

50

GETTING STARTED

USER RESEARCH

UI DESIGN

CONTENT STRATEGY

FRONT-END DEVELOPMENT

USER EXPERIENCE BEST PRACTICES

by above the fold.®



i Introduction

GETTING STARTED

- 1 Think big picture
- 2 Connect the goals
- 3 Get everyone involved
- 4 Hold fewer (more relevant) meetings
- 5 Hear every voice
- 6 Look beyond feature requests
- 7 Design collaboratively
- 8 Over-communicate
- 9 Share knowledge early
- 10 Build relationships

USER RESEARCH

- 11 Build personas
- 12 Get face time with your users
- 13 Use paper prototypes
- 14 Test usability...online
- 15 Listen to actions
- 16 Vet your ideas
- 17 Choose your test wisely
- 18 Follow the flow

UI DESIGN

- 19 Prioritize top tasks
- 20 Obey real-world rules
- 21 Sketch your ideas
- 22 Research UI patterns
- 23 Follow the MAYA principle
- 24 Document your UI patterns
- 25 Use tabs effectively
- 26 Reduce clutter: show actions on hover

- 27 Provide lazy registration
- 28 Take advantage of web typography
- 29 Design your color palette
- 30 Provide instant feedback

CONTENT STRATEGY

- 31 Tell a story
- 32 Remember the microcopy
- 33 Learn the jargon
- 34 Plan a content audit
- 35 Rewrite, reuse, republish
- 36 Supplement with synonyms
- 37 Vary your medium
- 38 Split test your content
- 39 Account for edge cases
- 40 Anticipate errors

FRONT-END DEVELOPMENT

- 41 Highlight real-time changes
 - 42 Load lazily
 - 43 Build with progressive enhancement
 - 44 Strategize for the small screen
 - 45 Research your UI approach
 - 46 Use sprites to improve performance
 - 47 Take advantage of HTML5
 - 48 Organize your CSS
 - 49 Consolidate your files
 - 50 Separate CSS classes from IDs
- ii Learn more
- iii About ATF
- iv Appendix

Introduction

Creating a good **user experience** (UX) is often the key to a product's success. After all, how many customers will give a product or service a second chance if their first experience was a bad one? If you're involved in creating a product, it pays to design each aspect of the user experience so that your customers come away satisfied and delighted.

Of course, UX design has many facets; trying to learn everything at once, it's easy to get overwhelmed.

That's why we wrote this eBook. Our daily work in the UX design field has led us to some tried-and-true best practices that you can apply to most software design situations, whether you're a designer, copywriter, developer, or a mix of many roles. We hope that this book will teach you some new tricks, while reinforcing concepts you're already implementing today.

Use this book as best fits your needs. Skim through it on your own; incorporate a few practices into your next project; share these tips with your team; add them to your UX toolkit. And if you find this book useful, please pass it along to others as well.

— The Above the Fold team



GETTING STARTED

How do you kick off?

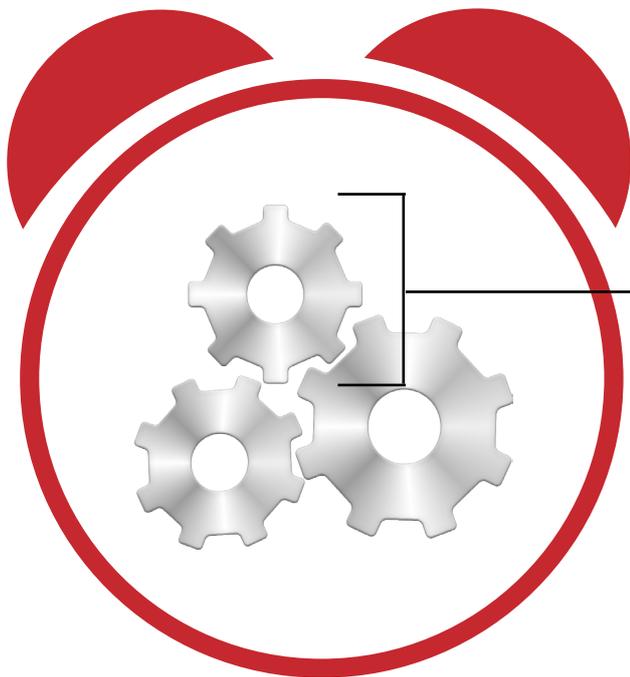
- 1 Think big picture
- 2 Connect the goals
- 3 Get everyone involved
- 4 Hold fewer (more relevant) meetings
- 5 Hear every voice
- 6 Look beyond feature requests
- 7 Design collaboratively
- 8 Over-communicate
- 9 Share knowledge early
- 10 Build relationships

Think big picture

The details of a project are clearly important. But getting too caught up in the details can lead to a disjointed and inconsistent user experience over time.

To keep the project on track, you need a shared vision of the project's future. Start by envisioning what the experience of using the product will be like 5 years from now – once all the kinks have been worked out. Keep that goal in mind while working on the day-to-day elements.

Document the short and long term goals early on in the process, and review them daily or weekly to keep them on the team's mind.



Think of the gears as the details...

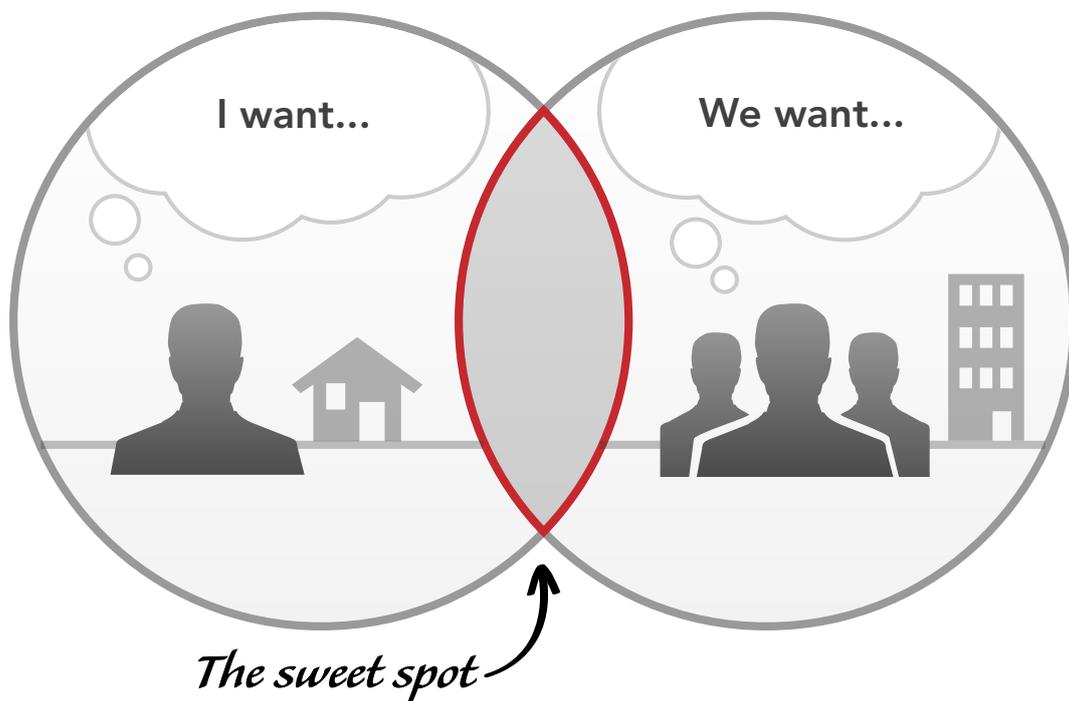
...and the big picture as the clock – all the gears need to fit together to make the clock work.

Connect the goals

Many projects start without a clear sense of the final objectives. The basic project goal may be known, but often the business goals and user goals are not articulated or shared. Additionally, the goals of a company don't always play nicely with user goals. This leads to designs that either frustrate users or don't align with business needs.

The solution? Identify all goals up front, and resolve conflicts early on.

Conduct interviews with stakeholders and C-level executives at the start of a project to learn their expectations. Then compare user goals and business goals, to help identify conflicts and create applicable design solutions.

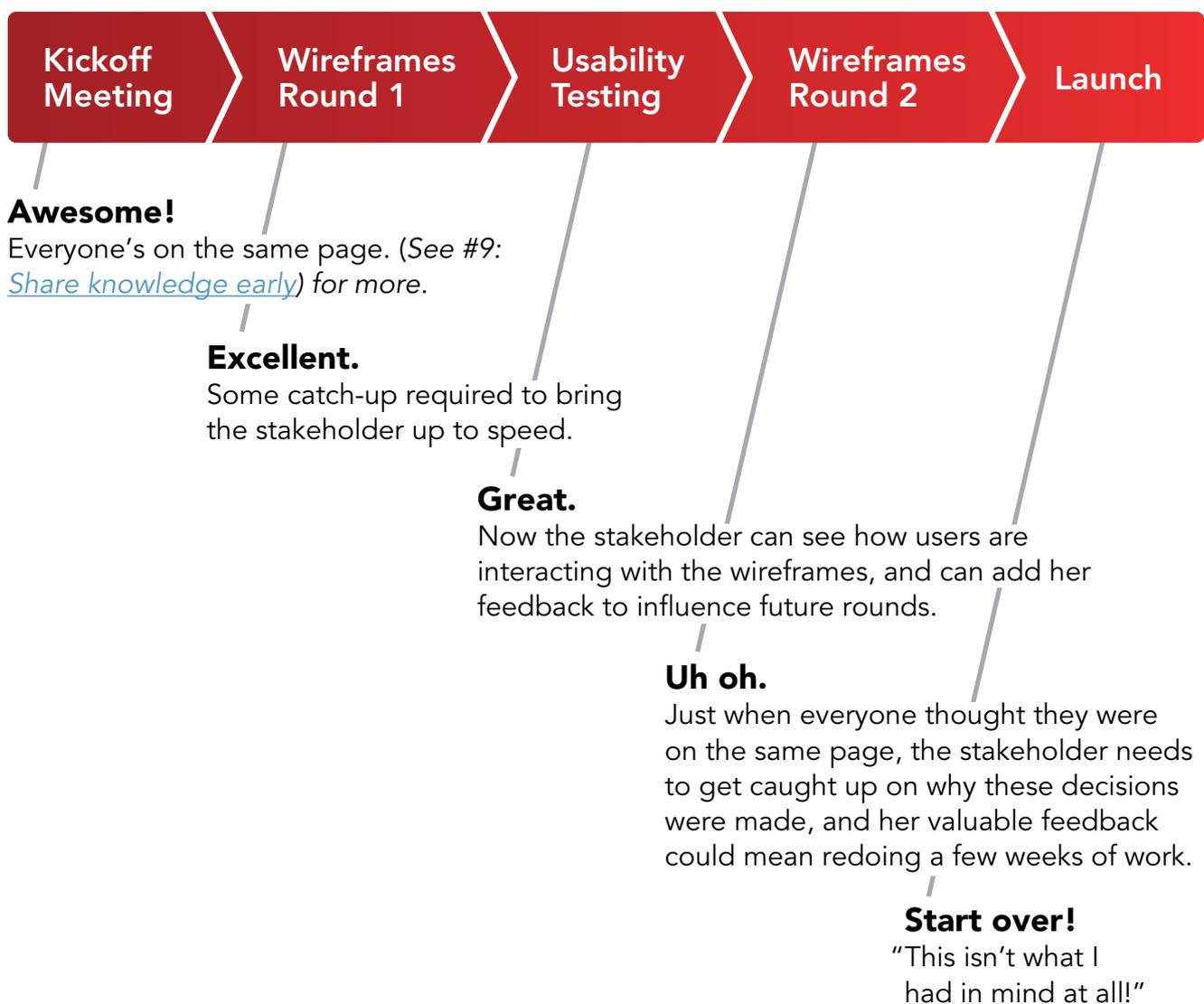
**Tip**

For more insight on how to balance user and business goals, read Paul Boag's article "[Business Objectives vs. User Experience](#)" in Smashing Magazine.

Get everyone involved

Stakeholders (those who are not directly working on the project every day) bring different perspectives to the goals and needs of a project.

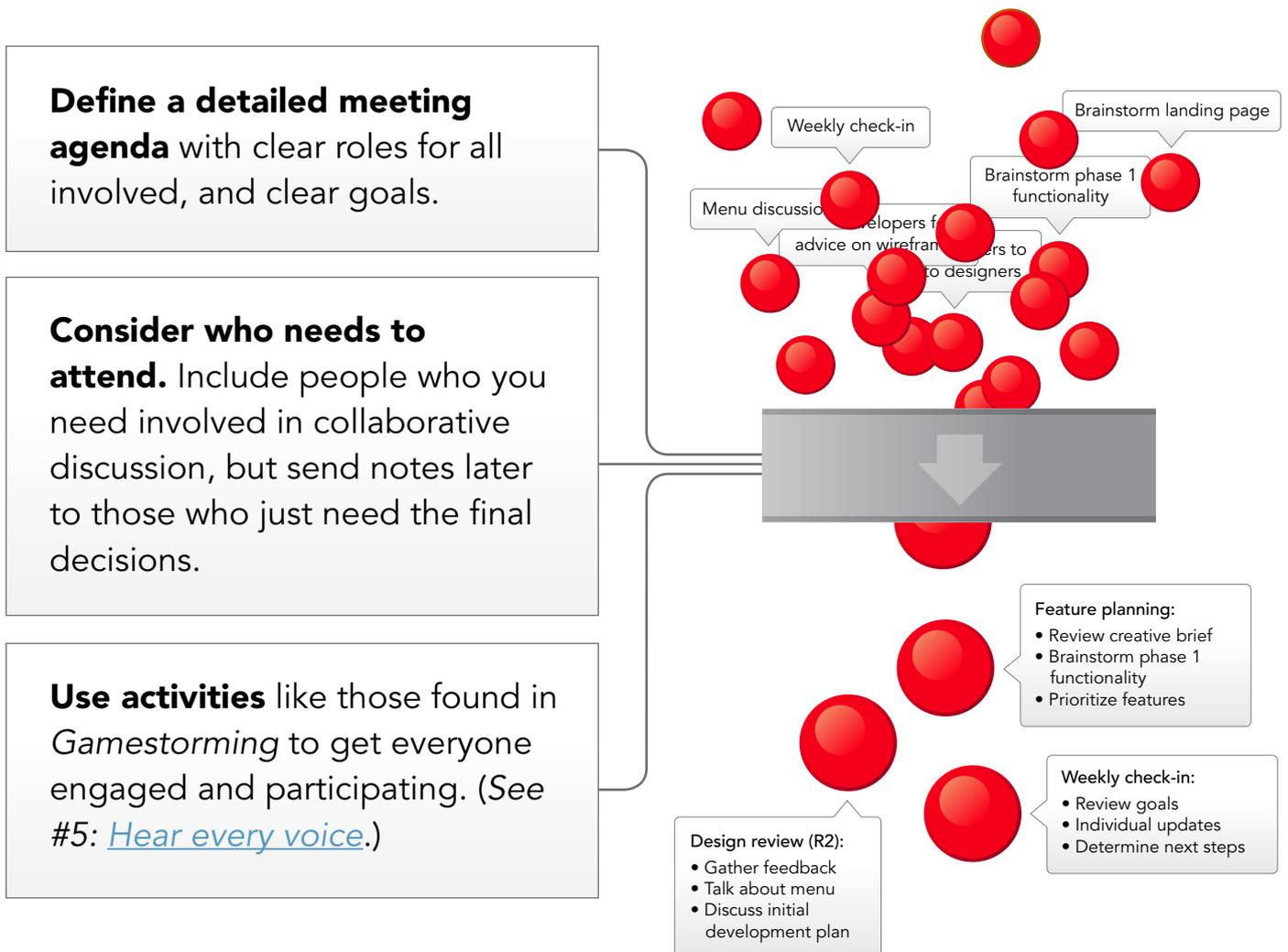
Where in the timeline do your stakeholders get involved?



Avoid the risk of having to start over by getting everyone involved as early as possible.

Hold fewer (more relevant) meetings

Meetings get a bad rap as a time sink, an interruption from “real” work, or an opportunity for some boss to hear the sound of his own voice. You can add a valuable opportunity for your team to collaborate and make meetings a positive thing with just three steps:



Hear every voice

Problem Meetings can be overpowered by one or two people with strong opinions and voices.

Solution Use brainstorming activities to hear every voice during the opening discussions, and make a point of asking the quieter people to provide their thoughts, whether in the moment or via email later.



Use 5-Finger Voting to determine group priorities.

Tip

Dave Gray's [Gamestorming](#) book is filled with group collaboration ideas, and new ones appear regularly on the [Gamestorming website](#).

Look beyond feature requests

When users start requesting features, take a step back and ask: *What is the problem they are trying to solve?*

For each feature request a user makes, identify the problem; from here you can brainstorm a well-considered solution. Even in the case where the user's suggestion is the best solution, you still gain better understanding of your customer and their needs.

"Can you add a tutorial on how to use the app?"



What users say

"Can you ensure first-time users will understand how to use the app?"



What users really mean

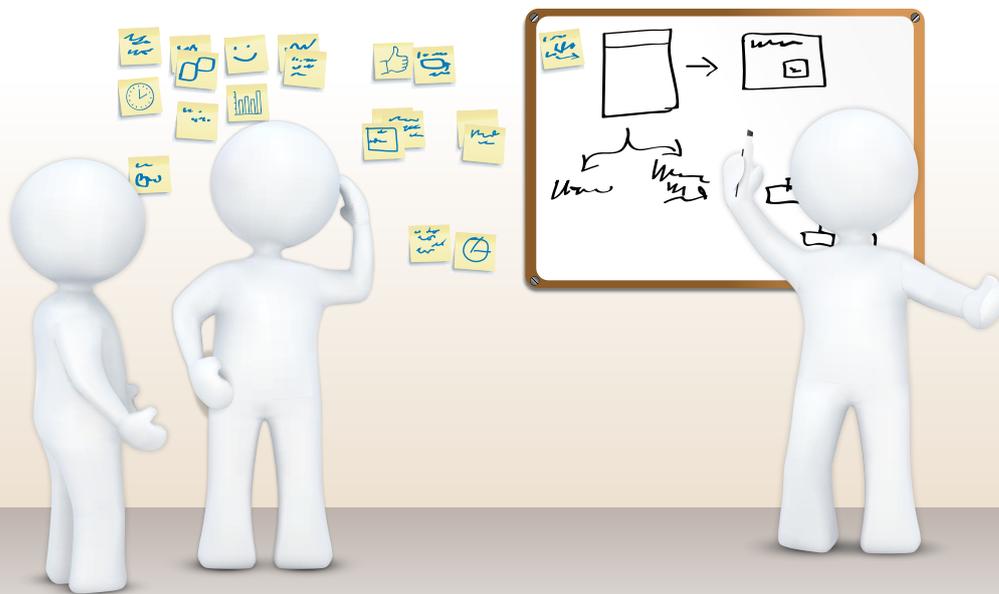
Tip

Keep your usability tests going regularly to keep the feedback loop open and continuous.

Design collaboratively

Collaboration requires a little extra time, but can vastly improve the overall design of the application if done effectively.

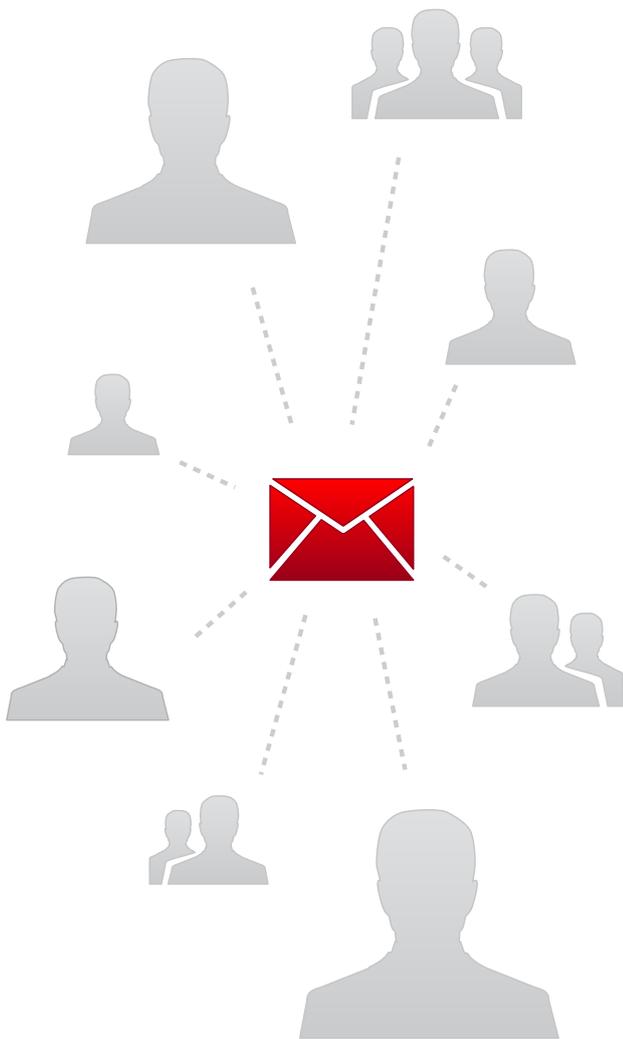
For example, the first round of wireframes is as much about discovering the best flow as it is designing the screen layouts, so it's a great time to get your team brainstorming.

**Tip**

Give the room 30 seconds to write features on sticky notes. Then, as a group, decide what features belong together, and watch the screens take shape.

Over-communicate

What would happen if you were hit by a bus? Or, less drastically, what happens during a team disagreement about a past decision, when no one has a written record?



Email recaps and regular check-ins feel like a huge waste of time when everything's going smoothly, but on complex projects, having a written record of team decisions and keeping everyone up-to-date regularly can prove invaluable.

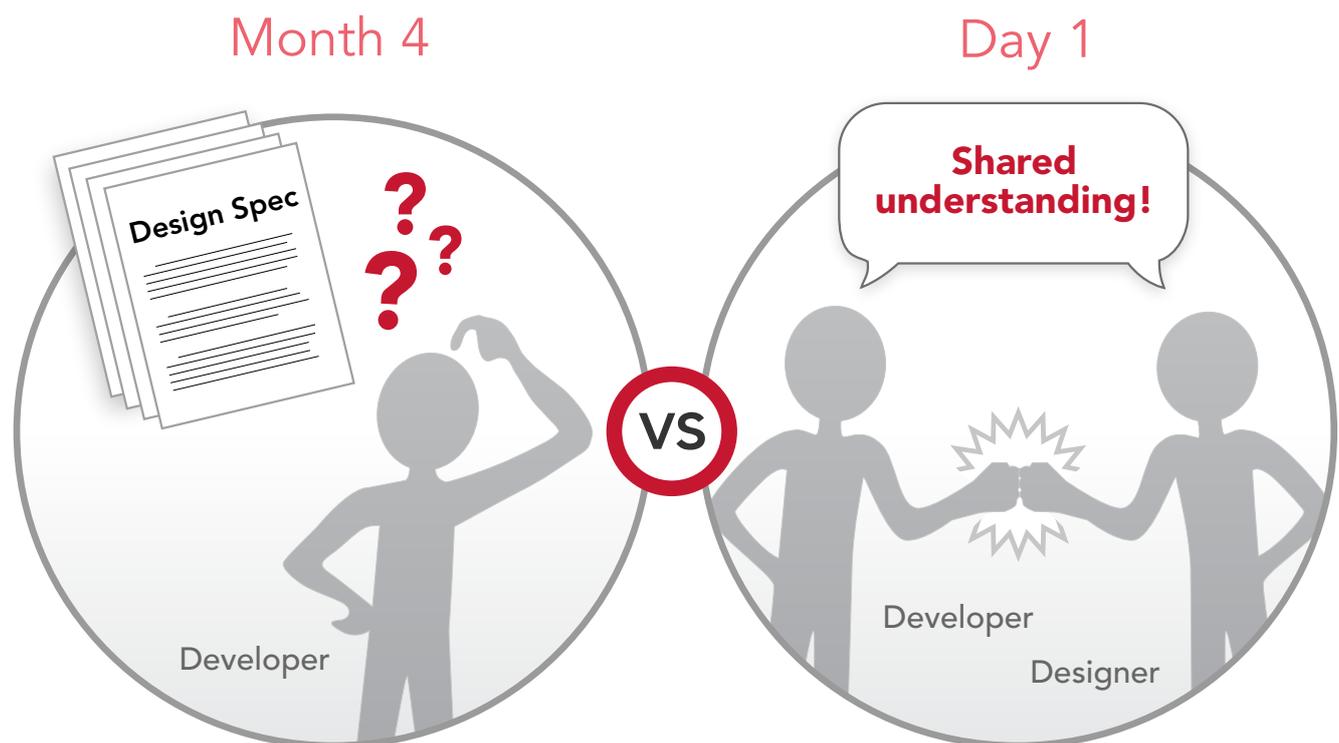
To save time, ask the team to spend 15 minutes at the end of each day to send out a recap of all decisions that influenced their work, or to bring a brief recap of their progress to any check-in meetings.

The whole team will benefit, as misunderstandings can be caught earlier in the process. And should anyone need to call in sick, there will be a written record of where they left off work.

Share knowledge early

Traditional design specification (spec) documentation requires a lot of effort, and it frequently can't keep pace with design and functionality changes. Shared understanding brings together multiple perspectives and sheds light on possible snags early in the process.

"[Designers] are in the problem-solving business, and [they] don't solve problems with design documentation. [They] solve them with elegant, efficient and sophisticated software." —Jeff Gothelf, [Lean UX](#)

**Tip**

See #8 ([Over-communicate](#)) for more information on where and how documentation can be beneficial.

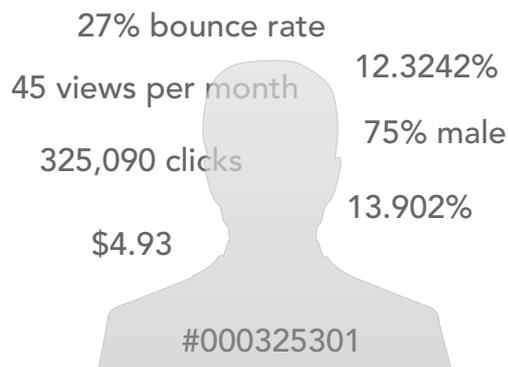
Build relationships

True or false?

Everything you need to know about your users you can learn from analytics.

False. Analytics are important, as are the results of any quantitative tests, but the best companies are built on **relationships**.

By focusing on your customers and getting to know them as individuals, you'll not only build brand loyalty, you'll also be able to improve your product with a steady supply of persona details (see #11: [Build personas](#)), usability testers, and users to interview.





USER RESEARCH

Learn what your users know

- 11 Build personas
- 12 Get face time with your users
- 13 Use paper prototypes
- 14 Test usability...online
- 15 Listen to actions
- 16 Vet your ideas
- 17 Choose your test wisely
- 18 Follow the flow

Build personas

Mike Jones manages products at Company X. His goal is to find the right audience for his product, and he learns about his audience through online forums and by interviewing users.

He has a team of two subordinates who help him collect and analyze data on their audiences. He holds weekly check-ins with his team, and is rigorous in sending feedback and updates.

This is an excerpt from a persona. Personas are created through interviewing multiple individuals in the same position, and compiling their personalities, needs, challenges, and expectations into an imaginary user. Through developing personas your team can identify and prioritize the specific goals of your target audience.



Mike Jones
Product Manager

"I love helping people"

Mike Jones manages products at Company X. His goal is to find the right audience for his product, and he learns about his audience through online forums and by interviewing users.

He has a team of two subordinates who help him collect and analyze data on their audiences. He holds weekly check-ins with his team, and is rigorous in sending feedback and updates.

Tip

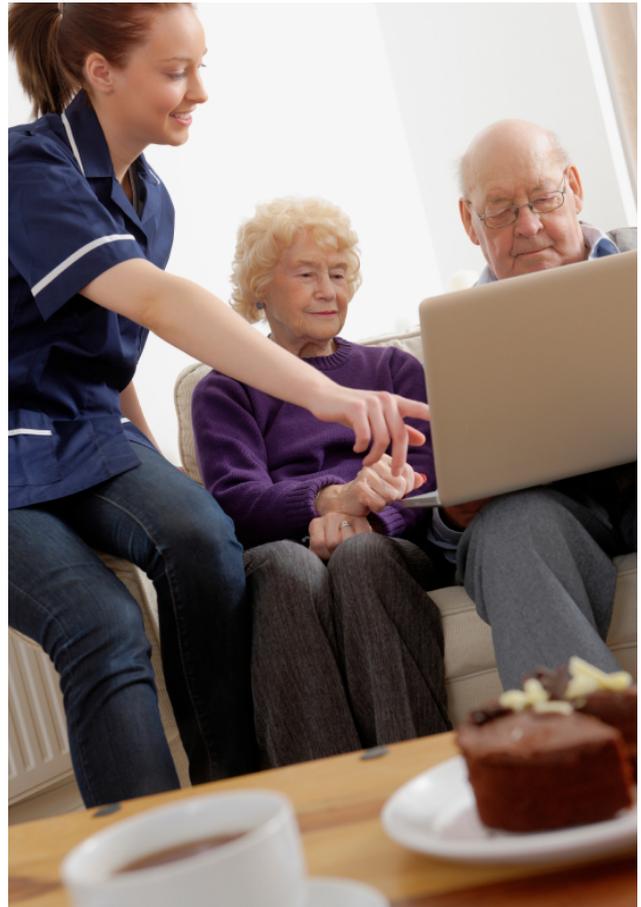
After building your personas, put them to work – prioritize features according to what your personas need. It's easier to make decisions for a specific user (or persona) than for a generalized audience.

Get face time with your users

Every product developer knows his market, but without user research he often lacks concrete data from which to form personas or prioritize features. Just remember this mantra:

A little face time is better than none at all!

User research can take many forms, including surveys, interviews, ethnographic studies, and usability testing – so do whatever you can, trying some of each if possible.



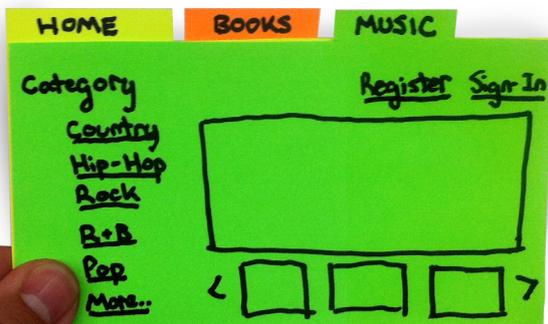
Tip

If you don't have access to users in your target market, go to your local coffee shop and offer gift cards in exchange for quick usability tests. It's surprising what insights you may gain!

Use paper prototypes

Design debates arise with any team; luckily most can be resolved with a paper prototype. Paper prototypes involve the same components as other prototypes: screens, users, interactions... everything but the computer!

- 1 Sketch the screens for the interaction you want to test, including buttons, modal windows, and other interactive pieces.
- 2 Ask an objective person to act as the "user."
- 3 As the user "clicks" on the "screen," act as the computer, simulating actions and interactivity. Your team members can observe successes and struggles, as in a traditional usability test.



This paper usability test offers immediate feedback in several ways:

- The team will receive clear, real-time feedback, solving design debates.
- You'll discover insights that no one thought of previously.
- Your objective user will provide you with a new perspective.

Tip

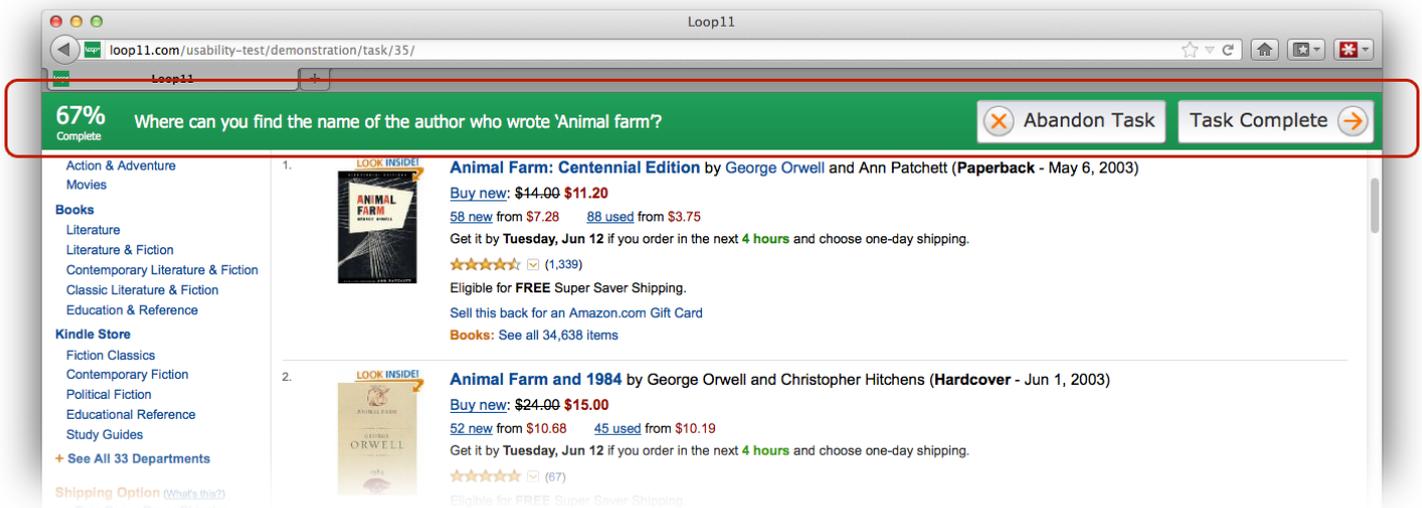
If possible, find a user who fits your target audience. At the very least, use someone who hasn't seen or heard about the in-progress application.

Test usability...online

Problem Traditional usability testing can become cost prohibitive, time consuming, and challenging to execute.

Solution Traditional usability testing is not the only way to catch the sticking points in your application. If you're pressed for time or short on funds, you can use an online service such as UserTesting.com or Loop11 for quick and dirty feedback within days.

Keep in mind that this solution works best when you're testing an application among general users rather than specific target audiences.



The screenshot shows a browser window with the URL loop11.com/usability-test/demonstration/task/35/. The task progress bar is green and shows 67% completion. The task question is "Where can you find the name of the author who wrote 'Animal farm?'". The results list two items:

- Animal Farm: Centennial Edition** by George Orwell and Ann Patchett (Paperback - May 6, 2003)
Buy new: \$14.00 \$11.20
58 new from \$7.28 88 used from \$3.75
Get it by **Tuesday, Jun 12** if you order in the next **4 hours** and choose one-day shipping.
★★★★☆ (1,339)
Eligible for **FREE Super Saver Shipping**.
Sell this back for an Amazon.com Gift Card
Books: See all 34,638 items
- Animal Farm and 1984** by George Orwell and Christopher Hitchens (Hardcover - Jun 1, 2003)
Buy new: \$24.00 \$15.00
52 new from \$10.68 45 used from \$10.19
Get it by **Tuesday, Jun 12** if you order in the next **4 hours** and choose one-day shipping.
★★★★☆ (67)
Eligible for **FREE Super Saver Shipping**.

Loop11 gives the user a simple way to complete usability tasks.

Tip

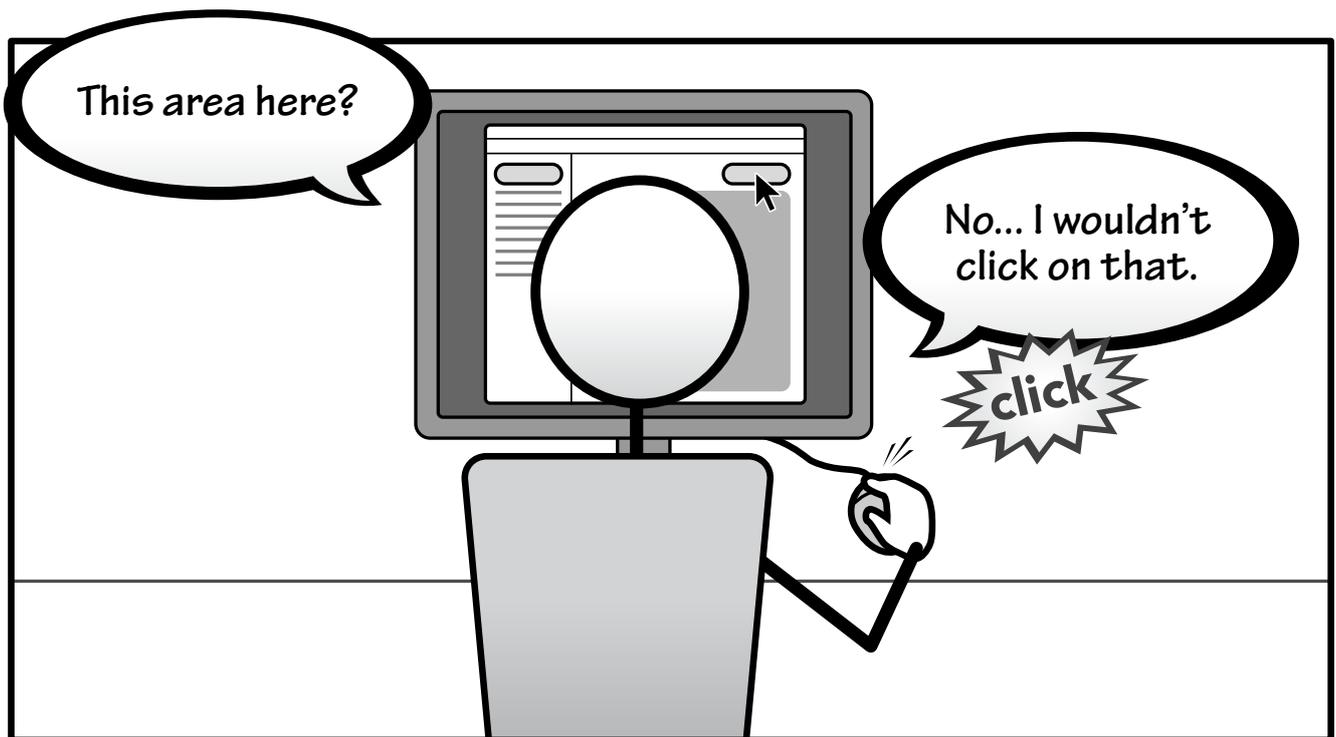
Use an online service for your regular monthly testing, but conduct the targeted, in person usability testing when rolling out new features or making major updates to the UI.

Listen to actions

During a usability test, users are asked to share their thoughts as they go. Sometimes they state a strong opinion, but their actions don't match.

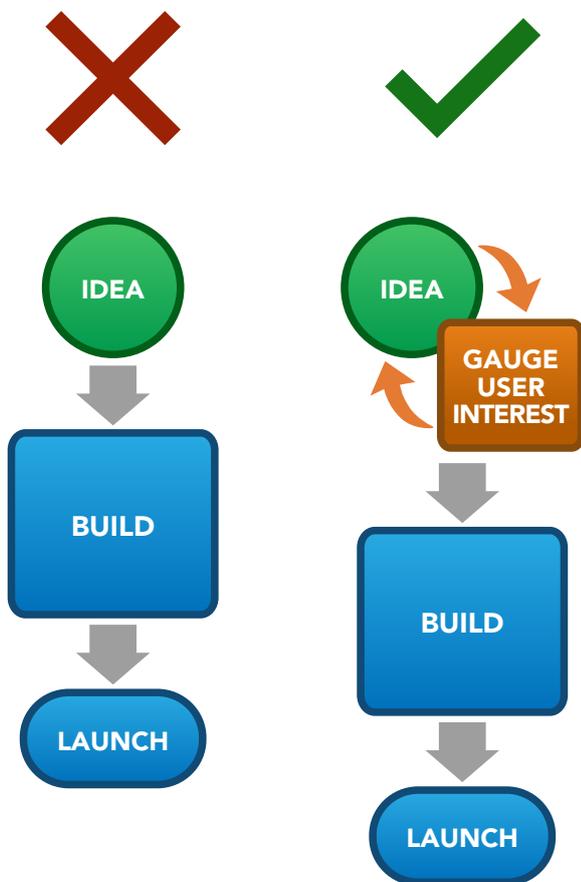
It is up to your team to observe their reactions and behavior, rather than focusing on the spoken statements. In particular, any time you hear a user say "I would do X..." take it with a grain of salt.

Help users along by asking them to talk about the decision making process, and what they think as they choose their course of action through your application. Listen to what they say, but pay closer attention to what they do.



Vet your ideas

Any entrepreneur can tell you that life isn't fair: no sooner do you come up with a brilliant idea, than you need to prove its worth. *Will users be interested in it? Is it worth being funded? Will it hit your target market?*



Startups know the story well: without a way to validate user interest, funding can be hard to come by and you can spend months or even years spinning your wheels. But now there are online services that measure user interest, and help you learn more about your target market by enticing your potential users to sign up for updates and more information!

Tip

The Internet provides many places to vet your ideas. Try [LaunchRock](#), for one.

Choose your test wisely

Do you need to know...

HOW MANY

If your question is:

- ✓ What price points are most effective?
- ✓ Which Call-To-Action gets the most click-throughs?
- ✓ Which design results in the most sign-ups?
- ✓ Which layout generates more sales?
- ✓ Which email campaign performs better?



A/B Testing

WHY

If your question is:

- ✓ Why don't people use my application/product?
- ✓ Why are users navigating to this page?
- ✓ Where are people getting stuck on my site? Why?
- ✓ Why aren't people buying?
- ✓ What is most effective about my site? Why?

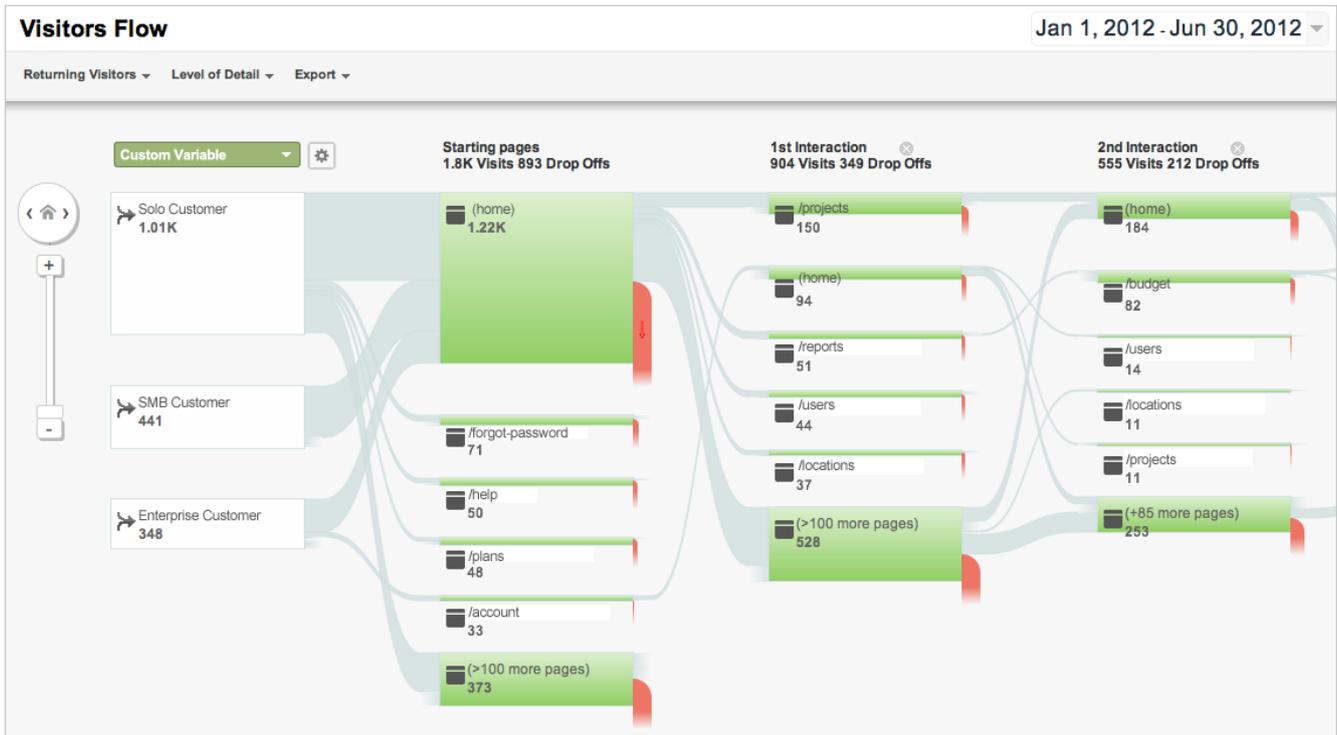


Usability Testing

Which test is right for your needs?

Follow the flow

Using Custom Variables in Google Analytics will help you segment your customer base and track where customers navigate to in your application.



- 1 Set a custom variable whenever you sign up a new customer.
- 2 View the Visitor Flow report in Google Analytics to see the flows your customers are following.
- 3 Cross reference the report with your expected navigation flow. If the two don't align, then it's time to rethink terminology or redesign the page where users are stumbling.

Tip

Take a look at the [Custom Variables setup guide](#) from Google Analytics.



UI DESIGN

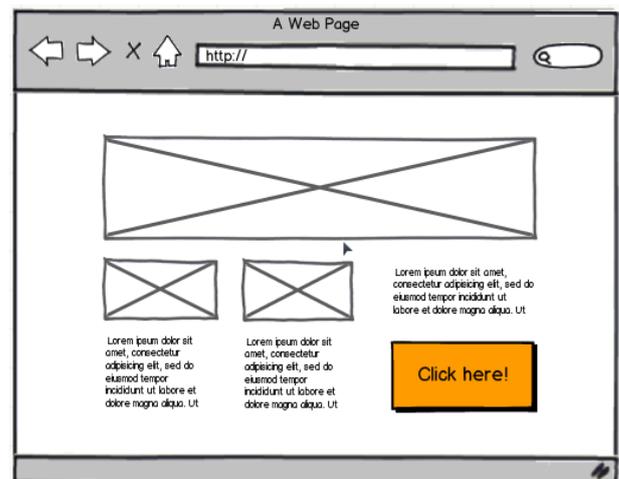
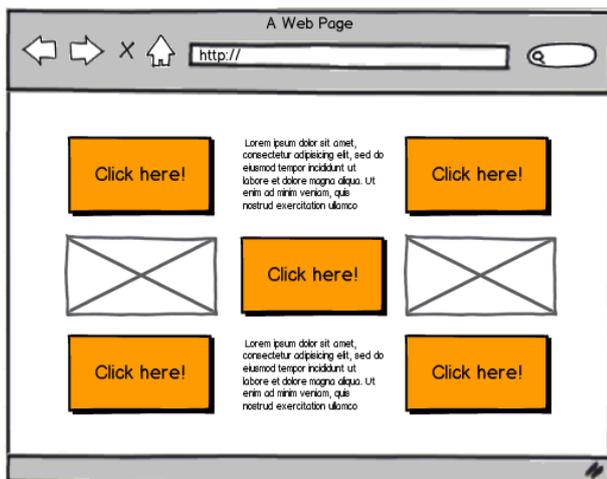
Designing with data

- 19 Prioritize top tasks
- 20 Obey real-world rules
- 21 Sketch your ideas
- 22 Research UI patterns
- 23 Follow the MAYA principle
- 24 Document your UI patterns
- 25 Use tabs effectively
- 26 Reduce clutter: show actions on hover
- 27 Provide lazy registration
- 28 Take advantage of web typography
- 29 Design your color palette (but not from scratch)
- 30 Provide instant feedback

Prioritize top tasks

Set one primary goal for your users on each screen. If you have multiple goals in mind, identify the primary goal by asking these questions:

- Who** is my ideal audience?
- Where** do I want to send them on my application/site?
- What** call to action will best direct users to the primary goal?
- How** do the secondary goals relate to the primary goal? (You may find that your secondary goals are so unrelated that they deserve a separate page altogether!) How many goals do I have altogether? Are they all for the same audience?

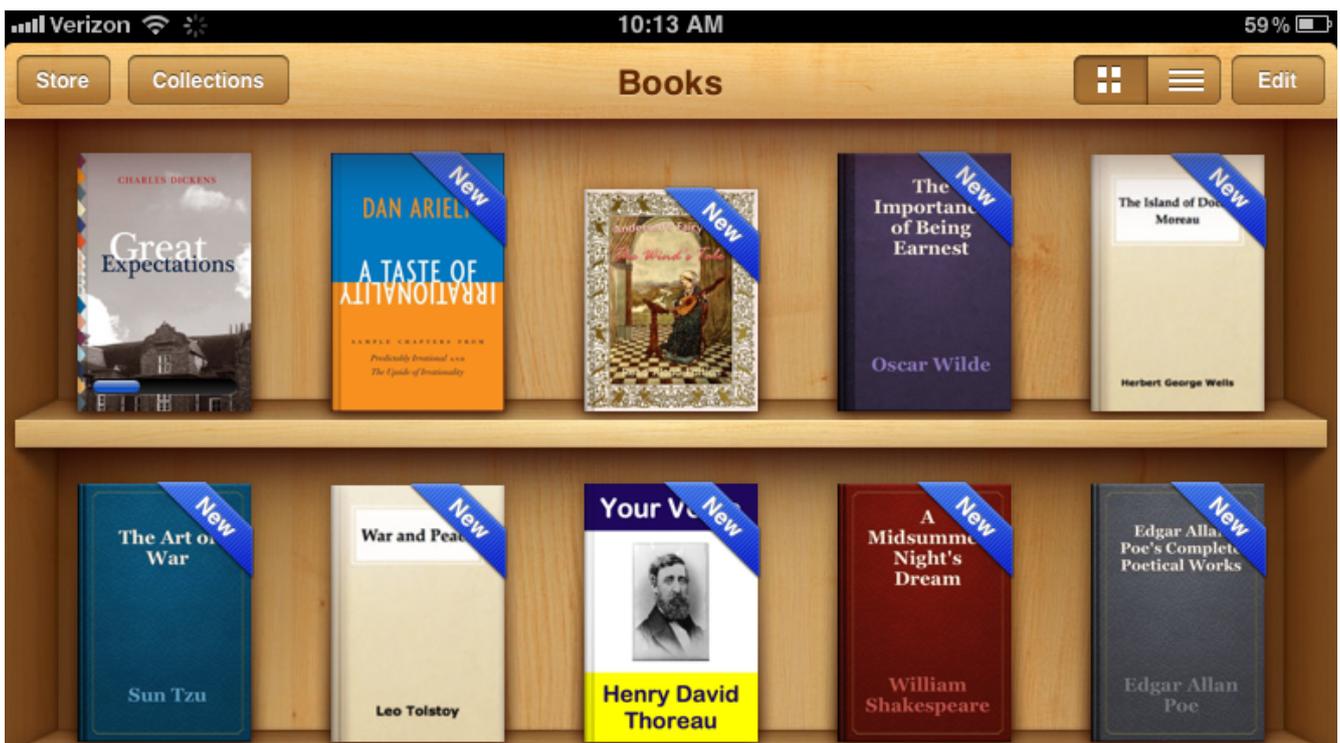


Prioritizing is much easier once you know your audience and their goals. With a little prioritization, your screens can become cleaner, crisper, and far more commanding.

Obey real-world rules

Even if you don't favor faux-real interfaces like Apple's iBooks, almost every interface you design will take some cues from real-world physical effects like lighting and perspective. Make sure those effects are applied consistently and sensibly; otherwise it's easy for users to become disoriented.

For example, if your buttons have a highlight along the top edge, and a gradient that darkens at the bottom, that implies a light source shining from above. Don't switch it up and include other elements on the same screen that look like they're lit from below.



Apple's iBooks app uses realistic lighting to depict a bookshelf.

Tip

Looking for more details? Check out the chapter on visual principles in Stephen P. Anderson's book, [Seductive Interaction Design](#).

Sketch your ideas

Don't be afraid of sketching! A quick drawing is often the best way to visually articulate goals and requirements, and you don't need art classes to do it.

Simple

Most interactions can be represented by simple rectangles, circles and triangles.

Complex

Digital tools such as [Balsamiq](#), [Mockingbird](#), and [Omnigraffle](#) come pre-loaded with basic shapes, text tools, form elements and annotation graphics.

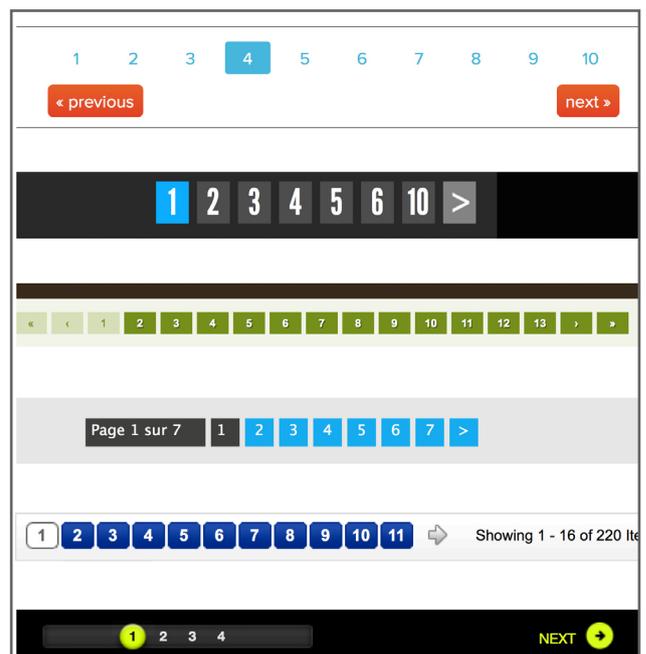


Research UI patterns

When you need to create a new piece of UI functionality, it's best to do some research first, to find out if someone else has already solved the problem you are facing.

Many sites exist to showcase UI patterns (reusable widgets that solve common problems) such as pagination, tabs, or faceted navigation.

In addition to saving you time and effort, UI patterns make use of existing conventions. Since these patterns are already familiar to users, they are usually preferable from a UX perspective, no matter how innovative your prospective new solution might be.



[UI-Patterns.com](#) shows examples of pagination.

Tip

[UI-Patterns.com](#), [PatternTap](#), and the [Yahoo! UI Design Pattern Library](#) are some examples of well-maintained UI pattern galleries.

Follow the MAYA principle

Problem

You can only raise the bar as high as your audience will let you; deliver an experience that's too far from what they're comfortable with, and they'll walk away.

Solution

MAYA (Most Advanced, Yet Acceptable) is a maxim to live by. Designers and innovators must pave the way toward the future, but in gradual steps—delivering experiences that break new ground, but still contain enough of the familiar to be acceptable to the people for whom they're designed.



If computers existed before anyone could understand them, no one would use them.

Tip

User interviews will help you to understand the daily life of the people for whom you're designing, and give you insight into what will be "acceptable" to them.

Read Jim O'Neill's article "[Deliver the Future, Gradually](#)" to learn more about the MAYA principle.

Document your UI patterns

Doing something right once saves you the trouble of needing to do it again. So when you build a UI widget that you (or others on your team) may want to reuse, don't keep it hidden – put it somewhere that everyone can find it when they need to.

Create a UI pattern library for your project – a living document (preferably in code) that compiles all the reusable patterns you have created, making them clear and easily available to everyone on the team, and to people who join in the future.

Your library could be as robust as Bootstrap (see right), or much simpler, depending on your needs.

Button dropdowns

Overview and examples

Use any button to trigger a dropdown menu by placing it within a `.btn-group` and providing the proper menu markup.



Works with all button sizes

Button dropdowns work at any size. your button sizes to `.btn-large`, `.btn-small`, or `.btn-mini`.



Split button dropdowns

Overview and examples



[Bootstrap](#), from the folks at Twitter, is a publicly-available UI pattern library.

Use tabs effectively

Tabs are a sorely misunderstood UI element. They are frequently employed as a navigation device, even in such popular places as Amazon.com or in the System Properties of your PC. Rarely are tabs used to do what they do best: alternating between views.

The debate rages on over whether or not tabs-as-navigation is actually “misuse,” but it can easily become overused. By following Jakob Nielsen’s [13 Guidelines for Tabs](#), you can ensure your tabs stay in their rightful place.



Tip

You can use UI pattern galleries to explore alternatives to tabbed navigation (such as scrolling, sidebar navigation, or dropdown menus).

Reduce clutter: show actions on hover

When each row in a data set or table contains a set of actions (such as “edit” and “delete”) the table can become cluttered.

Clean up the interface by only showing those actions when the user hovers over the row with her cursor.

ATF To-Dos

This is a compilation of to-dos from our staff meeting

- Talk about operating agreement
- Transfer from GoDaddy to a different domain
- Track the Udemy course Joe Baz 12 Jun 2012
- Content: dealing with conflicting usability feedback
- Create job description for consultant/biz dev
- Update Quarterly presentation deck based on feedback Joe Baz
- Update domain information per ICANN request

[Basecamp](#) shows Edit, Delete, and Move action icons when hovering over each item in a to-do list.

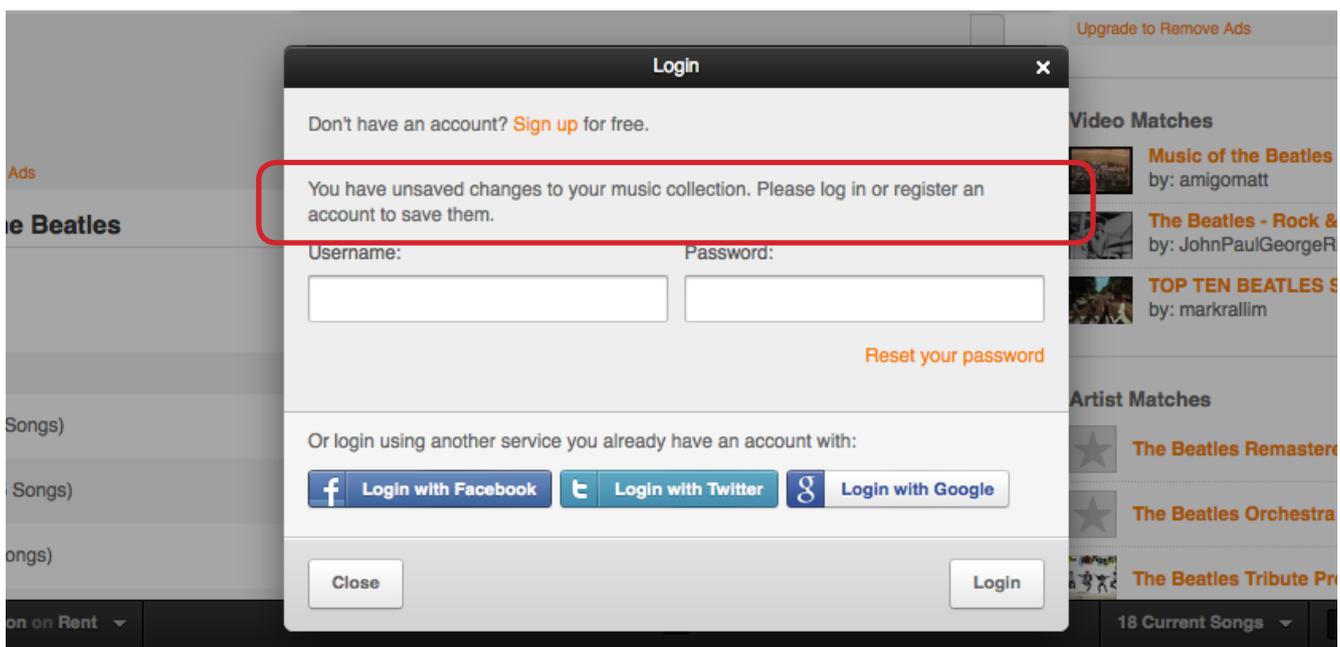
Tip

Hover doesn't work for mobile devices. Instead, consider cutting down functionality to allow only the basics, or filter the functionality into a few different pages.

Provide lazy registration

You wouldn't buy a car without taking it for a test drive. However, when you ask users to create an account before giving them a chance to use your service, you are asking them to do just that. This approach can scare people away from using your product.

Instead of asking for an immediate sign up, let users enjoy the service first, and then provide them a way to save their information and "lazily" create an account.



[Grooveshark](#) lets you create playlists, but to save them you must register.

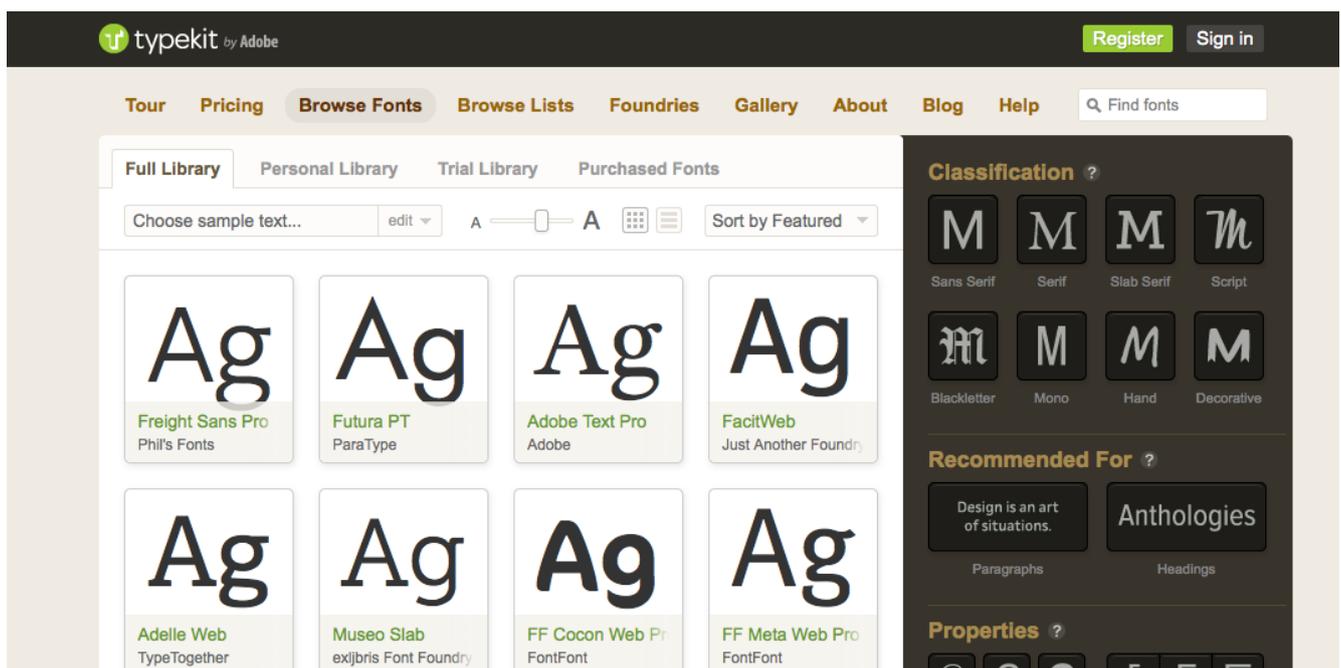
Tip

To make sign-up even lazier, allow integration with other platforms such as Facebook, Twitter, and Google+.

Take advantage of web typography

There was a time when Arial, Verdana, Georgia, Trebuchet and Times New Roman were the only fonts available to web designers. Putting text into images provided more options, but at the cost of SEO and accessibility.

Today, web typography is emerging as a solution. TypeKit and Google Web Fonts, among others, allow designers to use a wide range of fonts while leaving the text in HTML, where it can be read by search engines and screen readers.



[TypeKit](#) allows you to browse and try various typefaces.

Tip

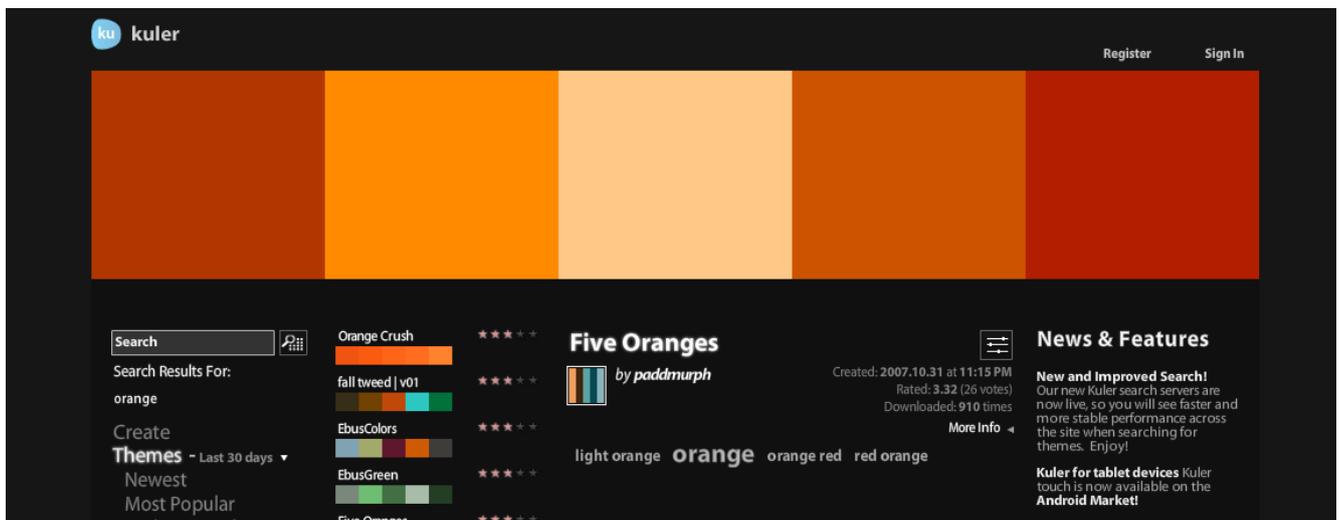
With great power comes great responsibility. Be judicious with typography choices to preserve the aesthetic of your website or application.

Design your color palette (but not from scratch)

When you need to create a color scheme for your new product or application, don't just make it up as you go along. Instead, take the time to plan out a palette, but design smart: start from a pre-made palette and tweak as needed.

Some tools to get you started include:

- [Adobe Kuler for Designers](#)
- [Color Scheme Designer](#)
- [21 Stimulating Color Palette Tools](#)



Adobe Kuler offers a constantly-updating library of color palettes.

Tip

Whatever color scheme you design, include one or two neutral tones for the text-heavy areas.

Provide instant feedback

Form fields should be designed with the best possible user experience in mind:

- 1 Fill out a form.
- 2 Receive an alert "Something on this form is incorrect!"
- 3 Give up and go home.

Wait. What? Let's try that again...

- 1 Fill out a form, and see clear indicators of whether your information is being accepted.
- 2 Correct errors quickly as you go.
- 3 Success!

User Name *	<input type="text" value="abovethefold"/>	
Email *	<input type="text" value="atf.com"/>	
Phone	<input type="text" value="Please enter a 10 digit number"/>	

Please enter a valid email address
e.g. ryan@example.com

The example above provides a checkmark icon for correct fields and a warning icon with a message for invalid entries.

Tip

Use JavaScript animations to make form field interactions friendly and fun. Example: [Tumblr's sign up form box](#) shakes to indicate incorrect field values (like a password that's too short).



CONTENT STRATEGY

Tell a story to create context

- 31 Tell a story
- 32 Remember the microcopy
- 33 Learn the jargon
- 34 Plan a content audit
- 35 Rewrite, reuse, republish
- 36 Supplement with synonyms
- 37 Vary your medium
- 38 Split test your content
- 39 Account for edge case
- 40 Anticipate errors

Tell a story

Storytelling has been a compelling form of knowledge transfer since before the written word.

From Plato to Steve Jobs, fairytales to business pitches, stories are an integral part of content strategy.

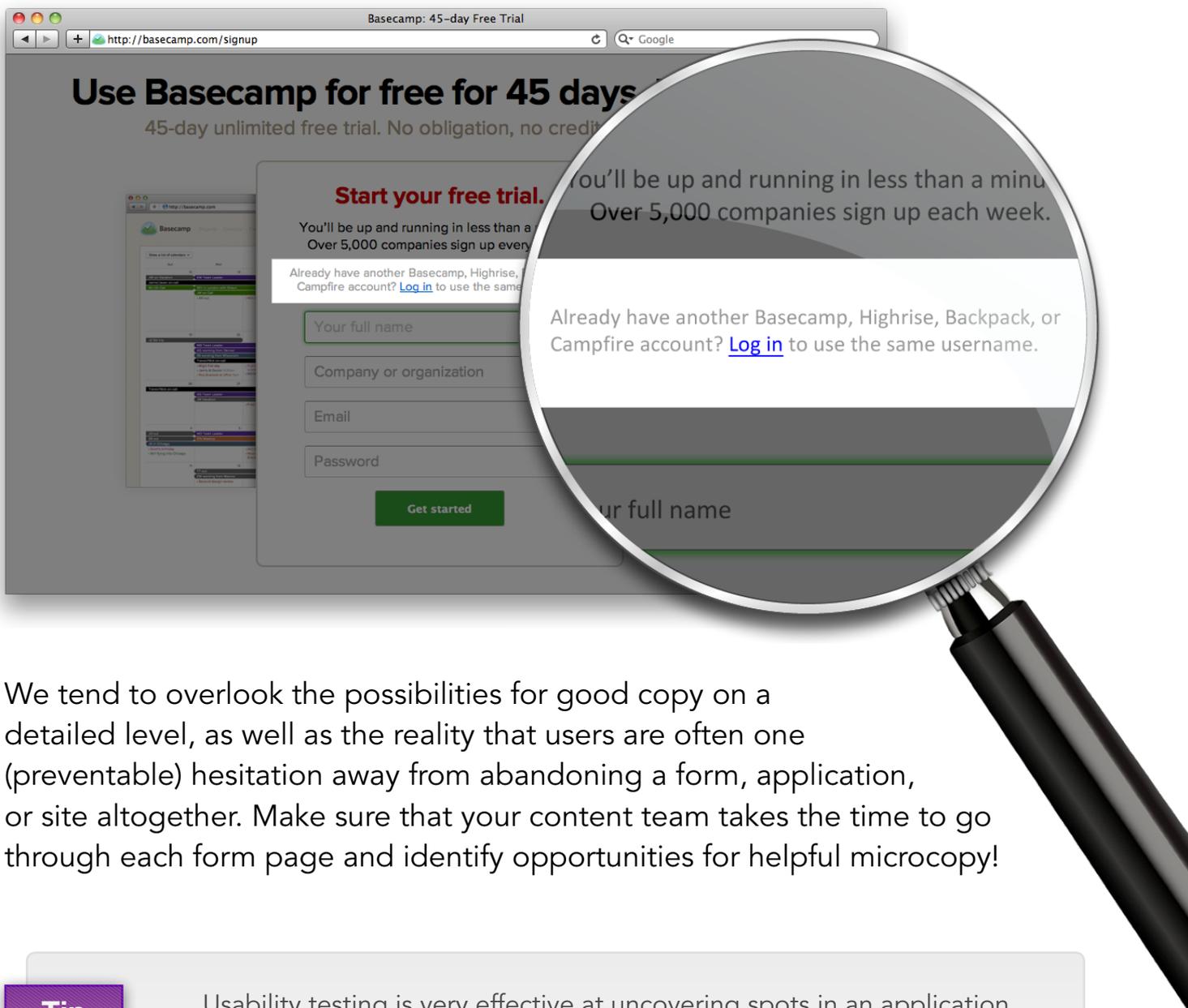
When you tell a story you engage your users. It can be the story of a journey, or a lesson learned, or a triumph, or a challenge, or nostalgia, relaxation, achievement, or many other possibilities.

It can be told through the colors you choose, through any medium, and in multiple ways.

By telling a story, you put your product in context and give meaning to your users.

Remember the microcopy

Microcopy refers to the tiny snippets of copy that often appear near form fields and inside complicated processes. These provide contextual help, clarification, or reassurance at moments of potential confusion.



We tend to overlook the possibilities for good copy on a detailed level, as well as the reality that users are often one (preventable) hesitation away from abandoning a form, application, or site altogether. Make sure that your content team takes the time to go through each form page and identify opportunities for helpful microcopy!

Tip

Usability testing is very effective at uncovering spots in an application where microcopy would be useful – it's often painfully easy to see where users hesitate or go wrong due to confusion.

Learn the jargon

We're not talking about business jargon... unless your target users are businessmen. The people who use your product have a language all their own. When you learn to speak their language, you can ensure your application doesn't miss the mark.

Starbucks coffee grande caffeine barista espresso chai triple-shot brownie frappuccino venti

Starbucks customers and employees share a certain specialized language.

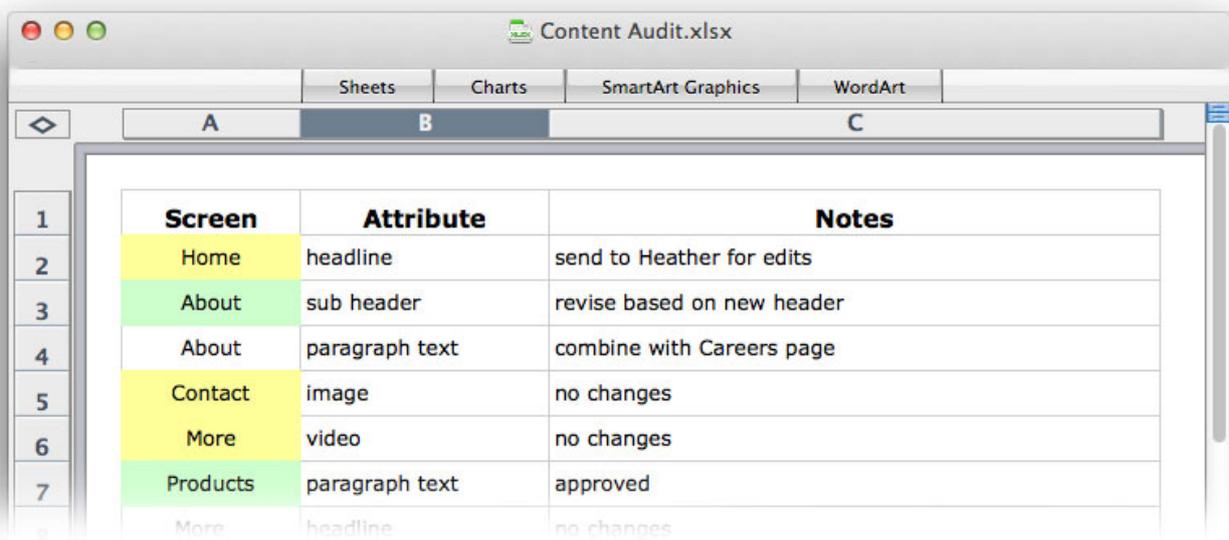
Tip

Interview your users and listen to how they express themselves, so you can match their tone. (See #12, [Get face time with your users.](#))

Plan a content audit

A content audit is a critical part of building or redesigning any complex application or site. Planning and organizing your content audit can be as easy as 1-2-3:

- 1 Set up a spreadsheet with sections for Screen, Attribute, Messaging, Call to Action, Audience, and Notes
- 2 Consider each attribute on each screen (Attributes include headline, sub header, paragraph text, image, video) and the relevant messaging.
- 3 Use the completed spreadsheet to help make decisions on what content to keep, and then track the creation of new content.



The screenshot shows a spreadsheet window titled "Content Audit.xlsx" with a table containing the following data:

	Screen	Attribute	Notes
1			
2	Home	headline	send to Heather for edits
3	About	sub header	revise based on new header
4	About	paragraph text	combine with Careers page
5	Contact	image	no changes
6	More	video	no changes
7	Products	paragraph text	approved
8	More	headline	no changes

Rewrite, reuse, republish

Creating quality content can be a lot of work. Writing social media updates, blog posts, forum discussions, site content, and videos in a consistent voice on a regular basis can cause burnout. So what do you do? Rewrite, Reuse, and Republish.

Quote your website. Make a blog post into a video. Make a discussion into a SlideShare by adding images to illustrate the main points. Quote blog posts as forum responses.

With some savvy marketing, a little content can go a long way.

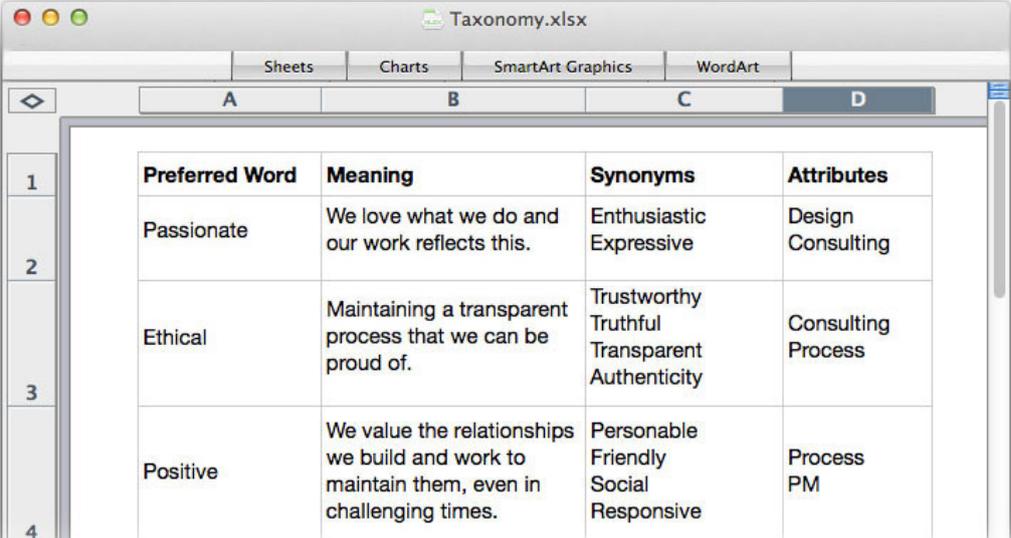
**Tip**

For more suggestions on reusing content, view [Barbra Gago's slideshare, "15 Ways to Distribute Content."](#)

Supplement with synonyms

Problem Different people use different terms when they think about your product – even members of your target audience will interpret one word in multiple ways. Ensuring that users can use different words to find what they're looking for on your site is a content strategy challenge.

Solution Create a taxonomy! By writing up a list of synonyms for each important term relating to your company, product, and application, you will learn to identify with different segments of your audience. Once you have a basic taxonomy, all future content will come together much more easily.



	A	B	C	D
	Preferred Word	Meaning	Synonyms	Attributes
1	Passionate	We love what we do and our work reflects this.	Enthusiastic Expressive	Design Consulting
2	Ethical	Maintaining a transparent process that we can be proud of.	Trustworthy Truthful Transparent Authenticity	Consulting Process
3	Positive	We value the relationships we build and work to maintain them, even in challenging times.	Personable Friendly Social Responsive	Process PM
4				

A sample taxonomy used by Above the Fold.

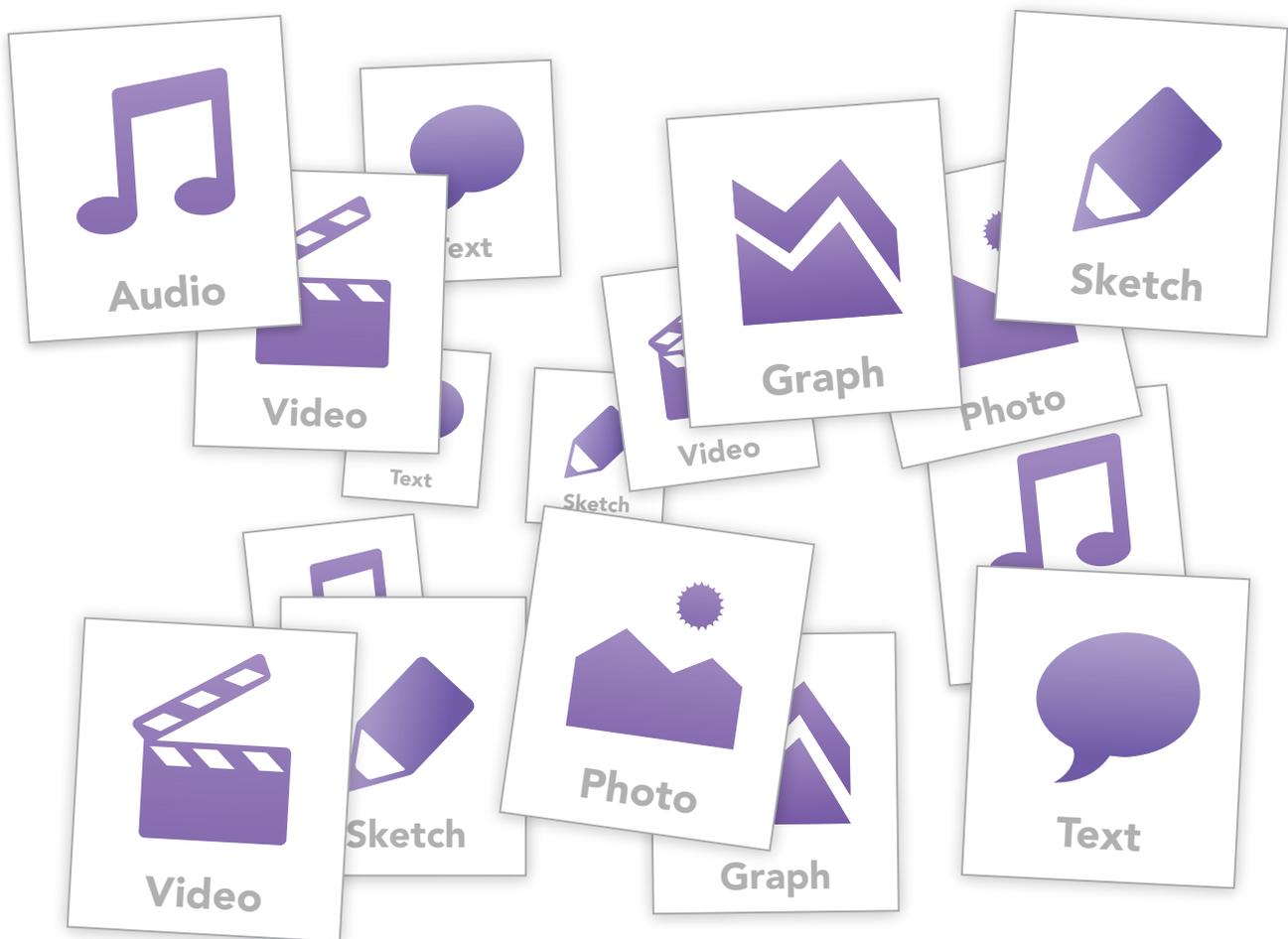
Tip

Read Heather Hedden's article, "[Better Living Through Taxonomies](#)" for a more detailed explanation on taxonomies.

Vary your medium

Content is more than just copy! Consider where your audience spends time, and how they access information.

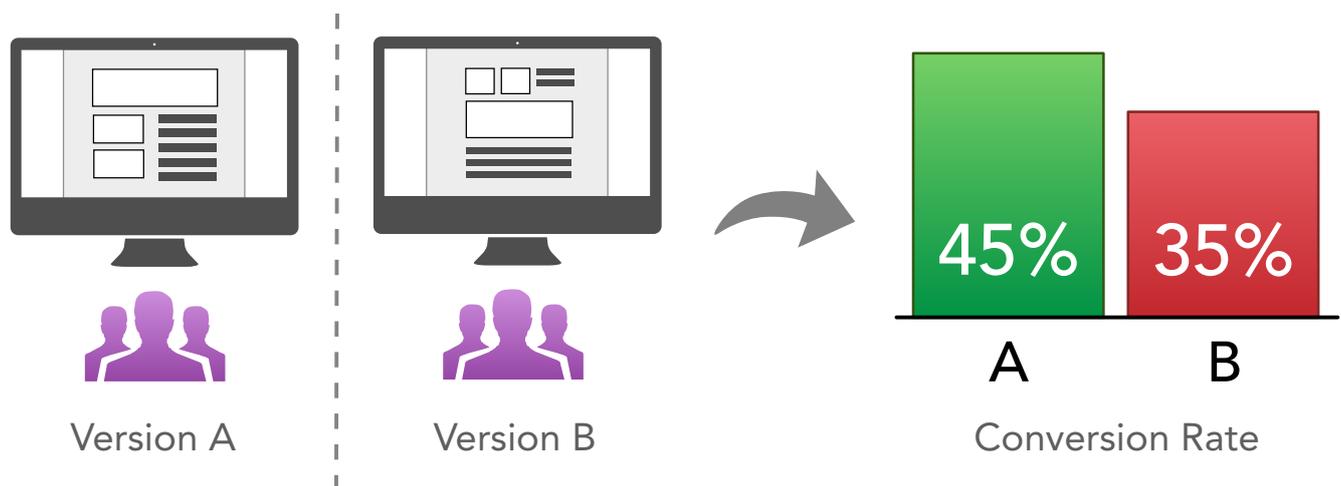
If your audience tends to surf the web online during work hours, they may find a blog post more helpful than a noisy video. Or if you have a particularly complex piece of information to get across, an infographic can help clarify it.



Split test your content

After creating new content, instead of simply replacing your old content, conduct an A/B split test to see which delivers better results in the real world.

In a split test, each user is served one of the two versions at random, without knowledge of the other one. Users' behavior on each of the versions can then be tracked and measured.

**Tip**

Tools like [Google Website Optimizer](#) can help you automate the testing process, allowing you to run tests easily by installing a simple tracking code.

Account for edge cases

Edge cases are a reality in every design. When dealing with a database application, understanding the real data that goes into the system is key to identifying potential edge cases which can cause the application to break.

Provide your team with real or simulated data early in the design process. With actual data (not “lorem ipsum” or filler data), the team can see how the interface might break if the design doesn’t accommodate the edge cases.



Address
1234 Lorem Ipsum Street



Address
686 Massachusetts Ave, Suite 304, Cambridge, MA 02139
9674 Broad St, Chicago, IL 60625
31 Spooner Street, North Kingstown, RI 02852

While designers should design for the most practical use cases first, these edge cases can’t be ignored. The more complex the application, the more edge cases need to be considered.

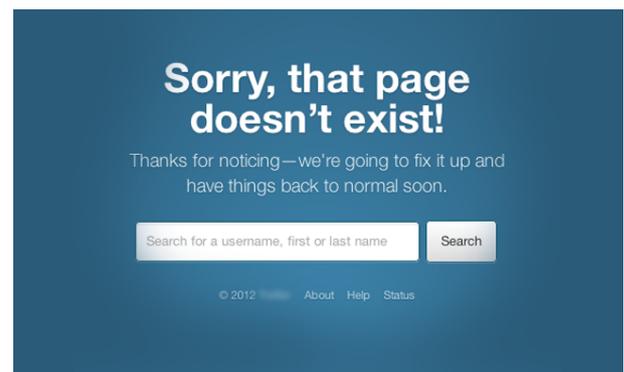
Tip

Check out Luke Wroblewski’s [interview series](#) with designers who give their perspective on edge cases.

Struggling with edge cases? Ask your development team for some help interpreting the content and structure of your databases.

Anticipate errors

Which of the following two error messages makes you feel confident about the website you're visiting?



Considering the user experience includes considering how we present the user with error messages. Detailed and appropriate (and sometimes even humorous) copy helps the user to move forward.

As your team designs the application, think through the possible errors a user might encounter at each step of the process. Then craft a response or solution for that error to help the user get back on track.

Tip

Ask developers to provide a list of error messages, along with context such as when users will see those messages. Then you can create new error message copy appropriately.

FRONT-END DEVELOPMENT

Build a strong base

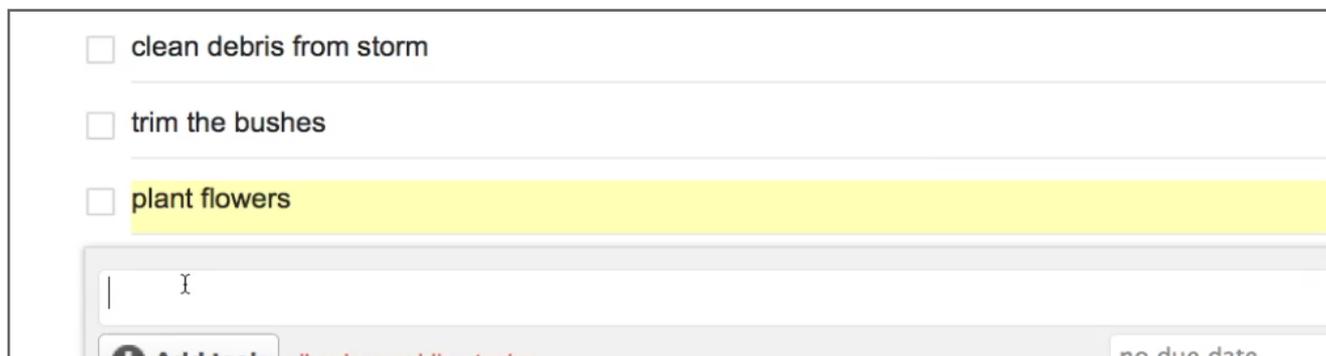
- 41 Highlight real-time changes
- 42 Load lazily
- 43 Build with progressive enhancement
- 44 Strategize for the small screen
- 45 Research your UI approach
- 46 Use sprites to improve performance
- 47 Take advantage of HTML
- 48 Organize your CSS
- 49 Consolidate your files
- 50 Separate CSS classes from IDs

Highlight real-time changes

Failing to notice changing information has consequences. When you fail to notice a friend's haircut, you only risk social embarrassment. But when a user fails to notice information that just updated on a page in your application, the consequences are more dire – she may not know what to do next, and you may lose her as a customer.

Build in visual cues to draw attention to information that has just changed. Here are some ways to transform the text on your page:

- Change the background color surrounding the item in question
- Change the font color or font weight
- Consider adding in some animations, either via CSS3 or JavaScript



[Todoist](#) briefly highlights text that was just added.

Tip

Regardless of your choice of transformation, you can gradually “fade” the effect back to normal to signal to the user to continue on her journey.

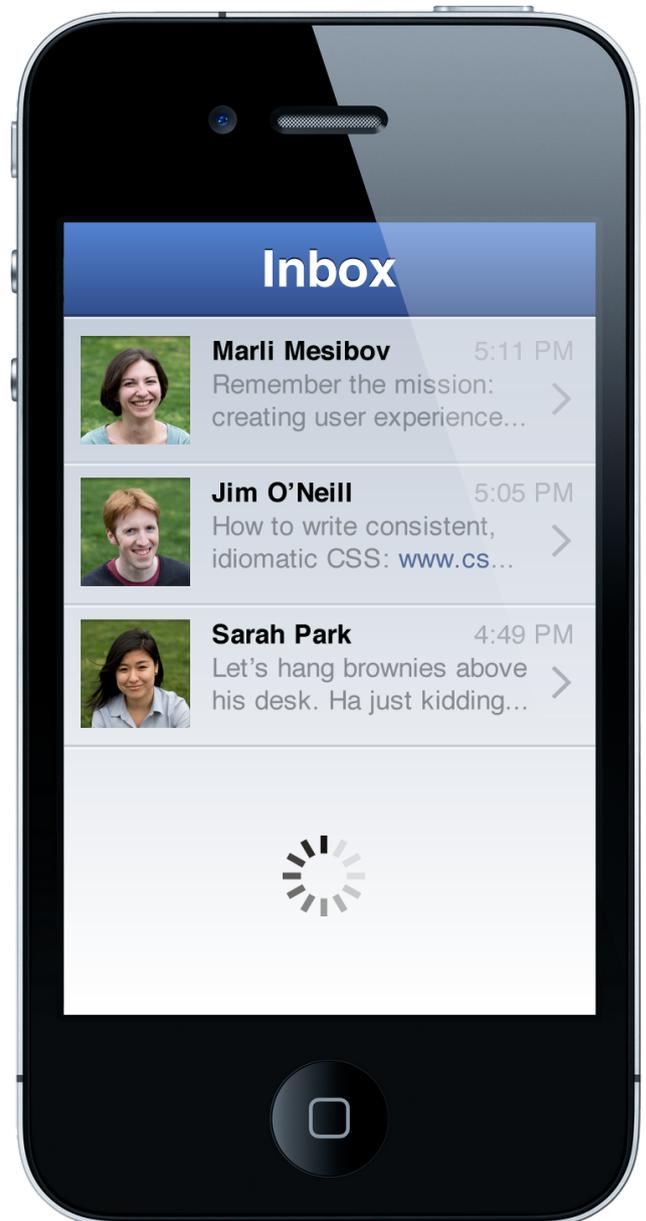
Load lazily

Problem

Traditionally loading pages allows all the data to load before anything appears onscreen. This often leaves users waiting, which nobody likes.

Solution

Use **lazy loading**: get all the static page elements loaded first, and then load the immediately visible data. As the user scrolls down, more elements can load, progressively.



Build with progressive enhancement

Anyone with an older Internet-enabled phone has encountered the major problem with “graceful degradation.” Older devices and browsers can’t handle new features and interactivity, and it is often painfully clear that features are missing, because older devices were a low priority.

When a user can tell that they are missing out on features because the developers didn’t bother, that’s a bad experience – and a preventable one.

To provide a good experience for everyone:

- 1 **Focus on content first.** Build a basic experience that everyone can access, which delivers your content in a clear, readable way.
- 2 **Apply the visual presentation next,** for devices that can handle it.
- 3 **Layer the interactivity on top,** for those capable devices.

This approach, done right, leaves no one behind.



Tip

For more on progressive enhancement, read [Aaron Gustafson's article](#) in A List Apart, or [Filament Group's book](#) on the subject.

Strategize for the small screen

The use of mobile devices for browsing the web is so prevalent, every single website and application should be usable on a small screen. But far too many aren't, either due to lack of understanding or lack of resources.

If your organization doesn't have the resources to design a full mobile solution, you can still achieve mobile-friendliness quickly and easily.

Consider using CSS media queries to make a few small layout and navigation adjustments for smaller screens, or fall back on a nicely-designed text-only stylesheet if you need to. These considerations need to be on your radar, especially if you don't have buy-in to go forward with a full-scale mobile effort.



Tip

Get started with Responsive Web Design by reading [Ethan Marcotte's article](#) on A List Apart.

Research your UI approach

Which UI development approach is better for you: programmatic or declarative?

Programmatic

- Develop the UI in code.
- Can be harder to learn, read and understand.
- UI is coupled with your code.

Declarative

- Create the UI in markup.
- May not provide much out of the box functionality.
- Flexible -- often the better choice.

Ultimately, the best choice for you depends on your application and your team. Evaluate the [pros and cons of each approach](#) before making a final decision.

```
// Javascript
<script type="text/javascript">
  new Ext.Application ({
    launch: function() {
      new Ext.Panel ({
        fullscreen: true,
        html: "Hello world!"
      });
    }
  });
</script>
```

A programmatic "Hello World" app in Sencha.

```
<!-- HTML -->
<div>Hello world!</div>
```

A declarative "Hello World" app in HTML/jQuery.

Take advantage of HTML5

Did you know that HTML5 provides many new input types? Browsers on most smartphones and tablets often change the virtual keyboard layout based on the input type in a way that saves users time and frustration.



New field types include:

- url
- email
- datetime
- datetime-local
- date
- month
- week
- number
- range
- color
- search
- tel
- time

Tip

If you're worried about browser support, never fear. Browsers ignore any input type they don't understand and render it as a normal `<input type="text" />`.

Organize your CSS

Applying a standard organization plan to your CSS code will prevent it from becoming bloated and disorganized, as is all too common in large projects.

- Use comments to create **headings** for different sections of the document, and comment liberally throughout the file.
- Include a **table of contents** within comments at the beginning of the file, listing all the different sections in order.
- To make searching within a CSS file fast and easy, precede each heading with a **search flag** – a special character that is uncommon in CSS code, such as an equals sign (=). For example, if the page layout styles are in a section named “=Layout,” then you only have to search for “=La” to jump directly to that section.

```
/* Table of Contents
=Global Styles
=Font Sizes
=Header Styles
=Body Styles
=Footer Styles
*/

/* =Global Styles */
html, body, h1, h2, h3, h4 {
    margin: 0;
    padding: 0;
    border: 0;
}

/* =Font Sizes */
body {
    font-size: 100%;
}
```

Consolidate your files

Maintaining a set of well-commented CSS and JavaScript files that are separated according to their purpose is great for development, but slows down performance on a production website or application. How can you consolidate?



As part of your standard process for pushing code to the production site, merge all CSS into one file and all JavaScript into another file. Then minify the code by stripping out all extraneous comments and white space. This can result in significant performance savings, due to smaller file sizes and fewer HTTP requests.

Tip

Consolidation and minification can be done automatically using a tool such as [Minify](#).

Distinguish CSS classes from IDs

Combining CSS selector types without a structure produces messy, unmanageable code. Keep your selector types in line with a general guideline.

Use CSS classes (e.g. ".page-title") to apply visual styles. This works well because classes are infinitely combinable and they cascade easily.



Use IDs (e.g. "#cta-donate") mainly to pinpoint specific elements in the DOM in order to trigger interactive behavior. This is great for performance, since the browser can find an ID faster than it can find a class name.

**Tip**

Plan how to structure your markup in advance, but accept that every situation is unique and that you will encounter exceptions.

Learn more

We hope the tips in this book give you a good starting point for improving your UX. If you find you have more questions on user experience, best practices or anything relating to this book, contact us! We love conversation and we're happy to help with all your UX needs.

Questions or comments?
Get in touch with ATF:



Email us



Visit us

Above the Fold is located at:
686 Massachusetts Ave, Suite 304
Cambridge, MA 02139

Web: www.abovethefolddesign.com

Email: hello@abovethefolddesign.com

Visit us on Facebook:
www.facebook.com/abovethefold

Follow us on Twitter:
www.twitter.com/abovethefold

Did you like this eBook?
Share it with your friends:

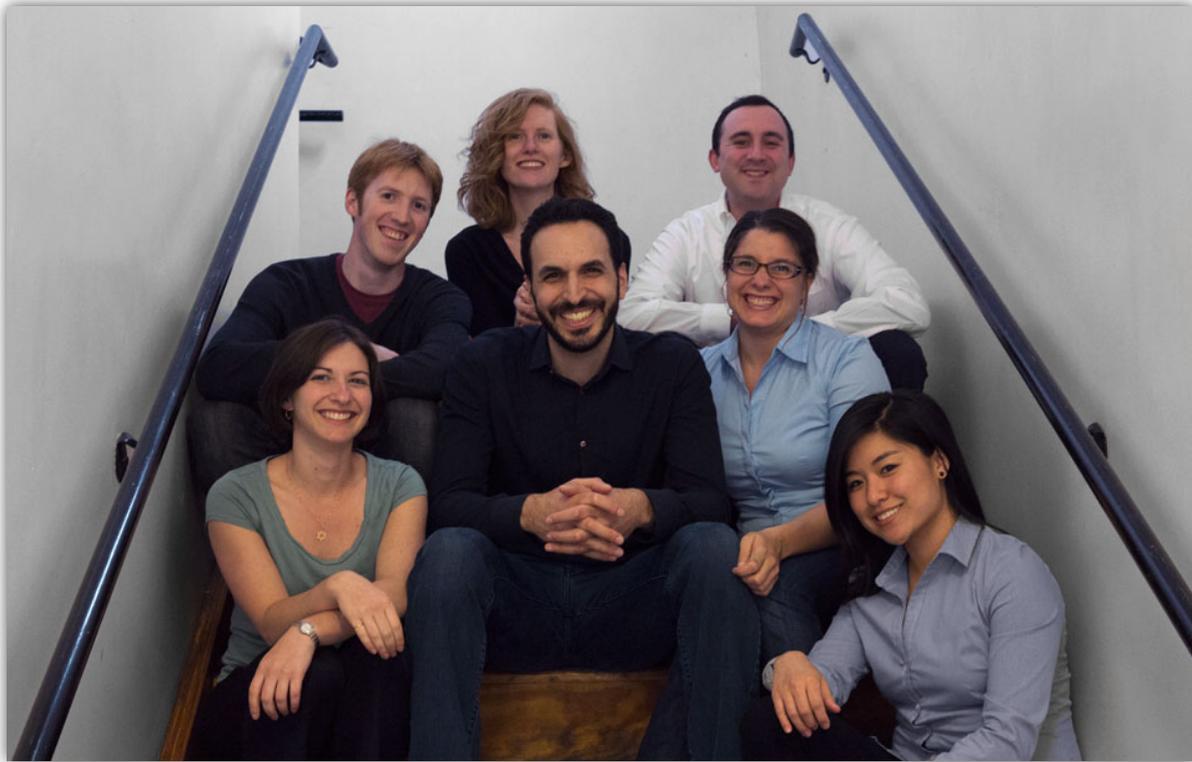


Share us



Tweet us

About ATF



Above the Fold is a user experience design agency that values people – that means you, our customers, and your customers too.

From the first consultation through the final line of HTML, our team delves into your goals and your users' needs to create a custom design solution for your business. Each project is more than just a set of deliverables; it's a chance to create long-lasting relationships – for you, your customers, and us.

Appendix

Web resources in this eBook

- 2 **Business objectives vs. user experience:**
<http://uxdesign.smashingmagazine.com/2011/02/04/business-objectives-vs-user-experience/>
- 5 **Gamestorming website:** <http://www.gogamestorm.com/>
- 9 **Lean UX:**
<http://uxdesign.smashingmagazine.com/2011/03/07/lean-ux-getting-out-of-the-deliverables-business/>
- 14 **UserTesting.com:** <http://www.usertesting.com>
Loop11: <http://www.loop11.com>
- 16 **LaunchRock:** <http://launchrock.com/>
- 18 **Custom Variables setup guide:**
<https://developers.google.com/analytics/devguides/collection/gajs/gaTrackingCustomVariables>
- 20 **Seductive Interaction Design:**
<http://www.amazon.com/Seductive-Interaction-Design-Effective-Experiences/dp/0321725522/>
- 21 **Balsamiq:** <http://builds.balsamiq.com/b/mockups-web-demo/>
Mockingbird: <https://gomockingbird.com/>
OmniGraffle: <http://www.omnigroup.com/products/omnigraffle/>
- 22 **UI-Patterns.com:** <http://www.ui-patterns.com>
PatternTap: <http://www.patterntap.com>
Yahoo! UI Design Pattern Library: <http://developer.yahoo.com/ypatterns/>
- 23 **Deliver the Future, Gradually:**
<http://www.uxbooth.com/blog/deliver-the-future-gradually/>
- 24 **Bootstrap:** <http://twitter.github.com/bootstrap/components.html>
- 25 **13 Guidelines for Tabs:** <http://www.useit.com/alertbox/tabs.html>
- 26 **Basecamp:** <http://basecamp.com/>

Web resources in this eBook, continued

- 27 **Grooveshark:** <http://grooveshark.com/>
- 28 **TypeKit:** <https://typekit.com/fonts>
- 29 **Adobe Kuler for Designers:** <http://kuler.adobe.com/>
Color Scheme Designer: <http://colorschemedesigner.com/>
21 Stimulating Color Palette Tools:
<http://www.sitepoint.com/21-stimulating-color-palette-tools-for-designers/>
- 30 **Tumblr's sign up form box:** <https://www.tumblr.com/>
- 35 **Barbra Gago's slideshare, "15 Ways to Distribute Content":**
<http://www.slideshare.net/LeftBrainDGA/15-ways-to-distribute-content>
- 36 **Better Living Through Taxonomies:**
http://www.digital-web.com/articles/better_living_through_taxonomies/
- 38 **Google Website Optimizer:** <https://www.google.com/websiteoptimizer>
- 39 **Luke Wroblewski's series of interviews:**
<http://www.lukew.com/ff/entry.asp?554>
- 41 **Todoist:** <http://www.todoist.com>
- 43 **Aaron Gustafson's article on progressive enhancement:**
<http://www.alistapart.com/articles/understandingprogressiveenhancement/>
Filament Group's book on progressive enhancement:
<http://filamentgroup.com/dwpe/>
- 44 **Ethan Marcotte's article on responsive web design:**
<http://www.alistapart.com/articles/responsive-web-design/>
- 45 **Pros and cons of each UI approach:**
<http://candanny.wordpress.com/2011/08/14/declarative-vs-programmatic-in-javascript-mobile-app-frameworks/>
- 46 **CSS Sprite Generator:** <http://spritegen.website-performance.org/>
CSS sprites explained on CSS Tricks: <http://css-tricks.com/css-sprites/>
- 49 **Minify:** <http://code.google.com/p/minify/>