

# UX Walkthrough Protocols and Checklists

## The Fluid Approach

The Fluid approach to UX Walkthroughs involves performing a [Heuristic Evaluation](#) and [Cognitive Walkthrough](#), often at the same time. By using a combined approach, an evaluator can gain insight beyond that found by performing a single method in isolation. A UX Walkthrough can also involve an under-the-covers accessibility [Accessibility Markup Review](#) of the HTML.

Our aim for Fluid is **Software that works - for everyone**. This means that we assess *accessibility* as well as *usability*. Additionally, rather than having two separate inspections for usability and accessibility, we use a unified approach that addresses both areas.

Either of the primary methods - heuristic evaluation and cognitive walkthrough - may be used to address usability, accessibility, or both. The only difference is the set of principles applied or questions asked during inspection. Checklists for both usability and accessibility are offered here for each method.

When evaluating an application or website using a Fluid UX Walkthrough, you will be able to determine both:

- How well the application fulfills the basic requirements for usability and/or accessibility.
- How the interface accommodates a particular user - based on a [persona](#) or detailed knowledge about a particular user type.

## Before you Begin:

- Read the page [UX Walkthrough Preparation and Execution](#)
- Print a copy of [Usability Evaluation Questions](#)
- Print a copy of this page to use as a reference during your inspection.

## Fluid UX Walkthrough Protocols & Checklists

### Checklist: Heuristic Principles for Usability <sup>1</sup>

A heuristic evaluation is an expert review of a user interface with reference to explicit principles termed "heuristics." During the evaluation, reviewers see how well the service fulfills the basic requirements for usability or accessibility.

To conduct a heuristic evaluation of a website or application, travel through the pages of the site, examining each one while reflecting on each of the heuristic principles. Make a record of where you find compliance and where you find violations. Note that a heuristic evaluation may also be done in combination with a cognitive walkthrough.

#### Heuristic Principles

##### \_\_\_ Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

##### \_\_\_ Match between system and the real world

The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

##### \_\_\_ User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

##### \_\_\_ Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

##### \_\_\_ Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

##### \_\_\_ Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

### \_\_\_ Flexibility and efficiency of use

Accelerators - unseen by the novice user - may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

### \_\_\_ Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

### \_\_\_ Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

### \_\_\_ Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

---

<sup>1</sup> The above section lists the original Nielsen and Molich heuristics, as refined by Nielsen.

## Checklist: Heuristic Principles for Accessibility

The following is an abridged version of the [Simple Accessibility Review Protocol](#). You may want to read the full version to gain a better understanding of how the principles apply to accessibility.

### \_\_\_ Assess the overall layout, structure and content of the page

- Is the page structured into logical sections?
- Are those sections clearly labeled, and is their function apparent?
  - Do paragraphs, sections, and tables have meaningful headings?
  - Is the information in forms grouped in relevant "clumps"?
- Is site content helpful to users of assistive technology?
  - Can headings, captions and links be understood independently of their surrounding context?
  - Does sentence structure follow writing conventions in a consistent way?
- Are there sufficient non-visual cues to convey the overall structure? (I.e., Are there sufficient text and/or explanations on the page to understand its purpose and functionality without relying on graphic elements).
- Are there sufficient visual cues to support page content?
  - Are visual cues recognizable without being differentiated by color?
  - Does there seem to be sufficient contrast between foreground (text) and background colors?
- Are the most important information and navigational elements prioritized high on the page?
- Is navigation consistent from page to page? Is the structure of content consistent from page to page?

### \_\_\_ Play around with the layout: enlarge the font size; change the size of the window (bigger and smaller); adjust your resolution

- Is all the text still visible? Does it overlap?
- Does visual context break down at large text sizes?
- Are headers, labels, and visual cues still correctly associated with their content?
- Do columns shift or realign correctly?

### \_\_\_ Use the Tab key to navigate through the entire page.

- Do all links and controls receive focus as you tab through the page?
- Can all controls be activated with the Enter or Space keys?
  - Is movement possible using arrow keys within the object (for example, to move around cells within a table, or within a form)?
- Does tabbing follow a logical sequence through the page?
- Is content easily accessible and not secondary to navigational "chrome"? Are navigation and content formats consistent from page to page?
- Are there any areas where you get stuck and have to use the mouse?
- Are shortcuts or skip links provided to make keyboard access quicker? Are they visible and clear?

\_\_\_ **Use Internet Explorer (6 or 7) or Firefox with Popup Alt Attributes Extension to check for alternative text for all images and title text for links.**

- Do images have text alternatives for visual content?
- Are descriptions short (around 20 characters maximum) and using the `alt` attribute?
- Are decorative or meaningless images such as spacer graphics using empty alt text (`alt=""`) so they will be ignored by screen readers?
- If a description cannot be adequately provided with 20 characters, does a graphic have a supplemental `longdesc` attribute?
- Is title text specified for links? When link text cannot be made descriptive enough by itself, more of an explanation can be provided through the use of the `title` attribute.

If you wish to do a more comprehensive accessibility inspection than is described here, you will find detailed instructions at the following links:

- [Comprehensive Accessibility Review Protocol for PC](#)
- [Comprehensive Accessibility Review Protocol for Macintosh](#)

## Protocol: Cognitive Walkthrough

Sometimes performed at the same time as the [heuristic evaluation](#) (see left column), a [cognitive walkthrough](#) is a step-by-step exploration of a user interface to see how well a particular type of user (sometimes represented by a [persona](#)) will be able to perform a task or set of tasks.

### \_\_\_ Choose a user

- Choose a user from whose perspective you will perform the walkthrough. This may be easiest to do by selecting a [persona](#) - see [UX Walkthrough Preparation and Execution](#) for more information on this.
- If using a persona, ensure that the persona:
  - is adequate to judge what knowledge the user may plausibly be expected to have
  - specifies the particular needs, preferences, and limitations the user may have
- Note: separate walkthroughs may be needed for each persona or user type, although some issues will likely show up in more than one walkthrough, resulting in later walkthroughs going more quickly than earlier ones.

### \_\_\_ Define a goal and a set of tasks

- Determine the specific result desired by the user and motivating the interaction
- Define a set of tasks required to accomplish the goal
- Lay out the sequence of steps the user/persona should go through to perform the tasks and achieve the goal. (Information on defining goals can be found in [UX Walkthrough Preparation and Execution](#)).

### \_\_\_ Perform the tasks

- For each step in accomplishing the tasks, ask:
  1. Will the user know what to do at this step? Is complex problem solving needed to figure out what to do?
  2. If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal? Is complex problem solving needed to interpret the feedback?

#### \_\_\_ Record results

## Protocol: Cognitive Walkthrough with Accessibility

In an accessibility walkthrough, the main consideration is how low vision, blindness, impaired hearing, motor control limitations, or cognitive issues affect the use of a website or application. The cognitive walkthrough protocol is followed, with an emphasis on accessibility cues.

#### \_\_\_ Choose a user

- Choose a user from whose perspective you will perform the walkthrough. This may be easiest to do by selecting a [persona](#) - see [UX Walkthrough Preparation and Execution](#) for more information on this.
- If using a persona, ensure that the persona:
  - is adequate to judge what knowledge the user may plausibly be expected to have
  - specifies the particular needs, preferences, and limitations the user may have
- Note: separate walkthroughs may be needed for each persona or user type, although some issues will likely show up in more than one walkthrough, resulting in later walkthroughs going more quickly than earlier ones.

#### \_\_\_ Define the goal & tasks

- Determine the specific result desired by the user and motivating the interaction
- Lay out the sequence of steps the user/persona should go through to accomplish their goal. (Information on defining goals can be found in [UX Walkthrough Preparation and Execution](#)).

#### \_\_\_ Initial Assessment

1. Assess the overall layout and structure of each page.
2. Play around with the layout: enlarge the font size; change the size of the window (bigger and smaller); adjust the resolution.
3. Use the Tab key to traverse the entire page.

#### \_\_\_ Perform the tasks

For each step in accomplishing the task, ask:

1. Will the user know what to do at this step?
  - Is complex problem solving needed to figure out what to do?
  - Are the cues provided accessible to the user?
  - Can the cues be understood by someone who doesn't process text well?
  - Can the cues be found by someone who can't scan the screen easily?
2. Will the user be able to carry out the required action?
  - Can it be performed easily by a keyboard-only persona?
  - Can it be performed without visual monitoring?
3. If the user does the right thing, will they know that they did the right thing, and are making progress towards their goal?
  - Is complex problem solving needed to interpret the feedback?
  - Is the feedback accessible to the user, and can they find it (as for cues)?

Considerations:

- Blind persons and some persons with limited motor control need keyboard-only operation
- Persons with cognitive issues need visuals that reinforce text
- Persons with low vision must enlarge page content; deaf people require video captioning and visual, rather than auditory, prompts.
- You also must consider the assistive technology the user will use. Examples include screen magnifiers such as ZoomText, screen readers such as JAWS, or combination screen readers /enlargers such as the Kurzweil 3000.

#### \_\_\_ Record results

Unable to render {include} The included page could not be found.

## Wrapping Up

- Use the appropriate [UX Walkthrough Report Template](#) to report your findings

## References

- More detail on each of the heuristic principles can be found in an expanded list from Deniese Pierotti of Xerox, which itemizes specific things to look for when evaluating a system with Nielsen's Heuristics. See [Heuristic Evaluation - A System Checklist](#).
- A paper from Claire Paddison and Paul Englefield provides a list of nine heuristic principles for accessibility evaluations: [Applying Heuristics to Perform a Rigorous Accessibility Inspection in a Commercial Context](#)  
(Click on the Full Text PDF link and view pages 129-130.)  
Paddison and Englefield include in their paper a general discussion of the heuristic approach. This is recommended reading for all reviewers.