COGA’s Feedback on Collaboration Tools Accessibility User requirements (CTAUR)

This documents starts with a list of COGA’s suggestions and then has the same comment inline

List of suggestions from coga

Suggestion 1

Add the following user need and requirements

Suggested User need **20:** Users with learning or cognitive disabilities or who use assistive technologies need to be able t**o use efficiently,** including software with without getting disorientated and confused.

- REQ (requirement/pattern) 20 A: Keep it simple with as few steps as possible. Keep key processes and workflow as clear and simple as possible. This includes: editing, adding comments, viewing changes and comments and saving your work. Have short critical paths with as few steps as possible.
- REQ (requirement/pattern) 20 B: , Do not require the user to open multiple tabs, windows or panels to complete a single task. Remembering and navigating through this content can impossible or disorientate the user. For example, for some github editing, to review the changes inline one needs to do a pull request and then generate a version for people to review.

Suggestion 2

Add the following user need and requirements

Suggested User need 21: As a user who finds some web sites hard to use and struggles with remembering and following instructions, I sometimes need in-page instructions so that I can do the correct task.

- REQ (requirement/pattern) 21 A: Instructions from the document owner can be placed in a reserved consistent location that is easy to find.
- REQ (requirement/pattern) 21 B: Context sensitive help is available for tasks, workflow and any non-standard elements and support information for screen reader capabilities.

Suggestion 3

We have problems with requirements for user need 5: as we feel the requirements suggested would be confusing rather than helpful. For REQ 5A, I would turn it off and then forget to turn it on again and work on an old version.

For REQ 5B Personally, I would reload the document as I would assume it was just not working well

Text follows:

User Need 5: Users with vision, cognitive or physical disabilities need to be able to **edit content without the distraction** of changes introduced by collaborators.

- **REQ (requirement/pattern) 5A:** Provide a mode of operation in which changes made by collaborators are not displayed while the user is editing, or
- **REQ (requirement/pattern) 5B:** Provide a mode of operation in which the component of the content that the user is editing (e.g., the paragraph, section, semantic unit of source code, or graphical object) can only be changed by one collaborator at a time, preventing others from making simultaneous modifications to the same component.

Suggestion 4

A wysiwyg should be available for editing . Add that editing to not require multiple screens or multi step. For example I can add content and see its effect. without toggling tabs or screen.

Suggestion 5

Regarding User Need 11. Add to the requirement that it should be clear to the viewer WHO made suggested changes and WHEN and the option to note WHY the change.

Suggestion 6

add to 11 - version control

Original text:) Assistive technology users need to be able to read the text with information included about suggested changes (i.e., insertions, deletions or formatting modifications proposed by collaborators).

Suggestion: add that people with cognitive disabilities also need to

- see these changes,

- hide the changes, and
- easily get the latest clean version

Suggestion 7

We need it to be easy to see changes made.

For example, github diffs are impossible to track without a visual memory, as you jump between the sections and need to reoriented yourself to see if it makes sense. Without a good memory where we can @see@ the old bit, we get lost. As coga editors- we change into googledocs, copy it over, and use suggestions mode which puts the changes in green.

Suggestion add the following requirement

- REQ (requirement/pattern) 20C: Enable a mode for inline changes to be seen visibly such as color changes strike though etc. Changes should be visually marked and easy to reach for people with impaired visual and working memory, low vision users for Assistive technologies. Having the tags for start and end point would be helpful with a high contrast icon, for low vision users. Note that looking and comparing sections of a screen will not work for many users. For example, looking at diffs in different sections, users with impaired working memory, comparing part of a screen or across screens is not doable.

Suggestion 8

In user need 12, add this is also for also with vision loss such as with the aging

Suggestion 9

In User Need 14:
Add that when AI is used it needs to be marked as AI or that it is automatically generated. AI often makes mistakes and coga users can be confused by incorrect information. In fact, no information is better than incorrect information. However tagging automatically generated is a good compromise.

Suggestion 10

Add that the summary needs to contain the context, such as who made the changes and when. AI might summarize the document. But we need to know the changes, and the context of the changes. For example, this contains john’s changes to section of annotation. In another example a good summary may be “this is our plain language review”.

Suggestion 11

Original text: REQ (requirement/pattern) 16B: If e-mail or a similar medium is used to deliver notifications, ensure that the subject of the message clearly specifies the project, document or issue relevant to the notification.

We do not like this requirement at all. do not create a wall of text before giving us the label that means something to us. that is not a document name!

For example I get many emails like “issue github 1073 has been changed”.
Or ”the document revving cogas comments as a new comment”.
Assume I can only open 10 of these and assume I get 60. How do I know what to open? If one is important, but I can not deal with it now, how will I find it again?

Compair with a subject line like “assistive technology and coga -github issue by phill jenkins, has a new comment from city mouse” This is trackable and understandable. Here, the first thing is the meaningful label.

Suggestion, make short meaningful subjects that summarize the main point in the message and not the document name or auto generated tags that can apply to multiple messages. The subject / label/ header need to change when new issues or topics start to get discussed so I can find the new topic.

Suggestion 12

To requirement 16 C

The grouped threads often repeat the original message which is hard for slow readers. It can be very hard to tell what I have read already. This is often why I lose threads or emails, and the grouping often repeats the original message so I can not tell what I have read already.

Suggestion: Make a clear distinction between new and old parts of threads, without repeating them, so the user can track what they have already read. Use headings where possible.
The subject / label/ header needs to change when issues start to get discussed so that the user can find the new topic and track what they have read.

Suggestion 13

Original text: User Need 19: Users with learning or cognitive disabilities or who use assistive technologies need to be able t**o learn collaboration tools efficiently,** including software with which they are unfamiliar and which is necessary to the task at hand.

Suggestion: Replace the word “to learn” with “familiar with”. Learning is often impossible such as with early stage dementia as MCI. With a familiar interface learning is not net needed!

Suggestion 14

Add requirement

REQ (requirement/pattern) 19 D: When new processes or designs are added, keep them as familiar to the user as possible. Always allow users to roll back to an older interface where they exist. This allows older users to continue being productive.

Suggestion 15

Original text: REQ (requirement/pattern) 19A: In implementing collaboration features, follow established user interface conventions and design patterns. For example, use conventional terminology, labels, or icons for functionality that may be familiar to users.

For example a person with MCI may be unable to learn new infaces but can retain old skills and knowledge.

Suggestion: Add the term “ familiar terms “ to the requirement . as :

- **REQ (requirement/pattern) 19A:** In implementing collaboration features, follow established user interface conventions, **familiar terms, and familiar** design patterns

Suggestion 16

Change to formatting: Use embedded list structure as bellow in the document so that the semantics show what requirement is under what user need.

I have formatted the document bellow so you can see what we meen. and to make it easier to follow so we can review it.

1. user needs distinguishable from requirements (The user needs are not bulleted). So you can jump from user need to user need, and see to which user need each REQ (requirement/pattern)uirement belongs to.

2. Key works in userneeds in bold
3 Explained what req is

_end of list -----

Intro for coga

The latest version of Collaboration Tools Accessibility User Requirement (CTAUR) is ready for a further review. This is the planned last draft before it is published as a W3C Group Note.

Related announcement here:
https://lists.w3.org/Archives/Public/public-wai-announce/2024JulSep/0001.html

COGA has already provided substantial feedback on previous versions of CTAUR, which led to the addition of significant requirements and clarification of the document's scope. As a result, the document should now be fairly aligned with COGA's objectives.

The latest version can be found here: https://www.w3.org/TR/ctaur/ However, I have pasted the text of the CTAUR document below. If you have any further feedback on this latest version of CTAUR, please leave it below, either using the comment facility, or as inline text (but please clearly indicate where your comment starts and ends).

The Research Questions Task Force specifically Requirement comments on the following issues and questions:

1. We came to understand the collaborative editing environment in terms of managing complexity. We observed that many word processing, spread sheet, software development, and media development environments are themselves intrinsically complex. To this, collaborative tooling adds a further layer of complexity: the management of proposed, accepted, and rejected edits from multiple participants. Does this framing make sense? Is its importance clearly communicated by the document?
2. Do we delineate between the content creation elements of software and those relating to managing collaboration sufficiently? Is the distinction meaningfully communicated? Do you agree with this scoping?
3. We inserted a section in our Introduction on Social Considerations. This brief section is included to communicate which stakeholders we regard responsible for which aspects of collaborative efforts. Is this helpful?

4. We created a glossary to define the term "WYSIWYG" in response to a comment. Are there other terms we use you would like defined in the glossary?

We will then discuss the feedback in the **first week of October**.

Lisa: I put my comments at the end and reformatted the document to make it easier to follow) press COGA’s Feedback on CTAUR (August 2024) to jump to them.

I suggest we review the document compair to what we think is missing.
Also should we make comments separately?

Collaboration Tools Accessibility User requirements

W3C Group Draft Note 10 July 2024

More details about this document
This version:
https://www.w3.org/TR/2024/DNOTE-ctaur-20240710/
Latest published version:
https://www.w3.org/TR/ctaur/
Latest editor's draft:
https://w3c.github.io/ctaur/
History:
https://www.w3.org/standards/history/ctaur/
Commit history
Editors:
Jason White
Janina Sajka
Scott Hollier (Centre for Accessibility Australia)
Feedback:
GitHub w3c/ctaur (pull REQ (requirement/pattern)uests, new issue, open issues)

Abstract

This document outlines various accessibility-related user needs and requirements for both synchronous and asynchronous web-based collaboration tools based on various collaborative engagement scenarios. These tools typically include one or more specific collaborative features such as content editing by multiple authors, support for comments annotations, and revision control in real-time synchronous sessions, or asynchronously. Asynchronous tools are more commonly known as revision control systems rather than collaboration tools though their functionality is otherwise very similar. The Cloud-based office application suites from Google and Microsoft are well-known examples of synchronous, real-time collaboration tools, though they also support asynchronous collaboration. Web tools built on git are well-known example of asynchronous collaboration tools.

The accessibility user needs and requirements described in this document may be implemented in the collaboration tool itself, or in an assistive technology application such as a screen reader

or screen magnifier. Widely used desktop applications such as word processors and spread sheets also commonly support collaboration features. We take a holistic approach to give foremost priority to the user's perspective, leading to the identification of features and solutions that may be implemented by different components of the software stack involved in performing a collaborative task.

Although the user needs and requirements identified in this document *are non-normative*, they may influence the development of future accessibility guidelines, normative technical specifications, or features of collaboration tools and assistive technologies. They are relevant to software developers who contribute to the collaborative experience.

Status of This Document

This section describes the status of this document at the time of its publication. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at https://www.w3.org/TR/.

This document was published by the Accessible Platform Architectures Working Group as a Group Draft Note using the Note track.

Group Draft Notes are not endorsed by W3C nor its Members.

This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

The W3C Patent Policy does not carry any licensing requirements or commitments on this document.

This document is governed by the 03 November 2023 W3C Process Document.

Editor's note

Editor's Note: Contributing to this Document

This updated working draft publication is a document intended to become an Accessible Platform Architectures (APA) Note. The intent of this (and all) APA working draft publications is to gain a wider review of its content and solicit feedback on user needs that may have been missed, underrepresented, or sub-optimally described.

APA encourages review and feedback from all accessibility perspectives to ensure future drafts are as comprehensive as possible.

Table of Contents

1. Introduction

1.1 What are collaboration tools?

For the purposes of this document, a collaboration tool is any software that supports features designed to facilitate the interactive creation, editing or annotation of content by multiple contributors, whether working in simultaneous collaboration or asynchronously. Examples of collaboration tools include

- A Web-based text editor or word processor that enables multiple authors to edit content simultaneously while discussing their edits during a Real Time Communications teleconference, with each contributor's changes being integrated into the resulting text and propagated in real time to the collaborators.
- A Web-based text editor or word processor that enables multiple authors to edit content asynchronously over many days and week, with each contributor's changes being integrated into the resulting text and propagated on next load to the collaborators.
- A tool that enables Web pages to be annotated with comments that are automatically made available to other users of the annotation service who access the same pages with suitable software. The software may be included in a user agent, or it may be supplied as an extension.

- An Integrated Development Environment (IDE) that supports the collaborative editing of program source code in real time (or asynchronously).
- A wiki that supports version control, for example by enabling authors to revert to prior versions of a page or to view the differences between two versions.
- A midi or audio editing application that allows real-time synchronous audio content editing, or asynchronous content editing. Collaborators hear edited content on demand during a teleconference editing session or on content refresh when editing asynchronously.

1.2 Distinctive features of collaboration tools

This document focuses primarily on features unique to collaboration tools, rather than features which they share with other Web applications or with application software in general. Indeed, any tool that provides one or more of the features enumerated here may benefit from the user needs and corresponding requirements elaborated in the sections that follow.

The distinctive capabilities of collaboration tools are illustrated by the examples described in the section: 1.1 What are collaboration tools?. It is important to consider how these features are manifested in the tool's user interface. From this perspective, the distinguishing features may be described as follows.

Real-time and asynchronous co-editing
A feature enabling multiple authors to edit the same content simultaneously or over days, weeks months, and years. In synchronous co-editing, the changes introduced by different authors in real-time are combined almost immediately, using algorithms such as operational transformation [concurrency-control]. The combined changes are then made immediately visible in all of the participating authors' editing sessions. The effect is that each author may perceive, in real time,

- Edits proposed by collaborators,
- The location of other editors' focus within content.

Asynchronous edits, on the other hand, are made visible on document reload.
Annotation of content with comments
Some tools enable users to associate comments with parts of the content that is being read or edited. In systems such as word processors, replying to comments is supported, allowing threads of discussion to be associated with parts of a document.
Comparing revisions
Some systems can display the differences between revisions of content for purposes of comparison.
Suggested changes
Some word processors can show changes (insertions, deletions and formatting-related modifications) made by collaborators, which an editor can choose to accept or reject. These revisions are sometimes referred to as suggested changes or as tracked changes. Each change may be accompanied by metadata, for example the identity of the author who made the change, and a time stamp.
Access controls

Some collaborative environments support *access controls*, allowing restrictions to be imposed on modification of part or all of the content. Permission to modify content may be granted on a granular basis to specific individuals or to groups of users. For example, in a collaborative tool for creating fillable forms, some users may only be allowed to change the values of input fields (i.e., to complete a form), whereas others may be free to edit any aspect of the document, including the addition, deletion and rearrangement of form fields.

Collaboration tools differ widely in the nature of content that may be edited. They also differ widely in the user interfaces presented to users. For example:

- Word processors typically provide a what you see is what you get (wysiwyg) interface based on a rendered view of the content.
- Editors designed for source code or markup language text development do not provide a rendered view. In these applications,, indentation and syntax highlighting may be the only visual cues provided to the structure of the code or markup available in the editing environment.
- In collaborative video editors, candidate video renditions may be played in side by side windows for purposes of comparison and editing decisioning.
- By their very nature, audio alternatives presented in music or sound editing environments must be played serially, and their distinctions remembered by the user in the comparison and decisioning process.

As the preceding cases suggest, collaboration tools are not restricted by the kind of content that may be edited. Thus, tools that support editing of static images, mathematical notation, or other content types are also within the scope of this document. However, only the collaboration-related aspects of such systems are addressed here. The accessibility issues arising from creating and editing different types of content are not considered in this document, as they are separate problems from the user needs associated with the collaborative features of the tools.

Nevertheless it is important that collaborative tools support the full range of editing functions associated with making web content accessible. Among others this would include the ability to add headings, provide alternative text for images and add captions to videos.

Some collaboration tools support accessibility by mapping unique keyboard commands. Some also organize their feature options in unique menus or uniquely located menus. We prefer collaboration tools that utilize standard menu organization and typical keyboard commands now well known to users from the stand-alone desktop environment. Standard controls REQ (requirement/pattern)uire far less learning from the user, whereas specific accessibility modes with custom keyboard commands, and with menus that shift their location on screen pose significantly steep learning challenges to most users with disabilities, not just users with cognitive and learning disabilities.

1.3 Defining User needs

Specific user needs are fREQ (requirement/pattern)uently defined both by task REQ (requirement/pattern)uired to achieve a particular goal and also by environmental conditions. Context matters. For example, the cognitive demands imposed by interacting with the collaboration-related features of an application depend not only on the needs and capabilities of the user, including the possible presence of assistive technology, but also on the context. A collaborative task that the user can perform independently while working alone in a distraction-free environment may quickly become cognitively burdensome when performed during a working teleconference session. Working with comments and suggested changes becomes more cognitively demanding when other authors are simultaneously editing the same content, and the user needs to be aware of their activities (e.g., to avoid introducing conflicting changes) while still performing the editing task. The use of different input types and methods, such as speech input or switch-based input, can significantly affect the amount of time REQ (requirement/pattern)uired to enter and edit text, as well as the user's ability to respond to potentially disruptive changes introduced by collaborators.'

1.4 Collaboration tools and accessibility

By following established guidance, notably that of Web Content Accessibility Guidelines (WCAG) [wcag22], designers of collaboration tools can help ensure that their user interfaces are *perceivable* to and *operable* by a wide range of users with disabilities. Following the Guidelines also enables user interfaces to be more *understandable*, and to be *robust* in their support for a range of user agents and assistive technologies. In addition, broadly applicable guidance on improving accessibility for people with cognitive and learning disabilities has been published in [coga-usable]. However, implementing current guidelines and suggested practices is not sufficient by itself to ensure that the user interface of a collaboration tool can be understood and used efficiently by people with disabilities. Thus, conforming to WCAG may well be insufficient for collaborative environments. For example WCAG does not inform automated interface simplification — a general web accessibility REQ (requirement/pattern)uirement being considered in APA's WAI-Adapt Task Force.

The collaboration features of these tools are necessarily complex. This can impose significant cognitive demands on many users, not only users with specialized accessibility requirements. This is especially true for users of screen readers, screen magnification and color contrast assistive technologies, as well as for persons living with various cognitive and learning disabilities. For this reason, the unique cognitive demands established by collaborative content creation applications can impose barriers to access which are addressable, in part, by making appropriate software design and implementation choices. Additional control of cognitive demands can be achieved by using the application and any assistive technologies appropriately in a collaborative setting, and by ensuring that the social context in which the collaboration occurs supports participation by contributors with disabilities (see section 1.5 Social Considerations).

Many users cannot track updates on multiple locations simultaneously, rather, they must view and comprehend the interactive elements of the application's features sequentially, for example in speech or braille for screen reader users. A screen reader or magnifier used in a collaborative application may well present suggested changes and comments in one section of the screen while the user is reading a document in a word processor. The user may also be expected to be communicating verbally with fellow collaborators (e.g., in a meeting) while undertaking editing tasks or comparing multiple revisions of content. Moreover, in applications supporting real-time collaborative editing, incoming changes made by other contributors may alter the content that the user is reading or editing in real time. These cognitive demands can be particularly challenging if a person is working with a user interface that is unfamiliar, whether it be an application or an assistive technology.

Due to the cognitive demands created by collaboration tools in the practical and social contexts in which they are used, strategies for improving accessibility are desirable that extend beyond current W3C guidance as documented elsewhere.

Thus when we talk about collaborative tools we must consider accessibility burdens imposed by their concomitant complexity. In truth, collaborative tools are necessarily complex interfaces for all users, and not only persons with various disabilities. The salient point here is that a failure to design to accomodate persons with disabilities appropriately will inevitably prevent their participation in collaborative work. What constitutes challenging complexity for most users will inevitably become an insurmountable barrier for some persons with disabilities.

A fairly common accessibility failure is the use of arbitrary color to flag edits put forth by different collaborators. However, identifying collaborators only by colorization violates WCAG 2.2 Success Criterion 1.4.1 as described below in User Need 11.

1.5 Social Considerations

Collaborative tools should support all identified accessibility features in order to provide comprehensive accessibility. However, it is unlikely all features will be needed by any individual collaborative team effort. We assume persons with disabilities are brought into collaborative teams because of the contributions they are expected to make in the project. Teams are encouraged to focus on accommodating the specific accessibility needs of participating team members in order to operate most efficiently and productively.

1.6 Scope and Applicability of this Document

Accessibility-related guidance provided in this document is applicable to a wide variety of tools. No unnecessary restriction is placed on the types of Web-based software to which it may reasonably be applied.

If a tool implements one or more of the distinctive features described in section 1.2 Distinctive features of collaboration tools, then the guidance in this document which addresses each such supported feature is relevant and applicable to the tool. Thus, the scope of the document includes any tool implemented using Web technologies that implements at least one of the distinctive features for which guidance is offered in the sections that follow.

For example, an annotation tool supporting the association of shared comments with selected text in Web pages would offer only a single feature described in this document. For this reason, only section Annotations would be relevant to the tool.

2. Terms

WYSIWYG
An acronym for What You See Is What You Get. As described by Wikipedia, it refers to software which allows content to be edited in a form that resembles its appearance when printed or displayed as a finished product, such as a printed document, web page, or slide presentation. WYSIWYG implies a user interface that allows the user to view something very similar to the result while the document is being created.

3. Real-Time co-editing

User Need 1: Users need to be able to **discover the presence of collaborators** who are reading or editing the content.

- **REQ (requirement/pattern) 1:** Provide a mode of operation in which *status messages* alert the user whenever a collaborator opens or closes an interactive session involving the same content that the user is accessing (e.g., the same document).

Note

WCAG REQ (requirement/pattern)uires that status messages be made available to assistive technologies. See Web Content Accessibility Guidelines (WCAG) 2.2 [wcag22], success criterion 4.1.3, and the associated definition of *status message*.

User Need 2: An assistive technology user needs to be **informed in real time of change**s to the content being made by collaborators.

- **REQ (requirement/pattern) 2:** Provide a mode of operation in which status messages inform the assistive technology user of insertions, deletions or formatting-related changes made by collaborators as they occur. Limiting these notifications to a content span currently focused by the user is also an advisable option.

User Need 3: Assistive technology users need the ability to **read or edit without being distracted** by status messages.

- **REQ (requirement/pattern) 3:** Provide a mode of operation in which status messages informing the user of the presence or activities of collaborators are suppressed. This may be achieved by allowing the user selectively to enable and disable specific types of status message, to messages relating to a specific span of content, or all such messages.

User Need 4: An assistive technology user needs the ability to **track changes introduced by a specific collaborator** as they are made.

- **REQ (requirement/pattern) 4:** Provide a function that moves and tracks user's focus to the location where a specific collaborator is editing. If there are multiple active collaborators, then multiple such commands, or a menu of active content editors, should be available.

User Need 5: Users with vision, cognitive or physical disabilities need to be able to **edit content without the distraction** of changes introduced by collaborators.

- **REQ (requirement/pattern) 5A:** Provide a mode of operation in which changes made by collaborators are not displayed while the user is editing, or
- **REQ (requirement/pattern) 5B:** Provide a mode of operation in which the component of the content that the user is editing (e.g., the paragraph, section, semantic unit of source code, or graphical object) can only be changed by one collaborator at a time, preventing others from making simultaneous modifications to the same component.

Coga suggest missing items

4. Annotations

User Need 6: An assistive technology user needs to be informed of the **presence of annotations along with the specific part of the content being annotated,** such as words, sentences or paragraphs. This also applies to lines of code in a software development project.

- **REQ (requirement/pattern) 6:** Ensure that information about annotations is conveyed to assistive technologies, together with the boundaries of the text to which the annotation applies, along with any metadata associated with the annotation, and any comment text.

Note

See Web Content Accessibility Guidelines (WCAG) 2.2 [wcag22], success criterion 1.3.1.

User Need 7: Assistive technology users need to be able to read text **without being distracted** by information about annotations.

- **REQ (requirement/pattern) 7:** Provide a mode of operation in which information about annotations is suppressed. This mode might be activated by an application setting, such as a toggle switch controlling the presence or absence of annotations.
- **User Need 8:** An assistive technology user needs to be able to navigate between annotations (from previous to next) and to obtain a navigable list of annotations (e.g., a list of comments in a word processor document or on a Web page), in order to read and respond to annotations efficiently.
- **REQ (requirement/pattern) 8:** Provide navigation functions and a means of obtaining a list of all the annotations associated with the content.

User Need 9: Assistive technology users need to be able to **control the amount of information** presented about annotations to prevent becoming overwhelmed while they are reading, navigating and editing content.

- **REQ (requirement/pattern) 9:** Provide options for the user to limit the amount of detail presented as each annotation is encountered. For example, it should be possible to suppress presentation of metadata, or replies to comments, or to alert the user only to the presence of the annotation without presenting the metadata or comment text.

User Need 10: Assistive technology users and users with learning or cognitive disabilities sometimes need **support in understanding and navigating annotations that represent comments organized as *threads* of conversation.**

- **REQ (requirement/pattern) 10A:** Ensure that the structure of comment threads is unambiguously presented in the user interface, both via visual cues such as icons and color changes, and to assistive technologies.
- **REQ (requirement/pattern) 10B:** Enable threads to be expanded or collapsed by the user, and ensure the expanded or collapsed state is disclosed to assistive technologies.

Note

REQ (requirement/pattern) 8 may be valuable to users in general, and it should be considered for inclusion as a feature of collaboration tools themselves.

5. Version control features

5.1 Suggested changes

User Need 11: Assistive technology users need to be able to **read the text with information included about *suggested changes*** (i.e., insertions, deletions or formatting modifications proposed by collaborators).

- **REQ (requirement/pattern) 11:** Provide a mode of operation in which details of insertions, deletions and formatting changes are appropriately presented by the assistive technology as the user reads the content.

Three suggestions and

add that cognitive need to see these changes, and

can also hide them. and

easily get the latest clean version

User Need 12: Users with color blindness need to be able to distinguish insertions, deletions, and unaltered text effectively.

- **REQ (requirement/pattern) 12:** Use distinctions other than color to identify inserted and deleted text in the visual interface as REQ (requirement/pattern)uired by WCAG 2.2 Success Criterion 1.4.1: Use of Color.

5.2 Presenting Differences Between Revisions

User Need 13: Users need to be able to compare revisions in **meaningful units (words, sentences, lines, etc.), according to the nature of the content, to maximize com**prehension.

- **REQ (requirement/pattern) 13:** Present differences in a manner that is appropriate to the type of content. For example, software source code might be presented with line-by-line differences, whereas natural language documents might be presented with differences shown word-by-word or sentence-by-sentence.

5.3 Summarizations

User Need 14: Users with learning or cognitive disabilities, and users of assistive technologies sometimes **need support in identifying revisions a**nd understanding their effects.

- **REQ (requirement/pattern) 14A:** Provide a mechanism to identify and summarize a series of changes to content. When artificial intelligence (AI) rather than human authoring is used to generate summaries, provide the ability to edit the generated summary.

- **REQ (requirement/pattern) 14B:** Provide a mechanism to identify and summarize individual comment threads. When AI rather than human authoring is used to generate summaries, provide the ability to edit the generated summary.
- **REQ (requirement/pattern) 14C:** Provide, if technically feasible, a mechanism allowing any user automatically to generate a summary for themselves of content or of a comment thread at any time. Project editors should have the ability to revise and commit a summary to the content in collaborative development as a permanent feature of the collaborative effort.

6. Notifications and Messages

Collaboration tools may send notifications to the user for a variety of reasons. For example, a user may be notified if a collaborator asynchronously submits changes to a document or project, or adds a comment. These notifications may be delivered via operating system facilities, or by a messaging service, such as e-mail or an instant message protocol. Moreover, the collaboration tool may support commenting, issue tracking, or other forms of interaction via external messaging. These optional capabilities are addressed in the following user needs and system requirements.

User Need 15: Users who are easily distracted need to receive only notifications that are crucially important to their collaborative activity.

- **REQ (requirement/pattern) 15:** Ensure that users can choose which types of notification are delivered, and which are suppressed, according to the nature of the information conveyed.

User Need 16: Users for whom reading text is slow or difficult need **information that is important to the task at hand to be clearly distinguished and prioritized**.

- **REQ (requirement/pattern) 16A:** Provide a mode of operation in which notifications are short, and links to more detailed information are included. In this mode, full details are not provided in the notification. For example, a user could be notified that a comment or issue has been created, with the full text being available only via a link rather than as part of the notificational message itself.
- **REQ (requirement/pattern) 16B:** If e-mail or a similar medium is used to deliver notifications, ensure that the *subject* of the message clearly specifies the project, document or issue relevant to the notification.
- **REQ (requirement/pattern) 16C:** If multiple notifications are provided together (e.g., in a single message), ensure that the user can sort the notifications according to reasonable preferences, for example, most recent first, or oldest first. This is applicable, for example, to a series of comments organized as threads of discussion, all delivered in a single summary message to the user.

- Missing - important

7. Access Controls

A collaborative environment may provide access controls to restrict the modification of content to specified individuals or groups of users. Moreover, access controls may be applied to the entire content, as in a document which is marked as *read-only* in a text editor or office application, or they may restrict editing to designated parts. Depending on the capabilities of the application, permissions may be changed by an authorized user during a collaborative editing session.

User Need 17: Users of assistive technologies and those with cognitive or learning disabilities need to be able to **find out whether they have permission** to edit content, in whole or in part.

- **REQ (requirement/pattern) 17:** Ensure that the applicable permission (e.g., read-only state) affecting content currently in focus is prominently presented in the user interface, and that it is made available to assistive technologies.
- **User Need 18:** Users, including those with learning or cognitive disabilities and those with assistive technologies, need to be informed of changes made to access permissions that take effect during an editing session.
- **REQ (requirement/pattern) 18:** If an access control affecting the content currently in focus is changed during an editing session, for example by a collaborator, a notification of the change is presented in the user interface, and made available to assistive technologies.

Note

Web Content Accessibility Guidelines (WCAG) 2.2 [wcag22] should be consulted for guidance on ensuring that the user interface for configuring access controls meets appropriate accessibility requirements.

8. General Guidance on Implementing Accessibility Features of Collaborative Environments

To facilitate effective collaboration, applications should be designed to respect conventions of user interface design that are likely to be expected by users, including those who have disabilities.

User Need 19: Users with learning or cognitive disabilities or who use assistive technologies need to be able t**o learn collaboration tools efficiently,** including software with which they are unfamiliar and which is necessary to the task at hand.

- **REQ (requirement/pattern) 19A:** In implementing collaboration features, follow established user interface conventions and design patterns. For example, use conventional terminology, labels, or icons for functionality that may be familiar to users.
- **REQ (requirement/pattern) 19B:** Support the accessibility-related features of the user's operating system, user agent, and assistive technology. For example, some assistive technologies, such as screen readers, have features designed specifically for reading comments and suggested changes in textual content, which should be supported instead of defining application-specific functionality or keyboard commands that achieve the same purpose.
- **REQ (requirement/pattern) 19C:**Make collaborative features available via an API to allow interoperability with tools with which the user may be familiar, and which may better satisfy a person's specific accessibility-related needs. For example, a revision control system could interoperate via an API with a user's chosen text editor or integrated development environment. 2 missing - we propose bekllwo user requirement 20

Note

Provision of an API does not diminish the importance of ensuring that user interfaces provided by the application satisfy accessibility requirements. Nor is REQ (requirement/pattern) 19C applicable to all collaborative environments. For example, some applications are highly dependent on editing functions specific to the tool itself, and should not be expected to interoperate with external editors.

Suggested User need **20:** Users with learning or cognitive disabilities or who use assistive technologies need to be able t**o use efficiently,** including software with which they are unfamiliar and which is necessary to the task at hand.

- REQ (requirement/pattern)20 A: Keep it simple. Keep process and workflow as simple as possible. Have short critical paths with as few steps as possible
- REQ (requirement/pattern)20 B: , Do not require the user to open multiple tabs, windows or panels to complete a single task as remembering and navigating through this content can disorientate the user.
- REQ (requirement/pattern)20 C: Enable a mode for inline changes. Even looking and comparing sections of a screen will not work for many users. Changes should be visually marked and easy to reach for Assistive technologies. Having the tags for start and end point would be helpful.

- REQ (requirement/pattern) 20 D: When new processes or designs are added, keep them as familiar to the user as possible. Always allow users to roll back to an older interface. This allows older users to continue being productive.
-

Suggested User need 21 :As a user who finds some web sites hard to use and struggles with remembering and following instructions , I sometimes need in-page instructions so that I can do the correct task

- REQ (requirement/pattern) 21 A)Instructions from the author can be place in a reserved consistent location that is easy to find.
- Context sensitive help is available for tasks , workflow and any non-standard elements.
- support info for google doc screen reader capabilities

A. References

A.1 Informative references

[coga-usable]
Making Content Usable for People with Cognitive and Learning Disabilities. Lisa Seeman-Horwitz; Rachael Bradley Montgomery; Steve Lee; Ruoxi Ran. W3C. 29 April 2021. W3C Working Group Note. URL: https://www.w3.org/TR/coga-usable/
[concurrency-control]
Concurrency control in groupware systems. Clarence A. Ellis; Simon J. Gibbs. Proceedings of the 1989 ACM SIGMOD international conference on Management of data. 1989.
[wcag22]
Web Content Accessibility Guidelines (WCAG) 2.2. Michael Cooper; Andrew Kirkpatrick; Alastair Campbell; Rachael Bradley Montgomery; Charles Adams. W3C. 5 October 2023. W3C Recommendation. URL: https://www.w3.org/TR/WCAG22/

Lisa (from COGA) comments

LS comment 1: i need the content to be eiser to understand
1. Wall of text is a problem. (full call)
I have had to reformat the document to make it easier to follow so we can review it.

2. user needs distinguishable from requirements (The user needs are not bulleted). So you can jump from user need to user need, and see to which user need each REQ (requirement/pattern)uirement belongs to.

2. Key works in userneeds in bold

3 Explained what req is
Note that - Even with that the language is hard to understand

LS comment2 : this is in section 11, but hard to understand. Things in bold are missing but they are also added as in user need 20 above.

Coga Suggested User Need 1: As a user with a language, processing, or memory impairment, I need the interface and language used to be clear and easy for me to understand. As a user with short and medium-term memory impairment and impaired executive function <https://www.w3.org/TR/coga-usable/#dfn-executive-function>, I need a familiar interface so that I do not need to figure out and remember new interfaces. This may take a few weeks of repetition and I may not manage to learn it all if I have a condition affecting learning new things, such as dementia.

Coga Suggested REQ (requirement/pattern) (REQ (requirement/pattern)uirement) 1: Help users understand what things are and how to use them. Use things that are familiar to the user so that they do not have to learn new icons, symbols, terms, or design patterns. People with cognitive and learning disabilities often need common behavior and design patterns. For example, they may know the standard convention for links (underlined and blue for unvisited; purple for visited).. This includes:

- **Keep it simple. Keep process and workflow as simple as possible**
- **Have short critical paths with as few steps as possible, Do not REQ (requirement/pattern)uire the user to open multiple tabs, windows or panels to complete a task as remembering and navigating through this content can disorientate the user.**

- Use familiar terms, such as save, Help, Copy. Avoid making the user need to learn new terms such as mode or Fork.
- Do not make the user learn new terms or language for editing, such as a markdown, html or other syntax that they may not know. For example, provide an edit button and WYSIWYG (you see is what you get) HTML editor. Familiar icons, such as the help icon, with the word help.
- Use common design patterns when you can. Do not REQ (requirement/pattern)uire a new way of thinking just to something simple like create/edit/make available a simply formatted text document

- **When new processes are necessary, keep them as familiar to the user as possible. Always allow users to roll back to an older interface. This allows older users to continue being productive.**

LS comment3 : Where is the need for help? Added as user need 21

Suggested User Need5 :As a user who finds some web sites hard to use and struggles with remembering and following instructions , I sometimes need in-page instructions so that I can do the correct task

Coga Suggested REQ (requirement/pattern) (REQ (requirement/pattern)uirement 5)Instructions from the author are easy to find. For example, an icon and link to a readme document is always available on the main toolbar.

LS comment4 : Addition user need (added to as user need 20)

As a user who stooges with sequencing,orientation and/or complex multi steps task.

One page , no need to look at multiple pages to complete a task and skip between tabs.