Accessibility and Usability Heuristic Review for Responsible Interface Design

Introduction

This evaluation template is a living document, an attempt to bridge best practices in security-minded usability reviews with best practices in accessibility reviews. We seek to empower tool teams to proactively take steps to improve the accessibility and usability of their tools, and believe that this review template is an effective first step.

The usability heuristics are adapted from an industry standard, the Nielsen Norman Usability Heuristics. They incorporate relevant recommendations developed by the Tech Policy Design Lab's project Deceptive Design: Moving Towards Trusted Design Patterns, which are strategies and recommendations created in collaboration between the World Wide Web Foundation, 3x3, and Superbloom Design.

We worked with Accessibility Lab to adapt their audit procedure into this self-guided review. Not all WCAG standards can be realized through this review, and we recommend contracting professionals like Accessibility Lab to do a full review of your product if you feel unsure about your compliance with current accessibility standards.

How to use this resource

What is a heuristic evaluation?

A traditional heuristic evaluation is a way for tool teams and designers to locate usability and design issues within an interface. Heuristics act as guidelines or ideals toward which we hope all interfaces can strive that make systems safe and easy to use.

An evaluation of this type is, in essence, a reaction to a designed interface, which can be in many different states of "doneness" – from a lofi prototype to an existing, established app. As with all User Experience (UX) interventions, the project will benefit from being evaluated as early in its development as possible. The same goes for evaluating the accessibility of your project: the earlier you consider accessibility best practices, the better for your users and for your development team. Doing an evaluation early in a development cycle also helps you as a tool builder to not invest development and implementation time and effort into a design or build that may not be accessible or usable for users might lead to eventual overhaul of the tool down the line.

These kinds of evaluations are helpful in a few ways:

- If you don't have much resourcing for design or UX, it can be an accessible way to make progress in usability and accessibility
- Getting an overview of usability and accessibility issues that may be present in a piece of software
- Seeing thematic gaps in your team's usability and accessibility practices

Heuristic evaluations cannot replace user research or user testing, which are both methods and processes that will give much more nuanced and comprehensive insights into the usability of your system.

Set up and execution

Heuristic evaluations work best when multiple people evaluate the same interface. Evaluators do not need to be UX experts or even part of your tool team – in fact, it can be very beneficial to have an "outsider" review the usability of your system. Just make sure that they're reviewed and understand the heuristics you're using.

Each evaluator should collect their insights in a common style, but without influence from others' evaluations. Some options include:

- An evaluation template.
- A spreadsheet: Evaluators can capture one observation per line, along with its corresponding heuristic.
- A tool like Miro with workspaces for each evaluator. This enables evaluators to include screenshots and other

It can be quite daunting to review an entire system, website, or application. Narrowing the scope can help make the evaluation more robust and complete. Using materials you may already have, like personas or user journey maps, you can look at a particular user task or pathway, a particular page of your site, its use on a particular browser or device, etc.

Evaluations of this type, performed by a variety of evaluators, can help combat bias in your work. They can help to interrogate your reasoning and work to ensure you do not deepen existing bias or entrench structural discrimination. They can also help increase digital equity by foregrounding accessibility, combatting the common issue of accessibility as an afterthought in software development.

Of course, not all of the tips and guidelines listed below will apply to every software project, which each has its own goals, objectives, and orientations towards their users.

Heuristics and guidelines can never replace user input

While we believe that the principles and guidelines here can help you to design more usable, accessible, and security- and privacy-minded software products and tools, they are not guarantees. A tool or website can be 100% WCAG compliant and still have accessibility challenges that went unforeseen. **The only way to get ahead of these surprise issues is to test your tools with real and diverse sets of users early and often.** Evaluations can be paired with user testing cycles within your development processes, working very harmoniously with each other.

Having diverse sets of users from multiple perspectives is important to cover the bases of usability, accessibility, and security:

- Users representing your different personas will improve usability issues
- Users that employ different assistive technologies will improve accessibility issues
- Users that have a range of threat models will improve security and privacy issues.

Principles of respectful interface design

This evaluation intends to pull together best practices in responsible interface design from the perspectives of security and privacy, accessibility, and usability. For example, some of the insights that follow are adapted from Nielsen Norman's guide. We want to highlight some of the core tenets of the leaders in these fields to keep in mind as you evaluate your interface based on the heuristics in the next section.

WCAG guidelines

While specific accessibility requirements are woven into the numbered heuristics below, it is helpful to keep in mind the guiding principles of WCAG compliance as you evaluate your system:

1. **Perceivable**: Information and user interface components must be presentable to users in ways they can perceive.

2. **Operable**: User interface components and navigation must be operable.
3. **Understandable**: Information and the operation of an user interface must be understandable.
4. **Robust**: Content must be robust enough that it can be interpreted by a wide variety of user agents, including assistive technologies.

Please note that Accessibility Compliance is measured with the following system:

- A – minimum level of compliance (critical for all)
- AA – intermediate level (essential) **-** this is the requirement for governments and telecommunications operators
- AAA – highest level (essential for some)

Trusted design guidelines

We have weaved in implementation recommendations from the Tech Policy Design Lab's project entitled Deceptive Design: Moving Towards Trusted Design Patterns. Deceptive designs (also known as "Dark Patterns") are built into website or app interfaces and can manipulate users into making choices they may not want to make. Trusted Design, by contrast, respects people's human rights. It prioritizes people, not platforms, and empowers them to make informed choices. Trusted Design is important because it expands autonomy and consent for users on the internet, particularly vulnerable populations. While specific recommendations are listed under the heuristics below, these three principles are also useful to keep in mind when reviewing an interface:

1. **Build for at-risk communities first**: The people experiencing the most serious harms from deceptive design are from vulnerable and marginalized groups. Start by designing for them.
2. **Deliver user agency and control**: Most digital products and services are developed by companies with a duty to make money for their shareholders. Deceptive design manipulates people using technology into acting against their own interest for the benefit of those who made the product. Combatting deceptive design requires acknowledging the agency of end-users, and respecting their right to make decisions that are less profitable for companies.
3. **Free people from cognitive burdens**: Overwhelming users with extraneous information is a dubious business practice. Rights-Respecting Design should not put the burden of avoiding harms on the user. An interface may make something technically possible, but weaponizing the difficulty level may make it impossible in practice.

Security and privacy considerations

Creating privacy and security respecting systems is a complex responsibility that goes far beyond what is covered in this document. However, we wanted to highlight some considerations from the user experience perspective that may help improve the privacy and security of your users, especially to those on your team responsible for Terms and Conditions, Privacy Policies, and GDPR compliance:

- Advocate for security best practices such as following privacy-by-design and privacy-by-default principles that protect people's digital rights.
- Stay updated on laws and policies that impact product design and implementation, such as privacy, data protection, consumer rights, and human rights in general. Advocate for going beyond mere compliance.
- Product teams often rely on user data, such as crash data, to improve their products. Yet use of the data can entail privacy risks. Communicate those risks clearly and openly. Make potential harm visible.
- Ensure your team provides discussion space and transparency for product decisions.
- Consider asking an Institutional Review Board or data protection officer to review your site's balance of meeting research objectives with robust public disclosures of the risks/benefits for participants.
- Consider developing a recognizable and uniform logo or button to promote awareness of your data use and privacy policy, ideally a short-form privacy notice.
- Devote extra attention and resources to your request for consent.
- Forefront in plain language what data you collect, for what purpose, and how you use and store it without double negatives and ambiguous descriptions.
- Test to see whether your visitors or users may want to temporarily disable data collection. If that's a common desire, give them an easy way to opt out for a period of time.

Heuristics and principles for responsible interface design

Note: for each of the accessibility guidelines referenced below, please see their footnote to read more about which WCAG 2.1 standard they are referring to.

#1: Wayfinding & visibility

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

Tips for evaluation

- Avoid "black boxes" or "magic" when possible.
- Avoid misleading cues that lead users to make false assumptions about what the system is doing, like loading animations that take longer than necessary.
- Don't bury key information that affects an individual's personal data or opt-out process in the long scrolls of a privacy / personal data policy.

Related Accessibility Guidelines to follow:

- Always provide alternative texts to images where possible.[1]
- Ensure that information and relationships that are implied by visual or auditory formatting are preserved.[2]
- Ensure a correct reading sequence.[3]
- Allow devices can be used in any orientation (portrait or landscape).[4]
- Let the user pause or stop the automatic audio.[5]
- Avoid background for the prerecorded audio.[6]
- Check that color contrast is sufficient.[7]
- Allow font and button size to be large enough.[8]
- Ensure visual displays meet accessibility compliance.[9]
- Responsive web design.[10]

[1] See WCAG 1.1.1, 1.4.5, and 1.4.9.
[2] See WCAG 1.3.1.
[3] See WCAG 1.3.2.
[4] See WCAG 1.3.4.
[5] See WCAG 1.4.2.
[6] See WCAG 1.4.7.
[7] See WCAG 1.4.3, 1.4.6, and 1.4.11.
[8] See WCAG 1.4.4.
[9] See WCAG 1.4.8 and 1.4.12.
[10] See WCAG 1.4.10.

- Ensure interaction is predictable if hover or focus causes content changes.[11]
- Ensure that target sizes are large enough to easily activate them.[12]

[11] See WCAG 1.4.13.

[12] See WCAG 2.5.5.

#2: Human language

The system should speak the users' 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.

Tips for evaluation

- Make complex text available for review later, for example by emailing a copy.
- Consider an 80% visual to 20% written ratio when presenting new concepts or new information to the user for the first time.
- Be aware of how different cultural contexts give different meanings to words, phrases, shapes, and colors, and adjust to avoid misunderstanding.
- Use plain language to communicate your privacy policy. Communicate clearly what the necessity or utility is to the people of sharing this specific information.
- Avoid using language that creates a false sense of urgency or necessity.
- Avoid making right-to-left language interfaces more complicated than left-to-right.
- People with cognitive disabilities should be able to understand what they're consenting to.

Related Accessibility Guidelines to follow:

- Use clear language and content that is readable.[13]
- Simplify complex technical information and avoid legal jargon, and incrementally disclose more complex details.[14]
- Write for an elementary reading level with language that accommodates a wide range of literacy and non-native speakers.[15]

#3: User control and agency

People may end up making unintended choices, or outright mistakes, as they use or explore your tool. Choices should be easily undoable and reversible, so that nobody gets stuck in a virtual cul-de-sac. This encourages learning by exploration, and leads to fewer errors. It also gives users the ability to make informed decisions, especially about privacy and security related settings and calls to action.

[13] See WCAG 3.1.1 and 3.1.2.
[14] See WCAG 3.1.3, 3.1.4, and 3.1.6.
[15] See WCAG 3.1.5.

Tips for evaluation

- Bring attention to the part of the interface where visitors have to make a choice instead of using subtle links to conceal the option.
- Add risk and benefits information for choices people make.
- Minimize the use of infinite scroll and/or recommendations based on algorithms that prolong, delay, or otherwise influence a person's choices by overwhelming them with information.

Related Accessibility Guidelines to follow:

- If someone uses accessibility software or assistive technology, they should have a comparable experience to someone who does not.[16]

#4: Consistency and standards

Take a systematic approach to words, visuals, and images. Users should not have to wonder whether different words, situations, or actions mean the same thing. Don't invent new vocabulary, verbal or visual, unless necessary.

Related Accessibility Guidelines to follow:

- Make the content appear and operate in predictable ways.[17]
- Use of consistent presentation, layout and identification of functional components.[18]
- Encourage design that gives users full control of changes of context.[19]

[16] See WCAG 4.1.2 and 4.1.3.
[17] See WCAG 3.2.1 and 3.2.2.
[18] See WCAG 3.2.3 and 3.2.4.
[19] See WCAG 3.2.5.

#5: Error prevention, diagnosis, and recovery

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. Error messages should be expressed in plain language, precisely indicate the problem, and constructively suggest a solution.

Tips for evaluation

- Check that no components are in areas that people are likely to hit by mistake.
- Have clear options for users to report problems and get help.

Related Accessibility Guidelines to follow:

- Make it easier to fill out forms, indicating the purpose of common inputs (autofill).[20]
- Make it easier to operate and navigate content, indicating the meaning of all controls.[21]
- Help users avoid mistakes.[22]
- Help users correct mistakes.[23]

#6: Efficiency and just-in-time information

Minimize the user's mental load by making objects, actions, and options visible. The user should not have to remember information in order to use the tool. Instructions should be visible or easily retrievable whenever appropriate. Provide help and documentation when and where people are likely to need it. Offer information freely and openly, rather than making people search for it. Use defaults, hints, and informative empty states in order to both smooth the process for experienced users and "teach" less-experienced users.

Tips for evaluation

- Count how many clicks or taps it takes to complete a privacy-protecting action, and make sure it is not more complicated than a privacy-ambivalent choice.
- Approach respectfully with an on-screen nudge instead of disrupting an important workflow or process. Give people the option to decide later.
- Approach the site visitors or app users when what you are asking for is related to the view or task they are engaged in.

[20] See WCAG 1.3.5.
[21] See WCAG 1.3.6.
[22] See WCAG 3.3.2, 3.3.4, and 3.3.6.
[23] See WCAG 3.3.1 and 3.3.3.

Related Accessibility Guidelines to follow:

- Offer users help text that provides information related to the function currently being performed.[24]

#7: Clean, task-centered interface

Every word, visual flourish, or other piece of information should have a purpose related to what the tool is trying to help people accomplish. Respect the task at hand and avoid unneeded urgency: alerts, notifications, and banners hurt the user's ability to focus. Do not demand more of people's time and attention than they need in order to get the job done. Avoid or be careful with time limits.

Tips for evaluation

- Make sure action buttons or banners don't take up half the screen and hide important information on all devices, operating systems, browsers, and apps.
- Present "yes" and "no" options equally by creating a symmetrical design with the same font, color, and readability.

Related Accessibility Guidelines to follow:

- Provide users enough time to read and use content or avoid using time limits.[25]
- Avoid moving, blinking, scrolling, or auto-updating information or let the user pause, stop or hide it.[26]
- Let the user postpone or suppress interruptions, except in an emergency.[27]
- Let the user continue the activity without loss of data after re-authenticating.[28]
- Warn of the duration of any user inactivity that could cause data loss.[29]

#8: Consent and opt-in

Particularly when it comes to sharing data or changing settings, users should be presented with words, images, and design tools that encourage them to understand what they are doing in each specific case. Users should be defaulted to settings that do not compromise their privacy, being prompted to opt into these features (like ad tracking, for example) that are often hidden

[24] See WCAG 3.3.5.
[25] See WCAG 2.2.1 and 2.2.3.
[26] See WCAG 2.2.2.
[27] See WCAG 2.2.4.
[28] See WCAG 2.2.5.
[29] See WCAG 2.2.6.

with deceptive designs that force users to have to opt-out in confusing ways. This may add friction to the user experience, but it is essential to making sure the user is able to act in their own best interest. Choosing to opt out should carry minimal penalty, and the consequences of a choice should be knowable in advance as well as reversible.

Tips for evaluation - there is a lot to consider here! [30]

- Walk people through your data policy when they sign up or start using an app.
- Use opt-ins instead of opt-outs. Don't pre-check the consent box.
- Consider letting the site visitors or app users choose what data they want to share.
- Don't use broad, vague language to bully people into consent. For example, when people choose to withdraw their consent for use of their data, state clearly how that will impact site services or functionality.
- In e-commerce, avoid pre-adding items to a shopping basket or otherwise include purchases in a sneaky way. Avoid setting up recurring purchases automatically.
- Don't make the process for unsubscribing significantly harder or longer than the process for subscribing.
- Use legible fonts and clear displays when asking for consent or provide important information.
- Avoid using colors to trick people into making mistakes or selecting options that are against their own best interest, such as using green in contexts where "green" means "go" and "red" means "stop.".
- Present "yes" and "no" options equally by creating symmetrical designs with the same font, color, and readability.
- Limit the delay between the time the user selects a choice to when they confirm, without adding unnecessary steps to apply guilt and fear of missing out, such as "Are you sure? We hate to see you go."
- Don't bury key information that affects an individual's personal data or opt-out process in the long scrolls of a privacy / personal data policy.
- Lay out a simple and intuitive process for people to opt-in and opt-out as they prefer
- Ensure the process and design for deleting accounts or restricting access to data makes those choices easy to see and simple to accomplish.

Related Accessibility Guidelines to follow:

- Use text and not only sensory characteristics such as shape, color, size, visual location, orientation, or sound.[31]

[30] Evaluators should be considering user consent through opt-in or opt-out in the context of privacy regulations like GDPR and CCPA, in order to have a fully compliant understanding of the implications for the user's experience; e.g. GDPR's Article 7(3) is explicit about the right to opt-out, and the CCPA's Section 1798.110 requires businesses to provide California residents with the right to opt-out with a clear conspicuous link on their homepage titled "*Do Not Sell My Personal Information*".

[31] See WCAG 1.3.3.

#9: Emotionally appropriate tone

The visual and verbal tone of an interaction should match the content of the interaction. Sometimes a tool should seem playful, sometimes it should seem highly technical, sometimes it should seem very serious. This is especially true in error states, where tone helps communicate the severity of the error, and in areas involving privacy or security, where the wrong tone can cause serious mistakes. Be sure to consider accessibility if you use multimedia content.

Tips for evaluation

- Avoid using language that creates a false sense of urgency or necessity.

Related Accessibility Guidelines to follow:

- Offer alternatives like transcripts to describe the multimedia or offer an audio track for the Video-only content.[32]
- Offer Captions for the videos[33]
- Offer Audio Description or transcript for the videos[34]
- Offer Sign Language for the videos[35]
- Avoid content that can cause seizures or physical reactions.[36]

[32] See WCAG 1.2.1.
[33] See WCAG 1.2.2 and 1.2.4.
[34] See WCAG 1.2.3, 1.2.5, 1.2.7, 1.2.8, and 1.2.9.
[35] See WCAG 1.2.6.
[36] See WCAG 2.3.1, 2.3.2, and 2.3.3.

Additional accessibility considerations to keep in mind:

- Avoid color being used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.[37]
- Make sure the content is operable through a keyboard.[38]
- Ensure keyboard navigation is visible and in the right order.[39]
- Reduce accidental activation if keyboard shortcuts are offered.[40]
- Allow users to reach the main content quickly and easily.[41]
- Use descriptive titles and links.[42]
- Use descriptive headings and labels.[43]
- Use headings to organize the content.[44]
- Make it possible for users to locate content (for example, a search mechanism).[45]
- Offer information about the user location (for example, a breadcrumb trail).[46]
- Let users operate with a single pointer or control.[47]
- Ensure the visual label for controls is a trigger for speech activation.[48]
- Allow users to choose different ways of inputting content.[49]

[37] See WCAG 1.4.1.
[38] See WCAG 2.1.1, 2.1.2, and 2.1.3.
[39] See WCAG 2.4.3 and 2.4.7.
[40] See WCAG 2.1.4.
[41] See WCAG 2.4.1.
[42] See WCAG 2.4.2, 2.4.4, and 2.4.9.
[43] See WCAG 2.4.6.
[44] See WCAG 2.4.10.
[45] See WCAG 2.4.5.
[46] See WCAG 2.4.8.
[47] See WCAG 2.5.1, 2.5.2, and 2.5.4.
[48] See WCAG 2.5.3.
[49] See WCAG 2.5.6.