements and understanding of what will happen, why, with whom and when.
- Set the right expectations among stakeholders.
- Try to keep the plan to one page.
- Secure buy-in for the UX research by making it a team effort.
- The core of the plan is the list of questions you are trying to answer. Choose the right ones.

Happy planning! ❧

About The Authors

Charles Hannon

Charles Hannon is a professor of Computing and Information Studies at Washington & Jefferson College in Washington, PA. He teaches courses in the history of computing, human computer interaction, project management, and information visualization. He blogs on user experience design at www.uxappeal.com[76].

Damon Dimmick

Damon Dimmick is a user experience designer, agile ux advocate, and ux evangelist based in the metro Boston area. He has spent his career advocating for and designing efficient and pleasant experiences for complex products built by global companies. He is currently the Manager of Interaction Design (Global UX) for Monster.com[77] where he helps jobseekers and employers find each other through the use of sophisticated job search and hiring tools. Damon is also a founding partner of the Subforum[78] design group, a think-tank of UX professionals that advocates grassroots ux research, continuing design innovation, and design responsibility. Damon also shares his thoughts via Twitter[79].

Eva-Lotta Lamm

Eva-Lotta Lamm is a designer doing mainly web and interface design[80] but enjoying graphic design[81] and illustration[82] work as well. If you are after her sketchnotes, have a look at her sketchnotes set on flickr[83] and the book she made[84].

---

76. http://www.uxappeal.com
77. http://www.monster.com/
78. http://subforum.org/
79. http://twitter.com/damondimmick
80. http://www.evalotta.net/work/?c=web
81. http://www.evalotta.net/work/?c=visu
82. http://www.evalotta.net/work/?c=illu
83. http://www.flickr.com/photos/evalottchen/sets/72157607235674386/
84. http://www.evalotta.net/sketchnotes

Erik Perotti

Erik has worked in human-computer interaction since the late 1990s. Currently, he works as a Senior Product Manager, User Experience at Plantronics, working on mobile apps inter-operating with computers. He has co-authored multiple patents on mobile phone notifications, and holds a Masters Degree in HCI from Carnegie Mellon University.

Louis Rosenfeld

Louis Rosenfeld wears two hats: he works as an information therapist for large, messy organizations with findability problems, like PayPal, Lowes, and Ford; and he's founder of Rosenfeld Media[85], the user experience publishing house. A librarian by training, Lou is co-author of Information Architecture for the World Wide Web[86] (O'Reilly; 3rd edition 2006) and Search Analytics For Your Site[87] (Rosenfeld Media; 2011), co-founder of the Information Architecture Institute[88], and a former columnist for Internet World, CIO, and Web Review magazines. He blogs[89] occasionally and tweets (@louisrosenfeld[90]) frequently. If you liked this article, you really should attend Lou's workshop on adaptive information architecture[91].

Marcin Treder

Marcin Treder is a design enthusiast that literally lives for creating the best user experience possible. After years working as a UX Designer and UX Manager he focused on his own start-up UXPin[92] that provides tools for UX Designers all over the world. UXPin tools are used by designers in companies like Google, Apple, Microsoft, IBM, Salesforce. UXPin was recently voted the best start-up in Central and Eastern Europe. Marcin enjoys writing (e.g. UXMag), blogging (Blog UXPin[93], UXAid[94], Startup Pirate[95]) and tweeting (@uxpin[96], @marcintreder[97]).

---

85. http://www.rosenfeldmedia.com/
86. http://www.oreillynet.com/catalog/9780596527341/
87. http://rosenfeldmedia.com/books/searchanalytics/
88. http://iainstitute.org/
89. http://louisrosenfeld.com/
90. http://twitter.com/louisrosenfeld
91. http://rosenfeldmedia.com/workshops/adaptable-ia/
92. http://uxpin.com/
93. http://blog.uxpin.com/
94. http://uxaid.com/
95. http://startuppirate.tumblr.com/
96. http://www.twitter.com/uxpin
97. http://www.twitter.com/marcintreder

Rian van der Merwe

Rian is passionate about designing and building software that people love to use. After spending several years working in Silicon Valley, he is currently Director of User Experience at consultancy Flow Interactive[98] in South Africa. He also blogs[99] and tweets[100] regularly about User Experience and Product Management.

Stefan Klocek

As a Director of Interaction Design at Cooper Stefan leads teams, manages engagements and sparks new initiatives. He's adamant about working in the practice and signs up for solving the really hard complex problems. As a designer he's reflective about thinking and manic about making. He also teaches the practice and method of design for Cooper U.

Tomer Sharon

Tomer is a user experience researcher at Google Search in New York City and author of the book, It's Our Research: Getting stakeholder buy-in for user experience research projects[101] (2012). He founded and led UPA Israel and is the co-founder and organizer of leanUXmachine[102], a weekend of UX learning, collaboration, and mentorship for Israeli startups. Tomer holds a master's degree in Human Factors in Information Design from Bentley University[103].

---

[98] http://www.userexperience.co.za/
[99] http://www.elezea.com/
[100] http://twitter.com/rianvdm
[101] http://itsourresear.ch/
[102] http://leanuxmachine2011.tumblr.com/
[103] http://graduate.bentley.edu/ms/hfid

About Smashing Magazine

Smashing Magazine[104] is an online magazine dedicated to Web designers and developers worldwide. Its rigorous quality control and thorough editorial work has gathered a devoted community exceeding half a million subscribers, followers and fans. Each and every published article is carefully prepared, edited, reviewed and curated according to the high quality standards set in Smashing Magazine's own publishing policy.[105]

Smashing Magazine publishes articles on a daily basis with topics ranging from business, visual design, typography, front-end as well as back-end development, all the way to usability and user experience design. The magazine is—and always has been—a professional and independent online publication neither controlled nor influenced by any third parties, delivering content in the best interest of its readers. These guidelines are continually revised and updated to assure that the quality of the published content is never compromised.

About Smashing Media GmbH

Smashing Media GmbH[106] is one of the world's leading online publishing companies in the field of Web design. Founded in 2009 by Sven Lennartz and Vitaly Friedman, the company's headquarters is situated in southern Germany, in the sunny city of Freiburg im Breisgau. Smashing Media's lead publication, Smashing Magazine, has gained worldwide attention since its emergence back in 2006, and is supported by the vast, global Smashing community and readership. Smashing Magazine had proven to be a trustworthy online source containing high quality articles on progressive design and coding techniques as well as recent developments in the Web design industry. ❧

---

104. http://www.smashingmagazine.com
105. http://www.smashingmagazine.com/publishing-policy/
106. http://www.smashing-media.com