

Agile Web UI Readings

Contents

1. Good UI and Bad UI
2. Characteristics of Successful UI
3. UI Spec Doc
4. UX Flow Chart
5. Adaptive_vs_Responsive_Design
6. Guide for UI Design Process

The Difference Between Good UI and Bad UI

So, what exactly is user interface, anyway?

When I visit your page (just like if I visit your house for dinner), I should feel welcomed. Good user interface makes me feel at home. Everything makes sense, and the cat isn't hopping on the dinner table. Bad user interface is cat hair in your soup—it leaves a bad taste in your mouth and you're ready to exit stage left quickly.

Put simply, user interface (UI) is how the site interacts with the user. It's less about a site's beauty and more about its usefulness in delivering the product to the user. If you're a blogger, the product is your point-of-view; if you're an online candle store, the product is a candle. The purpose of UI is to get the user to the product as efficiently and quickly as possible.

Consider the 4 E's of Good UI design. UI should always be the following:

- -Easy to use
- -Easy to understand
- -Error-free
- -Effective for the end-goal (or product)

Conversely, bad UI design is sluggish, complicated, and generic. And, surprisingly, there's little grey area. Either you have a thoughtful UI design, or you have a generic blob that doesn't meet the requirements of your users.

What Does Good UI do?

An effective UI design is intuitive, both in how it interacts with the user and how the user interacts with the site. Good UI design has threads of familiarity. Even if I'm visiting your page or app for the first time, I should understand how it works—and quickly.

A good user interface holds my hand and takes me where I should go. Customers like hand-holding. And, that's not an insult to your customers. Of course, what works for one site most definitely will not work for another, and that's why testing is so important. More on that a little later.

What Does Bad UI do?

Bad UI drops you off in the middle of the desert and expects you to make it on your own to the rain forest. It does not lead you where you need to go. Too often, this is because websites try to cater to an impossibly huge demographic, so the core audience is marginalized in favour of appealing to the broader audience.

Here are 10 Examples of UI gone wrong (and how to avoid it on your site):

1. UI is not responsive.

He's dead, Jim. In this day and age, having a website that users must pinch to zoom on mobile web devices is embarrassing. Although zooming in isn't difficult, it indicates that you are out of touch with your users. If you notice that your website gets a fair amount of traffic from mobile devices, you should consider updating your UI to answer the call.

Consider fat fingers and failing eyesight in your design. Not all of us are blessed with nimble finger tips.

2. UI is not intuitive.

When you create your website or app, you should already have in mind your target customer. The usability of your design is determined by how easily your target customer can navigate around it. It doesn't matter if adults aged 18-49 can use an app that's targeted to ages 6-10. When your UI doesn't make sense to the user, it will be abandoned.

Strive to implement platform conventions whenever possible. Most users expect the location of the search box to be in the upper right hand corner a website. Or visited links to change colour. Users will feel at home with design elements that they expect. By all means, be creative, but don't sacrifice user experience.

3. Design is inconsistent.

The tone of your website should be fairly consistent on every page. A reader shouldn't feel as if they are on a totally different website from one page to the next. Maintain uniform navigation and a decisive tone throughout.

4. There is no target.

Who is your target user? If that is not immediately evident on the first page, you've lost. Many generic websites are offered to the masses, but very few succeed.

Remember an important principle, known as Pareto's Principle aka 80/20 rule.

In this case: 80% of your sales come from 20% of your customers.

This principle is true for whatever type of website or app you have, whether you're monetizing it or it's just for fun. It's infinitely more important to find out who those 20% are, and tailor your website just for them.

5. Social interaction is lacking.

Users need to feel validated in choosing to spend time on your site. One of the most powerful ways to fill that need is through touting your popularity. Implement social tools on your website, such as comment forms. Showcase your number of subscribers on your newsletter sign-up. Post recommendations or reviews from satisfied users, with images and links to their websites.

6. There is no direction.

This point goes back to hand-holding. Figure out what your bottom line is. **What do you want users to get from your site or app?** Is it entertainment, a good, or a service? Whatever the end-goal, you

must lead them to it. When I visit your site, I shouldn't be met with a lot of information that will possibly distract me from the goal.

This is precisely why a blog plastered with ads doesn't work nearly as well as a blog with one or two carefully placed endorsements. When you narrow the focus, you control the path.

7. The forms are too needy.

The only thing worse than a long, drawn-out form is one with unclear error messages. As emphasized in the 4 E's of Good UI, you want the user interface to be error-free, but form fields will challenge everything you know about usability.

It's important to allow your users to be humans, and humans make errors. Your form fields should accommodate the user by offering to correct misspelled words or inline error validation. Another good UI design is pre-filled forms, populated from information filled out on other pages within your site.

Also, **consider requiring less information from users upfront**. It's already a hassle to register for one more site, so make your registration be as painless as possible, which leads me to the next step.

8. Lack of social login.

Social login is so important on the web, right now. The seemingly disparate parts are coming together. If you've configured your website as its own isolated community, you've done a disservice to yourself and your users. **Everything is connected**.

Instead of asking users to sign up cold, integrate the social login, where you allow them to connect to your site through social media, such as:

- Facebook
- Twitter
- Google+
- LinkedIn
- Etc,

The benefits are that it's less work for the user, and it allows you to have more information about that user had they registered to your website cold, like photos.

Of course, be sure to allow users the option to sign up the traditional way. Some users do not use social media, or may prefer to have a separate interaction with your site

9. It is slow.

Slow is evil. We're talking seconds, but it can make a huge difference in whether the user sticks around to wait for the page to finish loading. **A slow website can hurt you**.

To counteract this, implement best practices for website speed. These best practices include:

- minimizing your HTTP requests by unifying elements and using CSS Sprites
- combining style sheets
- enabling compression to reduce bandwidth.
- Use .jpg and .png and compress those images as much as possible without losing quality. For .png specifically, this is a nice tool I use to reduce the file size: tinypng.com

10. There are readability issues.

Good user interface also tackles content. It's nice to have engaging content, but if it's presented in the wrong way, users will not read it, and it may come back to haunt you.

The most important thing you need to know about usability is that **most people don't read, they scan**; and if the scan seems interesting, they'll skim; and if the content is bursting with personality (ahem), then they'll read. But, the first point is creating scannable text. Do this by using subheadings, bullets and highlighted text.

Next comes the appropriate use of font and size. **Text size should always be bigger than what you think**, generally hovering around 16 pixels, but compensate for your font.

Last, but not least, is content. Make your content relatable for your target audience and easy to read. Remember, user interface is how the site interacts with the user.

So, now that you've had a rundown of the main points of good and bad UI, **here are 5 questions you should ask yourself about your current design:**

1. Who is your target audience?
2. Why are users coming to your site? What is the user hoping to solve by visiting?
3. Can Grandma use it? Is it painfully simple? If not, start over.
4. Are you effectively leading users to your target?
5. What is the clear solution?

After you answer these questions, **it's time to start testing out your site**. Testing is a necessary component to efficient UI. Fortunately, testing can be done easily and inexpensively with automated testing. The key rules to follow when testing, as outlined by Jakob Nielsen, are: Get representative users; ask them to perform representative tasks; shut up and let them do the talking.

Good UI is actually not complicated, at all. It's about simplifying the focus. What are your favorite examples of good UI?

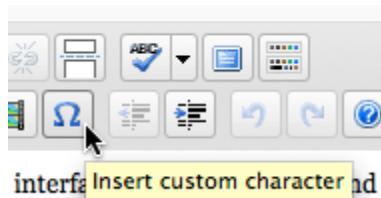
End of document

Eight user interface design principles

There is a lot of information out there about various interface design techniques and patterns you can use when crafting your user interfaces and websites, solutions to common problems and general usability recommendations. Following guidelines from experts will likely lead you towards creating a good user interface — but what exactly is a good interface? What are the characteristics of an effective user interface?

Clarity

Clarity is the most important element of user interface design. Indeed, the whole purpose of user interface design is to enable people to interact with your system by communicating meaning and function. If people can't figure out how your application works or where to go on your website they'll get confused and frustrated.



What does that do? Hover over buttons in WordPress and a tooltip will pop up explaining their functions.

Information density

Clarity in a user interface is great, however, you should be careful not to fall into the trap of over-clarifying. It is easy to add definitions and explanations, but every time you do that you add mass. Your interface grows. Add too many explanations and your users will have to spend too much time reading through them.

Keep things clear but also keep things concise. When you can explain a feature in one sentence instead of three, do it. When you can label an item with one word instead of two, do it. Save the valuable time of your users by keeping things concise. Keeping things clear and concise at the same time isn't easy and takes time and effort to achieve, but the rewards are great.



The volume controls in OS X use little icons to show each side of the scale from low to high.

Intuitiveness

Many designers strive to make their interfaces ‘intuitive’. But what does intuitive really mean? It means something that can be naturally and instinctively understood and comprehended. But how can you make something intuitive? You do it by making it ‘familiar’.

Familiar is just that: something which appears like something else you've encountered before. When you're familiar with something, you know how it behaves — you know what to expect. Identify things that are familiar to your users and integrate them into your user interface.



GoPlan's tabbed interface. Tabs are familiar because they mimic tabs on folders. You figure out that clicking on a tab will navigate you to that section and that the rest of the tabs will remain there for further navigation.



Responsive means a couple of things. First of all, responsive means fast. The interface, if not the software behind it, should work fast. Waiting for things to load and using laggy and slow interfaces is frustrating. Seeing things load quickly, or at the very least, an interface that loads quickly (even if the content is yet to catch up) improves the user experience.

Responsive also means the interface provides some form of feedback. The interface should talk back to the user to inform them about what's happening. Have you pressed that button successfully? How would you know? The button should display a 'pressed' state to give that feedback. Perhaps the button text could change to "Loading..." and it's state disabled. Is the software stuck or is the content loading? Play a spinning wheel or show a progress bar to keep the user in the loop.

Loading dmitry@usabilitypost.com...



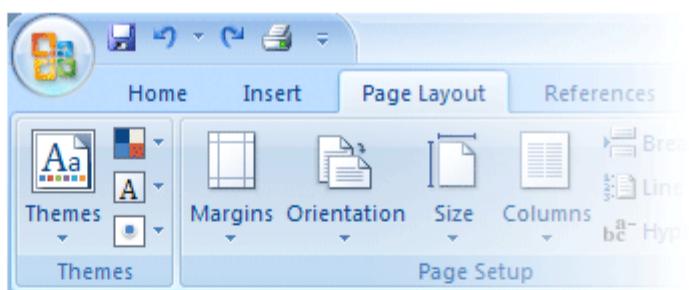
Instead of gradually loading the page, Gmail shows a progress bar when you first go to your inbox. This allows for the whole page to be shown instantly once everything is ready.



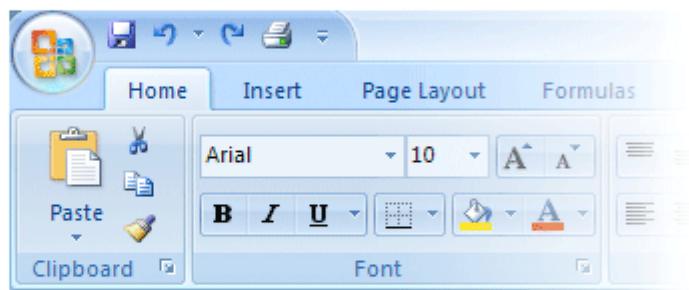
Now, I've talked before about the importance of context and how it should guide your design decisions. I think that adapting to any given context is smart, however, there is still a level of consistency that an interface should maintain throughout.

Consistent interfaces allow users to develop usage patterns — they'll learn what the different buttons, tabs, icons and other interface elements look like and will recognize them and realize what they do in different contexts. They'll also learn how certain things work, and will be able to work out how to operate new features quicker, extrapolating from those previous experiences.

Microsoft Word



Microsoft Excel



Microsoft Powerpoint

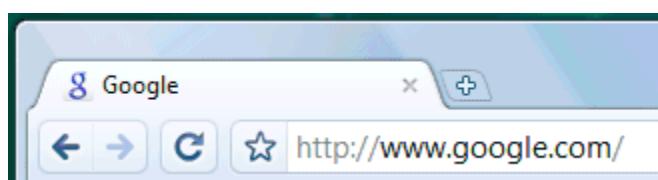


The Microsoft Office user interface is consistent for a reason.

□□□ttr□ti□□□

This one may be a little controversial but I believe a good interface should be attractive. Attractive in a sense that it makes the use of that interface enjoyable. Yes, you can make your UI simple, easy to use, efficient and responsive, and it will do its job well — but if you can go that extra step further and make it attractive, then you will make the experience of using that interface truly satisfying. When your software is pleasant to use, your customers or staff will not simply be using it — they'll look forward to using it.

There are of course many different types of software and websites, all produced for different markets and audiences. What looks 'good' for any one particular audience will vary. This means that you should fashion the look and feel of your interface for your audience. Also, aesthetics should be used in moderation and to reinforce function. Adding a level of polish to the interface is different to loading it with superfluous eye-candy.



Google are known for their minimalist interfaces that focus on function over form, yet they clearly spent time polishing off the Chrome user interface elements like buttons and icons to make them look just right as evident by the subtle gradients and pixel thin highlights.



A user interface is the vehicle that takes you places. Those places are the different functions of the software application or website. A good interface should allow you to perform those functions faster and with less effort. Now, ‘efficient’ sounds like a fairly vague attribute — if you combine all of the other things on this list, surely the interface will end up being efficient? Almost, but not quite.

What you really need to do to make an interface efficient is to figure out what exactly the user is trying to achieve, and then let them do exactly that without any fuss. You have to identify how your application should ‘work’ — what functions does it need to have, what are the goals you’re trying to achieve? Implement an interface that lets people easily accomplish what they want instead of simply implementing access to a list of features.



Apple has identified three key things people want to do with photos on their iPhone, and provides buttons to accomplish each of them in the photo controls.



Nobody is perfect, and people are bound to make mistakes when using your software or website. How well you can handle those mistakes will be an important indicator of your software’s quality. Don’t punish the user — build a forgiving interface to remedy issues that come up.

A forgiving interface is one that can save your users from costly mistakes. For example, if someone deletes an important piece of information, can they easily retrieve it or undo this action? When someone navigates to a broken or non-existent page on your website, what do they see? Are they greeted with a cryptic error or do they get a helpful list of alternative destinations?

The conversation has been moved to the Trash. [Learn more](#) [Undo](#)

Trashed the wrong email by mistake? Gmail lets you quickly undo your last action.

How to create a user interface specifications document (UI Spec)

After you have reviewed and examined the *Overview Criteria* and before you create a *UI Flow Chart* the next step is the UI Spec. It is an essential step for the product lead (usually the same person as the CEO/project lead in a small startup), as it forces them to visualize a rough draft of the app *and put it in writing* for the team to kick around.

A Quick Note On 'How' To Keep the UI Spec Simple

In general, everyone says *keep it simple*, but few people know what that really means. In the context of building a consumer-facing app, know that even you don't know the answer yet. You have to test things rigorously before you even *begin* to know what 'simple' is for your app.

Your job here for the UI Spec is to dump out a first draft of features, view-by-view (or page-by-page). Don't stray too far from the Overview Spec. Ask yourself for every feature: **"is this absolutely essential in order for us to test our core value hypothesis?"**

If you successfully get through a thorough first draft of your UI Spec asking yourself this question, then you'll be in good shape. Don't second guess yourself. Build this product that you come up with and test/iterate later.

Suggested Format for the UI Spec

Think through every possible view/page of the app, including tool-tips, lightboxes, etc... and just dump out your ideas in bullet-point list within a Google Doc (so your team can easily edit/modify/comment).

The following pages show an Example...

Pages:

- Home Page (/)
 - Header (logo, menu -> about, contact, [profile <- if logged in], login/logout)
 - Simple hero shot and marketing message
 - Super obvious call to action above the fold to enter your email address to get started
 - Simple thank you message that appears in place of the field after a valid email is entered.
 - “About us” content below the fold
 - Intro content (title/couple sentences)
 - 4 sections (either two by two - or 4 columns) - Meetings, Agenda Items, Action Items, Users - each with their own visual icon - each allowing a paragraph of text
 - Social Media Widgets (FB facepile & Latest Tweets)
 - Footer
- Contact Us (/contact)
 - Address / Email / Social Media / Contact Form
- Login (/login)
 - Simple, centered login form - email & password
- Finalize Signup (/finalize_signup)
 - Simple, centered form like the login page, but for First, Last, Password, Password Confirmation

- Dashboard (/dashboard)
 - 4 columns filling the height and width of the content section of the screen (i.e. between the header and footer). Then each column is scrollable in itself if/when there are more rows than there is height.
 - Agenda Items
 - Users
 - Meetings
 - Action Items
 - Each column will have a “new” button on the top or bottom (whatever looks best).
 - Clicking “new” (or doing many other actions with the app) will open a lightbox that has the necessary fields/instructions to execute the action.
 - Each row in a column will be selectable, yielding some kind of UI to take an action (e.g. “Schedule Meeting” after multiple users - and/or multiple agenda items - are selected)
 - Agenda Item Column
 - Each row
 - Title / Started By (avatar/name) / Applicable Users (facepile - would be cool to do a horizontal scroll here if more than 6 pics or something)
 - When expanded (comes up in lightbox)
 - Shows the meta data, notes, associated action items & meetings, and its comment stream
 - Has ability to add a comment to the comment stream.

- Meetings Column
 - Each row
 - Title / Starts At / Ends At / Location / Users (facepile thing)
 - When expanded into lightbox (this is what will be viewed before/during the meeting)
 - Meta data, notes, Agenda items (with the ability to rearrange order, and expand 'em to take notes during the meeting), ability to assign quick action items to users during the meeting (either linked to an agenda item or not).
- Action Items Column
 - Each row
 - Title / Assigned To / Due Date / Followed By (facepile thing)
 - When expanded into lightbox
 - Meta data, notes, comment stream
- Users Column
 - Each row
 - Avatar, Name, Email
 - When expanded into lightbox
 - Action Items, Meetings, Agenda Items that you have permissions to see (i.e. that you are a part of) - plus a simple comment thread where you can have a private message with this user.
- Edit Profile (/dashboard/users/[id]/edit)
 - Simple form (like the login page) for a user to edit their first/last/avatar/email/password.

Here are some final thoughts:

- We are moving towards a multi platform environment so you wanted to, ideally, think of a UI that would **port easily to small screens**. These four columns will make simple mobile views that a user can swipe between, and much of the UX (button locations and such) can probably be similar to the desktop version so there will be *some* continuity between UIs.
- Professional designers will likely make this much better than what you can visualize in your head, so **didn't bother to over-think it**. In this game you can't be a perfectionist. You want to make sure that the four main resources (Users, ActionItems, Agendalitems, and Meetings) were CRUDable (create, read, update, destroy) and that their relationships could be — in theory — understood visually.
- It's a good idea to **focus a user on their task at hand**. Thinking through the UX could be pushing the limits of your skills, so — again — you will rely heavily on professional designers to finalize the way we release this for your client (aka MVP, Minimum Viable Product).

You'll want to make sure each member of your team pours over the UI Spec. It's hard to overstate how important it is that everyone understands the details and is on the same page before you move forward.

End of Document

How to create a user experience flow chart (UX Flow Chart)

Next we examine the process of moving from a User Interface Spec to the construction of a flow chart that describes the detailed user experience (UX) through the app. This is where the engineer(s) and creative(s) must work closely together to decide what kind of software tools can best support the intended interfaces and click/swipe behaviour, down to every last tab, tooltip, lightbox, icon, etc...

Flow Charting Software

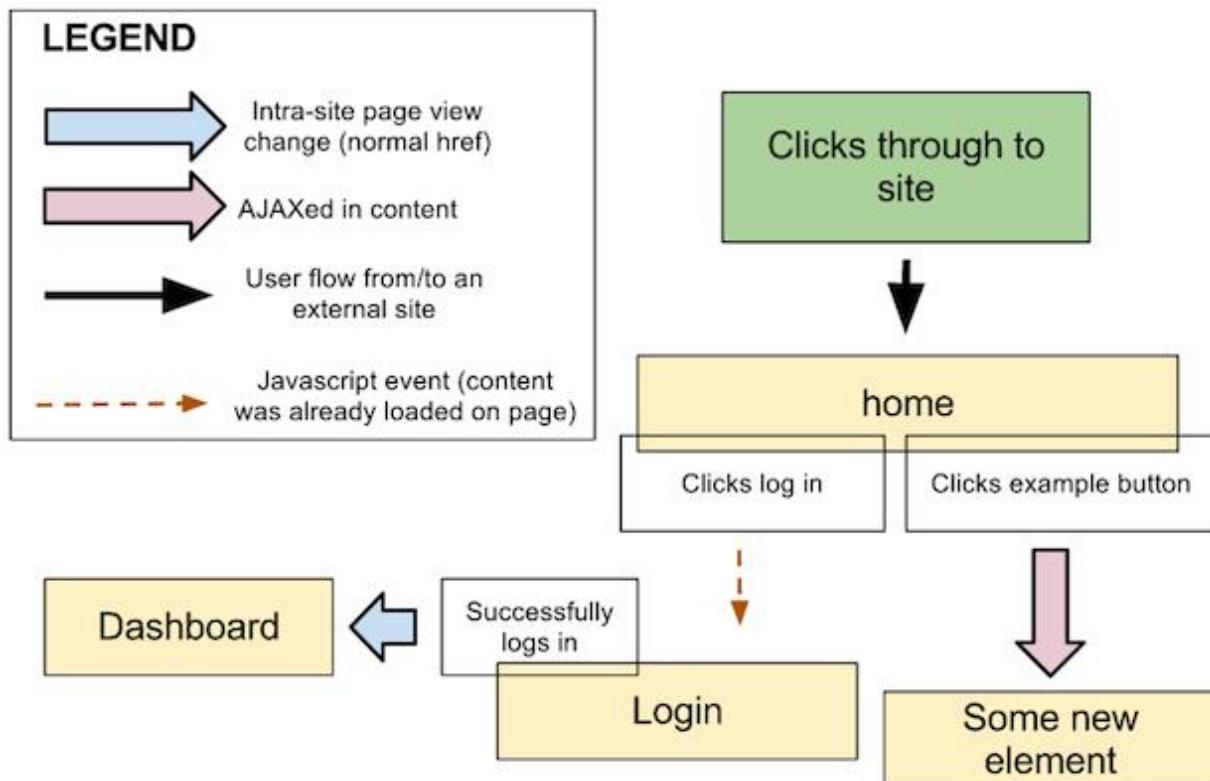
While there are plenty of flow charting tools available out there, we've found that using Google Docs' Drawings App extension is more than suitable for our needs because:

- It's super easy to use and free.
- The built-in collaboration tools are world-class and make comment-tracking and direct email replying to comments simple.
- The ability to link to Google Drive files (or any URL) from 'nodes' of the flow chart facilitates real-time progress-tracking for remote teams.

While Google Drawings aren't explicitly designed for flow-charting we've found that the benefits outweigh the drawbacks.

A Flow Chart Format Friendly to UI Designers and Developers

Below is the simple format we use in a Google Drawing that forces the engineer(s) and creative(s) to think carefully through exactly how content (for web apps) will be presented to users, and allows everyone to click on a node to view the latest wireframe or mockup:



In this example, each of the arrow types represent a critical decision that developers (especially) need to speak into.

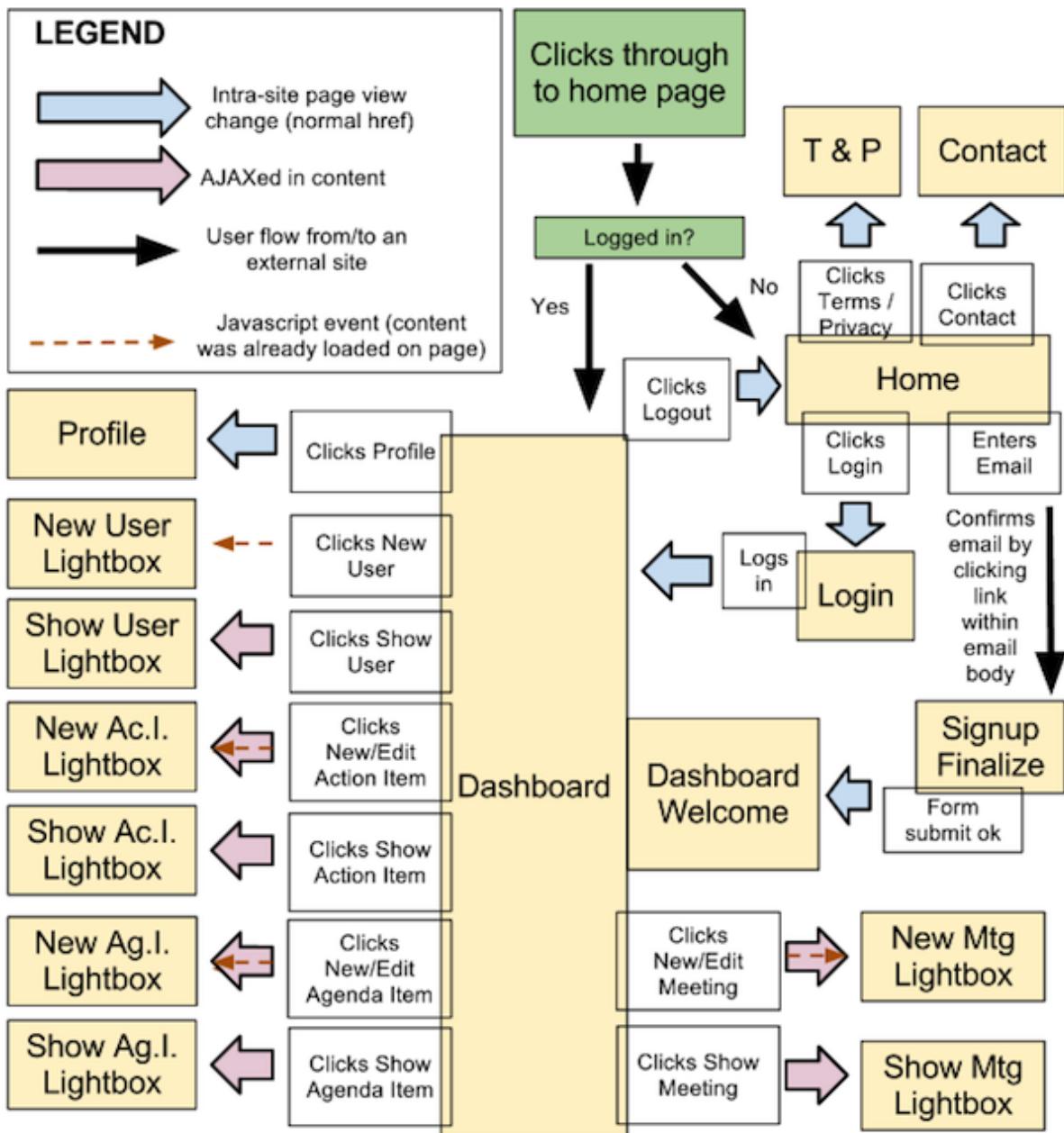
- **Blue-coloured arrows** represent normal page-to-page transitions. These are relatively slow user experiences but are often appropriate, of course.
- **Lavender-coloured arrows** represent Asynchronous JavaScript and XML (AJAX) transitions, which pulls content in behind-the-scenes from the server and presents it as desired into one or more elements of the UI.
- **Brown-coloured dashed arrows** represent on-page JavaScript events such as tooltip presentations, lightbox window displays, tab switches, etc... that do not require AJAX calls (i.e. the content is already present in the browser but in most cases hidden).

What is also important to note is that **yellow boxes** (nodes) represent new pages or elements of the UX that, in our preferred system, always link to URLs of the current wireframe or mockup of that node.

The purpose here is linking nodes directly to URLs of images hosted on Google Drive is helpful because the collaboration tools to comment on specific parts of the image are excellent (e.g. you can draw a box around a specific element and start a discussion about it that your team can have over direct email ... and Google will keep record of the conversation in the comment window for that file.

The 'Our Agenda App' Flow Chart

Here is the specific example of our flow chart based on the UI Spec for a fictitious OUR AGENDA APP:



Here are the key things while constructing this flow chart:

- We wanted to ensure **every top-level component of the UI Spec was covered** so we didn't miss anything.
 - One of the main purposes of this flow chart is to cover **every possible element/tab/icon/etc...** so that our UI designer won't have to come back later and wireframe/mockup something for our minimum viable product (v0/MVP). Thus, some of these nodes can **combine multiple 'states' of the view** to avoid the creation of unnecessary wireframe/mockup images.
 - For example, since obviously the **Dashboard** is our main view, we can tell our designer to both fill in example resources in the four columns (Meetings, Users, Action Items, and

Agenda Items) AND show the UI for what happens when multiple resources from different columns are selected.

- The **Dashboard Welcome** node is attached to the Dashboard node (i.e. it will have all the same UX components) but is separate because this view should be mocked up independently — it has an empty set of columns and need to give the new user a quick “tour” of features, which is a common and helpful feature for apps.
- From the engineering perspective, knowing that this would be a Backbone.js app using Eco templates, we had to make a decision of what front-end code would already be loaded in the browser to facilitate JavaScript and/or AJAX transitions in the UX. This led us to place the blue, lavender, and/or brown-dashed arrows as appropriate so we knew more specifically how we would build out the app later. It is helpful to make these decisions as early as possible.
- The **overlapping JavaScript and AJAX arrows** in this flow chart are simply a shorthand way to note that the “new” views will be simple JavaScript events, and the “edit” views will require an AJAX call. These don't require separate wireframes/mockups.
- The standard Terms and Privacy pages for a web app — noted above as the single node “**T & P**” — will be a generic page layout that we can use down the line for multiple one-off pages as needed.
- And finally, it is important **not to introduce extra features here** beyond what we wrote down in the UI Spec. New ideas will inevitably come in this stage as you build out the flow chart, but you should note them for a future release. Now is the time to focus on building the v0, launching it ASAP, and testing how users interact with it so you can begin making data-driven decisions about new features.

Note carefully how we did both the “Logged in?” conditional and the “Dashboard Welcome -> Dashboard transition.” These types of relatively-complex scenarios can be visually explained in different ways, but the key idea is that all your nodes cover everything your UI designer needs to wireframe/mockup and that he/she understands exactly what's expected. As long as your team is on the same page, then you are good to go.

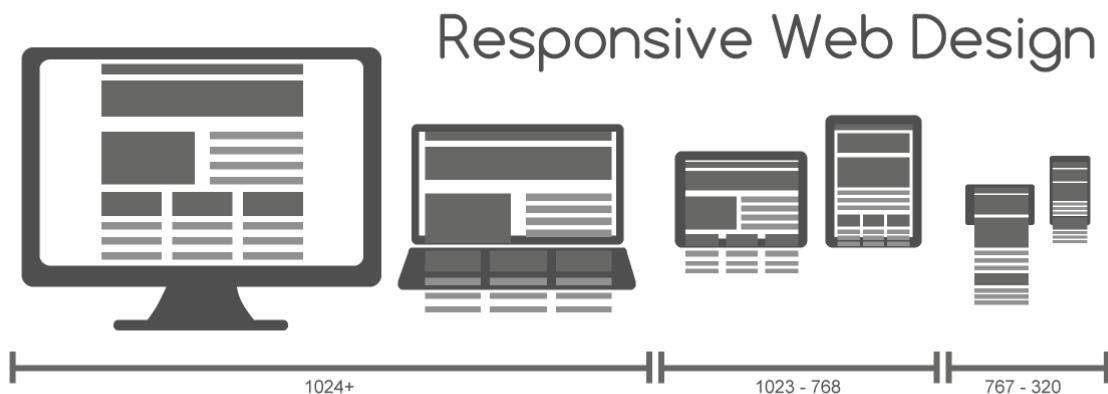
End of Document

Adaptive vs. Responsive Design

Responsive Design

The term Responsive Design was first coined by the web designer and developer Ethan Marcotte in his book, *Responsive Web Design*. Responsive designs respond to changes in browser width by adjusting the placement of design elements to fit in the available space.

A responsive website shows content based on the available browser space. If you open a responsive site on the desktop and then change the size of the browser window, the content will move dynamically to arrange itself (at least in theory) optimally for the browser window. On mobile phones, this process is automatic; the site checks for the available space and then presents itself in the ideal arrangement.



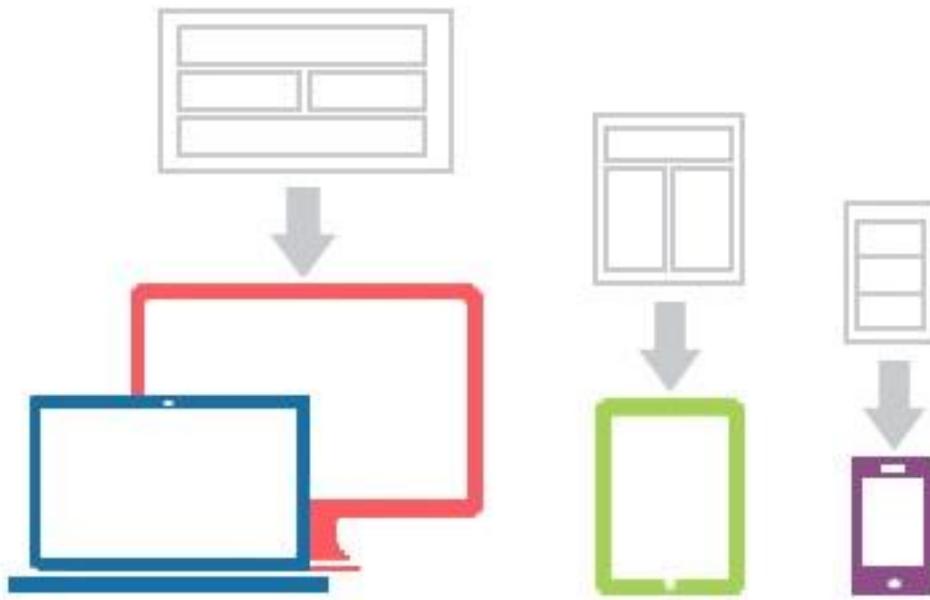
Responsive Design lets designers show content based on the browser space available. This allows consistency between what a site shows on a desktop and what it shows on a handheld device. Responsive design is the “traditional” option and remains the more popular approach to date.

Pros	Cons
Uniform and seamless = good UX. Abundance of templates to use. SEO friendly. Often easier to implement	Less screen size design control. Elements can move around Advertisements lost on screen. Longer mobile download times.

Adaptive Web Design

Adaptive Web Design was introduced in 2011 by web designer Aaron Gustafson in his book, Adaptive Web Design: Crafting Rich Experiences With Progressive Enhancement. It is also known as progressive enhancement of a website.

Where responsive design relies on changing the design pattern to fit the real estate available to it, adaptive design has multiple fixed layout sizes. When the site detects the available space, it selects the layout most appropriate for the screen. So, when you open a browser on the desktop, the site chooses the best layout for that desktop screen; resizing the browser has no impact on the design.



Adaptive Design, developed in 2011, is more involved in that the designer has several fixed layout sizes. It offers an alternative to the “one-size-stretches-to-all” approach.

Pros	Cons
<ul style="list-style-type: none"> Allows designers to build the best UX for the appropriate device. Mobile devices can sense their user's environment. Designers can optimize advertisements based on user data from smart devices. 	<ul style="list-style-type: none"> Labour-intensive to create – most adaptive designs are retrofitting traditional sites to make them more accessible. Tablets and netbooks can have trouble with site configuration tending to be smartphone- or desktop-oriented. Challenging to SEO — Search engines have trouble appreciating identical content on multiple sites.

A Complete Guide for UI Design Process



A Complete Guide for UI Design Process (User Interface)

UI stands for 'User Interface', UI Design Process is a composition of interface animation, visual element, screen layout and content. Essentially it is a graphical layout of an app. In UI Design Process, we will design the product according to the customers choice. The objective of user interface design is to make the user's interaction simple and efficient. The user interface design of any product is critical for its success.

Table of Contents

- Selecting User Interface Elements
- Best Implementation for UI Design Process
- UI Design Process
- Types of User Interface
- Benefits of Good User Interface
- Conclusion

Selecting User Interface Elements

Users are well known with the interface elements, so try to be uniform and excepted in your choice and their layout. This will help the task completion, efficiency and satisfaction.

User Interface elements include:

- Input control: checkboxes, text fields, buttons, radio boxes, data fields, toggles.
- Navigational components: search fields, pagination, tags, icons, slider.
- Informational components: icons, progress bar, message boxes, tooltips, modal windows.
- Containers: accordion.

Best Implementation for UI Design Process

To know your users, understand their objectives, goals, tendencies and priorities. Once you are clear about the user, consider the following while designing your interface.

- **Simple interface:** The interface should be simple to its users. They reject the unwanted elements and are clear with the language they use on the labels and in messaging.
- **Use common UI elements & create consistency:** When common elements are used, the user feels very comfortable to use and are able to get the things done quickly. To facilitate efficiency, it is important to create patterns in language, layout and design throughout the site.
- **Be determined in page layout:** Consider the relationship between the items on the page and structure the page based on importance. Place the items carefully to draw attention to the most important content of information and aid scanning and reading.
- **Attractively use colours and texture:** You can attract the users directly through the colour, texture, light, contrast, which should be used in the design.
- **Use composition to create clearance and grading:** You must carefully consider the typeface. Difference, fronts, size and text arrangement to help scannability and legibility.
- **Confirm that the system transmits what's occurring:** Inform your users of actions, changes in state or errors. Use various required UI elements to communicate status.
- **Avoid defaults:** By thoughtfully thinking and predicting the goals, clients get to your site. You can reduce the burden on the user, while you create a default. It is important when it comes to creating a design where there's a chance to have some fields pre-arranged.

UI Design Process

1. Context and user recognition:

Identify the ways of products application and convenience to the selected viewers. Know the grading of goals and the performance option for gathering selected viewers goals. Ranking the functionality allocate on how good they help to reach the goal. Considerations of the business and functionality related limitation and Choice of best products interface, allowing to reach the main business goal of the assignment.

2. Navigational and systemic interface design:

Designing plots sketch the “user-product” interplay in order to reach the goal. The grading of the scheme by the importance, dependence on the usage frequency and the client’s role. The growth of

the information architecture and the navigation interface design giving the best functionality content and user interactivity plot.

3. Distributional design of interface:

The layout design of the pages which must be present on the screen. Such design meets the needs toward the navigation, graphics, functional and text element of the screen forms the pages. The needs of the question meets the standards of the advantage of checklist.

4. Visual interface design:

In UI Design Process, we design the creative visual elements of the interface to meet the brand I standards and identity of the company, such as style, colour, fonts, graphics solutions. The designing of the icons and the graphic symbols. The design of the screen forms the key screen of content design.

5. Products prototype design:

Creating the work model for its user and combination of arranging the text is a prototype testing of products model on the engineering stage creates more needs towards functionality and user interface to the future product.

6. Testing usability of the prototype:

The experimental way to detect the user elements of the product and to find the problems which are faced by the users while the application is usability testing. The main aim of usability testing is to evaluate the usability of the product.

The methods of testing usability:

- Observation of users
- View of users
- Measurement of benchmark

7. Specification of UI:

The record construction of the “specification of user interface”, gives the outlook of the standards, structural composition and design of the product visually considering the endorsement on the bases of the usability testing results

Types of User Interface

Command-line:

A command-line interface needs a client to type in commands from a list of commands. This kind of interface can consume a lot of time to learn, it is not inborn. Command-line will be irritating for inexperienced users. But it is strong for the experienced users. The users of the command-line interface are the network managers and technicians. The system task and set up the task is required to perform. This task can be done only by this kind of interface.

Graphical user interface:

The interface which is graphical in nature is known as graphical user interface. It is a user interface which consists of graphical elements, such as icons, buttons and windows. It was created to differentiate the graphical interface from the command line interface. These types of interfaces are available in multi-tasking environments. Graphical user interface allows the user to interact with the

electronic device through audio indicator and graphical icons. Graphical user interface is basically functioned by direct manipulation of the graphical elements.

Form-based:

The operating system is developed for the business where an employee has to enter a lot of details. There are field names, where it should be entered next to the place where information must be entered. The radio buttons and down box are the other type of response fields. When the data is entered it is validated. Validation allows only the correct data to enter the system. The other data is rejected. Validated by using a range of methods, it includes: A range

- A character length check
- A data input mask
- A presence check
- Obtain the user to select from a list using combo boxes
- Using check digits.

Menu-driven:

The operating system is designed sometimes with a menu based user interface. The user IT skills cannot be assured in menu-based user interface. There will be a limited number of options on the screen to the system users of the menu-based interface. When the user selects the menu, they get a sub-menu, which gives furthermore options. The selection continues to the further sub-menu. This goes on until the user is able to select correctly, what they want from the choice which is displayed on the screen.

Natural – language user interface:

Natural language user interface is a kind of computer-human interface. Where linguistic facts such as verbs, clauses and phrases act as UI controls for innovating, selecting, changing data in software application. It provides natural human-like interaction with any kind of application. It makes the work efficient, as it rejects the necessity to study a special system of queries and authorize for detailed and precise illustration of the requested information.

Benefits of Good User Interface

The user interface design is a part of the wider user experience process. UI design is concerned with the look and feel of the interface that the user interacts with.

Make a clear vision:

Clarifying helps you to define precisely how to satisfy the needs and expectations of the audience.

Creating your mark:

Being visionary is very essential in design making. Due to incorrect assumptions of the user leads to poor online experience. We have to create the impact of the best design.

User confidence

The best UI design creates the prospective confidence in your user and the brand. If the work or the element won't work properly or looks poor-quality. They will interpret this as a direct reflection of your company. If you can't create a good looking website, it is not possible to deliver good products or service.

Navigation is a breeze:

When the website and web applications are in poor quality and design. It creates frustration for the clients. The website should look better in design, one of the major benefits of UI design is based on UX insights and best product. It helps the audience to look at what they are looking for.

Interlinkage acknowledgement:

Everything should be done in a meaningful manner, for example, with a button having its colour change will interact with your visitor that their action has been acknowledged and completed successfully.

Optimal on-screen experience:

There are various mobile devices on the market and your viewers could be browsing the internet from smartphone and large desktop screen. The benefits of user interface are various, but very easily one of the most important factors is to make sure that different screen sizes are created. You should make sure that your site is providing the optimum level of performance to every visitor.

Clear competitive advantage:

The UI design benefits the ability to convert customers with ease. If you make the life of your visitor easier by good UI design. They're going to show their appreciation by giving you their custom.

Retain custom:

If you design what people are urging for, they will continue to engage. It is a competitive market place so retain the loyal and honest customers.

Design for many users:

Not every client is equal as familiar as with the online world. The range of skill levels is important because you don't want to discourage skilled users in favour of casual users. You must ensure that your site or app works for everyone.

Simplify everything:

Involving in your website or web application should be delightful, not a burden. Simplify the process. Navigate their way through a complex network of pages to make an enquiry or purchase a product.

Conclusion

In this article, we have come across an array of UI design process, types, and benefits. The great user interface design can bring to your website and web applications. We are really passionate about it. We work on various projects where simplifying the user interface has brought massive changing results to the business.