Investigating Accessibility Barriers in Software Development Tools Through the Blind People's Perspective

Giselle Lorrane Nobre Melo[1], Suzane Santos dos Santos[2], Luciano Arruda Teran[1], Ingrid Teixeira Monteiro[3], Kamila da Hora Rios Rodrigues[2], Marcelle Pereira Mota[1]

[1]Universidade Federal do Pará (UFPA)
Belém, PA - Brazil

[2]Universidade de São Paulo (USP)
São Carlos, SP - Brazil

[3]Universidade Federal do Ceará (UFC)
Quixadá, CE - Brazil

{giselle.nobre@icen.,luciano.teran@icen.,mpmota}@ufpa.br,

{suzanesantos@,kamila.rios@icmc.}usp.br, ingrid@ufc.br

Abstract. *Even when using screen reader tools, blind software developers face digital accessibility issues in collaborative work. Our primary goal was to investigate the difficulties faced by software developers, specifically in team integration and communication with professional tools. For that, we carried out a Rapid Literature Review (RLR) and an accessibility inspection using the Semiotic Inspection Method mediated by Screen Reader (SIM-SR) to investigate the communicability of task management tools and of instructions as coordination tools in the 3C collaboration model. As a result, we proposed 16 collaborative work considerations to the equity of blind people in software teams and to improve their daily development experiences mediated by management tools.*

1. Introduction

The Information and Communications Technology (ICT) in Brazil has experienced significant growth recently, driven by both digital transformation and the post-pandemic demand for technological solutions [Atique et al. 2025]. Given this context, the formal workforce in the ICT sector has also expanded, transforming work dynamics and modalities. A clear example is the adoption of remote work, which was mainly driven by the COVID-19 pandemic, making the model an imperative necessity [Escorcer et al. 2024].

Remote work, also known as telework or distance work, is defined as the practice of performing duties outside the traditional office environment, often using technology to communicate and collaborate with colleagues and supervisors [Sin and Kathiarayan 2023]. The increased acceptance and availability of remote work represent a valuable opportunity to redefine the work environment and create more inclusive work cultures and practices for People with Disabilities (PwD), a diverse pool that has been systematically excluded for decades due to a lack of such accommodations [Lake and Maidment 2023].

However, it is crucial not to assume that remote working is automatically accessible and inclusive, especially in the ICT field, where the need for constant collaboration and the rapid evolution of technology can further exacerbate barriers for PwD, for example, tools that work correctly with screen readers [Escorcer et al. 2024]. To effectively integrate teleworkers with disabilities and overcome organizational norms that favor people without disabilities, it is essential to make adaptations and adjustments, especially given that teleworking policies hastily implemented during the COVID-19 may have neglected accessibility requirements [Lake and Maidment 2023]. Blind and low-vision people report accessibility issues in various contexts, such as using tools beyond programming, requiring greater dedication and effort to make meetings accessible and participate equitably, and sometimes omitting their disabilities from development teams to avoid being disadvantaged. In this context, social, organizational, and technical solutions to mitigate accessibility problems were presented [Cha et al. 2024a].

Considering the context of software development, there are still many barriers to overcome. In a remote environment, software development relies on tools that facilitate tasks. Considered a critical factor in the success of collaborative work, perception, or situational awareness, refers to understanding others' activities and providing context for one's own [Fuks et al. 2011]. For blind people, for example, keeping up with the rest of the team in remote environments can present specific challenges related to both the technical and social aspects involved in the collaboration process. Technical problems include accessibility issues in the various tools used in the development environment. This negatively impacts continuous collaboration with work colleagues, whether they are blind or have other disabilities.

Therefore, the need to investigate problems with tools used by developers from a user experience perspective becomes evident. The objective of this work was to answer the following questions: (1) What challenges do blind people face in software development teams when using task management tools? and (2) How can we make task management tools used by software development teams accessible to blind people? Aligned with this purpose, the present study investigates the adaptation of task management tools to promote the PwD's autonomy software development teams. Management tools, included in the 3C collaboration model as a cooperation tool, were chosen because they centralize the main development activities, aligning team objectives and distributing tasks. To achieve the research goal, a Rapid Literature Review (RLR) [Smela et al. 2023] was conducted to ascertain the obstacles encountered by the software developer community in the current state of the art. Subsequently, accessibility inspections were conducted using the Semiotic Inspection Method Mediated by Screen Reader (SIM-SR) [Carvalho 2021] in two task management tools, which are classified as coordination tools in the 3C collaboration model [Fuks et al. 2011].

The remainder of this paper is organized as follows: background with concepts of Computer-Supported Collaborative Work (CSCW) and Human-Computer Interaction (HCI) in Section 2, methodology in Section 3, rapid literature review in Section 4, accessibility inspection in Section 5, collaborative work considerations in Section 6, discussion in Section 7 and final remarks in Section 8.

2. Theoretical Background

This section presents concepts that support this study, including awareness, the 3C collaboration model, accessibility, communicability, SIM-SR, and developer experience (DX).

Awareness is an essential concept in collaborative systems and has been a valuable topic at CSCW since its inception. According to Dourish and Bellotti (1992), awareness is the understanding of others' activities, providing a context for one's own activity. This context is used to ensure that individual contributions are relevant to the group's overall activity and to evaluate individual actions in relation to the group's goals and progress. Therefore, allows groups to manage the work process in collaborative systems.

These systems are known for enabling group members to communicate clearly, coordinate their work, and cooperate with each other. These three components are the pillars of the collaboration process, which later termed the 3C collaboration model [Ellis et al. 1991, Fuks et al. 2011]. The model is divided into three concepts (C's): **1) communication**, i.e. the act of making knowledge common and shared; **2) cooperation**, i.e. the action of cooperating in a group or for the group; and **3) coordination**, i.e. the action of organizing together. In the software development landscape, communication tools enable development team members to communicate, for example, via text channels, voice channels, or video conferencing. Cooperation tools enable members to interact collaboratively, such as code versioning, testing, and development tools. Coordination tools enable members to analyze team demands and on the progress of their activities, such as task management tools and shared documents.

DX is a construct derived from User Experience (UX), and although the two concepts intersect, DX differs from UX by focusing on how developers feel, think, and value their work [Fagerholm and Münch 2012]. From this perspective, consists of the developer's concentration, as reflected in the criteria of activities clarity and objectivity continuous feedback on the activities development to resolve difficulties as quickly as possible and the developer's skills [Kuusinen 2016]. However, aspects related to accessibility and developer experience have not been explored to this point in this research.

The communication between these professionals and the systems implemented in the remote collaborative environment should be improved. In this context, Semiotic Engineering proposes to investigate metacommunication, which the communication process between systems designers and users through the interface (communication about communication) [de Souza et al. 2006].

One of the methods provided by SemEng to evaluate communicability (which is the quality of the designer's metacomunication) is the Semiotic Inspection Method (SIM). Among the steps in this method are: to analyse the metalinguistic, static, and dynamic interface signs; to elaborate one "designer metamessage" (describing who is the system user considered by the designer and describing how the system works) to each sign. The Semiotic Inspection Method mediated by Screen Reader (SIM-SR) is a method for evaluating communicability by analyzing metalinguistic, dynamic and static signs with and without a screen reader (SR) [Carvalho 2021]. SIM-SR is therefore a way to connect two criteria of quality of use: communicability and accessibility. Accessibility means ensuring systems are usable by everyone without impediments in the interface [Barbosa et al. 2021].

3. Methodology

We carried out two studies to understand the issues faced by blind people in software development teams: in the first one, we conducted a RLR (phase 1); and in the second, we assessed the accessibility of a management tool (phase 2). The results of the two studies were consolidated in a set of collaborative work considerations (phase 3). All of our studies have been approved by our institution's research ethics CAAE: 74659423.6.0000.0018.

Phase 1. Rapid literature review

In phase 1, we conducted a RLR[1] [Smela et al. 2023] to understand from the literature faced by blind workers in software development teams when using management tools, defined in the 3C model [Fuks et al. 2011] as coordination tools. To address we defined inclusion and exclusion criteria, research questions, and a search string to select and filter relevant articles. four relevant national and international computer science scientific repositories: SBC Open Lib, Scopus, IEEExplorer, and ACM DL. The RLR steps are detailed in Section 4.

Phase 2. Accessibility inspection

In phase 2, the first and third authors conducted an accessibility inspection (using SIM-SR) on a task management tool Microsoft Planner, a tool widely used in software development teams [Hu et al. 2024] and which visually impaired people indicate has accessibility problems [Rocha et al. 2023]. We use SIM-SR because it allows performing systems analysis based on defined scenarios, with and without a screen reader, considering the main dashboard of the cards and two modals, related to editing the card and the task of moving the card, respectively. The method involves evaluating systems by static, dynamic, and metalinguistic signs with and without a screen reader, to verify accessibility and communication issues in this cases, since the system's behavior should be the same in both scenarios. The results are condensed and discussed in the metamessage contrast and communicability assessment stage. Information on the application of the method is detailed in Section 5.

In phase 3, based on the evidence from phases 1 and 2, we identified 17 collaborative work considerations to enable the development of accessible task management tools within a collaborative context, improving autonomy for blind people in software development teams. The considerations for collaborative work are detailed in Section 6.

4. Rapid Literature Review

The objective of this RLR is to understand, within the current state of the art, the problems faced by PwD when working as software developers, how these problems are overcome, and the proposed improvements to these problems and their applications.

Based on this goal, we the search string **(accessibility OR a11y OR ally OR inclus*) AND (software development OR tool*) AND ((disab* OR impair* OR blind*) AND ("software develop*" OR "software program*")**, designed to address accessibility issues, software development, and software development related to blind people and applied it to the SOL SBC, Scopus, IEEExplorer, and ACM DL computing science scientific repositories, with a time range from January 2019 to December 2024. The defined range aimed to observe years preceding the COVID-19 pandemic (2019), the three years the pandemic last (2020, 2021, and 2022), and post-pandemic years (2023 and 2024), as

[1] The rapid literature review (RLR) process involves executing the method by only one reviewer.

there was a need to adapt welfare services through hybrid or remote work, and as a consequence, the IT and software development market became heated for people with and without disabilities.

The research questions seek to understand what difficulties PwD face in the IT field when working as software developers **(RQ1)**, how these difficulties are overcome **(RQ2)**, what alternatives company managers provide for team members who are PwD **(RQ3)**, what methods are used to assist PwD in their tasks with autonomy **(RQ4)**, and what problems exist with tools used by blind in software development teams **(RQ5)**.

After applying the search string, duplicate articles, proceedings, and books were checked, resulting in 265 articles. In the first filtering, by reading the title, abstract, and applying the inclusion criteria (full or expanded articles in English or Portuguese related to at least one research question or addressing accessibility in software development teams) and exclusion criteria (duplicate articles, articles in languages other than English or Portuguese, short articles, and articles that did not address accessibility issues in software development teams), 26 articles were selected. In the second filtering, the articles were read in full to verify the research questions, with 13 related articles for the last filtering, as illustrated in Figure 1.

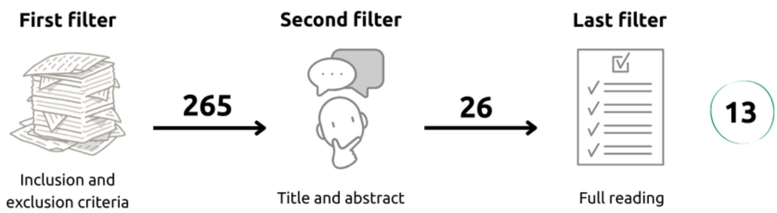

Figure 1. Filters of Literature Review.

RQ1. Difficulties faced by software developers PwD

RQ1 aims to understand the difficulties PwD face when working as software developers, based on their performances and experiences.

Akter et al. (2023) and Huff et al. (2020) explore the problems in video conferencing tools, classified as a communication tool in the 3C collaboration model [Fuks et al. 2011]. In this sense, simultaneous communication in meetings via voice and chat and the lack of accessibility of the resources implemented in these tools are cited as factors that can lead to information overload and difficulty of use for blind people.

Regarding problems with collaborative tools, Huff et al. (2020) reported difficulties with pair programming: interfaces are highly visual, with information illegible to screen readers and keyboard navigation. Nascimento et al. (2022) , Pereira et al. (2023) and Skura et al. (2024) reported problems in software development collaboration tools (IDEs), which were divided into blocks: code writing, code debugging, software versioning and deployment, navigation, diagrams, and learning materials.

Johnson et al. (2022) conducted an interaction mapping of the program-L community[2], an online community for asking questions and helping each other, aimed at

[2] [https://www.freelists.org/archive/program-l/]

blind software developers, with support for use in IDEs or hardware, indicating the screen reader used by the user to facilitate community collaboration. Pandey et al. (2022) highlighted that graphical interface builders are inaccessible to SR, and despite accessibility updates in IDEs, problems persisted, and the updates needed to be done. Problems with application emulators, user interface (UI) resources, layout managers, and code tab order are also cited, which made it difficult for the user to locate themselves in the interface.

In coordination tools, Cha et al. (2024b) indicated that, although the accessibility recommendations provided by the Web Accessibility Initiative (WAI) are implemented in the tools and they are technically accessible, problems persist when blind use them. In this case, the project management tools Jira and Trello, the whiteboard tools Miro, Bluescape, and LucidChart, and the design tool Envision were cited. Problems with screen readers in slide presentations and pair programming are also reported, as well as in interactions with hardware (touchscreen), which become unfeasible with screen readers and their users.

Melo et al. (2024) reported that deaf people report problems with communication tools, such as conferencing with automatically generated captions in meetings where participants interact simultaneously, as well as reading long and verbose texts. Blind people indicate problems using task management tools such as Jira and Microsoft Planner.

RQ2. Software developers PwD overcoming the barriers

Research question 2 seeks to understand how the problems faced by PwD are addressed in software development teams.

Akter et al. (2023) described that participants in their study simulated meetings in advance to test the resources available in video conferencing tools, and facilitators gathered information about the participants to determine whether the scenario needed to be adapted. Huff et al. (2020) reported that participants used text editors with simpler layouts to assist with software development rather than IDEs. Melo et al. (2024) highlighted that the participants used spreadsheets to update their tasks rather than task management tools. The studies relate data of blind people.

Johnson et al. (2022) indicated that online support with people in the same condition facilitates mutual help and promotes community interaction. Pandey et al. (2022) also indicated mutual collaboration among the blind in the program-L community, highlighting the participation of sighted people to assist in meetings where blind people serve as facilitators, to support interaction tasks at the interface, as well as the option to record and broadcast the training or meeting. Similarly, Cha et al. (2024b) highlighted the necessity of sighted people in blind activities to provide information that screen readers do not reproduce in meetings, as well as empathy through active participation.

RQ3. Company support methods for PwD autonomy in development teams

Research question 3 aims to understand how company managers propose improvements or adaptations for people with disabilities in software development teams.

The company featured in the study by Pereira et al. (2023) implemented a professional development training program for PwD to increase team diversity. Participants in the program are hired for eight months and receive remuneration, ongoing training, and access to a sign language interpreter during this period, but the facilitators are not PwD professionals.

RQ4. Support and empowerment methods for PwD in development teams

Research question 4 seeks to verify in the literature the methods aimed at adapting software development teams to mitigate the problems faced by PwD.

Silva et al. (2021) proposed recommendations to be applied in phases within agile methodologies, considering accessibility and usability: daily meeting, pair programming, planning meeting, evaluation meeting, retrospective meeting, effort estimation, workshops, code challenges, and maintenance. Miranda et al. (2024) proposed a conceptual model of agile development focused on accessibility, with criteria defined in class diagrams (UML), to ensure accessibility throughout the software development process. Similarly, Armas et al. (2020) proposes adaptation for continuous integration and continuous delivery (CI/CD) with accessibility requirements and, in the validation proposal, indicates the participation of PwD.

RQ5. Barriers and solutions related to tools for blind people

Research question 5 examines the problems blind people report in software development teams and the solutions proposed to mitigate them.

Silva et al. (2021) indicated that agile practices aimed at the blind need clarity and detail in information, dynamics with verbal communication, and physical objects in face-to-face meetings to facilitate integration. Nascimento et al. (2022) suggested that mentoring sessions should be conducted within companies, with team members, leaders, and managers to understand the importance of using accessible tools from the blind perspective. Skura et al. (2024) proposes accessibility improvements in collaboration tools: code writing and debugging, IDEs and diagrams, teaching and learning materials. Melo et al. (2024) proposes design considerations for task management tools from the blind perspective.

Huff et al. (2020) proposed flexibility in communication and work practices to include blind, understanding blind' strengths to assign tasks based on their skills, and support from blind and other team members to collaboratively build accessible software documentation and to facilitate the onboarding of new developers in the same condition. Cha et al. (2024b) indicated that physical interface models should be created to assist the blind developers, as well as the effective implementation of accessibility features in IDEs. Pereira et al. (2023) highlighted the importance of training programs in companies for the inclusion of PwD in development teams, with the help of accessible practices, organization, and creation of accessible material, and the use of accessible tools.

Pandey et al. (2022) indicated that problems related to the lack of software documentation should be discussed, particularly in the UI scenario, since this directly impacts code writing and debugging and, consequently, the use of software with screen readers. Therefore, it is recommended to consider accessibility in tools with screen readers, since there are various tools and distinct screen readers. Continuous help and support between PwD and the program-L community, as well as effective verification of accessibility features implemented in interfaces, are crucial, as they may indicate accessibility, but problems still occur in practice.

5. Accessibility and Communicability Inspection

Based on the RLR, we assessed Microsoft Planner to verify accessibility and communicability issues using SIM-SR. To this end, the first evaluator and first author of this, with prior knowledge in HCI and inspection, evaluated scenarios with and without a screen reader, as proposed by the method. To validate the evaluation, a second evaluator and the third author of this work, with the same experience in HCI and SIM, verified the signal evaluation performed by the first evaluator.

The scenarios defined for the Microsoft Planner inspection were: creating a card, which involves adding task information such as title, date, names of the users performing the task, and description; editing a card, which involves changing the data indicated when creating the card; and locating and moving a card, which involves finding a specific card in the task panel between columns and moving it to another column, corresponding to the task update. These scenarios are commonly performed by software development teams, based on updated activities. The complete SIM-SR steps are available in Melo (2026).

5.1. Evidence from the inspection in Microsoft Planner using SIM-SR

In Figure 2, it is possible to visualize the metalinguistic signs [M1 to M4] and the dynamic signs [D1] and [D4]. In the scenario of creating a card, across steps with and without a screen reader, it was observed that the metalinguistic sign [M1] and the dynamic sign [D1] have the same name but different functionalities. For users without a screen reader, the difference is apparent in the buttons' different colors. Still, the field reproduction with a screen reader is similar, which can confuse users with a screen reader about the purpose of the button actions and their location in the task pane.

The dynamic sign [D6] is a search bar in the panel, which can be used to locate a card by its title. The metalinguistic sign [M2] is a text field for entering the name of the card to be created; [M3] is a field where the user can indicate the task completion data; and [M4], labeled "assign", in Portuguese "atribuir", is used to add members to this card.

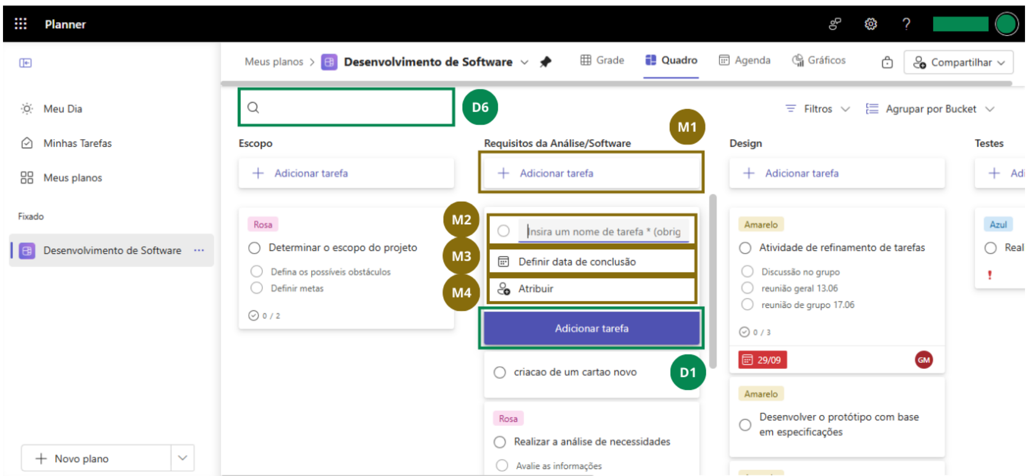

Figure 2. Action dashboard for creating the card

In Figure 3, in the scenario of editing a card, it is possible to observe the static sign [E1], with the English "bucket"; the entire panel is in Portuguese, and with a screen reader, it is reproduced in English. In both cases, the use of English without further information in a panel primarily in Portuguese can confuse the user.

The metalinguistic sign [M5] indicates reading the selection icon, followed by reading the text field with the task name. The metalinguistic sign [M6] corresponds to assigning users to a card. The metalinguistic sign [M7] indicates the insertion of a label on the card, represented by labels with names and colors. The dynamic sign [D2] indicates a selection box labeled “show on card”, which is displayed only after the user enters characters in the text field indicated by the dynamic sign [D3].

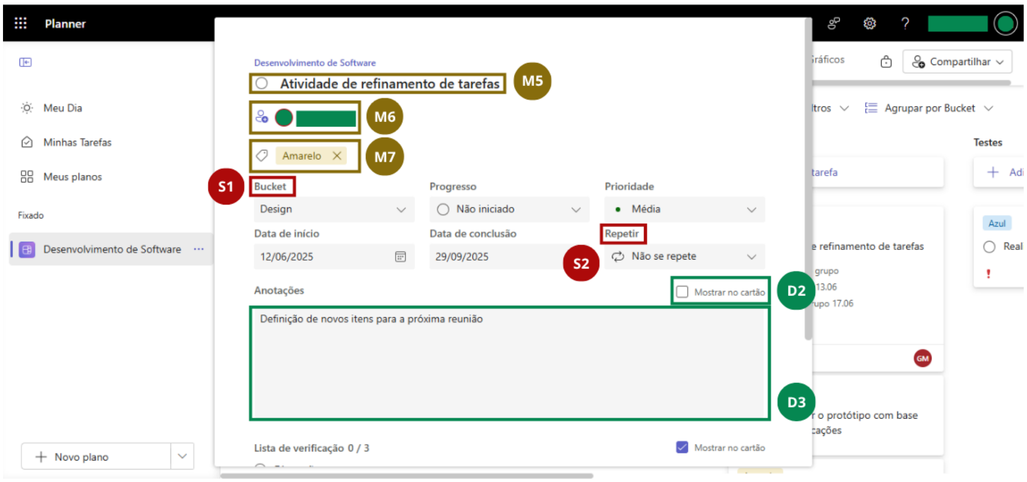

Figure 3. Action dashboard for editing the card

In Figure 4, the screen for moving the card shows the static sign [E3], which indicates the button to close the screen in the interface, and, when displayed by the screen reader, the information is mixed between Portuguese and English. The dynamic signs [D10] and [D11] indicate the panel where the card is located and the menu of columns to which the card can be moved. The dynamic sign [D12] indicates the options of columns the card can be moved to.

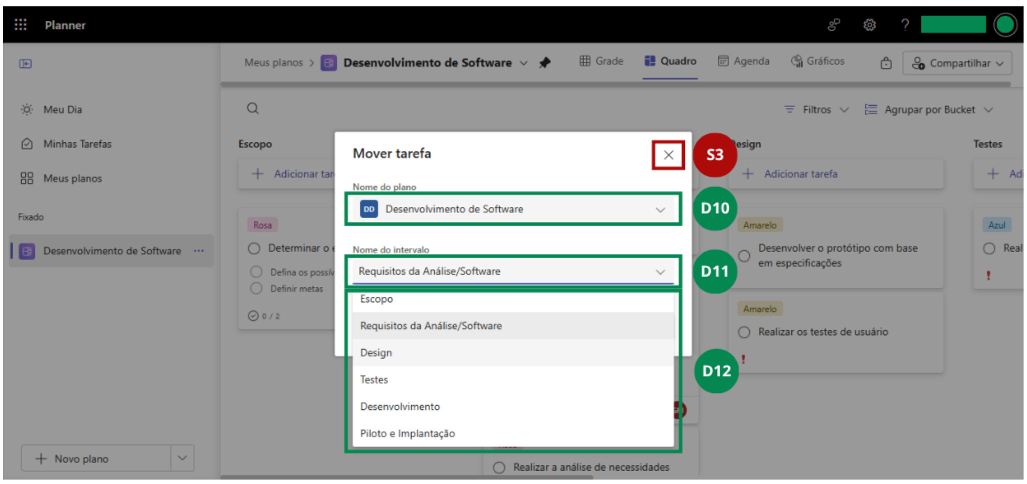

Figure 4. Action dashboard for moving the card

5.2. Final assessment

During the accessibility and communicability inspection of the Microsoft Planner tool, barriers were observed when performing the defined scenarios with and without a screen reader, as proposed by SIM-SR. The problems found are related to language mixing in the interface and, when reproduced by the screen reader, a lack of reading of some functionalities available in the interface, as well as an excess of information in functionalities when reproduced by the screen reader.

Based on data obtained in the other stages of the inspection, with and without a screen reader, it was possible to observe accessibility problems in the Microsoft Planner task management tool. The problems highlighted in this inspection are related to the amount of information the system provides. Occasionally, there is an excess of information, which can confuse users with a screen reader; a lack of information can lead to confusion when locating and defining actions in the system.

Another problem is how to complete actions in the tool. Microsoft Planner does not provide a button to complete editing the card being created, regardless of whether a screen reader is in use. Therefore, this task may not be completed due to insufficient information. In this case, the first author performed the test without a screen reader and completed the action by clicking outside the pop-up on the interface. When performing the tests with a screen reader, those who were already aware of the absence of this button had no major problems completing the task. However, if another evaluator had performed the task, this could indicate a problem with completing it.

6. Collaborative Work Considerations

Based on the analysis of data from the literature review and the accessibility and communication assessment of the Microsoft Planner, we proposed 16 collaborative work considerations to include blind software developers in the software development process. Considerations 1 through 10 were based on evidence obtained from the RLR (Section 4). Considerations 11 through 16 were based on evidence obtained from the accessibility and communication assessment of Microsoft Planner (Section 5). These considerations are both general and specific, focusing on fundamental characteristics in the development of accessible task management tools, as indicated in Table 1. The considerations were generated from recommendations indicated in the literature within the context of the research, mainly highlighting the requirements aimed at blind people and improvements in task management tools, based on problems found in Microsoft Planner.

7. Discussion

Based on the results of this work, we observe that the development of accessible software still requires improvements, discussions, and reflections to ensure effective application, mitigate accessibility problems, and promote equity and diversity in software development teams. This need also stems from the COVID-19 pandemic, which enabled the adaptation of common welfare services to remain functional in hybrid or remote environments, thereby prompting the inclusion of PwD in the IT area.

Accessibility in software development is still an emerging topic, as reflected in the gap between reports of accessibility in development teams and proposed solutions,

Table 1. Collaborative Work Considerations and Evidences.

Consideration	Evidence
Promote direct, detailed communication in interactions with blind people	In the phases of pair programming, planning, evaluation, retrospective, and effort assessment, rich detail about the activities performed and the effective inclusion of blind people in discussions are fundamental for understanding the problems and sharing their experiences in the activities [Silva et al. 2021, Pereira et al. 2023]. Akter et al. (2023) suggests implementing an optical resource to identify the person who begins speaking in a meeting, to facilitate identification and better comprehension by blind people.
Adapt all training sessions and workshop materials.	All materials used in training sessions and workshops must be accessible to blind people. It is recommended to make the materials available to participants in advance and to ensure remote support is effective [Silva et al. 2021, Akter et al. 2023, Pereira et al. 2023].
Allow blind people to serve as facilitators for training sessions and workshops.	Blind individuals should act as facilitators in training sessions and workshops, allowing team members to understand their difficulties and challenges in performing their tasks and to share best practices in the development environment that all team members can implement so that blind members feel part of the team's activities [Silva et al. 2021].
Respect each individual's turn to speak.	From a blind person's perspective, verbal communication is fundamental to understanding the context of the discussion. Therefore, it is necessary to respect speaking time in meetings so that communication is effective for all participants [Silva et al. 2021, Akter et al. 2023].
Provide accessible documentation with instructions for us screen readers.	System documentation is necessary so users can understand how it works. For the blind, it is important to provide information on keyboard shortcuts and the navigation implemented during the system's development [Silva et al. 2021, Akter et al. 2023, Miranda et al. 2024, Pandey et al. 2022, Pereira et al. 2023, Skura et al. 2024].
Provide a help and support channel for the blind users within the system.	Support within the system to understand how the implemented features work is fundamental to its use, and consequently, blind users can answer questions about access and navigation using screen readers [Johnson et al. 2022, Skura et al. 2024].
Use personas in the user journey that blind people.	The user journey is fundamental to understanding the system user. Therefore, blind personas must be considered so that requirements are conceived and developed in a way that makes system implementations accessible [Armas et al. 2020].
Review the prototypes created from an accessibility perspective.	It is necessary to verify whether the suggested functionalities include accessibility features before the requirements creation and system implementation stage [Armas et al. 2020, Silva et al. 2019].
Test the implemented requirements with blind users.	The implementations made to the system must be evaluated by real users within their usage context to verify the accessibility of the feature developed in the system. For this, validation with blind participants is necessary [Armas et al. 2020, Huff et al. 2020, Silva et al. 2019].
Consider the WCAG guidelines in all system requirements.	In the development process of a given software to enable the digital inclusion of PwD [Armas et al. 2020], as they are directly related to accessibility issues in the functionalities to be implemented.
Provide distinct text views for fields and buttons for each task.	In **Microsoft Planner**, there are two buttons with the same name but different functions, which can confuse understanding how to perform the task.
Clearly and objectively describe the text views fields and buttons.	Some text in **Microsoft Planner** lacks a clear, objective description of its functionality, leading to confusion for users.
Add captions to the system's icons.	In **Microsoft Planner**, some icons lack alternative text or captions, making them difficult for users to interpret, both with and without screen readers.
Set the system's default language to the user's preferred language.	In **Microsoft Planner**, with and without a screen reader, even when the user's preferred language has been set.
Provide objective information in the feature descriptions.	In **Microsoft Planner**, some symbols, during the screen reader inspection phases, provided a lot of information, for example, about card details. However, displaying too much information can confuse the user.
Use a screen reader to read aloud the important and necessary fields.	In **Microsoft Planner**, reading acronyms and other fields in the foreground that are not essential to interpreting the symbol can confuse a screen reader user.

and in the lack of practical application of these proposals in the papers returned from the literature review. It was observed that, within the context of the 3C collaboration model, the works primarily address studies focused on cooperation tools and communication tools, which makes this study important in addressing accessibility issues focusing in coordination tools.

In this sense, this work reflects the intersection of CSCW and HCI through task management tools as coordination tools in the software development environment, with an understanding of the 3C collaboration model and the observation of accessibility and communicability issues in collaborative systems. In addition, the characteristics of these tools and the impact of their individual and collective use are reflected in the concept of DX, derived from UX, in which user needs and ways of working directly shape the developer experience through the use of accessible software.

In the context of accessibility and communicability inspection, barriers were found that also affect the experience of blind software developers, who, even with the WCAG guidelines, found that accessible software development is not evident in the Microsoft Planner task management tool when applying SIM-SR, as is reported in the literature review articles, where participants report their difficulties in using tools in the context of software development.

In this context, collaborative work considerations propose improvements aimed at accessible software development from the perspective of blind people and their reports, combined with an accessibility inspection of a task management tool, to mitigate the problems faced by software developers who are blind and thus enable them to carry out their activities autonomously in software development teams.

In parallel with this work, an accessibility inspection was conducted on the Jira task management tool to understand the accessibility problems blind face when working in software development teams, which included two sighted evaluators and one blind evaluator [Melo et al. 2024].

8. Final Remarks

This study aimed to verify the accessibility problems faced by blind people in software development teams through a literature review based on RLR, followed by an inspection of accessibility and communicability in a task management tool using SIM-SR, classified as a collaboration tool in the 3C collaboration model. From these results, considerations for collaborative work were developed to minimize problems faced by visually impaired people and promote equity in software development.

The collaborative work considerations were generated from the problems reported in the studies returned from the RLR and from the accessibility and communicability inspection of Microsoft Planner. However, the 10th to 16th considerations have been addressed by the WCAG guidelines, as they are complementary and agree that new considerations should use more direct, concise language. Thus, this work is an intersection of HCI areas through accessibility and communicability improvements integrated into the SIM-SR method; in CSCW through the 3C collaboration concept and the perspective of coordination tools; and in DX, which impacts an effective experience for blind developers.

As limitations of this work, we highlight the short time frame of the literature review, the validation stage of the generated design considerations, and the inspection of other task management tools. As future work, we propose investigating additional tools to consolidate the considerations generated, classify them for better application, verify the perspectives on other deficiencies or conditions of software developers, and, include analysis with real users and validation with blind people in development teams.

Acknowledgments

We thank CAPES, finance code 001, for its support and the Grammarly tool for grammatical support in English. We thank Josef Kleber (*in memoriam*) for building the pdfcomment library, which made this paper, written in LaTeX, accessible to blind readers.

References

Akter, T., Cha, Y., Figueira, I., Branham, S. M., and Piper, A. M. (2023). “if i’m supposed to be the facilitator, i should be the host”: Understanding the accessibility of videoconferencing for blind and low vision meeting facilitators. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS ’23, New York, NY, USA. Association for Computing Machinery.

Armas, L., Rojas, H., and Renteria, R. (2020). Proposal for an accessible software development model. In *2020 3rd International Conference of Inclusive Technology and Education (CONTIE)*, pages 104–109.

Atique, A. B., Bertoni, A. P. S., da Fonte, A. R., Raya, Portilho, C. M., Neto, F., and Azevedo, T. (2025). Estudo de tic 2025 – indústria de software e serviços de tic no brasil. Technical report, Observatório Softex.

Barbosa, S. D. J., Silva, B. S. d., Silveira, M. S., Gasparini, I., Darin, T., and Barbosa, G. D. J. (2021). *Interação Humano-Computador e Experiência do Usuário*. Autopublição.

Carvalho, L. P. (2021). Mis-lt: Um método para avaliação de acessibilidade e comunicabilidade de sistemas interativos. Master’s thesis, Universidade Federal de Minas Gerais.

Cha, Y., Figueira, I., Ayala, J., Edwards, E. J., Garcia, J., van der Hoek, A., and Branham, S. M. (2024a). ”do you want me to participate or not?”: Investigating the accessibility of software development meetings for blind and low vision professionals. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI ’24, New York, NY, USA. Association for Computing Machinery.

Cha, Y., Jackson, V., Figueira, I., Branham, S. M., and van der Hoek, A. (2024b). Understanding the career mobility of blind and low vision software professionals. In *Proceedings of the 2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering*, CHASE ’24, page 170–181, New York, NY, USA. Association for Computing Machinery.

de Souza, C. S., Leitão, C. F., Prates, R. O., and da Silva, E. J. (2006). The semiotic inspection method. In *Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems*, IHC ’06, page 148–157, New York, NY, USA. Association for Computing Machinery.

Dourish, P. and Bellotti, V. (1992). Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*, CSCW ’92, page 107–114, New York, NY, USA. Association for Computing Machinery.

Ellis, C. A., Gibbs, S. J., and Rein, G. (1991). Groupware: some issues and experiences. *Commun. ACM*, 34(1):39–58.

Escorcer, M., de Queiroz, L., Lima, K., Passos, J. K., Neiva, A., and Soares, L. (2024). Remote work in it: Benefits, challenges and future directions. In *Anais do XV Workshop de Sistemas de Informação*, pages 120–124, Porto Alegre, RS, Brasil. SBC.

Fagerholm, F. and Münch, J. (2012). Developer experience: Concept and definition. In *2012 International Conference on Software and System Process (ICSSP)*, pages 73–77.

Fuks, H., Raposo, A. B., Gerosa, M. A., Pimentel, M., Filippo, D., and Lucena, C. d. (2011). Teorias e modelos de colaboração. *Sistemas colaborativos*, pages 16–33.

Hu, D., Bhuiyan, M. M., Lim, S., Wiese, J., and Lee, S. W. (2024). Unpacking task management tools, values, and worker dynamics. CHIWORK '24, New York, NY, USA. Association for Computing Machinery.

Huff, E. W., Boateng, K., Moster, M., Rodeghero, P., and Brinkley, J. (2020). Examining the work experience of programmers with visual impairments. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 707–711.

Johnson, J., Begel, A., Ladner, R., and Ford, D. (2022). Program-l: Online help seeking behaviors by blind and low vision programmers. In *2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–6.

Kuusinen, K. (2016). Are software developers just users of development tools? assessing developer experience of a graphical user interface designer. In Bogdan, C., Gulliksen, J., Sauer, S., Forbrig, P., Winckler, M., Johnson, C., Palanque, P., Bernhaupt, R., and Kis, F., editors, *Human-Centered and Error-Resilient Systems Development*, pages 215–233, Cham. Springer International Publishing.

Lake, B. and Maidment, D. W. (2023). "is this a new dawn for accessibility?" a qualitative interview study assessing remote working experiences in adults with physical disabilities post covid-19. *WORK*, 76(2):437–451. PMID: 37092205.

Melo, G. (2026). Supplementary inspection material: Accessibility and communicability inspection on microsoft planner. https://anonymous.4open.science/r/Accessibility-and-communicability-inspection-on-Microsoft-Planner-3074/.

Melo, G. L. N., da Silva Menezes, N., dos Santos Maciel, A. C., Teran, L. A., da Rocha, T. Á., de Souza, C. R. B., and Mota, M. P. (2024). Accessibility issues in establishing awareness on remote collaborative software development. *Journal on Interactive Systems*, 15(1):294–310.

Miranda, D., Araújo, J., and Liebel, G. (2024). A conceptual model for web accessibility requirements in agile development. In *2024 IEEE/ACM Workshop on Multidisciplinary, Open, and RElevant Requirements Engineering (MO2RE)*, pages 15–21.

Nascimento, F., Barbosa, P., and Viana, W. (2022). Programando às cegas: investigando a acessibilidade de ambientes de desenvolvimento de software. In *Anais do VII Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software*, pages 1–10, Porto Alegre, RS, Brasil. SBC.

Pandey, M., Bondre, S., O'Modhrain, S., and Oney, S. (2022). Accessibility of ui frameworks and libraries for programmers with visual impairments. In *2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–10.

Pereira, I. C., Gomes, G., Ribeiro, D. M., Souza, A. d., Lucena, D., and Pinto, G. (2023). Supporting the careers of developers with disabilities: Lessons from zup innovation. *IEEE Software*, 40(5):58–65.

Rocha, T., Teran, L., Silva, A., Souza, C., and Mota, M. (2023). Awareness no desenvolvimento colaborativo remoto: Uma perspectiva sobre acessibilidade. In *Anais do*

XVIII Simpósio Brasileiro de Sistemas Colaborativos, pages 140–153, Porto Alegre, RS, Brasil. SBC.

Silva, A. M. d., Sampaio, S. C. B., and Marinho, M. L. (2021). Using software engineering and agile methods to improve inclusion and team diversity. In *Proceedings of the XIX Brazilian Symposium on Software Quality*, SBQS '20, New York, NY, USA. Association for Computing Machinery.

Silva, J. S. e., Gonçalves, R., Branco, F., Pereira, A., Au-Yong-Oliveira, M., and Martins, J. (2019). Accessible software development: a conceptual model proposal. *Universal Access in the Information Society*, 18(3):703 – 716. Cited by: 16.

Sin, J. C. C. and Kathiarayan, V. (2023). The evolution of remote work: analysing strategies for effective virtual team management and collaboration. *Int J Manag Commer Innov*, 11(1):91–101.

Skura, S. d. F., Beal, M. M., and Franciele (2024). Desafios sobre acessibilidade em ferramentas case. In *Anais do XXXV Simpósio Brasileiro de Informática na Educação*, pages 1349–1362, Porto Alegre, RS, Brasil. SBC.

Smela, B., Toumi, M., Świerk, K., Francois, C., Biernikiewicz, M., Clay, E., and Boyer, L. (2023). Rapid literature review: Definition and methodology. *Journal of Market Access & Health Policy*, 11(1).