

The UXPin logo consists of the text "UXPin" in a white, sans-serif font, enclosed within a thin white square border. It is positioned in the upper right quadrant of a large orange circle.

UXPin

Content Wireframing for Responsive Design

UXPin

Content Wireframing for Responsive Design

Copyright © 2015 by UXPin Inc.

All rights reserved. No part of this publication text may be uploaded
or posted online without the prior written permission of the publisher.

For permission requests, write to the publisher, addressed “Attention:
Permissions Request,” to hello@uxpin.com.

Index

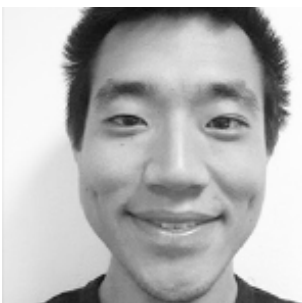
A Few Quick Words	6
What is A Wireframe?	8
Methods of wireframing	10
Wireframing in the process	12
A Mobile-First Approach to Content Wireframes	28
Creating Customizable Grids for Content Wireframes	33
Collaborating with clients and team members on your design	35



Tom Green is a professor of interactive multimedia at the Humber Institute of Technology and Advanced Learning in Toronto, Ontario. He is the author of several best-selling books in the area of Flash and Flash technologies. His latest book is [Foundation Flash CS5](#) for designers, coauthored with Tiago Dias. Tom has completed DVD videos for Lynda.com, InfiniteSkills, and Adobe Systems and is a regular contributor of tutorials to [webdesign.tutsplus.com](#), [com](#) and [Layersmagazine.com](#). He is also an active member of the Adobe Education Leaders and Adobe Community Professional groups, speaks at conferences and seminars around the world, and contributes regularly to the Adobe Developer Connection in the areas of Flash authoring and video technologies. [Follow me on Twitter](#)



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



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

[Follow me on Twitter](#)

A Few Quick Words

As I was leading my Responsive Web Design (RWD) classes between 2011 and 2012, I kept stumbling over the process of wireframing.

My students tended to focus on the wireframe as being the end game in the planning process and then spent an inordinate amount of time trying to wrestle their designs into the wireframes. What they weren't comprehending was the simple fact that RWD or designing for screens focuses on how the user will access the content.

They couldn't understand they had no control over the user pulling out a smartphone or tablet to access the content.

You can only imagine my relief when I happened to come across a [video by designer Stephen Hay](#) where he was speaking at the Beyond the Desktop conference in 2012 and there, in his talk based on his excellent book [Responsive Design Workflow](#), he presented the concept of the content wireframe. This was a huge relief for me.

I just knew there was a step before the wireframing process “got real” but I couldn’t articulate it. In this guide, I’ll explain the role of wireframing in UX design and the steps required to iterate concepts into an early content wireframe.

When you’re done reading, you’ll be able to take a more content-focused approach to UI and UX design. Content, after all, is what users care about most.

Sincerely,
Tom Green

Quick Overview of Wireframing

Let's clear up some confusion about terminology.

People often interchange terms like wireframes, sketches, and mockups to refer to rough design work. Sketches are the roughest, wireframes are bit more structured, and mockups are usually mid- to low-fidelity.

What is A Wireframe?

As described in the [Guide to Wireframing](#), wireframes are a framework of simple, low-fidelity designs that explore content structure before “formal” wireframes. They can range in fidelity from nothing more than boxes drawn on a napkin to something with actual photos and crisp typography.

You know those black-and-white boxes with Xs and text that look like the skeleton of a website? That's a wireframe. You know about carving out blocks of space for different categories of content? That's a *content* wireframe, which we'll cover later.

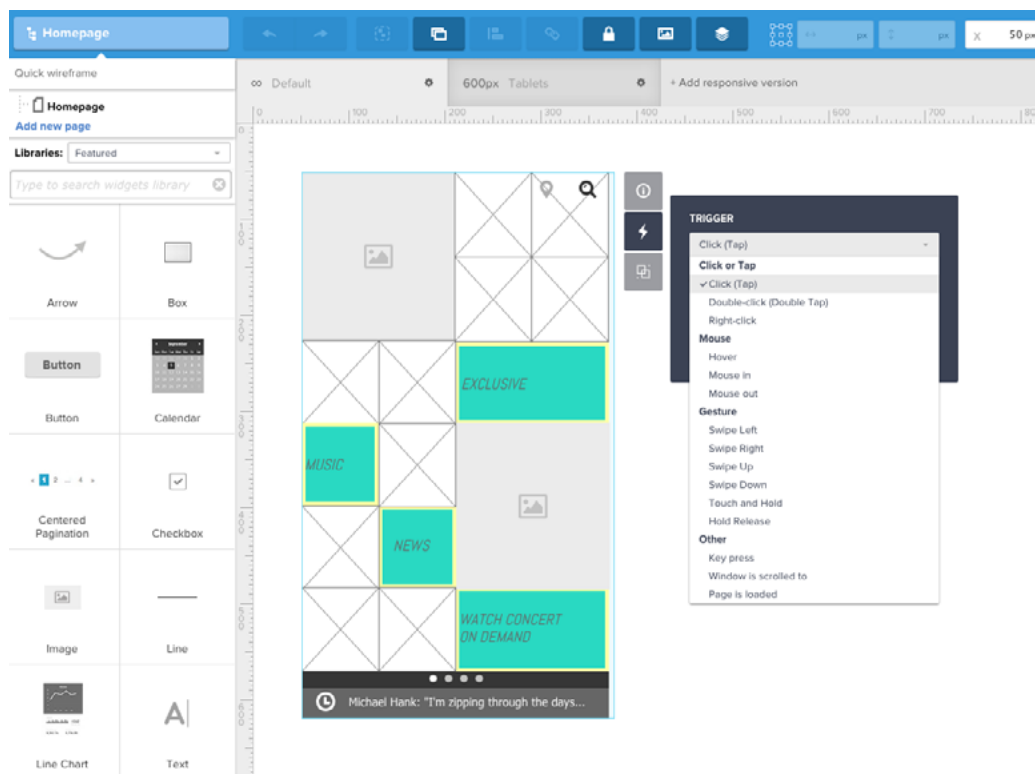


Photo credit: [UXPin](#)

In general, wireframes lean more towards the lower, more simplistic fidelity. It doesn't make much sense to worry about the paint when the concrete isn't even dry yet.

We find wireframes helps us focus on a site's structure so we know what we're going to build. Designers who rush into higher fidelity might easily overlook some aspect of the overall information architecture, leading to a Jenga-like redesign process. By planning with wireframes, designers can build a blueprint of the project without being distracted by pixel perfection too early in the process.

Methods of wireframing

Wireframes come in many forms, some of which aren't digital.

Personally, we prefer sketching on paper and then wireframing digitally: we fail quickly on paper, then improve ideas as wireframes (which can be shared and made interactive). You should choose the method that's most suitable to your product, accounting for your time and resources.

Here's a few common wireframing methods described in the free [Guide to Wireframing](#):

- **Sketching:** The most basic method is to draw your ideas on paper. Sketching your wireframe is fast and intuitive, though it lacks many online benefits. For instance, they're harder to share with other team members. Of course, sharing isn't too much of an issue if you sit near your team members (or aren't afraid to email a photo or scan).

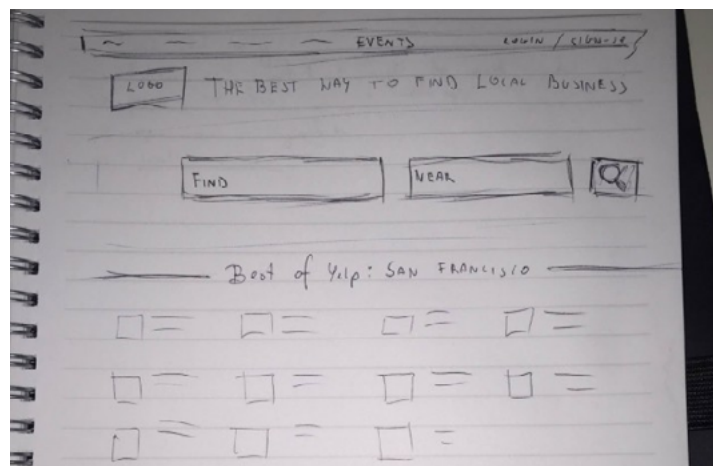


Photo credit: [User Testing & Design](#)

- **Presentation software:** Using Powerpoint or Keynote to make wireframes is easy. Just draw a few boxes on a blank slide. But while presentation software is very familiar and forces us to think about page flow (since slides are presented linearly), they lack substance. You won't be able to add complex interactions easily, so your wireframe becomes a dead-end deliverable.

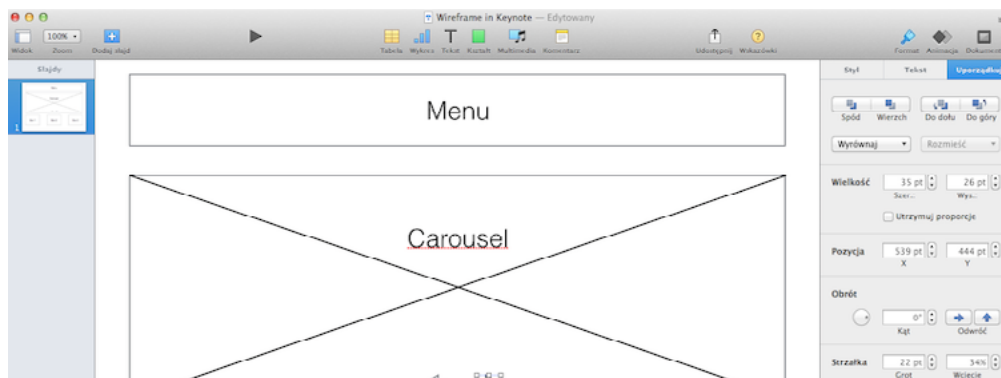


Photo credit: [Keynote](#)

- **Wireframing tools:** Specialized software designed specifically for wireframing facilitates the entire process and anticipates common concerns before they happen. The downside is they typically cost money – some more than others.

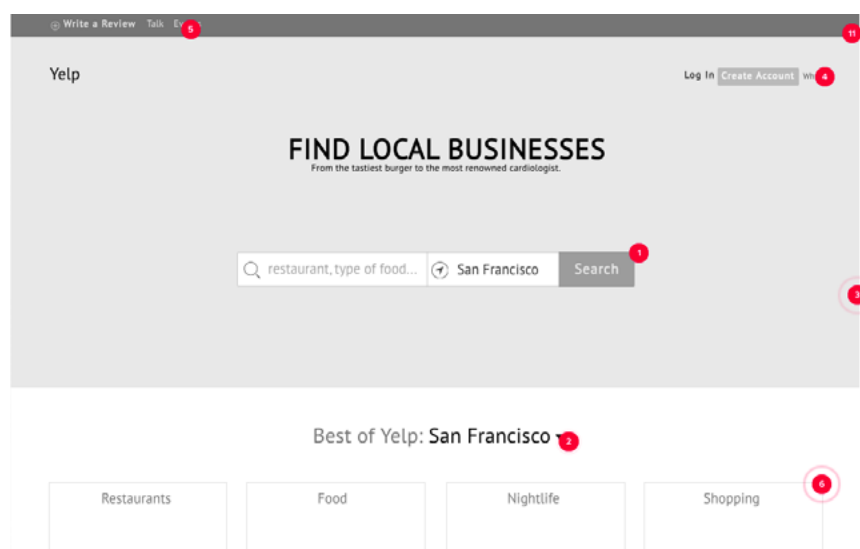


Photo credit: [User Testing & Design](#)

So which to use? Choosing the best method of building a wireframe always depends on how you intend to use it.

Wireframing in the process

There are as many variations in the design process as there are designers, so there's no one way to use a wireframe. However, if we summarize some of the processes, we can show you how to best use wireframes under different situations.

1. Traditional Process (Wireframe ► Mockup ► Prototype)

The wireframe originates from processes reminiscent of [waterfall design methods](#). In this process, fidelity and functionality increase linearly in phases.

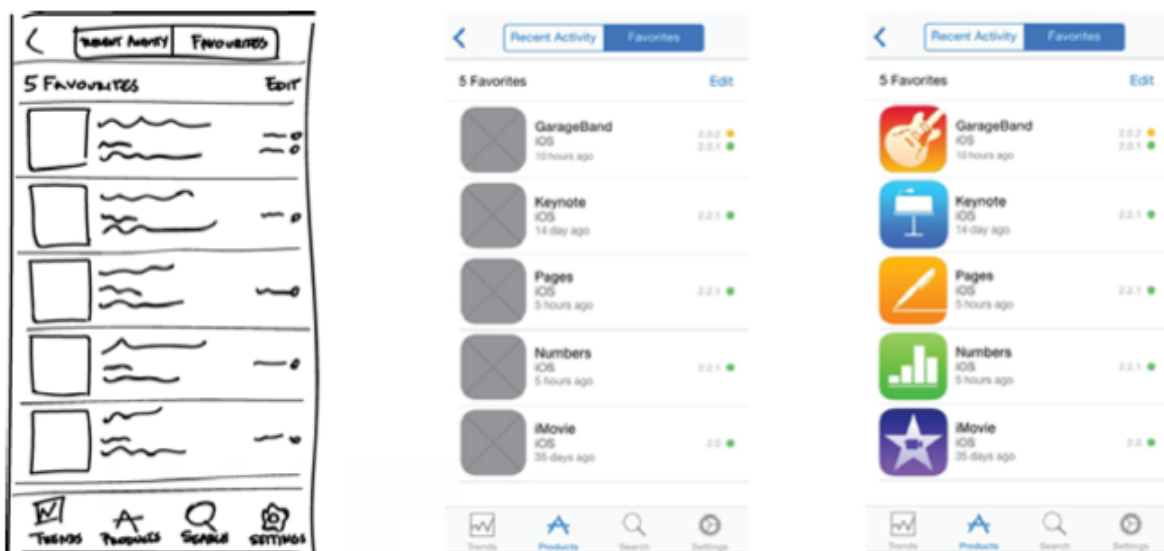


Photo credit: UXPin based on JFarny

This process relies on the specialization of each phase: the wireframing phase specializes in outlining and formalizing a big pic-

ture, the mockup phase specializes on the visual details, and the prototype phase specializes on usability and interactivity. In this case, the wireframing is the backbone. As the unifying document, it provides the foundation on which the mockup and prototype are built.

If you're stuck with the waterfall method, you can still use paper, Photoshop, or a wireframing tool. Just don't spend too much time, since change can occur when you least predict it.

2. Rapid Prototyping (Wireframe ► Prototype)

A result of the [Agile design strategy](#), rapid prototyping forgoes the traditional method of building one product to completion, and instead builds multiple, simpler [minimum viable products](#) for quick testing for smarter iteration.

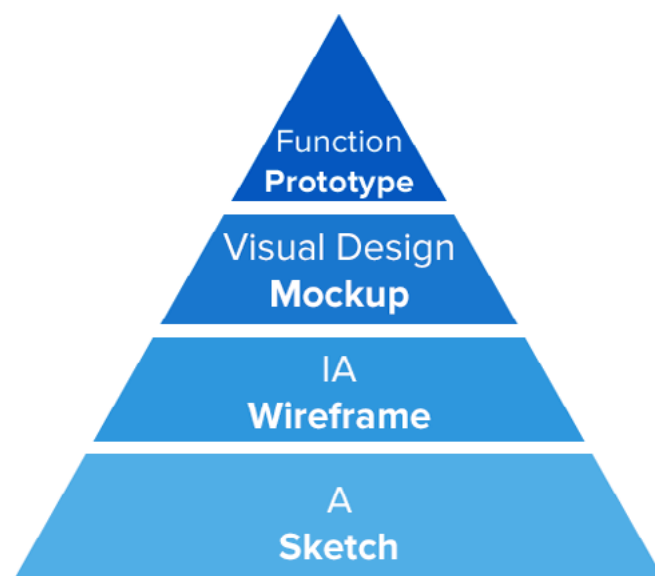


Photo credit: UXPin based on Tyler Tate via UXBooth

Despite claims to the contrary, wireframing makes rapid prototyping more effective. Due to the nature of quickly building, testing

and recycling prototypes, an interactive wireframe (lo-fi prototype) works better in this role than prototypes of higher fidelity. Interactive wireframes are easier and faster to create, but will still yield the same usability data from testing. They're so easy to build, in fact, that you can create and test several simultaneously.

3. Early Coding (Wireframes ► Coded Prototype)

Alternately, you can go straight from a wireframe into an early-stage coded prototype. This method works best for projects with unique technical requirements that need to consider coding as soon as possible, but is also a preference of designers who are confident in their coding ability and want to get the ball rolling.

Whatever the reason, wireframes still serve as a helpful visual guideline before committing to HTML prototypes. In most cases, the wireframe is the only blueprint from which to design, and so anything absent will have to be addressed during the actual development, which can be problematic (unless you are extremely proficient in code).

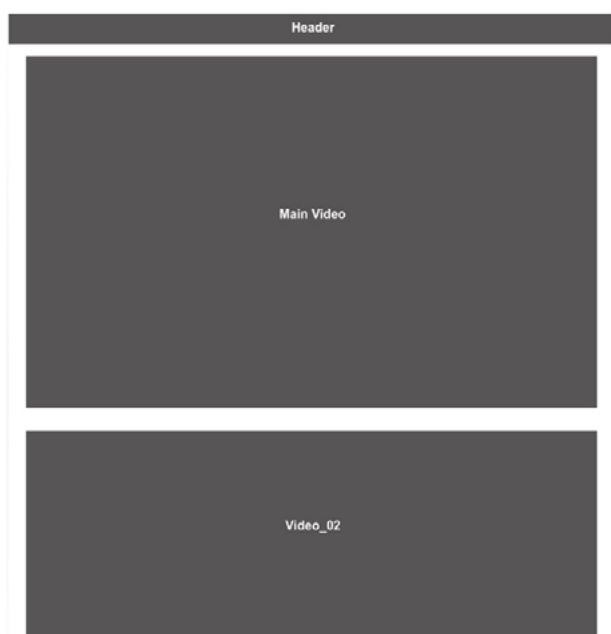
On the other hand, a thoughtfully-created outline will hold the project together and give you the chance to consider the details of layout before you're lost in the thick of coding.

If you'd like to learn more about jumping from wireframes to code, we recommend checking out Ash Maurya's piece describing his [mid-fidelity process](#). If you'd like to learn about wireframing *with* code, check out designer Matt Griffin's [article on A List Apart](#).

Introducing the Content Wireframe

The content wireframe concept is rather simple to understand because it is an offshoot of what I say in my classes, lectures and conferences: Fall in love with the user. Don't fall in love with the technology.

Content Wireframe



Low-Fi Wireframe

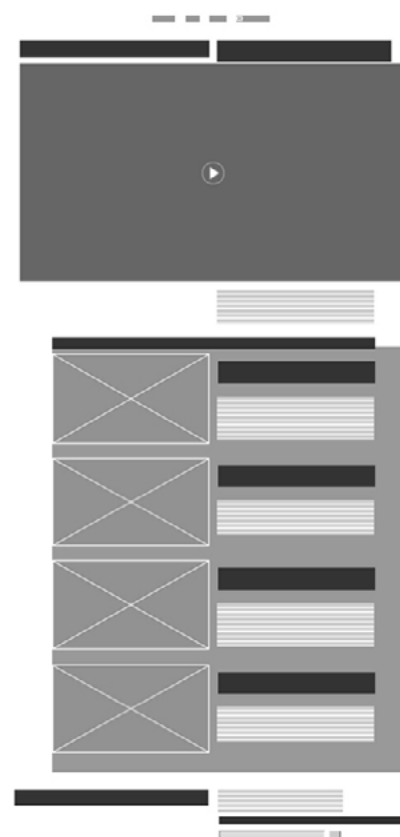


Figure 3: lo-fi wireframes introduce "context" to the process. Both created in UXPin.

Content wireframes do just that. They block out general content categories and force you, as designer Stephen Hay [so beautifully put it](#), to design from the content outwards.

And that's great for an unexpected reason.

After the iPhone debut in 2007, people realized that browser windows are unreliable places. Screens today range from 3" smartphones to 27" monitors and beyond. The landscape [gets more diverse every year](#). To make the same website work under unpredictable circumstances, designers needed a new mental model about design.

The technology was called the media query. The technique [was dubbed responsive web design](#).

Responsive web design forces us to consider design from a content-first approach. Not only do small devices have less screen space, but they also earn less attention. People use their devices on the go. They want information. They want it now. Websites built around content, with no extra frills or distractions, succeed because they deliver what people want with as little friction as possible.

This mentality called upon designers to rethink their approach. But it really isn't complicated. In fact, it involves three steps:

- Create a content inventory.
- Create a visual hierarchy of the survivors of that list.

- Construct the content wireframe based on the visual hierarchy

Do that and you have a potential deliverable for your client. What it tends to do is to pull the client's attention away from what "thing" goes where and focuses his or her attention on what is important: the flow of information.

Creating a Content Inventory

Wireframes are not what they used to be.

They're no longer about formalizing early-stage designs to soothe the panicky imaginations of stakeholders. It's about **divergent exploration**, creating a more structured sketch of concepts to bounce around with other product team members.

The key word is *structure*, or giving form to vague ideas. And the key to *that* is content.

	Navigation title	Page title	Files	Last updated	Owner	Comments	Delete?
0.0	Home	Wine Tasmania					
1.0	Wine Tasmania					No page at this level - displays 'History'	
1.1	History	History					
1.2	Touring Tasmania	Touring Tasmania					
1.3	Touring Links	Touring Links					
1.4	Wine Industry Tasmania	Wine Industry Tasmania					
1.5	Industry Statistics & Info	Industry Statistics & Info					
1.6	Investment	Investment					
1.7	Partners	Wine Industry Tasmania Partners					
2.0	The Wine Route					No page at this level - displays 'Overview'	
2.1	Wine route overview	The wine route					
2.2.0	Tamar Valley Wine Route	Tamar Valley Wine Route					
2.3.0	Southern Wine Region	Southern Wine Region					
2.4.0	East Coast Wine Region	East Coast Wine Region					
2.5.0	North West Wine Region	North West Wine Region					
3.0	Latest News	Latest News				No content on page	
4.0	Events					No page at this level - displays 'Overview'	
4.1	Overview	Events				No left-nav	
4.2	Booking	Event booking				No left-nav	
4.3	Privacy Policy	Privacy Policy				No left-nav	
4.4	Security and Refunds	Security and Refunds				No left-nav	
5.0	Members	Wine Industry Tasmania Members					
6.0	Resources	Resources	5 PDF files				
7.0	Contact Us	Contact Us				Email address & contact form	

Photo credit: [Maadmob](#)

No one browses to a site to admire its background image or the subtle gradients in its navigation bar. So designing around what people want is a smart strategy. A content wireframe outlines the material that people actually want when they visit your site. Even if the prose isn't final or the photos have yet to be shot, at least having an idea of what's to come is critical to keeping users engaged.

Luckily developing a content inventory isn't hard, though it may take some initial effort.

1. The first thing you need is to perform an audit of what you have, or plan to have. That's a list of all information on the site, often organized by URL. Organize each bit of information – not just the page it's on, but each section of a page – into topics.
2. Then decide which pieces of information should go on the same page, which may need new pages... and which should be rewritten or deleted altogether.
3. Finally, assign someone to be responsible for each page or piece of information.

There's no shortcut this eye-opening exercise that's essential to building out, well, everything. If you don't know what you're designing, then your design may not fit what it's supposed to communicate.

For more information, look at [this template](#) by Maadmob.

Determining a Visual Hierarchy

A visual hierarchy is the aesthetics that give information a sense of order. It leads users around a page, starting with an attention-grabbing element – usually an image or a heading – and suggests where they should look next. We use this to organize the content audit spreadsheet into a content wireframe.

Remember that content wireframes establish an information hierarchy and make people think more about the *concept* of information flow than the content itself. Having a content audit does more than put us in the mindset of building out wireframes – it gives us a rough idea of the space we need to allow for each piece of information.



Photo credit: [UXPin](#)

Above, a spreadsheet of information in color-coded categories helps us decide how to arrange information on a given page – and how much of each category is worth the user’s while. We can describe that in wide swaths based on category, or think more granular and decide which specific chunks of information belong on each page.

Point is, our decisions are based on known quantities, not fanciful guesswork.

We begin the process by setting priorities. What do people need to know most? Least? Will they browse until they find something interesting, or are they hunting for a particular fact? What does the business need to convey? The answers vary per project, but the process is universal:

1. Determine what the users should get from reading or using the website.
2. Use the content inventory we just created to frame “destinations.” or points in the user experience at which people get what they want.
3. Use design to solve the problem of how to get people from A to B.

That third step is why the second is so important. Without knowing where people want to go, we’re steering the design process without a map or compass.

Content Wireframes Aren't Rocket Science

The key issue around wireframes, for my students and other designers, is wireframes tend to take on unwarranted importance.

It is almost like the end of a renovation show on TV where everyone “oohs” and “ahhs” when the client gets to walk through the project.

Not for me. I prefer the scene where the contractor, during a kitchen renovation, pulls out a piece of paper and a carpenter's pencil and sits down with the client at the kitchen table. Then he or she draws a simple outline of the room and says, “How about we pull out this wall, put an island here and move the counter over there?”

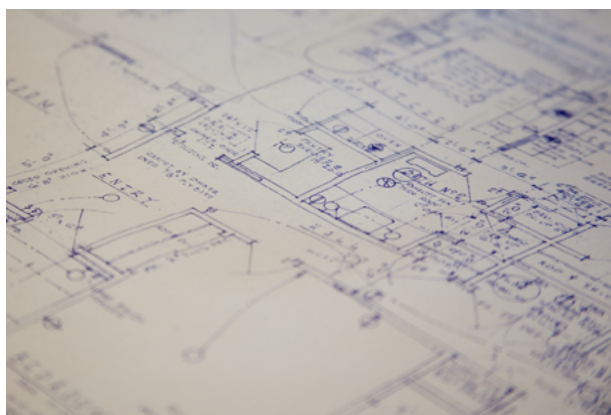


Photo credit: [Will Scullin](#). Creative Commons.

The carpenter is laying out his “intent” around his approach to the project and the client is focusing on the important parts of the project rather than focusing on tiles, fixtures, paint and lighting.

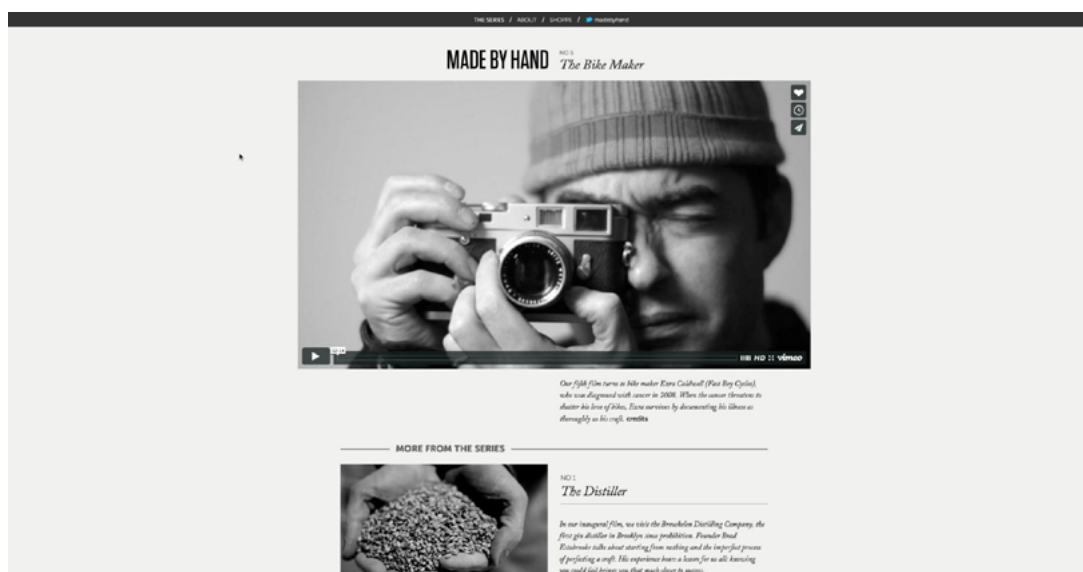
In his fantastic book, *Responsive Design Workflow* (worth buying, if you’re interested), Hay calls these things Content Reference Wireframes because, as he so eloquently puts it, they “describe how the wireframes deal with content: they simply *refer* to it as opposed to *depicting* it.”

Content wireframes, like our contractor’s sketch, depict placement – a wall, a counter, a header, a footer – and establish an information hierarchy and flow. Due to the fact these things are iterative, starting the conversation with a simple “How about...” rather than “Here’s how...” moves Content Reference Wireframes from the category of “Rocket Science” to where they originally started as “Old School Boxes and Arrows”.

Once that is established, we can then move into the LoFi and HiFi wireframe development process.

Case Study: Content Inventory & Visual Hierarchy

Let's see how how a Content Wireframe can focus the conversation. We'll start by deconstructing a simple page into its content inventory and visual hierarchy.



The Made By Hand home page has a distinct Information hierarchy.

I visit [This is Made By Hand](#) on a regular basis. As the name suggests, the site celebrates artisans who actually make things by hand. The home page features a series of videos when you first visit, it appears to be rather stark with very few flourishes. This makes sense because the films, not anything else, are the important content.

The content inventory for the homepage is, understandably, rather basic consisting of:

1. Header/Nav
2. Five videos (For the purposes of this article they will be labeled Main and Videos 1-4)
3. Footer

Still, we can see there is a definite visual hierarchy at play here. The list is a bit vague but once you understand the plan is to create a series of videos highlighting a particular craftsman, the priority of each piece of content establishes itself as:

1. The Main video
2. Videos 1-4
3. The Footer
4. The Header/Nav

It may strike you as a bit odd to see the Header/Nav content at the bottom of the stack. Actually, the whole purpose of the page is to get users to watch the videos and, if they like them, to subscribe to regular notifications about new content. The Header/Nav content is simply there to provide easy navigation around the site.

In many respects the old adage – “Content is king,” which is [generally attributed to Bill Gates](#) – drives this site. The important content is a

collection of videos and nothing else. This raises an important point: A content wireframe establishes content precedence, not page flow.

With that list in hand you can move on to actually creating the content wireframes.

Creating Content Wireframes

Although Stephen Hay is a big proponent of creating these content wireframes using a code or text editor such as Coda or Adobe's Brackets, for those of us who are, shall we say- "code challenged"-, a visual editor such as [UXPin](#) is a good choice.

In a pinch, you could also use non-design tools like Keynote or Powerpoint, but a specialized wireframing or prototyping tool helps speed up the process quite a bit.

Here's why:

- UXPin's Responsive feature let's you start with a "Mobile-First" approach and, using [preset or customizable breakpoints](#), you can quickly build out to a desktop version.
- Customizable grids allow for consistency of design.
- [Notes and comments](#) allow you to communicate your intent to the team and the client, making collaborative design much simpler.

Now let's actually create a Content Wireframe using our example from [ThisIsMadeByHand](#).

A Mobile-First Approach to Content Wireframes

For this quick tutorial, we'll take a mobile-first design approach to help us focus on just the most important content. To get an idea of what we're creating, you can take a quick glimpse at the [finished content wireframe](#).

If you haven't already, go ahead and create a free [UXPin](#) account. Now let's get started.

1. When you first create the project you are taken to a default surface.

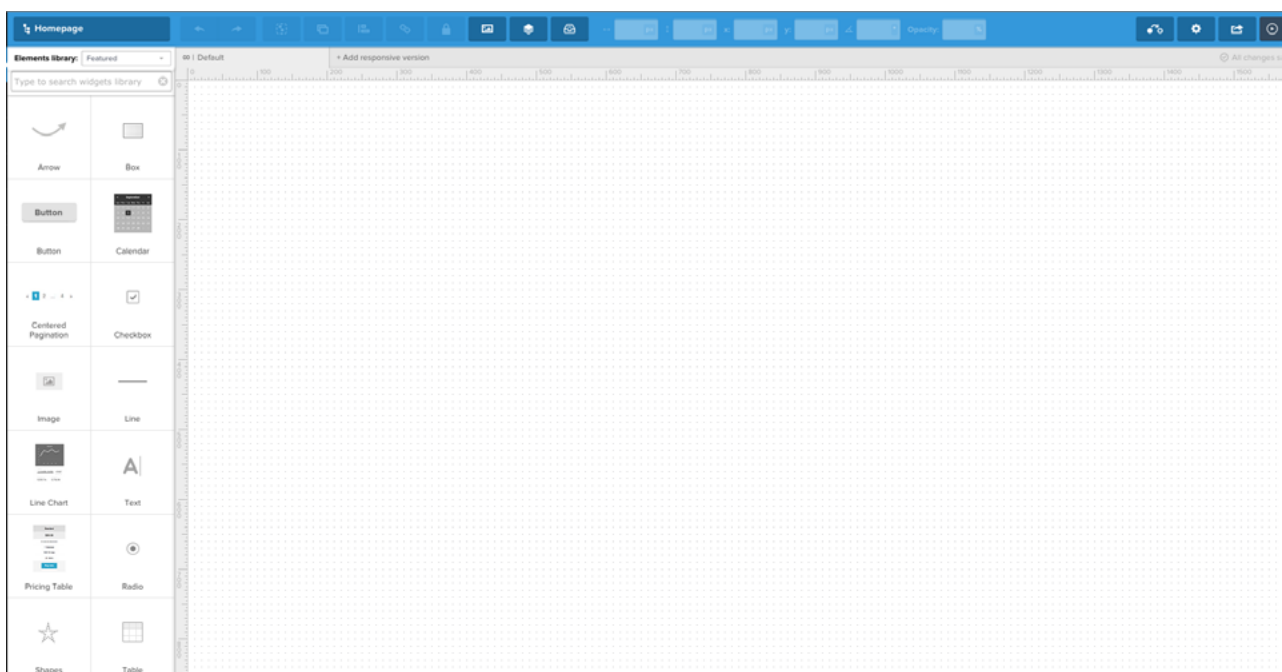


Figure 1: The UXPin interface

2. **UXPin** has an *Add responsive version* menu item. Click it and a pop down containing a number of presets will appear. I started with the *iPhone 320 px* option.

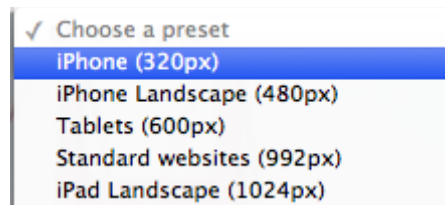


Figure 2: Adding a responsive version.

3. Next, I deleted the Default canvas. This left me with just a mobile canvas which was what I needed.

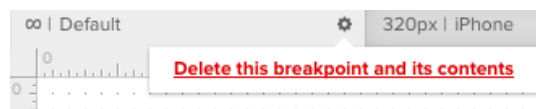


Figure 3: Breakpoints can be deleted.

4. I then added a Box and a Text element to the design surface. When you add an element, the Properties for the selected Element open in a dialog box.

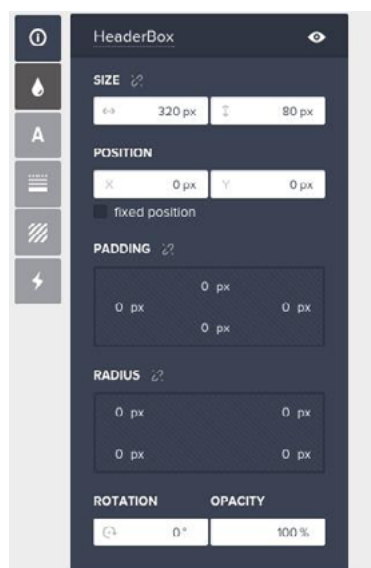


Figure 4: Setting the properties for the Box.

5. For the box, I filled it with a dark gray and set its width to 320 pixels. The text was set in a bold Arial , centered in the text area and colored white. For a content wireframe that’s all you need because you only want to Reference Content.

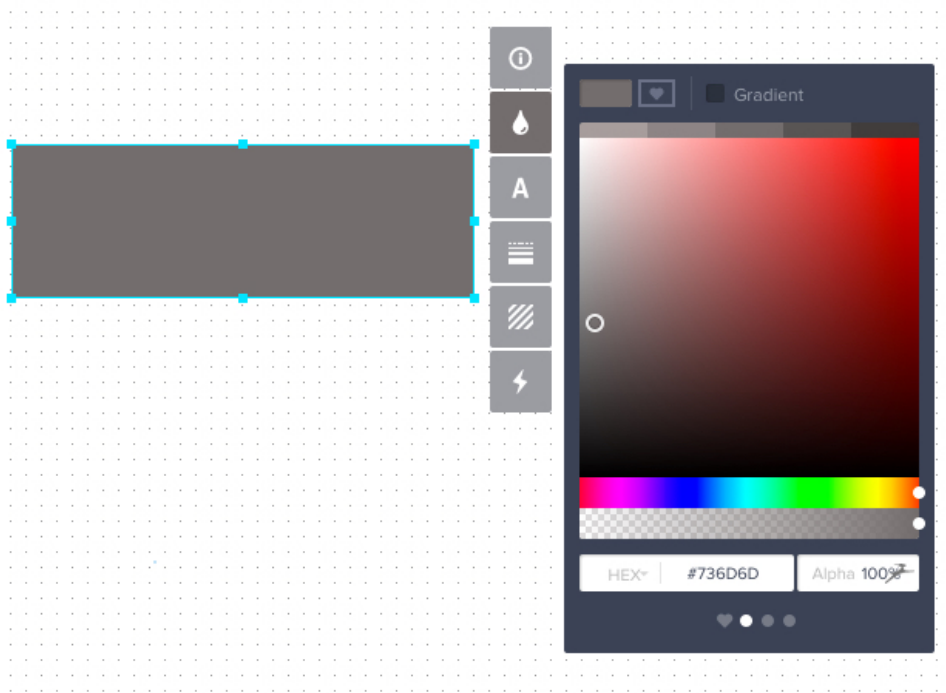


Figure 5: Setting an element’s color properties.

Everything else to be added to the page was simply a variation of this theme.

6. Instead of “reinventing the wheel”, with the Shift key held down, I selected both the box and the text and then, using the *Click-Option-Shift- Drag* technique (PC users can substitute the *Alt* key for the Mac’s *Option* key), dragged a copy of the header into place. I changed the text to “Main Video” and , to stress this element was at the head of the hierarchy, changed the box height to 320 pixels. I repeated this step for the next four boxes but changed their height to 255 pixels.



Figure 6: Setting the size for the secondary content.

7. The Footer was also important. To make it noticeable and to accommodate the eventual inclusion of a Subscribe feature, I changed the Box height to 345 pixels.

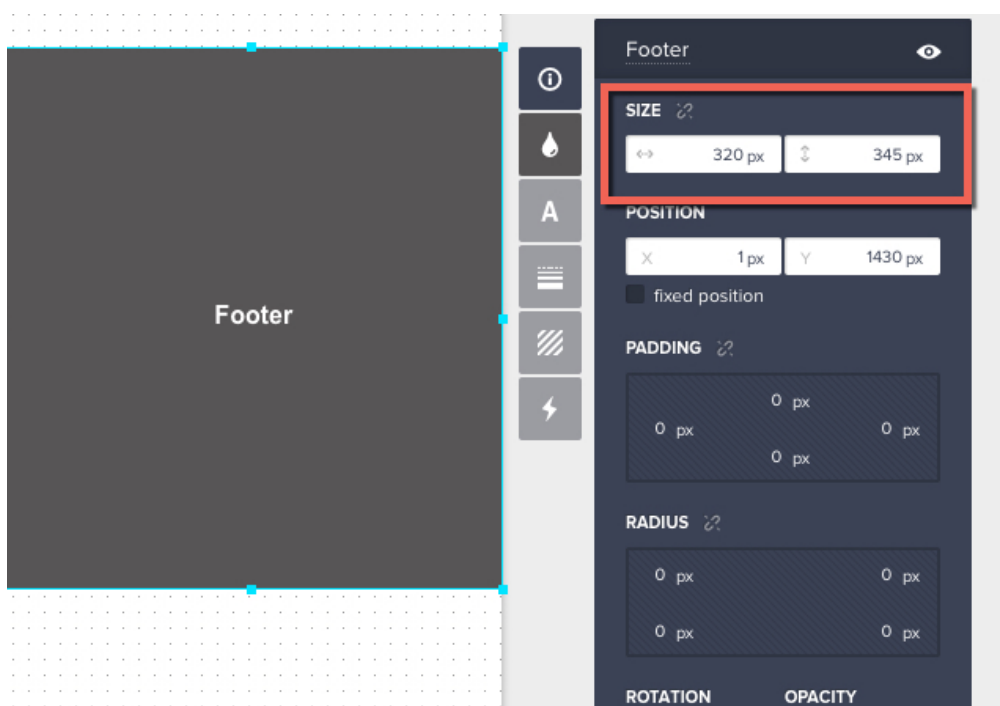


Figure 7: The footer is larger because it is important.

8. Adding the iPad version was a breeze.

I clicked the *Add responsive version* link, selected the *iPad 768 pixel* version and, in the dropdown, selected the *iPhone (320)* option to copy all of the boxes from the iPhone page to the iPad version. From there it was strictly a matter of rearranging and resizing the elements to reflect the priorities set earlier.

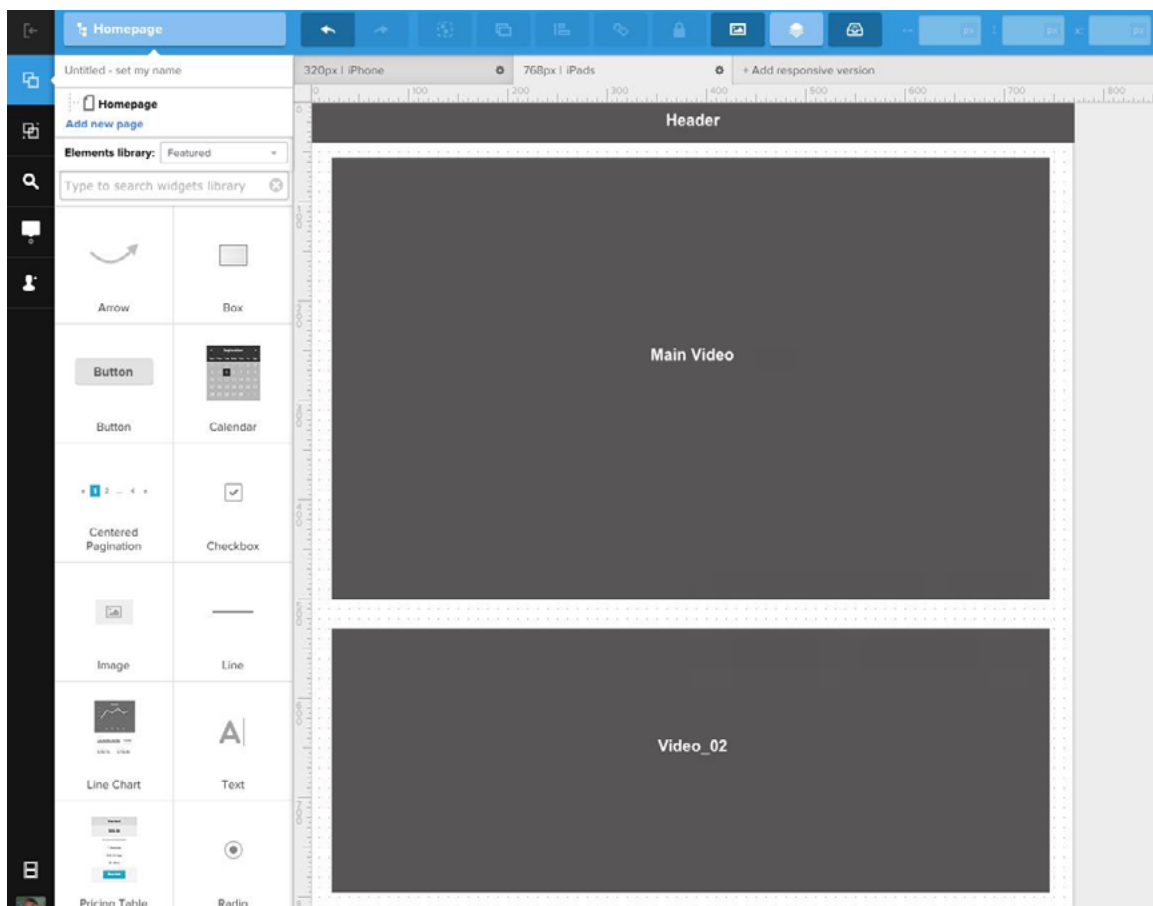


Figure 8: Creating the iPad content wireframe

Creating Customizable Grids for Content Wireframes

Now that I've created the rough layout, I'll add some grids to maintain alignment. I am not a huge fan of using "Mark One Eyeball" to place stuff on the canvas.

Though [UXPin](#) gives me a number of options – None, Dots and Grid – for the background, I much prefer a Grid with columns to help with content placement.

For the iPad layout I chose to work with a 4-column grid. To access this, you click on the Settings button (which looks like a gear) on the Toolbar. This opens a dialog box that allows me to set numerous grid options and, as shown below, to turn the grid on and off and to place it above or under the content.

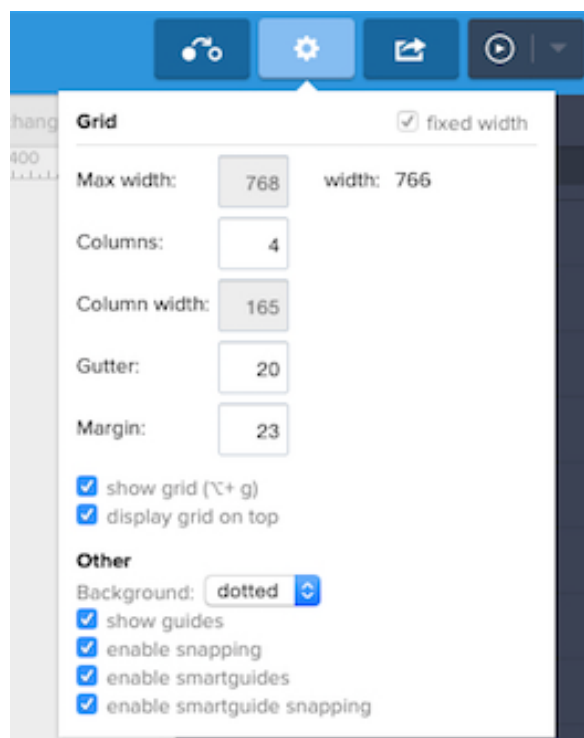


Figure 9: Adding a grid to the design.

The really useful aspect of the grids is there can be a separate and different grid for each breakpoint. For example, the wide page breakpoint uses an 8-column grid and the 1400 breakpoint uses a 12-column grid.

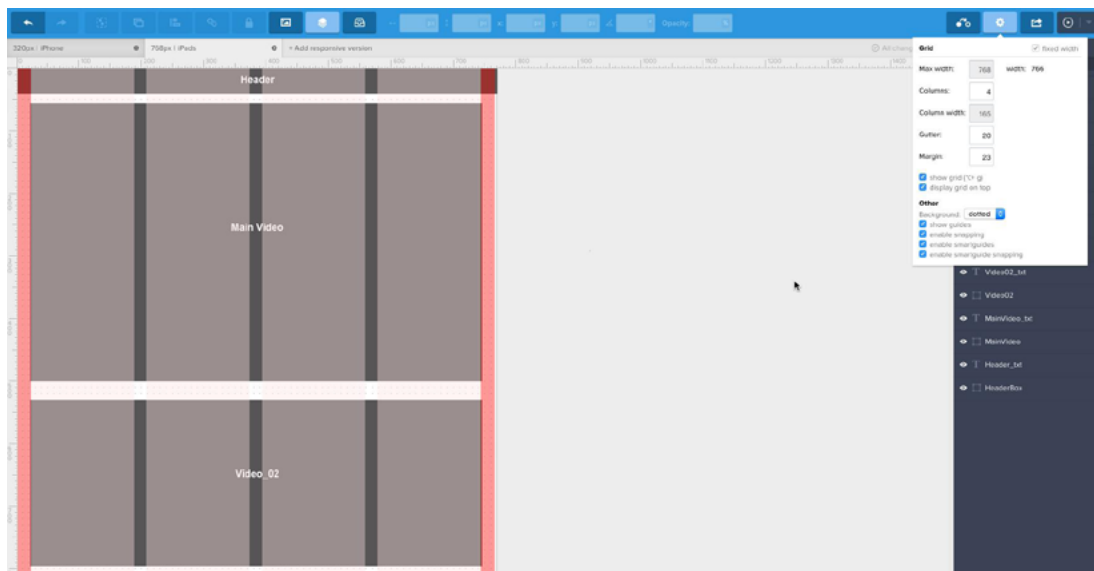


Figure 10: Use the Settings to apply a grid, create a customized grid and to turn it on and off.

On the surface this may look to be rather basic... and it is. The key takeaway here is the boxes just “reference” the content. We aren’t concerned with anything but where the content may be placed. Thus the real use of a grid here, is to ensure the boxes are roughly where they are supposed to be on the page.

The key word is “roughly”. The boxes will evolve from their rough form to interactive prototypes as decisions regarding content, colours, typography and so on are made and refined. Still, the boxes used in the Content Wireframe through their position and size will start the process of determining the project’s information hierarchy.

Collaborating with clients and team members on your design

UXPin contains an Add Note button on the left toolbar.

Click anywhere in the page and you can add a short description to the resulting Comment. Once you click the *Close* button, a pushpin icon indicates the location of a note. These notes are both visible in both the design and preview modes of UXPin. To open a note just click the push pin.

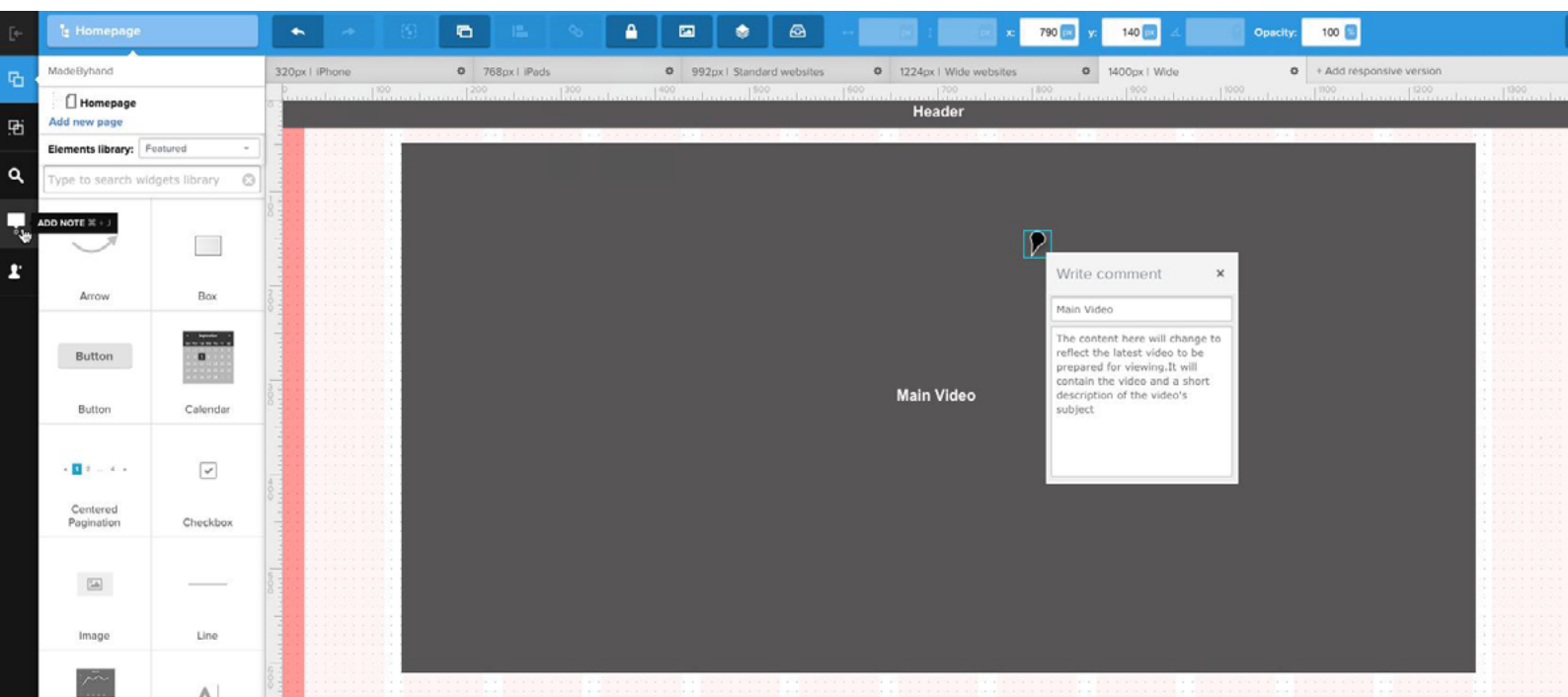


Figure 11: Adding a Note to an element on the page.

When the page is being viewed in the browser or [Preview mode](#), the team and even the client can also add comments or notes.

When a page is open in a browser, clicking the Comment mode button changes the cursor to a + sign. Click it, add the note and drag to its

location. When a Comment is attached to a page, it will be identified with a small numbered red dot. Click it and the comment will open. From there the reader can choose to reply.

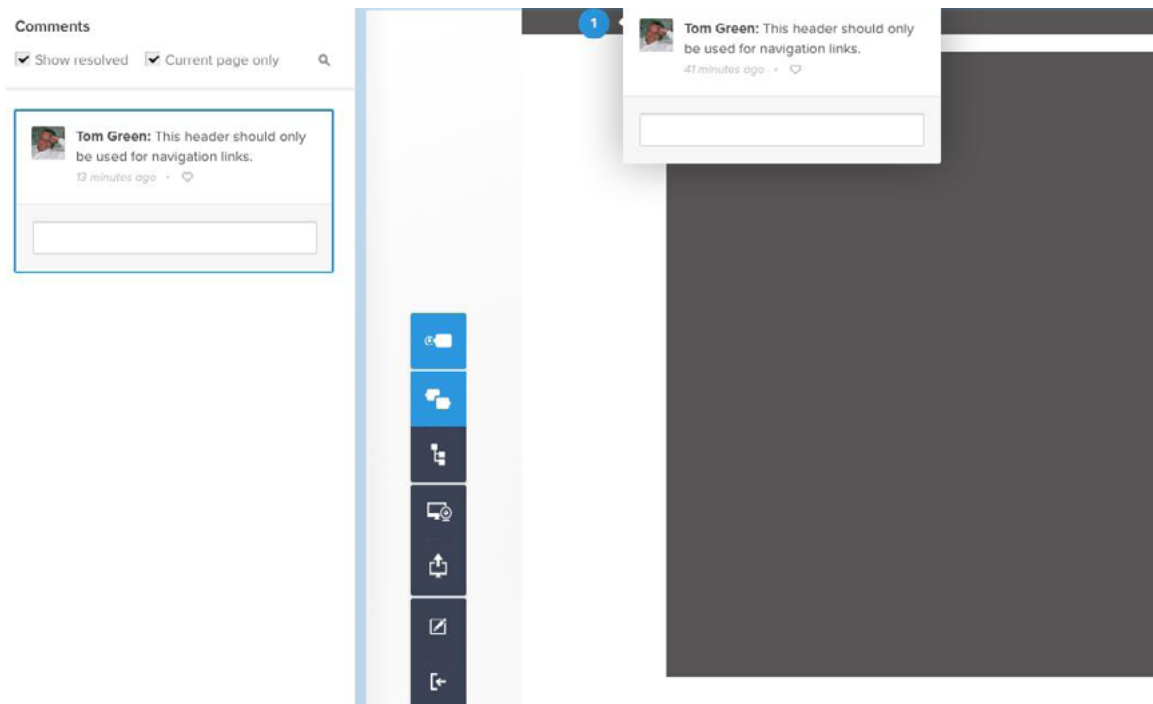


Figure 12: Notes can be added to a Content Wireframe when the link is viewed in a browser.

Conclusion

As you have seen in this piece, there is no great excitement around the creation of content wireframes.

They are in a place in the workflow between the Content Inventory and the LoFi wireframes. This is because their purpose is to simply establish an information hierarchy and, if the client is involved, to get him or her to think more about the “concept” of content and information flow than the actual content itself.

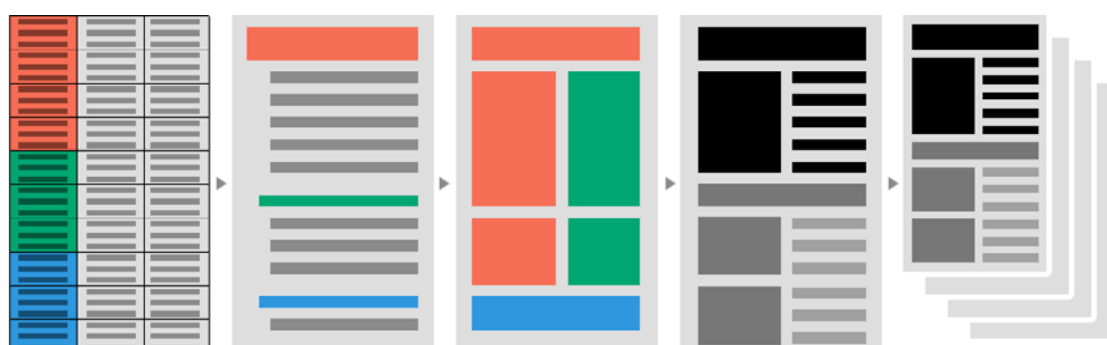


Photo credit: UXPin

The process above (from left to right):

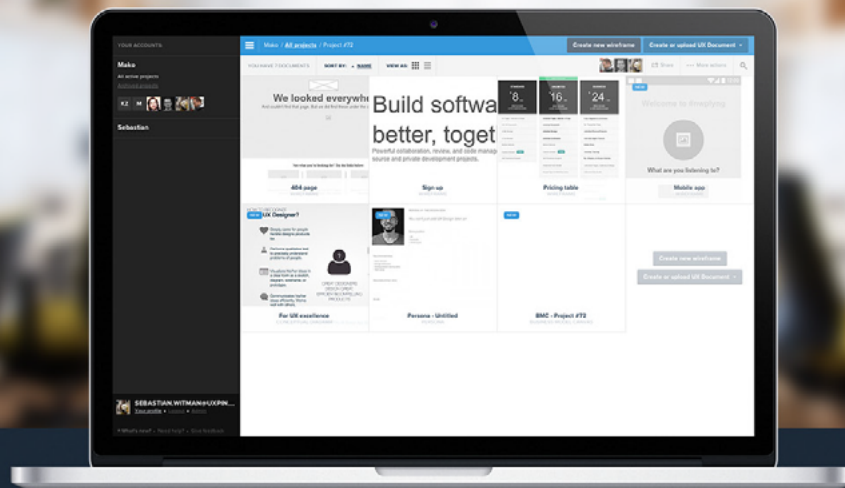
1. **Content Inventory:** Spreadsheet
2. **Visual Hierarchy:** Design that conveys order

3. **Content Wireframe:** Applying content to design
4. **Low-fi Wireframe:** Laying out the structure
5. **Interactive Wireframe:** Develop page flows by making parts of wireframe clickable

These things are a part of an iterative process that can and should change.

As I pointed out right at the top this entire process is more of a “How about...” conversation than a “Here’s how...” conversation. Once everyone is in agreement the next iteration of the project will involve pouring actual LoFi content into the boxes that make up the content wireframes.

[Start wireframing & prototyping in UXPin \(free trial\)](#)



- ✓ Complete prototyping framework for web, mobile, and wearables
- ✓ Collaboration and feedback for any team size
 - ✓ Lo-fi to hi-fi design in a single tool
- ✓ Integration with Photoshop and Sketch

UXPin

www.uxpin.com