

## UX Walkthroughs

A User Experience (UX) Walkthrough is a tool used to identify usability and accessibility issues in a website or application. It is a procedure for examining a user interface following a set protocol and making assessments based on predetermined criteria. It emphasizes paired or collaborative evaluation of user interfaces by designers and non-designers alike, and serves to bring a diversity of perspectives to bear on the design process. This tool is a synthesis of conventional methods (heuristic evaluation, cognitive walkthrough and code review) that enables the reviewer to make assessments from both a user's point of view and that of the designer.

The multifaceted nature of the UX Walkthrough enables the reviewer to make efficient and effective assessments across several dimensions (general design quality, task-oriented usability, assistive technology usability, accessibility standards compliance & code quality). A UX Walkthrough can be performed by novices as well as experienced evaluators. The result is a comprehensive and multidimensional report of usability and accessibility issues.



## Usability Testing

Usability testing, also sometimes referred to as user testing, is a technique for evaluating the usability of a design by working with an actual or potential user of a product or system. At a basic level, a usability test usually involves a facilitator asking a user to complete a series of tasks (which may or may not be predetermined). The findings are then fed back into the design and development cycle in order to improve the product or system. This does not mean that every suggestion made during testing should be directly incorporated into the design; rather it is the designer's responsibility to consolidate, interpret and parlay the information into the next stage of the design.

Usability testing can be performed at any point in the process, from the early stages of design to the final stages of production. It is most helpful to start early in the design process by performing usability tests with rough sketches or prototypes, and continuing to test more interactive prototypes as a product or system is iterated upon.



## Usability Testing

### Try

1. Before you start, reassure the participant that you are not testing them, rather, you are testing the design
2. Ask them to “think aloud” to understand their behaviour in the context of what they are trying to accomplish, determine whether or not the design is meeting their needs, and identify any confusion about the purpose of a design feature or anticipated outcome of an interaction.
3. Don’t help, react, interrupt, and draw attention to specific issues you care about.
4. Don’t ask leading questions
5. Take a note of any problems the user encounters, suggestions for improvements, and if useful, the amount of time it takes to complete a task

Please find more details about Usability Testing at the following link:  
<https://wiki.fluidproject.org/display/fluid/User+Testing>

## UX Walkthroughs

### Try

1. Choose a user
2. Specify an explicit user goal
3. Prepare working software or prototype
4. Choose an approach (for example, when testing a digital interface include screenreader users, keyboard-only users and users who have difficulty processing text)
5. Plan both the Heuristic Evaluation and the Cognitive Walkthrough
6. Perform the walkthrough and record the results

Please find more details about performing a UX Walkthrough at the following link:  
<https://wiki.fluidproject.org/display/fluid/User+Experience+Walkthroughs>

## Accessible Design Tools

Accessible design tools can be grouped into two different categories:

The first category includes design tools used most commonly during the early ideation and iteration phases of the design process. These tools enable team members to participate in the process of brainstorming, sketching, and ideation.

The second category of design tools are used to execute a design idea. Few of the most common design tools in use today have accessibility features (for example, the ability to attach descriptive text narratives to images, to enlarge the controls, or to navigate and execute designs using a screen reader). However, there are several tools that enable designers to check the accessibility of their final design.



## Accessible Communication Tools

In an inclusive design process, team members share ideas throughout the design/development process and at different stages of completion in order to solicit feedback from peers and stakeholders on an ongoing basis. This approach contrasts with the waterfall approach, a common process where pre-defined steps in the design and development process are followed in a strict order, requiring the completion of one step before moving to the next. With a waterfall approach members of different teams usually communicate only during the transition periods, when passing specific artifacts on from one step to another (e.g. handing off completed design mockups to developers).

An inclusive process encourages the exchange of ideas openly and fluidly, which helps the entire team to be aware of the decisions that are being made along the way, thus minimizing last-minute surprises and improving the final outcome. To maximize participation, the communication tools themselves should be available to all and accessible to individuals with a wide range of needs.



## Accessible Communication Tools

### Try

- Post relevant and timely content on an open wiki
- Use email lists and open channels or chat rooms to discuss issues
- Conduct meetings with audio/video conferencing to include remote participants
- Post events on an open, shared calendar
- Use open repositories to share designs artifacts and code
- Conduct open and regular community design and development critiques
- Participate in daily update meetings to share concerns, issues, items of interest

## Accessible Design Tools

### Try

There are a limited number of design tools available in the first category that are free, open source and/or relatively easy to learn, such as:

- Google Draw - free, with accessibility options
- Omnigraffle - not free or particularly accessible, but easy to learn and use
- Google Sketchup - free but not very accessible

There are several tools that enable designers to check the accessibility of their final design, such as

- ColorSafe - Text and color combination checker
- WCAG Contrast checker
- European Internet Inclusion Initiative - Web Page and PDF checkers

## User States & Contexts

User states & contexts is a tool for conceptualizing, designing, and evaluating the ability of a design to be perceived and operated by users in a wide range of states and contexts. This tool demonstrates all the many states and contexts users can be in at any given time and situation. It provides a way to determine common needs among users in different states and contexts.

A user states & context map can be used to demonstrate user needs that are represented by a particular persona, or that of a collection of personas. The map can also be used to consider how a user's state and context can change in the short term (e.g. on a daily basis) or the long term (e.g. over a lifetime).

By demonstrating the many states and contexts a user can be in at any given time and in any given situation, the map presents the "big picture" of the broad and ever-changing needs of a single user, and also allows a comparison of the needs of multiple users. This can reveal patterns, interesting outliers, or commonalities in needs that might not otherwise be obvious.



## Personas

Personas are models representing potential stakeholders who may use a product or service. Although they are fictional people, their needs, characteristics, goals and motivations are rooted in the insights and feedback collected from various sources including interviews/surveys or through familiarity with the needs and interests of self, co-workers, friends or family members. They begin as early, provisional sketches and often evolve through iterations as more information is gathered.

Personas are behavioural models; they do not represent the full demographics of any given population of complex and unique people. They enable designers, developers and evaluators across a project to keep a broad and diverse collection of stakeholders in mind. They must be developed and used with care in order to avoid stereotyping or fictionalizing the user, and they must be tempered with the awareness that no single persona or group of personae can independently determine the full range of potential uses of a product or service.



## Personas

### Try

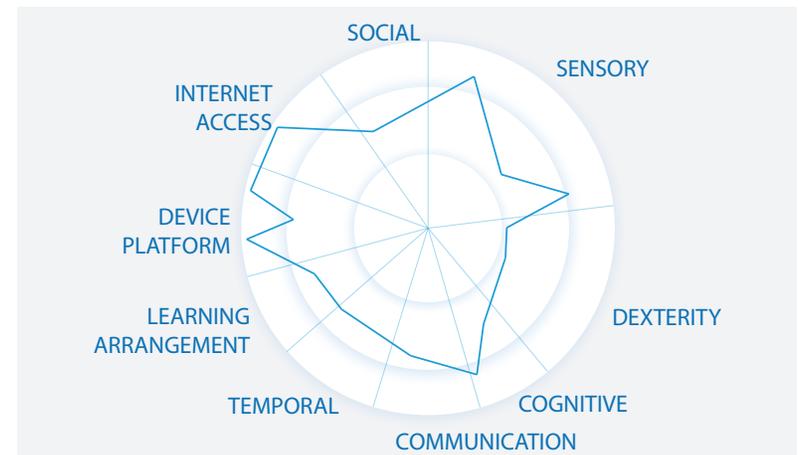
1. Think about various users in your domain with unmet needs
2. Imagine a user, inspired by people you know, that is unique and doesn't simply represent the norm, the average, or the typical
3. Draft the first version of your persona. Give them a name and describe their life, needs, preferences, likes and dislikes
4. Re-iterate and evolve your persona as you collect more feedback from potential users and/or stakeholders

### Combine with

- UX Walkthroughs
- User States & Contexts
- Use-Cases

## User States & Contexts

Needs that are commonly met are placed in the middle, and needs that are not well met are placed toward the edge. This demonstrates that the needs of users “on the edge” are greater in number, as well as the phenomena that by meeting edge needs, you also meet the mainstream needs of the middle.



For more information and examples of User States & Context maps please refer to the following link:  
<https://wiki.fluidproject.org/display/fluid/%28Floe%29+User+states+and+contexts>

### Combine with

- Personas
- Use-Cases

## Use-Cases

Use-cases describe particular scenarios in which a user or persona may encounter and use a product or service, providing more detail about specific tasks and goals as well as helping to map out the potential steps in a workflow. User personas and accompanying use-cases are not meant to exhaustively describe all potential stakeholders or situations; rather they help to illustrate key goals, the main steps that should be taken towards achieving that goal, and behaviour patterns related to the design in question.

In an Inclusive design process it is important to include edge cases. These are personas and use-cases that describe both users with needs that are not typically considered in the design process, as well non-typical or unexpected uses of a product or service. Use-cases present a picture of a person in a specific context, with available tools, existing constraints and potential distractions, who is hoping to achieve a specific goal using the product or service in question.



## Mindmaps

Mindmaps are visual diagrams used for organizing often complex information related to a specific topic or concept. They help designers, developers or others on the project to think through and reveal the interconnections between different pieces of information. Creating a mindmap can be useful in the early stages of a project to move from a loose narrative or vague concept into a more concrete picture of a system or workflow, prior to making more detailed design decisions about specific functions or interface layout etc.

Creating a mindmap is one way to dig deeper into the more complex layers of how a product or system might work. It can be used to explore different options, and/or to present a bigger picture of a complex system. In this way it enables designers or developers to better contextualize a specific topic, concept or system.



## Mindmaps

### Try

1. Start with the central concept or initial state.
2. List all the related sub-topics
3. Repeat this process for lower levels of topics and continue as needed
4. Begin to map out the connections between the items you have listed. These connections may cross over multiple levels of the hierarchy.

Note: Although mindmaps are traditionally visual tools, they can be created and presented in ways that are accessible to non-visual users. For example, they can be created using digital tools that are compatible with screen readers, such as Google Draw. Or they can be presented to an audience with an accompanying text description that includes a step-by-step description of the flow of information.

For an example of a mind-map, please see the following:  
<https://wiki.fluidproject.org/display/fluid/%28PGA%29+Mind+Mapping>

## Use-Cases

### Try

1. Determine what your user is hoping to achieve with your product/system (be specific)
2. Describe the context, available tools, constraints, potential distractions, etc.
3. Describe how the product can help your user achieve their goals
4. Re-iterate and evolve your use-cases as you collect more feedback from potential users and/or stakeholders as your project moves forward

### Combine with

- Personas
- User States & Contexts

## Metadata Authoring

Content authors can use metadata to provide information about what needs a learning resource can accommodate. Metadata can be used to describe, among other things: what the technical/environmental requirements are for a given resource (e.g. is a mouse required to interact with the resource?); what sensory modalities are required to perceive the content of the resource (e.g. does the learner need to be able to hear the content?); what learning style the resource accommodates (e.g. is this a highly visual learning resource?).

Describing Open Education Resources (OERs) and other resources with metadata allows a delivery system to automatically personalize the content and its delivery to meet the unique needs of each learner. Metadata can be recorded in a metadata record separate from the resource itself.



## Metadata Authoring

Authoring metadata promotes cumulative accessibility of resources as third parties create and associate alternatives to original resources. Together, the many resources in a system contain the features or educational materials that every student needs, but no single resource must be 100% accessible to every kind of learner. By creating metadata about each resource, it is easier to determine which resources might need to be adapted to meet an individual's needs.

For more information about metadata authoring please see <http://handbook.floeproject.org/Metadata.html>

### Combine with

-  Personas
-  Use-Cases



## Wireframes

Wireframes provide a way to develop and present design ideas in the early stages of a project. They can be made up of very rough paper sketches or can be more refined and detailed digital artifacts. In the early stages of designing an interface, they can be used to help plan the layout of the design. They are a good way to focus on the functionality of the design without getting into the details, such as style and colour.

Alongside the visual wireframes, it is best to also consider narratives, long descriptions, or written step-by-step "tours" through the wireframes as a way to create a textual alternative. Not only does this make the wireframes more accessible, but it is a good way to start imagining how the interface can be presented in alternative modes. Considering this in the early stages of design will result in more successfully integrated features.

The wireframes can be shared with different stakeholders and users to solicit feedback and refine your design prior to spending too much time and effort in implementation.

## Prototyping

Prototyping is an effective way to quickly communicate your ideas with others, solicit feedback, and learn through making and sharing your ideas. Prototypes can be used at different stages of your design and development process.

In the early stages, try rough and low fidelity prototypes to test a number of different ideas. Don't get bogged down with details and don't try to make them look good. Paper prototypes are good examples of rough and quick prototypes.

When your ideas are more refined, you can spend a little more time on details and the look and feel of the prototype. These prototypes focus more on the interactions between a user and a product, and help to refine the details and interaction patterns.

High fidelity prototypes come in handy in the final stages of the design and development processes. Their look, feel, and function more closely reflect the final product. They are often presented in a final round of usability testing.

## Prototyping

### Try

- There are several digital tools that help designers create interactive prototypes, many of which are available online. For example, InVision is a prototyping tool that is easy to learn and use.
- If you are designing a product, you can use 3D printing to build high fidelity prototypes.
- If you are working on a digital product, such as a website or an application, try publishing a beta version to get user feedback and fix the technical issues.
- If you are designing a service or a system, try roleplaying sessions to test your proposed solution.

### Combine with

- UX Walkthroughs
- Usability Testing

## Wireframes

### Try

1. Start by listing the global features of your interface as well as features that are specific to individual screens, in order to get a clear idea of the features and functions needed
2. Create a skeleton for each screen; features can be located within the screens and interaction behaviors assigned to each one
3. After several iterations and user testing sessions, create high fidelity wireframes that more closely reflect the look, feel and interactive behavior of your final product

### Combine with

- UX Walkthroughs
- Usability Testing
- Personas
- Use-Cases
- Accessible Design Tools

## Multimodal Testing

When designing and developing a product or service, consider all the different ways that a user might interact with it. For example, in the case of a digital interface, ensure that you integrate features into your product to make it compatible with screen readers, keyboard only users, and single switches. When you are ready, you can test your design in a number of different ways. It is best to run usability tests with participants who have a broad range of needs and experiences.

The audio experience of a digital interface can be tested in the very early stages of design with only sketches or paper prototypes, by having someone act as a human screen reader (by reading out the audio as a screen reader would). Once an interactive prototype is created, it can be tested with various input/output methods, including screen readers or other assistive technologies. Websites can be tested using various web accessibility checkers, however, these should be used only as an initial guide.

## Infusion

Infusion is a code framework for building personalizable and adaptive applications using JavaScript and other web technologies. It also includes a growing collection of user interface components, built on top of jQuery, that help make it easier to build user interfaces by mixing and matching accessible components and then customizing them to suit a unique context. Infusion's components are designed flexibly so that all their default features, styling and parts can be easily changed, replaced and added to.

The Infusion framework gives developers a way to build open, modular systems that can be changed and configured during use, so that anyone can modify and adapt it to create systems, applications, components, etc. that meet their specific needs. The Infusion framework reflects an accumulation of experience about how to make applications accessible, and is as much about community values, shared inclusive practices, and documentation as it is about code and components.

## Infusion

For more information about Infusion, please see <http://docs.fluidproject.org/infusion/development/>

## Multimodal testing

### Try

1. Use the native screen reader on your computer (e.g. NVDA on Windows or Voice Over on Mac - both of which are free of charge) to run initial tests on a digital interface
2. Navigate an interface without a mouse or track pad to test for keyboard-only interaction
3. Find a user who is familiar with the specific assistive technology in question to test your designs, as their experience will provide more realistic and nuanced feedback related to their expectations of typical behaviour.

### Combine with

- UX Walkthroughs
- Usability Testing
- Prototyping

## Synchronized Design Assets

When working collaboratively, designers need a platform that enables them to freely and easily share design artifacts. An ideal tool would allow versioning as well as structured commenting and discussion that is attached to the design artifacts (with associated archiving). The platforms that are currently available to developers, such as Github, are not compatible with most designers' needs and workflows. Thus, designers have often been left to come up with alternative methods of collaborative file sharing, each with its own benefits and limitations.

The main purpose of creating synchronized design assets is to provide the community with a common location where they can access the latest version of a file rather than searching through emails or chat logs. This helps the design teams establish more stable and shared archives for their design projects, and makes it easier to bring newcomers on board.



## Accessible Development Tools

Accessible development tools can be grouped into two categories:

The first category consists of tools that enable developers to share their projects, solicit peer feedback, and collaborate.

Development tools in the second category provide accessible environments for writing code. Most current code editors are highly customizable and developers can adjust the font type, size, text color, background color and brightness, etc. based on their preferences. There are few other accessible development tools that are made specifically for developers with unique needs and preferences.

Using a combination of these two categories can enable developers with a wider range of personal needs and preferences to contribute to the development process, collaborate, and share their projects with others.



## Accessible Development Tools

### Try

- One of the tools that provides an open platform for sharing projects and source code is Github. The Github web browser-based interface is compatible with screen readers and can be navigated by different means including keyboard, mouse and switches.
- An example of a development tool that provides an accessible environment for writing code is emacspeak. This is a “free” computer application, a speech interface and an audio desktop environment built for developers who rely on screen readers.

## Synchronized Design Assets

### Try

1. Store the latest version of a file on a cloud based sharing system, such as Dropbox.
2. Provide a link to this file on a wiki page that is available to all members of your community  
Note: The advantage of linking the file from a wiki page rather than sharing it directly via email, chat or other means is that people can leave comments and feedback about the design artifact in question, and a record of that discussion persists. In addition, it provides the community with a common location where they can access the latest version of a file
3. Notify your team or the larger community of your posting via email