# Writing Testable WCAG 3.0 Outcomes (Draft Revision)

(Approved by AGWG in January 2022. It was revised to meet the maturity level writing process in January of 2024. )

## Introduction

Writing testable outcomes is a highly iterative process. It may take dozens of iterations before an outcome is good enough, and the end result will likely not look like the original proposal. It is important to know that going in, and be prepared for this process to take many months, or even years.

This document outlines a methodology for writing outcomes for WCAG 3.0. It was created by the [Test Reliability](https://www.w3.org/WAI/GL/task-forces/silver/wiki/Test_Reliability) subgroup in 2022 and revised by former members of the subgroup in 2024. It is designed to create outcomes (and supporting methods) for WCAG 3.0 that are acceptable to organizations that may need to apply and enforce WCAG 3.0.

For the purpose of WCAG 3.0, a testable outcome is a requirement that **strives** for the following qualities:

* It addresses a clear user need
* There is little to no room for interpretation
* It can be tested quickly and inexpensively (possibly using tools)
* It is easy to learn and understand

## Why Testability Matters

Imagine a traffic law that said you're not allowed to drive dangerously fast. Your impression of what that means may be very different from that of the police. You can spend some time debating it by the side of the road, have it escalate to a court, then to a higher court, spending hundreds, if not thousands of hours arguing back and forth about what levels of danger are acceptable, how that is impacted by factors like quality of the road, the vehicle involved, driver skill, weather conditions, etc. Instead, many roads just have a speed limit.

Speed limits don't account for road condition, visibility, driver skill, how busy it is on the road, the safety of the vehicle you're driving, etc. In terms of optimally balancing traffic safety and road throughput, speed limits are terrible. But what they do have going for them is that it is really easy to test, both for the driver and for the enforcement agencies.

WCAG's strong emphasis on testability is part of what makes it so appealing for organizations to adopt. The key is to ensure that content producers can know whether or not they meet the expectation of those asking for WCAG conform content, without requiring extensive conversations about what different requirements actually mean and how those would have to be applied in different scenarios.

## Getting to the Exploratory Stage - Identifying User Needs and Outcomes

The Exploratory stage is the stage where the working group is exploring what direction to take with this outcome. This content is not refined, details and definitions may be missing. The Exploratory stage is focused on identifying the user needs of a topic based on the available research, sketching how to test if the user need has been met, and writing a preliminary outcome in response to that user need. Feedback should be about the proposed direction.

AGWG worked on getting all the proposed guidelines to the Exploratory stage in Q3 to Q4 2024. All of the existing Exploratory documents are linked in the guideline wiki page and can be accessed from the [Guideline List](https://github.com/w3c/wcag3/wiki/Guideline-List#guideline-list).

As part of the process, each guideline has a “scratchpad” document which lists the research, user needs, a sketch of the test, and the proposed early draft of the Outcome. If you are starting new Exploratory work, the documents that you will need are listed below.

Supporting documents:

* [Exploratory Guideline Template](https://docs.google.com/document/d/1GitKyTmNFiEmrN3Yogs81ZFzPMEERO2-7dTI9LKdwus/edit#heading=h.rpm5t115dztz) - Scratchpad template used in Q3 and Q4 of 2024
* [Identify User Needs through Research](https://docs.google.com/document/d/1GitKyTmNFiEmrN3Yogs81ZFzPMEERO2-7dTI9LKdwus/edit#heading=h.2xka14wsdfdo) - the section of the Scratchpad with instructions to identify the user needs
* [Template for Pre-Pull Request in Scratchpad template](https://docs.google.com/document/d/1GitKyTmNFiEmrN3Yogs81ZFzPMEERO2-7dTI9LKdwus/edit#heading=h.12gt6pip1eq0) - the section of the Scratchpad where you organize all the work in preparation for creating a Github pull request.

## Getting to the Developing Stage - Writing Initial Testable Outcomes

In the Developing stage, AGWG is in rough agreement on the direction of the outcome, although not all high-level concerns have been settled. The focus of the Developing stage is examining and recording the examples of how the outcome is met (or not) and the edge cases where there is ambiguity, conflict, or examples don’t quite fit. Early drafts of definitions of terms, tests, and assertions are part of this stage. The outcome goes through multiple iterations of wording as it gets refined based on the examples and edge cases.

Details have been filled in, but all issues are yet to be worked out. Feedback should be focused on ensuring the sections are usable and reasonable in a broad sense and include details such as

* Additional examples / edge cases.
* Categorization of particular examples.
* Gathering additional research related to the outcome

.

### Step 1: Review prior work and list user needs

Each outcome should have a scratchpad which includes research and user needs. Review the prior work and pull out the list of relevant user needs. There is a scratchpad for each guideline which can be accessed from the individual guideline page of the WCAG 3 Github wiki [Guideline List](https://github.com/w3c/wcag3/wiki/Guideline-List#guideline-list). If missing or incomplete, draft user needs. Review the current [Guideline and How to Template](https://docs.google.com/document/d/1Smly4XDxfzfXHa7AoUxoLXLy_3PdOXMkh0ZwtgksSPk/).

### Step 2: Come up with examples and edge cases

To build a shared understanding of what should and what should not be considered accessible, having a lot of examples is important. The examples need to be categorized, distinguishing between what passes and fails. If there are different levels of accessibility, those categories need to be broken down further. For example, separate very accessible from moderately accessible, and minor inconveniences from major and critical issues. Identify any critical errors where a user with a disability is blocked or harmed.

Add a brief description to each example, explaining why it is categorized the way it is. Most likely there are other possible categories too, such as “interactive image” and “decorative images”. Adding those can help create greater clarity later in the process. Each example should add something new to the list, and provide something new that can be learned from it.

Edge cases are the most interesting at this stage. Find examples that help discuss where to draw the line between passing and failing. Be careful to focus on plausible edge case examples. It is pretty easy to get sidetracked and come up with hypothetical problems that wouldn’t occur in real life.

Draft tests and assertions that could be used to evaluate each example and edge case.

* **Tests** are verifiable statements that allow testers to reliably determine if the content being evaluated satisfies the user needs identified in the Guideline.
* **Assertions** are attributable statements by a person or organization that they followed a procedure to improve accessibility.

Start by determining what portion of the example or edge case is quantitative and which is qualitative. Write a draft of the steps you would take to evaluate the example or edge case. How an example or edge case can be tested may vary by technology, in which case, record the technology (for example, web, iOS mobile, VR) that you are discussing.

This becomes a more complex problem when you are evaluating examples that rely on qualitative evaluation. You may be able to set break points on a scale of “how much is enough” or you may have to say, “if you follow this process, you will get toward better accessibility for this outcome.” See the section following on [Handling Ambiguity and Subjectivity](#_3vynaf63wngy). Determining a process to follow is (in WCAG 3 terms) creating an assertion. You can find more information in the [Assertions section](#_18tc5scae1y1).

Use the Developing Guideline Outcome Template to

* Capture sorted examples and edge cases
* Capture draft tests and assertions
* Capture and refine user needs

### Step 3: Create a rough outline of the outcome

With a new (or updated) list of examples, write the outcome in such a way that the categorization of all examples becomes clear. Passing examples must pass for some reason described in the outcome, similarly for failure examples. If some kind of level exists in the categorization, write an outcome for each level and avoid overlap between them.

Avoid discussions on readability, complexity or definitions. The focus at this stage should be on ensuring all the possible edge cases are described in the text of the outcome. Don’t feel compelled to write the outcome down in a single sentence. Outcomes do not need to be short. If more words are needed for a proper description, use them. In other words, WCAG 3 Outcomes do not need to look like WCAG 2 Success Criteria.

If you haven’t already, come up with a short title for the Outcome, so we have a name to talk about it.

Add the rough outline of the outcome to the Developing Guideline Template.

### Step 5: Define the terms used in the rough outline

This stage is focused on reducing ambiguity and subjectivity. Go through the outcome text word by word, and discuss different ways these terms can be understood and tested. See the section on [Handling ambiguity and subjectivity](#_wpuyj02kuf4h) for details.

Any word or phrase that plausibly could be misunderstood, or which can be explained to mean different things, needs to be either replaced, further specified, or defined. If at all possible, replace or define qualitative aspects with quantitative ones. Where this is not possible, ensure qualitative aspects are narrow and specific. Avoid qualitative aspects that depend on personal preferences or prior knowledge or experiences.

When using definitions, prefer existing definitions over new ones. When creating new definitions, do not redefine terms defined in other standards. Definition terms should be intuitive, so that the meaning of a definition could reasonably be guessed from its term. I.e. definition terms should not have unintuitive or surprising meanings. It also follows that if no intuitive term exists, that a definition is not appropriate.

Starting points for existing definitions:

* [WAI Glossary](https://www.w3.org/2021/09/draft-wai-glossary)
* [EN 301 549](https://www.etsi.org/deliver/etsi_en/301500_301599/301549/03.02.01_60/en_301549v030201p.pdf#page15)
* [Section 508's definitions](https://www.access-board.gov/ict/#E103-definitions)

Add the rough outline of the outcome to the Developing Guideline Outcome Template.

### Step 6: Simplify and Iterate

At this point, the outcome text may have gotten fairly complicated. There are several things that can be done to make an outcome easier to understand, and easier to test.

1. Take out anything that isn’t strictly necessary. If there are any words or phrases that can be removed without changing the meaning, take them out. Clarifications and examples should be in additional documentation such as notes, not the outcomes themselves.
2. Consider if the outcome can be split up. As a rule of thumb, if an outcome can be split up without increasing its complexity, it should be. Do try to avoid duplicating issues, so that each type of issue fits under only one outcome.
3. Based on changes, consider whether you need to go back to a previous step.

### Step 7: Get feedback, and iterate

Create a pull request.

The first line of feedback should come from AGWG. The first few iterations will likely come from them. Once AG has accepted the outcome for the working draft, public feedback comes in. Feedback can take a number of forms. At the developing stage, focus on requesting feedback on:

* Additional examples / edge cases.
* Categorization of particular examples.
* Gathering additional research related to the outcome

Once feedback has been gathered, the process may start again or continue onto the refining stage.

## Getting to the Refining Stage - Refining the Testable Outcome

The Refining stage is where the working group reaches consensus on this outcome. It is ready for broad public review and experimental adoption. There should be public feedback from the Developing stage to be considered or incorporated. The work focuses on completing the informative support information (the howto and methods), and adjusting the outcome for plain language and greater clarity and iterating through the simplification step. Requested feedback for the Refining stage should be focused on the feasibility and implementability.

### Step 1: Review feedback

Issues will be filed against the outcome in GitHub. Review these issues and based on the feedback, go back to earlier stages and complete the following:

* Add user needs and research
* Add or update examples and edge cases
* Update the outcome text
* Add or update definitions
* Fix editorial problems

For details, see steps in the defining stage. Once you have addressed the feedback, move onto step 2.

### Step 2: Simplify where you can

At this point, the outcome text may have gotten fairly complicated. There are several things that can be done to make an outcome easier to understand, and easier to test.

1. Take out anything that isn’t strictly necessary. If there are any words or phrases that can be removed without changing the meaning, take them out. Clarifications and examples should be in additional documentation such as notes, not the outcomes themselves.
2. Consider if the outcome can be split up. As a rule of thumb, if an outcome can be split up without increasing its complexity, it should be. Do try to avoid duplicating issues, so that each type of issue fits under only one outcome.
3. Remove any nuance that is not critical to the outcome. An outcome that is easier to understand, and easier to test is better than one that covers all possible edge cases. See: [How nuanced should an outcome be?](#_wpuyj02kuf4h)
4. Find an easy to scan presentation for the outcome text. A paragraph of text is harder to comprehend than a short text, followed by a bulleted list. Labeling each item on the list makes them even easier to scan, but only do this if meaningful short labels exist.
5. Carefully consider what could be automated. If there are words or phrases that can be removed or improved so that more of the outcome can be tested automatically, this should be done as long as it still meets the user need. The ability to automate should not override the user need if the user need can only be addressed by manual testing.

As part of simplification, examples may need to be reclassified. When doing this, document that this happened, and the motivation for reclassifying examples as an editor's note with the example.

### Step 3: Complete the How to & methods

Once there is draft text for the outcome, the How To document, and methods can be created. Writing these should happen before the feedback step, so that reviewers have the extra context they need about why the outcome is the way it is, and what that outcome means for specific technologies.

It is recommended to start with methods for HTML. If multiple methods are possible, it is acceptable to completely write the first method, and have placeholder content for other methods. In writing methods and how-to’s, follow the recommendations of its authoring guides. The examples and edge cases you wrote in Step 2 are the starting point for writing the examples and expectations.

* [Revised Method Template](https://docs.google.com/document/d/1l8iM8QSfzSkeRwJ6X9iArakrlYRc7oPKVfg1GxvCJ6Y/)
* [Guideline and How-To Template](https://docs.google.com/document/d/1Smly4XDxfzfXHa7AoUxoLXLy_3PdOXMkh0ZwtgksSPk)

As you complete the How to and methods, you may need to go back to an earlier step. That is a normal part of an iterative process.

### Step 4: Get feedback, and iterate

The first line of feedback should come from AGWG. The first few iterations will likely come from them. Once AG has accepted the outcome for the working draft, public feedback comes in. Feedback can take a number of forms. At the refining stage, focus on requesting feedback on:

* Additional examples / edge cases
* Feedback on content and readability of How To documents
* Feedback on wording
* Challenges with implementation

Once feedback has been gathered, the process may start again or continue onto the mature stage.

## Getting to the Mature Stage - Wrapping up

In the Mature stage the work is focused on finalizing the content of the outcome. This includes reviewing public feedback and finding implementations of the Outcomes. In this stage you hope to catch any edge cases or examples that were not previously identified.

### Step 1: Review feedback

Issues will be filed against the outcome in GitHub. Review these issues and based on the feedback, go back to earlier stages and complete the following:

### Add user needs and research

* Add and update examples and edge cases
* Update outcome text
* Update How Tos and Methods
* Add or update definitions
* Fix editorial problems

Step 2: Find Implementations

Finding publicly available sites, platforms, and assistive technology that have implemented the recommendations successfully is an important part of validating guidance.

Locate 3-5 sites that demonstrate a successful implementation of the guidance.

Consider how many of the most visited websites have or easily could implement the guidance. See [Wikipedia](https://en.wikipedia.org/wiki/List_of_most-visited_websites) or another source for a listing.

Based on the results, consider if any changes to the guidance is needed.

## Testing Qualitative Aspects

WCAG 3 tries to take the qualitative question of “How accessible is it?” and turns it into a quantitative measurement. In doing that, it takes an unbound quality like “accessibility” and constrains it in a way that can be quantified. This makes WCAG testing predictable. Unlike usability testing with people with disabilities, when testing WCAG there should be no surprise outcomes that pop out during testing.

The key to writing good outcomes for WCAG is to take qualitative aspects and quantify them, where possible. This is a matter of degrees (See [Handling ambiguity and subjectivity](#_wpuyj02kuf4h)). Only those aspects that lead to a lot of disagreement in interpretation need to be qualified. For example, testing if something is easy to understand will lead to far greater disagreement than whether or not a heading is descriptive.

Qualitative aspects that lead to interpretation differences need to be redefined in terms of more quantifiable aspects. That’s what WCAG itself does, redefining “accessible” as “conform to WCAG”. Outcomes within WCAG work that way too. A good example is legibility of text. Rather than require text to be legible, WCAG requires the foreground and background colors to contrast sufficiently. Thus taking a qualitative aspect and redefining it in a way that can be quantified.

There are no shortcuts available here. An outcome that leaves a lot of room for interpretation is simply incomplete. It cannot be used as a way to communicate requirements between organizations without additional documentation describing which interpretation to use.

## Handling Ambiguity and Subjectivity

It is helpful to work through this to separate these two problems. Something is ambiguous when there is more than one way to understand it. An outcome is subjective when there is a qualitative aspect to it.

### Ambiguity

An outcome is ambiguous when there is more than one way to explain what it means. An example of ambiguity is the word “heading”. Someone might take that to mean anything that is visually presented as a heading. Another person might decide that something is a heading because of its markup, or because of the function of that text in a page.

Because outcomes are not technology-specific, some level of ambiguity is unavoidable. Something called a “description” in one technology, may be called a “help message” in another. Interpreting technology-agnostic language of WCAG for specific-technologies like HTML and PDF requires some “translation”. Methods help with this, and ideally provide technology-specific definitions for technology-agnostic definitions of WCAG 3. This necessary “translation” is the only acceptable reason why an outcome may be ambiguous, and should therefore not persist in its methods.

There are a few ways to reduce ambiguity;

1. Replace the ambiguous word or phrase with something unambiguous
2. Remove wording that relies on individual perspective and experience, such as “according to users” or “commonly understood”.
3. Add a specifying word or phrase. For example, change “heading” to “semantic heading”.
4. Create a definition. For terms used only once this should be an inline definition. An inline definition is one where a term is used in a sentence, and its definition is used in the sentence following it. (As an example, spot the inline definition in this bullet).

Ambiguous terms can often be spotted by asking “who, what, how, where, when, why” questions. As an example, the phrase “perceived by users as a single control” is ambiguous, because it does not answer which users, and how many of them? What a user perceives as a single control can change substantially depending on if they use a pointer device, keyboard, screen reader, etc. Asking and providing answers to these questions is crucial to writing testable outcomes.

### Subjectivity

Subjectivity is different from ambiguity, in that an ambiguous outcome is one where the meaning can be disputed, a subjective outcome is one where the degree to which it is met can be a matter of opinion. For example, “The title describes the topic or purpose of the page” involves deciding if what is in the title closely enough matches what is on the page that it can be said to “describe” it.

Put in other words, an outcome is subjective when there is a qualitative aspect to it. Almost all accessibility testing has a subjective component to it. The key then is to only use subjective terms that leaves relatively little disagreement. There are a few strategies that can be used to reduce the subjectivity of an outcome:

1. Substitute terms that have a broad meaning for one or more terms that have a narrower meaning. For example replacing “open” with “expanded”, or replacing “descriptive” with “describing the topic or purpose”.
2. Break down the outcome into multiple separate outcomes. For example, rather than test all non-text content, separate outcomes could be created for visual and auditory content.
3. Adopt a rubric or an adjective rating. This helps particularly when there are a lot of “almost but not quite” and “passes, but barely” results in a particular test. Give specific bands that clearly define or show what goes in each adjective rating.
4. Use an assertion instead of a test to state that a process which would likely improve accessibility was completed. Assertions are evaluated based on whether the assertion itself meets requirements and do not require repeatable results from the process completed. See the section immediately following for more details on Assertions.

### Com

### Tracking Assertions

*See* [Assertions and Procedures](https://docs.google.com/document/d/10Rl3YJhc4kTyaQwBoQYFp-YvFRK_TdBKkA3kLQKHYwk/edit) *for the source of much of the following material*.

Assertions are new to WCAG 3. They are a way to address accessibility user needs that cannot be evaluated using a consistently repeatable test. An assertion is a formal claim of fact, attributed to a person or organization. In WCAG 3, an assertion is an attributable and documented statement of fact regarding procedures practiced in the development and maintenance of the content or product to improve accessibility. Assertions are only testable in that one can test that the assertion has been made correctly - not that any desired result has occurred. The results are always true/false.

Assertions are a statement that an organization followed a specific process in order to improve the accessibility or address a user need. An example of an assertion would be for an outcome related to using plain language. The early draft of an assertion might be: “An editor with documented training in plain language writing edited the content, making corrections to improve plain language, which were implemented.”

As the assertion is developed, look for standards or regulations that can be followed, like ISO or PlainLanguage.gov. However, we recognize that many Outcomes will require assertions that are written by the group.

Examples of procedures that may be addressed by assertions used during implementation might include:

* Training
* FTE (Full Time Equivalent) assignments
* Skills testing
* Coordination and documentation of accessibility processes
* Setting the priority for remediation

Examples of procedures that evaluate accessibility might include:

* Usability testing
* Heuristic evaluation
* Screen reader testing

Assertions should be documented as part of the conformance claim process. The required information may also be made available through the web site. Assertions include the following information:

* The statement being asserted
* The date of the assertion
* The date or date range the procedure was completed
* The scope of the assertion
* Contact information for the person or group making the assertion
* The outcome(s) or guideline(s) supported by the assertion

## Soundness of Outcomes

Accessibility solutions such as resizable user interfaces, accessibility trees, and video description can take years to develop. Along the way, there are many failures and dead ends. Eventually, these solutions come to sort of a natural conclusion. By that point, the edge cases and limitations of the solution will be properly understood. Creating outcomes based on well established accessibility solutions is fairly straightforward. However, doing so for newer accessibility solutions can be far more challenging.

Ideally, accessibility solutions are given time to develop and mature before they are codified in a standard. Premature standardization has two problems. Firstly, that the solution might have unintended consequences that can make accessibility worse, rather than improve it. And secondly, once a solution is standardized, any improvements that are made to it can take years before they get adopted, since it requires the creation of a new standard.

To minimize those risks when writing outcomes based on new accessibility solutions, the following questions should be discussed, and their answers documented:

1. How well does the solution address the user’s needs? What research exists to prove the soundness of the solution? What research exists that might contradict this?
2. What alternative solutions were considered? Why was the chosen solution used, rather than any of the alternatives?
3. What is the chance that in five or ten years new insights or new technologies will arrive that make the proposed solution obsolete or ineffective?
4. To what extent is the solution universally applicable? Does it work for different languages and cultures? Does it work for different types of content, and on different devices? Is it achievable for organisations of varying sizes?
5. In what ways might the solution hinder other accessibility needs?

## How Nuanced Should Outcomes Be?

If we let them, outcomes can become very complicated and nuanced. There are exceptions to every rule, and to every exception is another exception. This creates a tension, that comes from three different requirements:

1. Outcomes have to address the needs of people with disabilities.
2. Outcomes have to apply to all different types of content in scope of WCAG 3.
3. Outcomes have to be easy to learn and apply by content authors, testers, policy makers, legal experts, etc.

Each of these requirements pull the outcome into a different direction. There is no “right” way to balance these three. Instead, each outcome will be iterated over many times, until a point is reached where there are few, or no objections anymore to the outcome as written. How this works out is fairly unpredictable, and depends in large part on who is contributing to the conversation.

Even for outcomes based on existing WCAG 2.x success criteria, it is likely that this negotiation process will need to happen. While this may seem repetitive, going through this process is important in building broad consensus, both within and outside the W3C.

## Glossary