Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.6287

Assessing the awareness support in collaborative software development tools

Márcio J. Mantau [**Universidade do Estado de Santa Catarina (UDESC)** | ***marcio.mantau@udesc.br***]
Suelyn Concentius [**Universidade do Estado de Santa Catarina (UDESC)** | ***concentius.suelyn@gmail.com***]

Departamento de Engenharia de Software, Universidade do Estado de Santa Catarina (UDESC), Rua Dr. Getúlio Vargas, 2822, Ibirama, SC, 89140-000, Brasil.

Received: 30 July 2025 • **Accepted:** 16 July 2026 • **Published:** 13 August 2026

Abstract Agile methodologies and the growing geographic distribution of teams have driven significant transformations in the software development environment in recent decades. Collaborative tools have emerged as essential solutions for optimizing processes, enhancing communication, and coordinating efforts among team members. Therefore, implementing effective collaborative systems becomes crucial to guarantee project productivity, quality, and success. This study aims to evaluate the support provided by tools used in software development environments, with a particular focus on users' perspectives. We assessed some collaborative environments adopted by development teams, analyzing support for contextual collaboration and shared workspace aspects. We have noticed that an individual's ability to recognize and utilize awareness mechanisms depends on their skills and familiarity with the environment. As a contribution, this study provides a solid foundation for improving collaborative tools by highlighting strengths and weaknesses from the users' perspective. The data obtained can guide developers in implementing features that provide greater support for awareness, thereby facilitating informed decision-making and effective coordination among team members.

Keywords: Awareness, Collaboration, Assessment, Software development

1 Introduction

We can observe a significant evolution in the techniques, tools, and approaches used in the context of software development, especially with the growing adoption of agile practices and methodologies that favor collaboration. With geographically distributed teams, the need for tools that support collaborative environments has become evident. These tools not only optimize processes but also foster a sense of community among team members, which is essential for facing the complex challenges of contemporary development. In this scenario, various tools and resources have been utilized to support collaborative software development, including source code management and version control platforms, communication tools, and task organization and project management tools.

Effective collaboration among participants, especially those working in different geographical locations, requires a clear understanding of the interconnections between activities and tasks. We understand that the collaboration process is the union of individuals or groups to achieve a common goal George [2003] and, for this to occur within the collaborative context, it is necessary to provide aspects of communication, coordination, and cooperation Ellis *et al*. [1991]. Furthermore, to support collaboration, cues/information must be made available in the collaborative environment, allowing participants to communicate, coordinate, and cooperate. This support involves a fundamental element of a collaborative approach: the awareness Dourish and Bellotti [1992]. Awareness is defined as the "understanding of the activities of others, which provides a context for your own activity" Dourish and Bellotti [1992]. While foundational frameworks like Gutwin and Greenberg Gutwin and Greenberg [2002] and Dourish and Bellotti Dourish and Bellotti [1992] have established the theoretical pillars of workspace and group awareness, their practical assessment in modern, technical environments remains a challenge. Awareness is an essential concept in Collaborative Systems Tenenberg *et al*. [2016], a fundamental Gross [2013], or a backbone Mantau and Benitti [2024] aspect of these environments.

While models for assessing awareness have been validated in educational settings and general collaboration scenarios, the software development ecosystem presents unique challenges that necessitate a dedicated investigation. Unlike other domains, contemporary Software Engineering relies on intensely asynchronous collaboration, where awareness extends beyond teammate presence to the profound understanding of complex technical artifacts, such as commit flows, code dependencies, and build statuses. This study addresses a specific gap: the lack of a statistically rigorous metric to measure how technical tools—often strictly execution-oriented—support or neglect the cognitive load required to maintain group awareness.

By applying the Awareness Assessment Model to this scenario, this work contributes new insights by revealing that awareness in technical tools depends not only on the interface but on the maturity of the integration between communication mechanisms and software artifacts. This provides a calibrated scale that enables tool designers and team leads to diagnose collaborative bottlenecks that remain invisible under traditional analysis methods.

As teams adopt collaborative systems to improve how they work together, it is essential to understand how these tools contribute to their effectiveness and efficiency. Providing adequate support for awareness, including communication and coordination mechanisms, has been a research challenge since the emergence of the collaborative systems area Dourish and Bellotti [1992]; Schmidt [2002]. The study of support mechanisms and their influence on collaborative processes remains an open research question Mantau and Benitti [2022a]; Molina *et al*. [2025].

In a literature review on distributed pair programming da Silva Estácio and Prikladnicki [2015], the lack of empirical studies analyzing the use of collaborative systems in software development and the communication and coordination support provided was identified. In a later review, Silva *et al*. [2020] highlights the need for evaluation methods to determine which characteristics of group programming activities and the tools that support them are most effective. In this same context, Molina *et al*. [2025] investigated how communication, coordination, and awareness support mechanisms in collaborative environments can influence the programming task.

The focus of this study is to evaluate how collaborative tools offer support for awareness and, consequently, for collaboration from the user's perspective. For the evaluation, we will adopt the Awareness Assessment Model by Mantau and Benitti [2024]. The model employs a strong statistical approach that utilizes Item Response Theory (IRT), providing a robust strategy for measuring latent characteristics/traits, such as support for awareness and, consequently, collaboration.

IRT enables the calibration of evaluation items, ensuring that the user's perspective is central, thereby facilitating a more precise analysis of the variables that influence collaboration. This approach is valuable since user experiences can vary widely, and capturing these nuances is crucial for improving the support that systems offer. Furthermore, the literature reveals that many existing models lack scientific rigor and systematicity, resulting in superficial evaluations.

According to Collazos *et al*. [2019], few models are dedicated to analyzing awareness support from the user's perspective, with ad-hoc models predominating that do not offer standardized tests for this evaluation. By applying the Awareness Assessment Model presented by Mantau and Benitti [2023a, 2024], this work fills a significant gap in the evaluation of collaboration support in software development tools, as it enables an assessment that considers the complexity of collaborative interactions according to the different competencies of users (profiles). The collection and analysis of data on collaboration support allow us to identify strengths and areas for improvement in tools, thereby guiding development decisions and enhancing collaborative practices. This benefits organizations in terms of efficiency and productivity, serving as a basis for future innovations in cooperative systems.

2 Background

Collaborative platforms are essential for improving teamwork and communication processes, as well as facilitating more efficient information sharing. These systems can be categorized into various groups, each with its distinct characteristics and objectives.

Project management systems are crucial for planning and tracking activities, allowing team members to monitor and manage their tasks dynamically. Tools like **Trello**[1] and **Asana**[2] exemplify this category. According to Arce et al. (2013), the goal of agile development practices is to modify the tasks performed in software development and to add value where flexibility and collaboration prove more effective, efficient, and interesting than the standard rules and guidelines of current development planners.

Communication platforms, such as **Slack**[3] and **Microsoft Teams**[4], are essential for interaction among team members. According to Júnior et al. (2005), communication is the primary collaboration functionality of these systems, where there is a direct relationship between users. Notification services, chat, forums, and other systems enable project team members to converse simultaneously in the same room or channel, dedicating themselves to problem-solving and team decision-making.

Version control systems, such as **Git**[5] and **GitHub**[6], are crucial for software developers as they enable multiple team members to work in parallel without conflicts. In addition to promoting efficient collaboration, these systems also ensure the security and traceability of changes. Every commit made to the code is recorded with a unique signature, allowing the full development history to be consulted, from the most minor changes to major updates. This facilitates code auditing, bug identification, and reverting to previous versions if necessary.

Similarly, just as version control systems are essential for code management, **collaborative documentation tools** are fundamental for the success of software projects, allowing teams to create, share, and maintain information efficiently.

[1] **Trello** is a web-based tool for creating Kanban-style task lists. It is developed and maintained by Atlassian, Corp. Tool URL: `https://trello.com/`.

[2] **Asana** is a project management tool developed for web and mobile platforms. The tool is created and maintained by Asana, Inc. Tool URL: `https://asana.com/`.

[3] **Slack** is a cloud-based team communication platform. Slack Technologies initially developed it and is now maintained by Salesforce, Inc. Tool URL: `https://slack.com/`.

[4] **Microsoft Teams** is a team collaboration application developed by Microsoft Inc. as part of the Microsoft 365 product family, offering workplace chat and video conferencing, file storage, and integration of proprietary and third-party applications and services. Tool URL: `https://teams.microsoft.com/`.

[5] **Git** is a distributed version control system that tracks file versions, often used by programmers to collaboratively develop software by controlling source code. Git is considered free and open-source software, developed and maintained by a community. Tool URL: `https://git-scm.com/`. Community URL: `https://github.com/git/git/graphs/contributors/`

[6] **GitHub** is a developer platform that allows developers to create, store, manage, and share their code. It uses Git software, providing distributed version control with access control, bug tracking, software feature requests, task management, continuous integration, and wikis for each project. Platform URL: `https://github.com/`.

Tools like **Confluence**[7], **Google Docs**[8], and **Notion**[9] facilitate teamwork by enabling simultaneous document creation, comment insertion, and real-time editing. They help centralize knowledge, ensuring that all team members have access to the most up-to-date information, regardless of geographic location.

These tools not only enable real-time collaboration but also ensure the continuous and systematic organization and accessibility of information. Furthermore, similar to Git or GitHub, these platforms maintain a history of changes, allowing users to retrieve previous versions of documents if necessary. Thus, in a software project, both the code and technical documentation can be developed and maintained efficiently, ensuring that knowledge and guidelines are always aligned among all involved.

2.1 Related work

Various studies have addressed the evaluation of awareness and collaboration support in interactive systems, highlighting the importance of methodologies that integrate different types of evaluation. Molina *et al*. [2015] proposed an innovative approach that combines **subjective and objective evaluations**. This methodology enables a comprehensive analysis of collaborative systems, employing techniques such as satisfaction questionnaires, interviews, and eye-tracking tools. Molina et al.'s proposal emphasizes the relevance of **method triangulation**, where heuristic inspection and automatic data logging contribute to a deeper understanding of the user experience in collaborative environments.

Complementing this perspective, Lopez and Guerrero [2017] conducted a systematic review of **collaborative tools** that employ **ubiquitous mechanisms** to promote awareness in the collaborative context. Their work highlights the need for tools that not only facilitate collaboration but also foster a clear understanding of the dynamics and interactions among participants.

Recently, Mantau and Benitti [2024] developed the **"Awareness Assessment Model"**, which measures support for awareness and collaboration from the participants' perspective. In a subsequent study (Mantau and Benitti [2024]), they further developed this analysis, examining how users' awareness affects the effectiveness of interactions in collaborative environments. These works highlight the evolution in evaluation approaches, emphasizing the importance of methodologies that consider both users' subjective experience and objective data, thus contributing to advances in understanding how awareness impacts collaboration.

In this work, an evaluation of awareness support in collaborative environments will be conducted using the **Awareness Assessment Model by Mantau and Benitti [2024]**. The analysis will focus on the awareness mechanisms present in software development tools, which are essential to sustain the three pillars of the 3C collaboration model: communication, coordination, and cooperation. The objective is to understand how these mechanisms impact collaboration and team effectiveness during the development process.

2.2 Discussion

Through the research of related works, it was possible to enumerate some characteristics and compare them with this work. Table 1 presents this comparison, taking into account each aspect conceived at the beginning of the research.

3 The Awareness Assessment Model

The Awareness Assessment Model is explicitly developed for evaluating collaborative systems. It measures their quality by analyzing the awareness information provided by the application. At least one examiner conducts the assessment. Considering the participants' perception as a data source, this instrument enables us to classify the collaborative environment according to its level of awareness quality. The assessment quality scales have been developed adopting the Item Response Theory (IRT) statistical technique Baker and Kim [2017][10]. The model comprises the *Awareness Assessment Process* and *Conceptual View* (see Figure 1).

3.1 The Awareness Assessment Process

The *Awareness Assessment Process* is based on a set of HCI guidelines Barbosa and Silva [2010]; Rogers *et al*. [2013] and is inspired by the evaluation process defined by the standard ISO/IEC 25.040 [2011]. The assessment process comprises three phases: *planning*, *execution*, and *reflection* (see Figure 2).

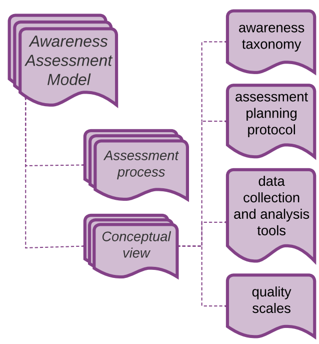


Figure 1. Awareness Assessment Model overview

[7]**Confluence** is a web-based enterprise wiki tool developed by Atlassian Corporation. In recent versions, the Confluence tool has evolved into a key component of an integrated collaboration platform. It has been adapted to work in conjunction with Jira and other Atlassian software products, including Bamboo, Clover, Crowd, Crucible, and Fisheye. Tool URL: `https://www.atlassian.com/software/confluence/`.

[8]**Google Docs** is an online word processor and part of the free Google Docs Editors suite. It is web-based and offered by Google, LLC. Google Docs is accessible via a web browser or mobile app. Tool URL: `https://docs.google.com/`.

[9]**Notion** is a web-based productivity and note-taking application developed by Notion Labs, Inc. Tool URL: `https://www.notion.so/`.

[10]The IRT refers to a family of mathematical models that relate observable variables (e.g., questionnaire items) and hypothetical unobservable traits or aptitudes (e.g., awareness quality). Thus, a stimulus (item) is presented to the subject, and he responds to it, and the response that the subject gives to the item depends on the subject's level in the latent trait or ability Pasquali [2020].

The IRT model is built by executing scripts in R source using the MIRT package (a multidimensional Item Response Theory package for the R environment) Chalmers [2012].

Table 1. Comparison of Proposed Tools

Authors	Considers 3C	Evaluates awareness	*Context-aware*	User perspective	Methodology	Evaluated Scenario(s)
Lopez and Guerrero [2017]	x	x	x	-	-	Systematic Literature Review on awareness support technologies
Mantau and Benitti [2023a]	x	x	x	x	CS	Collaborative learning tools (Moodle), video conferencing, and collaborative text editing tools
Mantau and Benitti [2024]	x	x	x	x	CS	Collaborative learning tools (Moodle), video conferencing, and collaborative text editing tools
This work	x	x	x	x	CS	Software development support tools

Legend: x → presents or contains; - → not observed or detailed; CS → case study.

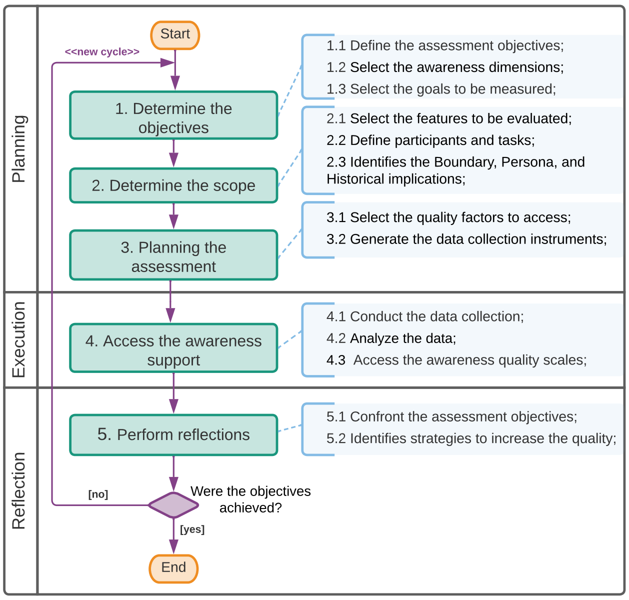


Figure 2. Awareness Assessment Process

Phase 1 - Planning. It refers to activities related to assessment planning and involves three basic steps: determining the assessment objectives, the assessment scope, and the assessment planning. First, the examiner determines the assessment objectives. This is the starting point for building the evaluation approach, aiming to select three essential components: assessment objectives, context, and goals.

- *Activity 1.1. Define the assessment objectives.* This step defines the evaluation goal in terms of the object of study, purpose, perspective, and context BASILI [1992]: the purpose defines the intention of the evaluation; the perspective tells the viewpoint from which the evaluation results are interpreted (e.g., users or experts); and the context is the environment in which the evaluation is performed.
- *Activity 1.2. Select the awareness dimensions.* Identify the related awareness dimensions that will be considered in the assessment. The comprehensive assessment model comprises three primary awareness perspectives, enabling us to evaluate the collaborative environment from each perspective.
- *Activity 1.3. Select the goals to be measured.* For each awareness dimension considered, select which design categories are relevant in the collaborative environment. These design categories represent the specific awareness assessment goals, thereby allowing the model to address the relevant aspects of the application with flexibility and precision.

Second, the examiner determines the scope. This phase represents the detailing of the context in which the evaluation will be carried out, the features of the environment that will be considered, the participants, their respective tasks, and finally, whether the boundary, persona, and historical implications will be considered:

- *Activity 2.1. Select the features to be evaluated.* Select the features or tasks of the collaborative environment to access. In some cases, the target environment can be complex, making it difficult to assess thoroughly, and some parts/features may not be relevant to the intervention. This activity allows building an assessment instrument focused on the relevant/exciting aspects.
- *Activity 2.2. Define participants and tasks.* Identify the participants involved in the evaluation process and the tasks that must be carried out within the collaborative environment. This evaluation instrument was designed to enable evaluation by specialists, involving users, or both. Thus, it is essential to clarify who is involved and what their roles and tasks are within the environment.
- *Activity 2.3. Identifies the Boundary, Persona, and Historical implications.* The awareness information can be categorized into the perspectives of Boundary, Persona, and Historical Awareness.

Third, the examiner determines the planning assessment. This phase represents the planning stage documentation, where it is established at what moment of the construction or use of the collaborative environment the evaluation will be carried out and which quality factors will be considered. Therefore, the data collection instrument is prepared, including the assessment purpose, methods, life cycle, and artifacts:

- *Activity 3.1. Select the quality factors to access.* The quality aspects define the additional quality factors under analysis in the evaluation, that is, to use this model together with another evaluation approach (e.g., usability, demographic, user experience).
- *Activity 3.2. Generate the data collection instrument.* This activity aims to prepare or customize the data collection instrument, considering the results of activities 1.2, 1.3, and 2.3.

Phase 2 - Execution. After the planning stage, the collaborative system assessment is done by adopting data collection and analysis instruments. In this phase, the examiner performs the awareness assessment of the target collaborative environment. The awareness support provided by the target environment is achieved through the use of data collection and analysis tools.

Phase 3 - Reflection. Once the evaluation is completed and the data is analyzed, the evaluator conducts reflections to gather feedback and identify strategies for improving awareness quality. The main objectives are confronted, and the awareness of quality factors is checked. If unmet, the examiner determines strategies to increase the quality of awareness mechanisms, and a new intervention can be planned. This process enables both the assessment of collaborative environments through awareness mechanisms and the improvement by prompting reflection on results.

3.2 The Conceptual View

The *Conceptual View* is a framework composed of the following artifacts [11]:

i) The awareness taxonomy is constituted of three main awareness dimensions, their respective design categories, and respective design elements, combined with three additional dimensions that directly imply the design categories and awareness elements: persona, boundary, and historical awareness dimensions (full reference can be found at Mantau and Benitti [2022b]);
ii) The assessment planning protocol represents an instrument for planning and executing the assessment process. This artifact helps in defining the assessment objectives, factors to be measured, awareness dimensions, life-cycle phases in which the awareness assessment will be applied, and so on Mantau and Benitti [2024];
iii) The data collection and analysis tools present a set of support artifacts for conducting the collection and compilation of data obtained by interventions Mantau and Benitti [2023b];
iv) The assessment scales and measurement items represent useful elements for analyzing and classifying the collaborative environment at an awareness quality level from the participants' perspective Mantau and Benitti [2023b, 2024].

[11] A full reference of *Conceptual View* artifacts can be found at our Awareness Assessment Model repository Mantau and Benitti [2023b].

4 Method

This study was structured into two steps.

First, we'll use an online form to identify a set of **collaborative systems** used in software development. These systems should support communication, cooperation, and coordination—the 3C model of collaboration Fuks *et al*. [2008]; Gutwin *et al*. [2016]; Collazos *et al*. [2019]. This form will also help us pinpoint the main features and functionalities commonly used in these systems to support software development activities.

Then, based on the identified systems, we'll evaluate the awareness support and, consequently, the collaboration provided from the user's perspective. This stage will involve a case study Wohlin *et al*. [2012]; Wohlin [2014], using the Awareness Assessment Model presented by Mantau and Benitti [2023a] as its foundation. To conduct this evaluation, we're using the process proposed by Mantau and Benitti [2024][12], as shown in Figure 2.

4.1 Data collection

The data collection process was conducted through a voluntary recruitment strategy, targeting both undergraduate students and market professionals from the regional software development ecosystem. Participation was strictly voluntary, and all respondents were informed about the study's objectives and the confidential nature of their data, following standard ethical procedures for academic research.

A total of 333 valid observations were collected, forming the basis for the Item Response Theory (IRT) analysis. It is important to distinguish the units of analysis: while the *number of participants* refers to the unique individuals who responded to the survey, the *number of observations* corresponds to the total instances of tool-user interactions evaluated, as each participant could evaluate multiple software tools. In this study, the *observation* serves as the primary unit of analysis for the IRT model to estimate the latent trait (θ) related to awareness perception. The sample characteristics include a predominant age group of 18 to 28 years (64%) and a diverse range of roles, including developers, managers, and students.

4.2 Test equating

To ensure that the parameters from the regional case study and the reference dataset are placed on a common scale, we employed IRT test equating. This statistical instrument allows for the production of a linkage between scores on different test forms so that they can be used interchangeably, as if they had originated from the same instrument Dorans *et al*. [2010]; Sansivieri *et al*. [2017].

Following the principles defined by [Holland *et al*., 2006], we integrated the current case study data ($n = 333$) with a larger reference dataset from [Mantau and Benitti, 2023b], which contains 1,280 previously compiled observations. This

[12] The complete documentation for the **Percepção** Evaluation Model, including all model artifacts and usage examples, can be accessed in Mantau and Benitti (2023b); The full specification of the evaluation process is presented in Mantau and Benitti (2023, 2024).

strategy allows us to place parameters from different populations or study instances on a common scale or metric de Andrade *et al*. [2000].

This approach satisfies the five fundamental requirements for score linking Holland *et al*. [2006]:

i) *The equal construct requirement*: Both the case study observations ($n = 333$) and the reference dataset Mantau and Benitti [2023b] ($n = 1280$) strictly measure the same latent trait: the awareness of collaboration support.
ii) *The equal reliability requirement*: The instruments were designed to maintain the same level of reliability across different applications, ensuring consistent measurement precision.
iii) *The symmetry requirement*: The equating transformation used to map the scores of the case study to the reference scale is the inverse of the transformation mapping the reference scores back to the case study.
iv) *The equity requirement*: The model ensures that it is a matter of indifference to a participant which specific set of items or tool evaluations they perform, as the latent trait estimation (θ) remains equitable.
v) *The population invariance requirement*: The equating function remains stable regardless of the subpopulation (students or professionals) from which it is derived within the regional software ecosystem.

By utilizing a Single Group (SG) design for the case study and integrating it through equating by common elements, we ensure that even with a targeted regional sample, the item parameters are expressed on a single, calibrated metric de Andrade *et al*. [2000]. This strategy provides the necessary statistical power to justify the sample size and allows for a precise estimation of difficulty (b_i) and discrimination (a_i) parameters.

5 Case study

The study was conducted in two stages. In the first stage, a questionnaire was administered to identify and list the most commonly used software development tools. Subsequently, the second stage focused on evaluating the awareness and collaboration support offered by these environments from the direct perspective of users.

5.1 Identifying the collaborative tools

First, we conducted a quantitative survey using a structured online questionnaire. The target audience comprised undergraduate students in the field of computing and professionals working in software development in the Alto Vale do Itajaí region, Santa Catarina. This choice aimed to ensure the participation of individuals directly involved in the software development context, capable of providing relevant information aligned with the research objectives.

The questionnaire was organized into four main sections. The first section collected demographic data, including age, gender, and job title, to characterize the participants' profiles and understand the diversity present in the sample. The sample consisted of 22 respondents, aged between 18 and 39, working in roles such as systems analyst, Quality Assurance (QA), and technical lead.

We structured the questionnaire around the pillars of collaboration, specifically Communication, Coordination, and Cooperation. For each of these aspects, participants were asked about the specific tools they use, which tools they use most frequently, and the most relevant functionalities for their professional activities.

In the section dedicated to Communication, the goal was to understand how teams efficiently exchange information, both synchronously and asynchronously. Among the most cited tools were Slack, Microsoft Teams, Google Meet, and Discord, with an emphasis on functionalities such as instant messaging, voice and video calls, as well as screen and file sharing.

Regarding Coordination, the objective was to identify how teams plan, organize, monitor, and control their tasks and deliveries. Approximately 73% of the participants reported using specific tools for these purposes, with Trello, Jira, and ClickUp being the most commonly used. These platforms offer essential features, including Kanban boards, backlog control, sprint management, priority definition, automatic notifications, and tracking dashboards, all of which contribute to the operational and strategic alignment of teams.

Finally, the section concerning Cooperation focused on mapping how team members collaborate in the creation, editing, and maintenance of shared artifacts, such as technical documentation and source code. In this context, Notion was widely mentioned for its collaborative document management capabilities, which enable simultaneous editing and version control. GitHub and GitLab stood out as predominant platforms for distributed versioning, continuous integration, collaborative code review, and pipeline automation, facilitating secure and traceable joint work.

The analysis of the data obtained revealed not only the consistent adoption of collaborative tools but also the value placed on the specific functionalities they offer. This understanding is crucial for evaluating the degree to which these tools meet the real needs of software development professionals, as well as for identifying possible opportunities for improvements in the collaborative support ecosystem.

5.2 Assessing the awareness support

The second stage of the research consisted of a case study conducted following the evaluation process outlined by Mantau and Benitti [2023a, 2024]. The focus was to investigate collaborative awareness support, specifically in the dimensions of workspace awareness and collaboration awareness, from the user's perspective. The process followed a three-stage structure: Planning, Execution, and Reflection, as detailed by Mantau and Benitti [2024].

During the Planning stage, the case study was structured according to the Awareness Assessment Model. The scope and objectives of the evaluation were defined, guided by the collaborative tools identified in the initial phase of the study. In these tools, the aim was to analyze and identify the awareness dimensions for the contexts of workspace awareness and collaboration awareness. Based on these definitions, data collection instruments were developed, consisting of demo-

Table 2. Planning protocol template

1. Determine the objectives	**Step 1.1** – Define the assessment objectives in terms of the object of study, purpose, perspective, and context. → *Object of study:* [Software development tools] → *Purpose:* Evaluate the collaboration support provided by the collaborative environment through its presented awareness mechanisms] → *Perspective:* [Participant's perspective (analysts, developers, designers, etc.).] → *Context:* [General-purpose software development environment, such as Slack, Jira, GitHub, etc.] **Step 1.2** – Select the awareness dimensions. Identify the related awareness dimensions that will be considered in the assessment. → *Awareness dimensions:* [Workspace and collaboration awareness] **Step 1.3** – Select the goals to be measured. We select which design categories are relevant in the collaborative environment for each awareness dimension. → *Goals:* [Access all categories of awareness support described in each dimension]
2. Determine the scope	**Step 2.1** – Select the features to evaluate. Select the functionalities or tasks within the collaborative environment that will be the object of the assessment. → *Features:* [communication tools, project management platforms, and code repositories] **Step 2.2** – Define the participants involved and the tasks that must be carried out within the collaborative environment. → *Participants:* [Undergraduate students in computing and professionals from software development companies] → *Tasks description:* [Participants answer the questions considering their perspectives on each tool.] **Step 2.3** – Identifies the Boundary (physical, virtual, or both), Persona (individual, other participants, group as a whole, or system), and Historical (past, present, future) implications. → *Implications:* [None]
3. Planning the intervention	**Step 3.1** – Select the additional factors to assess (like demographic, usability, and UX). → *Additional factors:* [None] **Step 3.2** – Generate the data collection instrument based on activities 1.2, 1.3, and 2.3.

graphic questions and 30 specific items from the Awareness Inventory, covering various elements of collaboration support.

The case study was developed according to the evaluation planning protocol model as described in Table 2 This protocol describes the execution plan for the proposed evaluation case study and other information regarding the selection of tools to be evaluated, the analysis criteria, and the data collection methods, such as interviews and questionnaires applied to users. During the execution of the case study, each tool was analyzed in terms of its awareness support, considering the elements of the 3C collaboration model: communication, coordination, and cooperation. All stages of this research, including data collection and analysis instruments, as well as other artifacts, are available in the Zenodo repository Concentius and Mantau [2025].

The Execution phase involved the application of eight distinct questionnaires, which generated a total of 333 observations. The participants were software development professionals who evaluated tools they use daily, such as Slack, Discord, Trello, and Google Meet. Demographic data indicated that 64% of participants were between 18 and 28 years old. Information was also collected on participants' familiarity with awareness and collaboration concepts, as well as with the evaluated environments. Among the analyzed environments, Skype stood out for the greater ease of recognizing awareness elements, according to the average scores. In contrast, tools like Jira proved more challenging for participants in identifying these elements.

During the Reflection stage, the data were examined concerning the objectives established in the planning. A clear trend was observed: participants with greater prior knowledge of awareness and collaboration concepts achieved superior results in identifying support mechanisms in the tools. Those with less familiarity, on the other hand, faced greater difficulties in recognizing these elements, highlighting the crucial importance of prior knowledge in perceiving collaboration in digital environments.

The analysis also revealed which tools offer more effective support for collaborative awareness, based on users' experiences. Tools that facilitate dynamic communication and provide clear visualization of progress and responsibilities received more positive evaluations. However, technically focused tools, such as task control and versioning, require greater knowledge for their support features to be identified appropriately.

5.3 Model calibration

In the IRT model, one of the most important steps is to estimate item parameters and participant abilities; this process is known as calibration de Andrade *et al*. [2000]. Calibration can be performed by knowing the item parameters, knowing the participant's abilities, or estimating both simultaneously (the most common situation). In the first strategy, only abilities can be estimated; in the second, only item parameters are estimated.

For this step, we combined the calibration dataset from Mantau and Benitti [2024] with the 333 observations obtained through the collection instruments (questionnaires). The observations were collected using eight evaluation questionnaires and eight groups of participants (software development teams), each group answering a specific questionnaire. We assumed the estimation of participant abilities and item parameters as a calibration strategy and the IRT multigroup estimation method Chalmers [2012].

Multigroup calibration was performed by maximum likelihood analysis for polytomous data (Samejima's graded response model Samejima [1969]) using the Metropolis-Hastings Robbins-Monro (MHRM) algorithm approach Cai [2010]. To calibrate the 333 observations collected to the Awareness Assessment Model, we considered each of the scenarios as a distinct evaluation group: calibration dataset from Mantau and Benitti [2024] (group 0), and groups 1 to 8 for each distinct participant group. In this scenario, two or more groups perform two or more tests, only partially different (with some common items).

In this configuration, the common items across different tests allow all parameters to be on the same scale at the end of the estimation process. Since a set of items connects the different populations, the observations can be evaluated under the same scale de Andrade *et al*. [2000].

We ran the IRT *script* available in the Awareness Model repository Mantau and Benitti [2023b]. Similar to the scenarios presented in Mantau and Benitti [2024], we interpreted the IRT discrimination (a) and difficulty (b) values, disregarding items outside the intervals $a = [0.65, 4.0]$ and $b = [-4.0, +4.0]$. We also analyzed the observed frequencies of each response category and, for all questionnaire items, the observed frequency was ≥ 10.

5.4 Demographic analysis

We generated frequency histograms based on participants' scores and demographic facets (Figures **??** to **??**). To verify if the model shows distinction in the discrimination (or mean) values of different groups, we also calculated the normal distribution and the average score grouped by each demographic perspective.

As demographic data, we collected information on age, gender, preferred environment, experience with collaborative environments, and individual knowledge of awareness and collaboration concepts. The histogram for each demographic facet is primarily within the individual score limits, where the model is representative (dotted vertical line).

5.4.1 Age facet

The evaluation of age distribution graphs enables verification of the demographic validity of the model, ensuring that the estimation of abilities remains consistent across different age groups. The age score graph, presented in Figure 3a, shows the frequency of participants by estimated ability levels, segmented by age group. The graph of normal and cumulative curves in Figure 3b displays the statistical distribution of these abilities, allowing observation of the shape and behavior of data dispersion among groups.

The results indicate a higher concentration of participants between 18 and 28 years old (214 observations), followed by the 29-39 years group (72 observations) and 40-50 years group (26 observations). The limits of the age groups (under 18 and over 50 years) appear with low frequency and more dispersed scores.

The visualization of the curves shows that the central ranges (18 to 50 years) follow a nearly normal distribution, with means close to zero and controlled variance, confirming the model's validity in these groups. Less representative groups, however, show curves slightly shifted to the left, suggesting a tendency for lower performance, which may be due to the reduced sample size, as statistically expected.

Overall, the data confirm good model stability in the main age groups of the sample, indicating that age does not significantly interfere with ability estimation, especially for the most representative ranges.

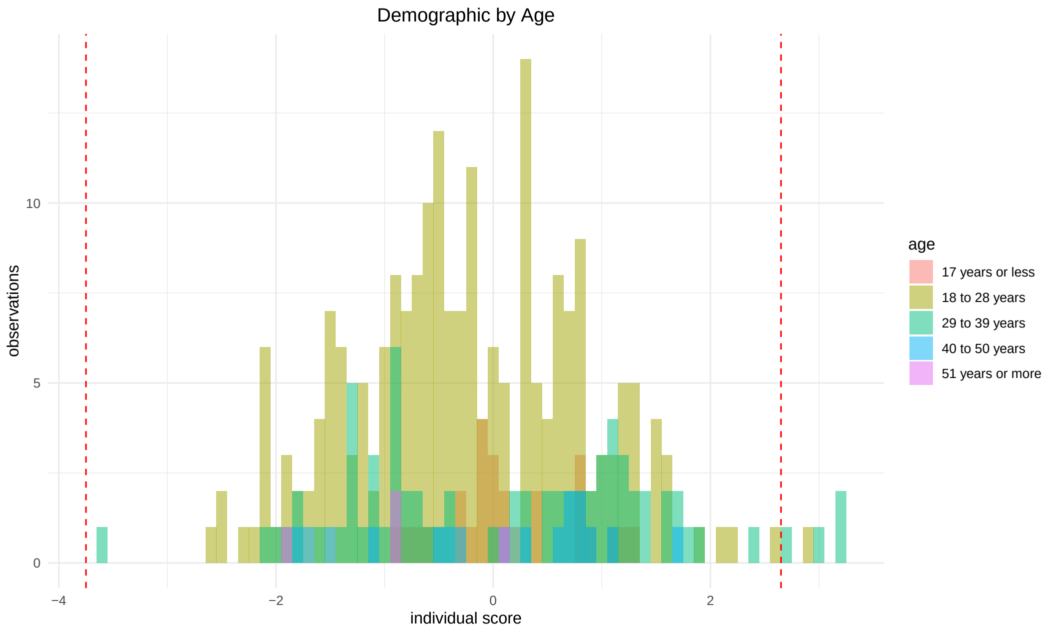
5.4.2 Gender facet

The analysis of frequency distribution by gender verifies the consistency of estimated abilities across groups, validating the model's impartiality. This ensures the model exhibits no gender bias, guaranteeing fair and representative estimates for each group. Figure 4a details participants' estimated ability levels by declared gender. Figure 4b further illustrates the score distribution's shape for each gender using normal and cumulative curves.

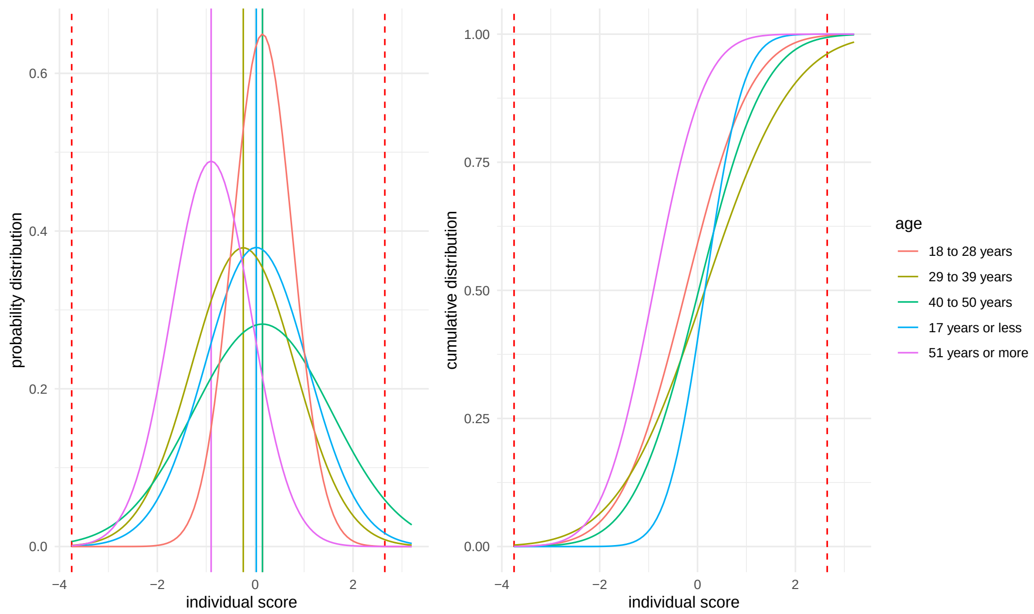
The data reveal that male and female groups are the most representative in the sample, showing a significant number of participants (203 and 128 observations, respectively). In these groups, a balanced distribution of mean score values is observed. The proximity between the distribution curves reinforces the absence of a relevant gender bias, demonstrating that the model evaluates abilities fairly, without discriminating against scores obtained due to this demographic facet.

On the other hand, the group identified as "other" has a considerably smaller sample, which naturally limits the precision of estimates for this segment. The scores of this group show greater dispersion and a slight shift of the curves to the left, suggesting a tendency towards lower scores compared to the male and female groups.

In summary, graphical analysis confirms the model's consistency and impartiality in evaluating abilities across most



(a) Score



(b) Normal curves

Figure 3. Demographic distribution of individual scores by age

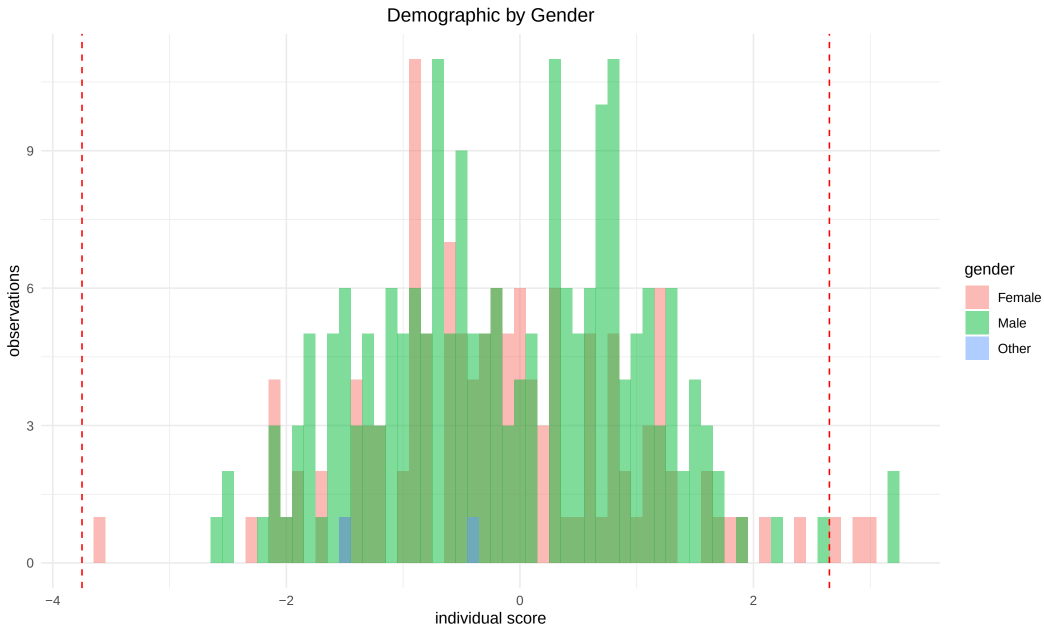
genders. This approach ensures transparent and rigorous data interpretation, enhancing the reliability of conclusions.

5.4.3 Evaluated Environment facet

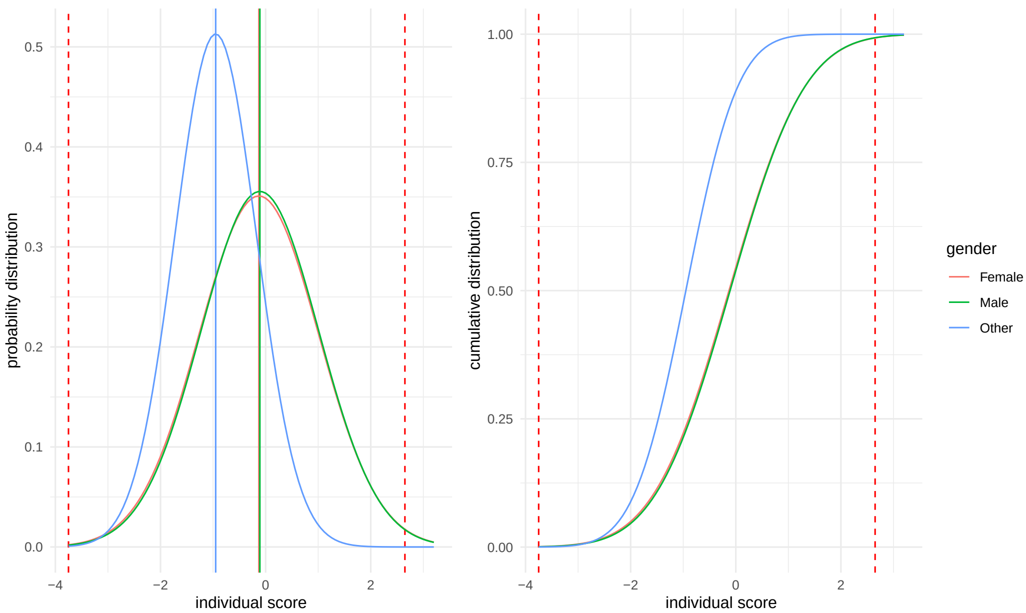
The demographic analysis of collaborative environments used by participants clarifies how abilities vary across different tools. Figure 5a displays individual scores grouped by environment (e.g., Slack, Jira, GitHub). Figure 5b further illustrates the statistical distribution of abilities within each group, highlighting significant variations.

Results indicate clear distinctions between collaborative environments. Tools like Discord, Google Meet, Microsoft Teams, and Skype show scores near zero or positive, suggesting that users of these platforms are more easily able to recognize awareness mechanisms. Conversely, Jira, GitHub/GitLab, Trello, and WhatsApp exhibited lower scores and left-shifted distributions, indicating greater difficulty for users in perceiving awareness elements. This is likely due to their more technical or asynchronous nature, which prioritizes task management and version control over continuous visual support and immediate interaction that boosts awareness.

The distribution curves reinforce these findings: better-performing environments show centrally concentrated curves, while lower-scoring environments have curves shifted towards negative values. This confirms significant differences in awareness support among collaborative tools, suggesting some platforms are more intuitive and effective at promoting situational awareness during group work.



(a) Score



(b) Normal curves

Figure 4. Demographic distribution of individual scores by gender

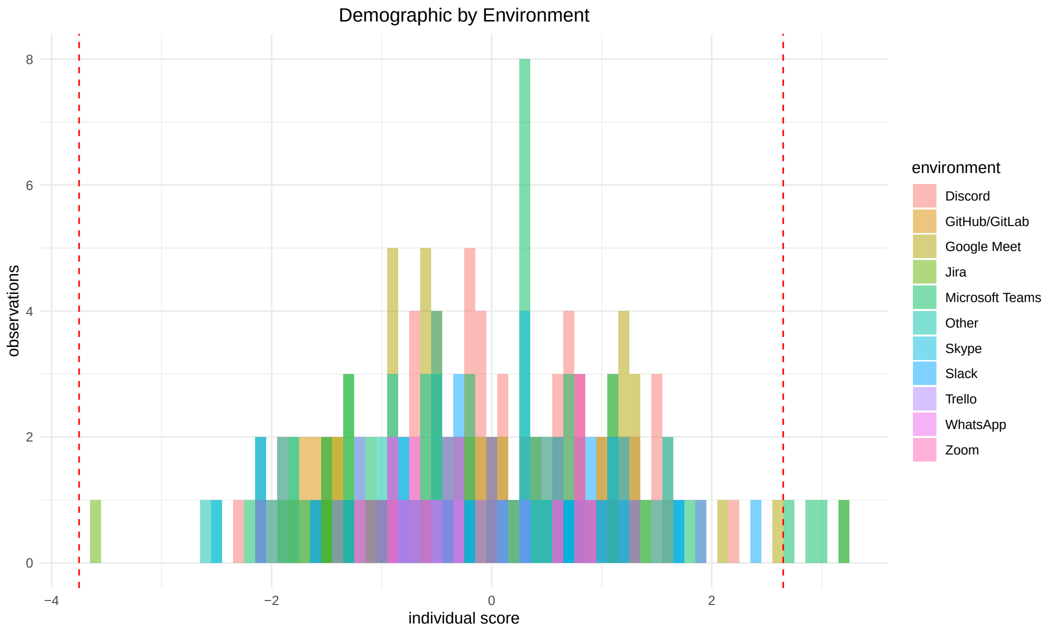
5.4.4 Familiarity with the Environment facet

Figure **??** illustrates how participants' familiarity with the collaborative environment impacts their abilities. Figure 6a shows scores by familiarity level (novices to experts) and participant frequency. Figure 6b presents a statistical analysis of these distributions using normal and cumulative curves, which reveal performance patterns across different familiarity levels.

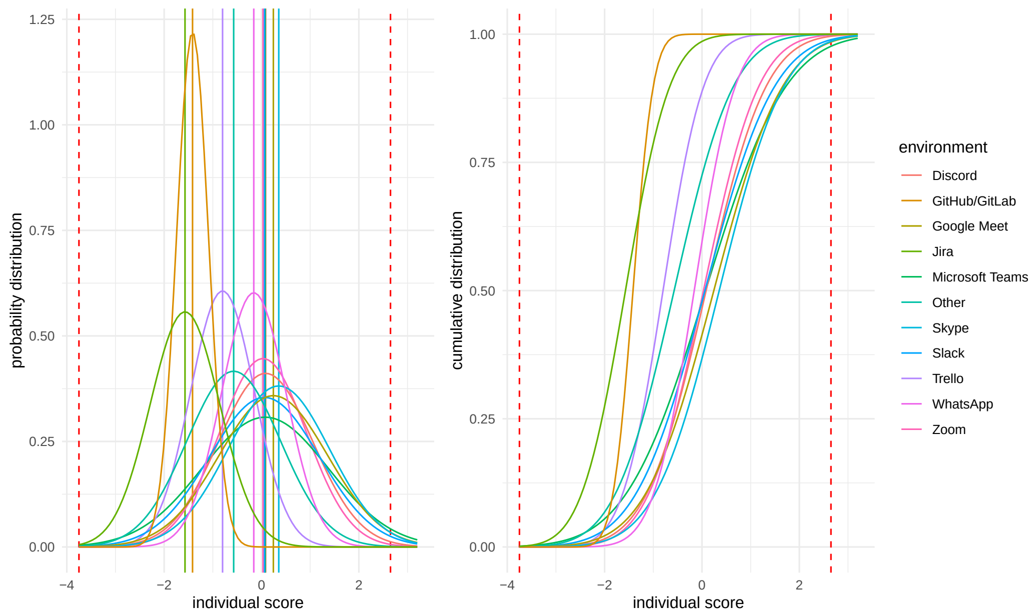
In summary, the data consistently shows a direct correlation between familiarity and ability scores. As participants' familiarity with the collaborative environment increases, their scores rise, causing the histograms to shift visibly to the right on the scale. This suggests that users with more experience achieve higher scores, indicating a better understanding of the evaluated mechanisms.

Delving deeper into the normal curves, the intermediate and advanced familiarity groups display mean scores clustering around zero or slightly positive values. This precisely illustrates their proficient understanding of the collaborative features and their awareness support. Conversely, initial familiarity groups consistently maintain negative mean scores, clearly demonstrating that less practical experience directly correlates with a reduced capacity to perceive and utilize the environment's awareness mechanisms.

Moreover, we observe the stability of the distributions for intermediate and advanced groups. Their density curves are closely aligned, underscoring a greater consistency and predictability in their estimated abilities. In contrast, participants



(a) Score



(b) Normal curves

Figure 5. Demographic distribution of individual scores by evaluated environment

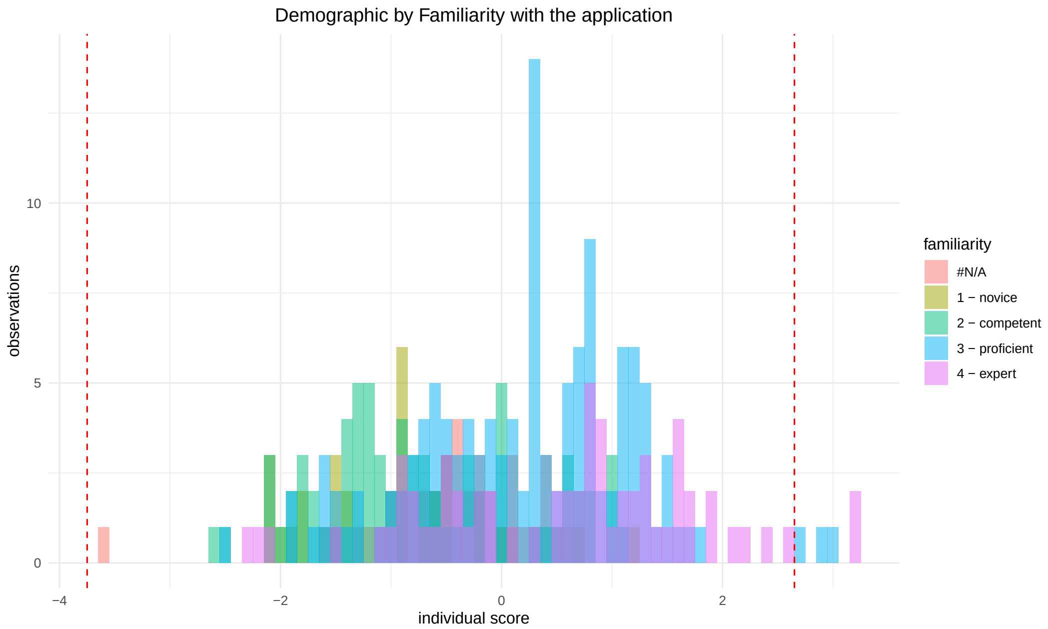
with less familiarity exhibit significantly greater dispersion in their scores, suggesting a more nascent and limited understanding of the collaborative resources they evaluated, and highlighting a broader range of performance within less experienced groups.

5.4.5 Familiarity with Collaboration Concepts facet

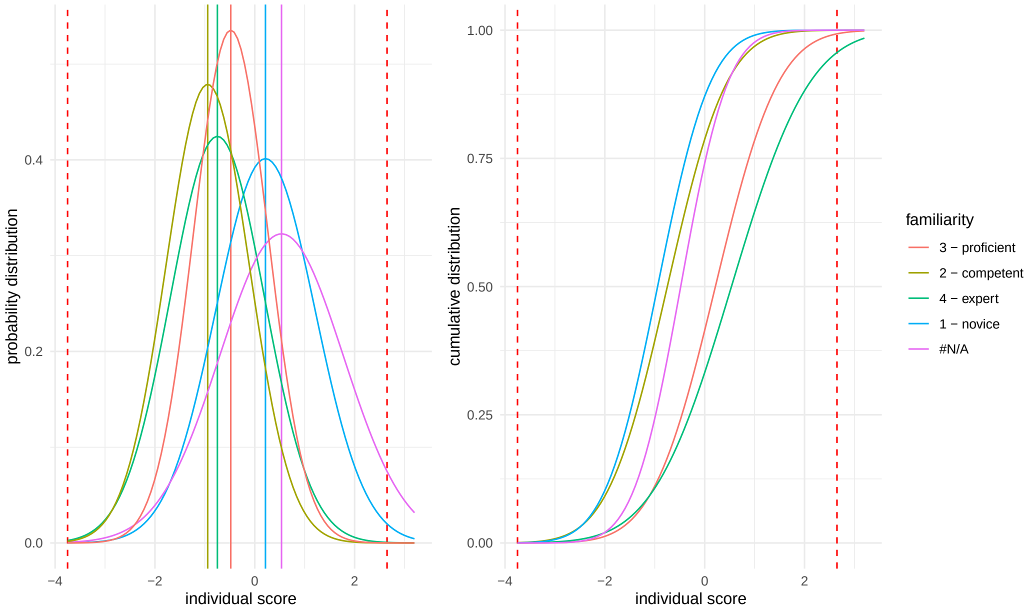
Figure 7a presents two graphs illustrating the relationship between participants' familiarity with collaboration concepts and their ability levels (categorized as novice, competent, proficient, expert, and N/A). Figure 7b shows the distribution of individual scores by familiarity category. The second graph, displaying normal and cumulative curves for each category, provides a detailed statistical visualization of the positioning of these categories on the scale.

As familiarity with collaboration concepts increases, scores shift rightward on the scale, indicating higher ability levels. Proficient and expert participants predominantly showed near-zero or positive scores, suggesting easier recognition of collaboration support mechanisms. Conversely, novice and competent groups concentrated their scores in negative ranges, indicating greater difficulty in identifying these mechanisms. For the N/A group (unanswered familiarity), results fall into an intermediate range, between the more and less familiarized groups, likely reflecting partial knowledge or difficulty in self-assessing understanding.

The probability distribution and cumulative curves reinforce these observations. More familiarized groups (profi-



(a) Score



(b) Normal curves

Figure 6. Demographic distribution of individual scores by familiarity with the environment

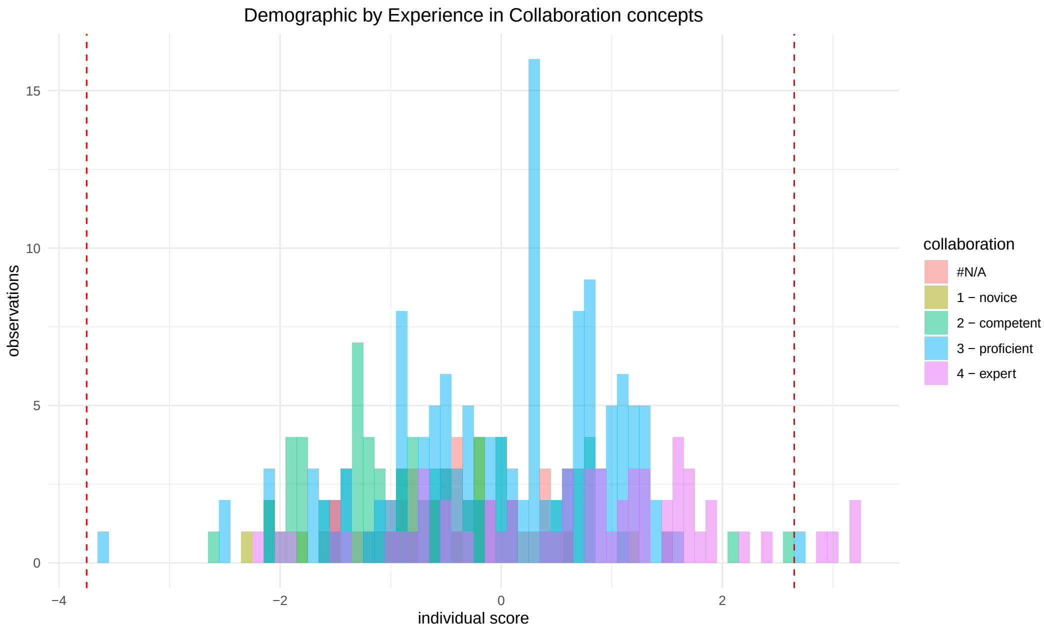
cient and expert) show curves notably shifted to the right with higher ability concentrations. In contrast, less familiar groups (novice and competent) have curves that are more concentrated to the left, indicating a lower perceived ability. Cumulative curves consistently follow this pattern, confirming clear distinctions between profiles and validating the impact of familiarity on performance.

5.4.6 Familiarity with Awareness Concepts facet

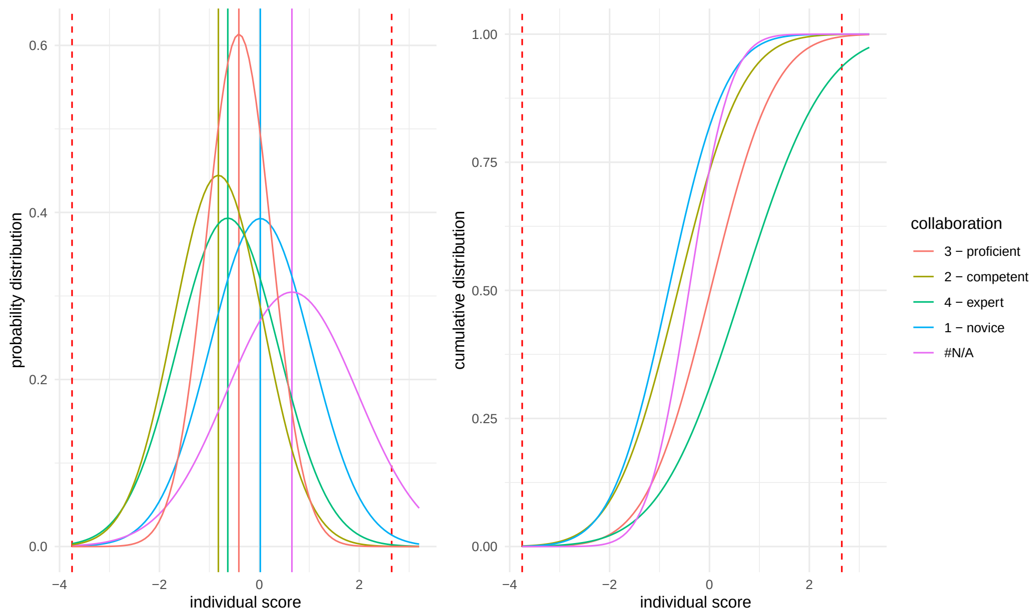
Figure 8a presents two graphs illustrating the relationship between participants' familiarity with awareness concepts and their ability levels (categorized as novice, competent, proficient, expert, and N/A). Figure 8b shows the distribution of individual scores by familiarity category. The second graph, displaying normal and cumulative curves for each category, provides a detailed statistical visualization of the positioning of these categories on the scale.

As familiarity with awareness concepts increases, scores shift rightward on the scale, indicating higher ability levels. Proficient and expert participants predominantly showed near-zero or positive scores, suggesting easier recognition of collaboration support mechanisms. Conversely, novice and competent groups concentrated their scores in negative ranges, indicating greater difficulty in identifying these mechanisms.

For the N/A group (unanswered familiarity), results fall into an intermediate range, between the more and less familiarized groups, likely reflecting partial knowledge or difficulty in self-assessing understanding.



(a) Score



(b) Normal curves

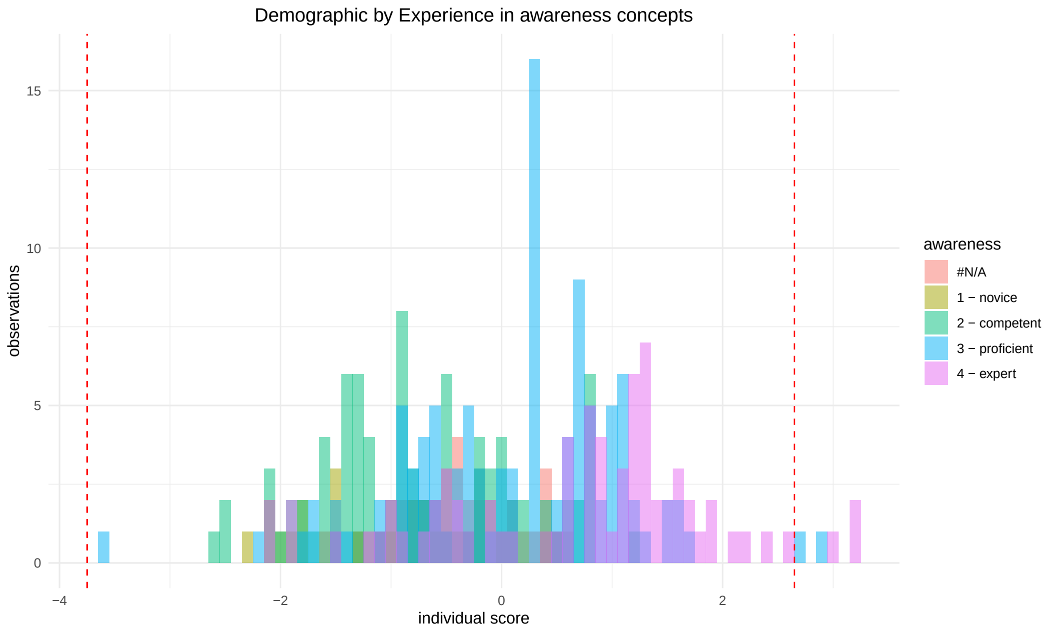
Figure 7. Demographic distribution of individual scores by familiarity with collaboration concepts

The probability distribution and cumulative curves reinforce these observations. More familiarized groups (proficient and expert) show curves notably shifted to the right with higher ability concentrations. In contrast, less familiar groups (novice and competent) have curves that are more concentrated to the left, indicating a lower perceived ability. Cumulative curves consistently follow this pattern, confirming clear distinctions between profiles and validating the impact of familiarity on performance.

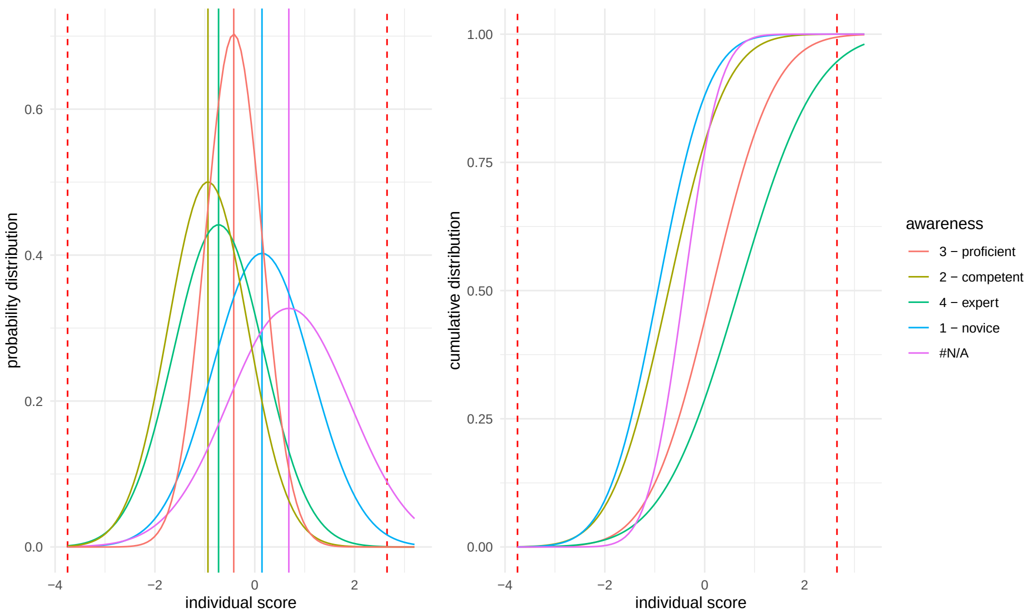
5.5 Demographic interpretation

As depicted in Figures **??** to **??**, the normal curves generated for each of the main demographic groups exhibited significant proximity, reinforcing the consistency of the developed model with behaviors previously observed in similar scenarios Mantau and Benitti [2024]. This analysis confirmed that, concerning age and gender, the model did not exhibit differential behaviors within these robustly sampled subgroups of individuals.

The exceptions were observed in groups with limited sample data (less than 1% of observations), specifically for the age facet (individuals aged 51 or more) and the gender facet (individuals who did not respond or selected ”other” options). In these cases, both the normal distribution and the cumulative probability functions (Figures 3b and 4b) displayed a pronounced leftward shift, indicating a slightly lower estimated ability compared to the other observed groups. This deviation is likely attributable to the small sample sizes, which can statistically limit the precision and generalizability of the



(a) Score



(b) Normal curves

Figure 8. Demographic distribution of individual scores by familiarity with awareness concepts

estimates for these specific segments, rather than suggesting an inherent lower ability.

For all other sufficiently represented groups, the uniform behavior of the sigmoid function suggests that the model does not significantly differentiate between the discrimination (a - the sigmoid slope) and difficulty (b - the sigmoid midpoint) parameters, underscoring the model's robustness and impartiality across the main demographic cohorts.

When comparing the scores grouped by preferred environment (Figure **??**), a clear distinction emerged among collaborative tools regarding the ease with which awareness mechanisms were perceived. Environments such as Jira, GitHub/GitLab, Trello, and WhatsApp exhibited slight differences in their average difficulty parameters (with mean θ values of -1.562, -1.404, -0.799, and -0.162, respectively). These lower, often negative, mean scores strongly suggest that users experienced greater difficulty in readily recognizing or interpreting awareness mechanisms within these specific tools.

This challenge can be fundamentally attributed to their design: these platforms frequently prioritize technical task management, version control, or asynchronous communication, where awareness cues might be less explicitly displayed or require more active user interpretation. Conversely, environments such as Skype, Google Meet, Discord, and Microsoft Teams showed sigmoid curves notably shifted to the right (with mean θ values of 0.340, 0.243, 0.089, and 0.072, respectively). This consistently positive average θ demonstrates that, in general, participants found it significantly easier to identify the available awareness elements in these communication-centric tools.

Their design, often emphasizing real-time presence, chat interactions, and visual cues, inherently promotes a more immediate and intuitive perception of activity and presence within the collaborative space. This insight is crucial for both tool selection, aligning platform capabilities with team awareness needs, and for designing targeted user training to optimize awareness perception across different tool types.

The analysis of participants' individual ability histograms (Figures 6a, 7a, and 8a), about their familiarity with the environment, collaboration concepts, or awareness concepts, revealed a compelling pattern. Both the normal distribution and cumulative probability curves (Figures 6b, 7b and 8b) consistently aligned with the participants' self-reported levels of familiarity. The observed frequencies in the histograms indicated a normal distribution for all groups, spanning the entire spectrum of the ability scale, suggesting that the model effectively captures the nuances of familiarity.

As familiarity increased, the histograms consistently shifted towards the right of the scale, signaling that more experienced users consistently achieved higher scores. Specifically, the intermediate and advanced familiarity groups predominantly displayed mean scores clustering around zero or slightly positive, indicating a proficient understanding and potentially a more nuanced interpretation and effective utilization of collaborative features, along with their underlying awareness support.

In contrast, initial familiarity groups consistently concentrated their scores in the negative ranges, clearly demonstrating that less practical experience or conceptual knowledge directly correlates with a reduced capacity to perceive and interpret the environment's awareness mechanisms. For participants who did not provide information for a particular familiarity facet (the N/A group), their results consistently fell into an intermediate range. This likely reflects a diverse underlying knowledge base, possibly comprising partial understanding or a general difficulty in accurately self-assessing their degree of familiarity with the concept.

This confirms that the model does not differentiate the scale based solely on demographic group membership; instead, the most significant distinguishing factor among individuals is the evaluated latent trait—their intrinsic ability to perceive and interpret awareness, heavily influenced by their familiarity and experience with both the tools and the underlying collaborative concepts.

5.6 Awareness scale

For the 333 observations considered, we estimated the mean scores for each of the collected demographic facets. Table 3 presents the average ability as a function of age, gender, familiarity (with awareness concepts, collaboration aspects, and the environment), and the evaluated environment.

Regarding gender, we collected 203 male observations (61%) and 128 female observations (38%); two participants did not answer this question or chose the option "other gender". A total of 214 observations were collected from individuals aged 18 to 28 (64%), 72 from individuals aged 29 to 39 (22%), and 17 from individuals aged 40 to 50 (5%). A minority sample included four individuals over 50 years old (<1%) and 26 observations from individuals under 18 years old (8%).

Applying the support scale generation formulas defined in the evaluation model Mantau and Benitti [2024], we estimated the probability scales $P_{i,k}(\theta_j)$ for each evaluation element. The generated scale spans a coverage interval of $[-4.0, +4.0]$. Notably, the model demonstrates representativeness primarily within the interval of $[-3.75, +2.65]$, effectively covering the scores of most individuals while accounting for outliers (less than 1%) with unusually lower or higher abilities. Figure **??** visually presents these probability scales, generated for each evaluation item across the considered awareness dimensions.

For each evaluation item, the scale illustrates the probability that a participant (j) with a specific competency score (θ) will recognize the available awareness information (k) within item (i). The distinct segments within the graph bars visually represent the participant's probable response ($P_{i,k}$) to each statement. We adopted the standard (μ, σ) scale[13], normalized with a mean (μ) of 0 and a standard deviation (σ) of 1.

As exemplified in Figure **??**, individuals demonstrating lower scores (reflecting lower ability or competency) generally face greater challenges in recognizing the available awareness elements. Consequently, they are more inclined to disagree with the perceived presence of these elements within the application. Conversely, participants with higher scores on the scale exhibit a greater likelihood of recognizing the support elements presented by the application, leading to a more pronounced agreement in their judgment of the items.

This observed gradual transition signifies a crucial insight: participants with greater familiarity or a deeper understanding of the evaluated mechanisms are more readily able to recognize both their presence and their practical utility within the analyzed tool.

5.7 Reliability

For each awareness mechanism within the model, we calculated the relationship between the probability of each response option (from "strongly disagree" to "strongly agree") and the individual's ability scale. Figure 10 displays the total information curve for the support of awareness mechanisms and its corresponding standard error (SE). The blue line represents the test information function $I(\theta)$, which is characterized by a normal (Gaussian) distribution Thissen and Wainer [2001]. Conversely, the red dotted line illustrates the standard error $SE(\theta)$. The intersection point between these two curves precisely delineates the interval within which the model demonstrates optimal representativeness.

The test information and SE curves collectively indicate the instrument's precision. Notably, the SE curve reaches its minimum value precisely at the point on the scale where the information curve achieves its peak. This signifies that the instrument is most precise for participants whose ability levels fall within this optimal range. Therefore, the instrument is particularly well-suited for participants with ability levels in the scale region where the information curve surpasses

[13] The (μ, σ) scale is used in IRT to represent, respectively, the mean value (μ) and the standard deviation (σ) of the individual abilities within the population de Andrade *et al*. [2000].

Table 3. Demographic Distribution

(a) Age

Obs.	Group	θ mean
26	17 years or less	0.147
214	18 to 28 years	-0.246
72	29 to 39 years	0.154
17	40 to 50 years	-0.037
4	51 years or more	-0.894

(b) Gender

Obs.	Group	θ mean
203	Male	-0.128
128	Female	-0.111
2	Other	-0.934

(c) Familiarity (awareness)

Obs.	Group	θ mean
29	1 - Novice	-0.936
87	2 - competent	-0.715
105	3 - proficient	0.144
74	4 - expert	0.680
38	#N/A	-0.418

(d) Familiarity (collaboration)

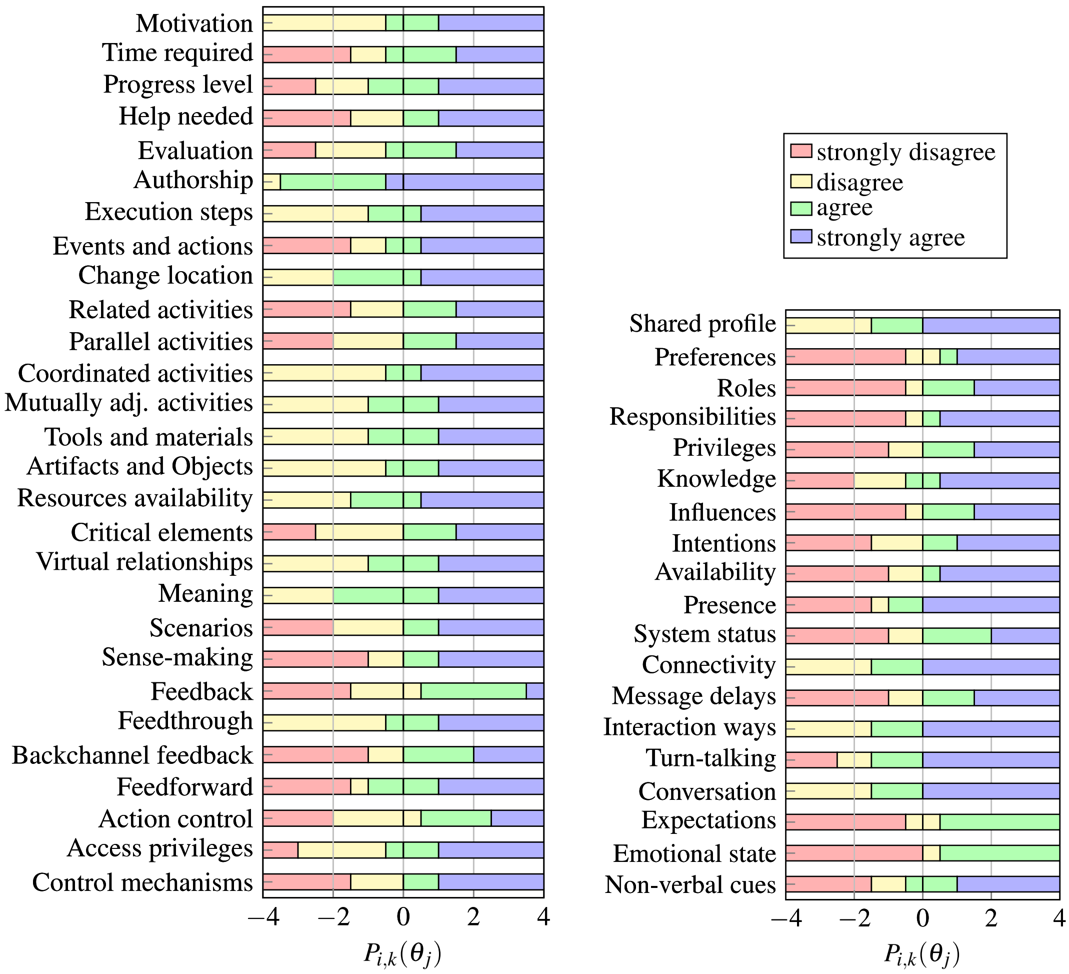
Obs.	Group	θ mean
20	1 - Novice	-0.816
77	2 - competent	-0.633
136	3 - proficient	0.018
59	4 - expert	0.648
41	#N/A	-0.405

(e) Familiarity (environment)

Obs.	Group	θ mean
30	1 - Novice	-0.939
72	2 - competent	-0.753
126	3 - proficient	0.214
64	4 - expert	0.531
41	#N/A	-0.476

(f) Environment

Obs.	Group	θ mean
66	Discord	0.089
12	GitHub/GitLab	-1.404
55	Google Meet	0.243
14	Jira	-1.562
52	Microsoft Teams	0.072
22	Skype	0.340
31	Slack	0.056
12	Trello	-0.799
17	WhatsApp	-0.162
13	Zoom	0.030
39	Other	-0.574



(a) Workspace awareness **(b)** Collaboration awareness

Figure 9. Awareness support scale

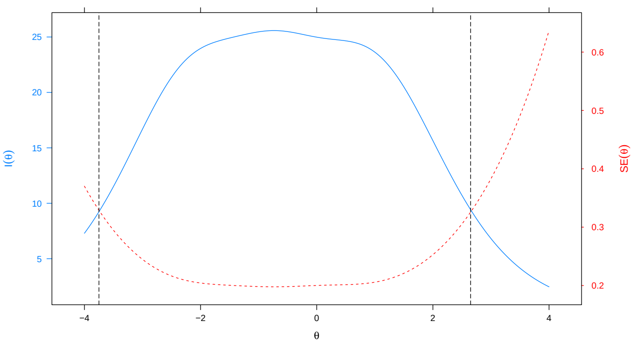


Figure 10. Test Information Function ($I(\theta)$) and Standard Error ($SE(\theta)$)

the standard error curve, which corresponds to the interval $[-3.75, +2.65]$.

The overall shape of the curve further confirms that the instrument comprehensively covers the entire latent trait, effectively assessing individuals ranging from those who struggle to comprehend awareness mechanisms ($\theta_j < -1$) to those who can rapidly identify these mechanisms ($\theta_j > 1$). Both the Cronbach's Alpha and IRT parameters provide strong empirical evidence for the validation of our proposed model.

Firstly, the adequate representation of the θ scale (spanning the interval $[-3.75, +2.65]$ as presented in Figure 10 serves as compelling evidence of the instrument's reliability. The internal reliability, assessed through Cronbach's Alpha coefficient DeVellis [2016], demonstrated robust internal consistency, consistently achieving values of $\alpha > 0.8$[14].

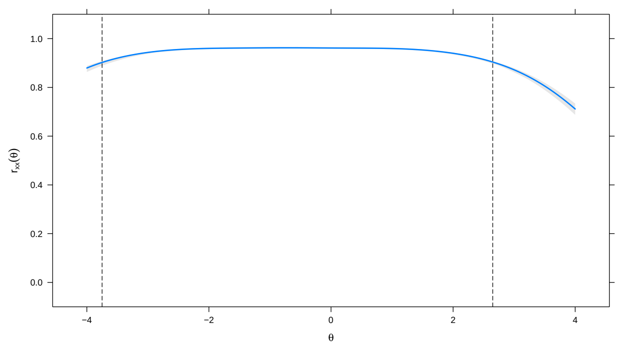


Figure 11. Test Reliability ($r_{xx}(\theta)$)

Secondly, we calculated the instrument's reliability function $r_{xx}(\theta)$ Thissen and Wainer [2001]; Chalmers [2012] based on the participants' latent traits. Our model exhibits excellent overall reliability, with the function reaching its highest value ($r_{xx} > 0.95$) precisely within the region of the scale where the test information function demonstrates optimal representativeness.

6 Conclusion

Collaborative approaches in software development have proven beneficial in both professional and educational settings. We conducted a survey of various systems supporting this approach. We evaluated their level of support for awareness and collaboration aspects, specifically related to the dimensions of collaboration awareness and shared workspace awareness. Participation was strictly voluntary, involving a combination of undergraduate students and market professionals from the regional software development ecosystem. We obtained 333 valid observations across eight distinct development groups/environments. As data collection and analysis instruments, we utilized the Awareness Assessment Model Mantau and Benitti [2023a, 2024]. The collected data were calibrated using the Item Response Theory (IRT) statistical approach, in conjunction with calibration data from the model itself Mantau and Benitti [2023b], which contains 1,280 previously compiled observations.

6.1 Robustness

To ensure the statistical validity and address the awareness gap across tools, we employed a test equating strategy. This statistical instrument produces a linkage between scores on different test forms so that they can be used interchangeably, as if they had originated from the same instrument Dorans *et al*. [2010]. Following the framework by [Holland *et al*., 2006], our approach satisfies five fundamental requirements for linking scores:

i) *Equal construct*: Both the case study and the reference dataset Mantau and Benitti [2023b] measure the same latent trait: awareness;
ii) *Equal reliability*: The instruments maintain consistent measurement precision;
iii) *Symmetry*: The transformation for mapping scores is reversible;
iv) *Equity*: It is a matter of indifference to a participant which specific set of items or tool evaluations they perform;
v) *Population invariance*: The equating function remains stable regardless of whether derived from students or professionals.

By integrating the regional data through equating by common elements, we ensure that item parameters are expressed on a single calibrated metric, justifying the sample size and providing accurate IRT indicators.

6.2 Discussion of Results

The evaluation results were positive, and environments more familiar to users showed better performance in terms of user perception. A clear distinction emerged based on tool design: technical or task-centric tools like Jira, GitHub/GitLab, and Trello exhibited lower average difficulty parameters (θ), indicating that participants found it harder to recognize the evaluated awareness mechanisms. This suggests an awareness gap in asynchronous environments where information is often fragmented within pull requests or tickets.

Conversely, communication-centric environments like Skype, Google Meet, Discord, and Microsoft Teams demonstrated higher recognition scores. These platforms' focus on high synchronicity and real-time presence reduces the cognitive effort required for users to perceive (awareness) group activities. The values of θ, a, and b, combined with Cronbach's alpha ($\alpha > 0.8$) and the reliability function $r_{xx}(\theta)$, attest to the precision of these findings.

[14] Cronbach's Alpha values between $0.7 < \alpha \leq 0.8$ are generally considered acceptable; values between $0.8 < \alpha \leq 0.9$ are considered good; and $\alpha \geq 0.9$ indicates excellent internal consistency DeVellis [2016].

6.3 Scientific Contribution and Practical Implications

The scientific contribution of this work lies in the empirical validation of the Awareness Assessment Model using IRT, providing a standardized metric to quantify how users perceive (awareness) collaboration support. The practical implications include:

- **For Development Teams:** Findings reveal that awareness recognition is tied to user proficiency. Teams can diagnose gaps in workflows and select tools with stronger visual support for asynchronous coordination.
- **For Tool Designers:** The study highlights a gap between communication-centric and technical platforms. Designers should focus on awareness-by-design, reducing cognitive load and implementing feedback loops that maintain group awareness without causing information fatigue.
- **For Training and Education Programs:** The correlation between theoretical knowledge and practical tool performance suggests that software engineering curricula should move beyond technical syntax. Programs should incorporate the 3C Model (Communication, Coordination, and Cooperation) to prepare students for modern development ecosystems.

6.4 Limitations

While this study provides valuable insights, it is important to acknowledge certain limitations:

Sample representativeness and self-report bias. Although a significant number of participants were involved, specific demographic subgroups were underrepresented. Furthermore, data collected via questionnaires rely on self-reported perceptions, which may be subject to biases such as social desirability.

Cross-sectional design. This study provided a snapshot of awareness at a single point in time. It does not capture how awareness evolves over prolonged use or impact long-term collaborative dynamics.

Focus on perceived awareness support. The research focused on participants' awareness rather than direct behavioral observation of actual outcomes like error rates or team cohesion[cite: 1]. While awareness is critical, direct behavioral data could offer a more comprehensive understanding.

6.5 Future Work

Based on the insights gained and the limitations identified, several promising avenues for future research emerge further to advance the understanding of awareness in collaborative environments:

Model generalization and enhanced support. Future work should expand the application of the Awareness Assessment Model to a broader range of environments, such as specialized IDEs with collaborative plugins and emerging VR/AR platforms. Given the awareness gap identified in technical platforms, research should focus on prototyping novel mechanisms, such as predictive awareness features and interactive feedback systems, to address the specific challenges of asynchronous coordination.

Triangulation with log data and productivity. A critical next step is to evaluate whether subjective user awareness aligns with actual interaction patterns captured in development tool logs. Furthermore, we suggest investigating the correlation between high levels of tool-provided awareness and objective productivity metrics, specifically the reduction of integration conflicts (merge conflicts) and rework. Understanding if improved awareness directly translates into smoother technical integration is vital for the software industry.

Experience-based awareness and developer roles. Future studies should analyze how developer experience levels (e.g., Junior vs. Senior) influence the awareness of awareness mechanisms. This includes identifying if certain features are more critical for novices or if experienced professionals prioritize different types of information. Utilizing IRT techniques, such as Differential Item Functioning (DIF), could reveal how the latent trait of awareness is manifested across these distinct demographic groups.

Longitudinal and mixed-methods studies. Future studies would benefit from longitudinal designs to observe how awareness and its impact on team performance evolve over extended periods. Supplementing quantitative IRT analysis with qualitative methods—such as in-depth interviews or ethnographic observations—will provide a more nuanced understanding of the cognitive processes involved in perceiving awareness cues in real-world collaborative settings.

Declarations

Authors' Contributions

MJM and SC contributed to the conception of this study. SC performed the experiments. MJM is the primary contributor and writer of this manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

A complete reference of *Conceptual View* artifacts can be found at our Awareness Assessment Model repository Concentius and Mantau [2025].

References

Baker, F. B. and Kim, S.-H. (2017). *The basics of item response theory using R*. Springer, Madision, WI, USA. DOI: 10.1007/978-3-319-54205-8.

Barbosa, S. and Silva, B. (2010). *Interação humano-computador*. Elsevier Brasil, Rio de Janeiro, RJ. Available at: `https://www.academia.edu/41857244/INTERA%C3%87%C3%83O_HUMANO_COMPUTADOR`.

BASILI, V. (1992). Software modeling and measurement: The goal/question/metric paradigm. *Technical Report*. Available at: `https://www.cs.umd.edu/~basili/publications/technical/T78.pdf`.

Cai, L. (2010). Metropolis-hastings robbins-monro algorithm for confirmatory item factor analysis. *Journal of Educational and Behavioral Statistics*, 35(3):307–335. DOI: 10.3102/1076998609353115.

Chalmers, R. P. (2012). mirt: A multidimensional item response theory package for therenvironment. *Journal of Statistical Software*, 48(6). DOI: 10.18637/jss.v048.i06.

Collazos, C. A., Gutierrez, F. L., Gallardo, J., Ortega, M., Fardoun, H. M., and Molina, A. I. (2019). Descriptive theory of awareness for groupware development. *Journal of Ambient Intelligence and Humanized Computing*, 10(12):4789–4818. DOI: 10.1007/s12652-018-1165-9.

Concentius, S. and Mantau, M. J. (2025). Avaliação do suporte à percepção em ferramentas colaborativas de apoio ao desenvolvimento de software: Repositório de artefatos. In *Zenodo Repository*. Zenodo. DOI: 10.5281/zenodo.15285691.

da Silva Estácio, B. J. and Prikladnicki, R. (2015). Distributed pair programming: A systematic literature review. *Information and Software Technology*, 63:1–10. DOI: 10.1016/j.infsof.2015.02.011.

de Andrade, D. F., Tavares, H. R., and Valle, R. d. C. (2000). Teoria da resposta ao item: conceitos e aplicações. *ABE, Sao Paulo*. Available at: `https://docs.ufpr.br/~aanjos/CE095/LivroTRI_DALTON.pdf`.

DeVellis, R. F. (2016). *Scale development: Theory and applications*, volume 26 of *Applied Social Research Methods 26*. Sage Publications, Inc, London, UK, 4th edition. Available at: `https://www.mpm.ac.in/News/Ebook_mpm_sadmin07755388.pdf`.

Dorans, N. J., Moses, T. P., and Eignor, D. R. (2010). Principles and practices of test score equating. *ETS Research Report Series*, 2010(2):1–41. DOI: 10.1002/j.2333-8504.2010.tb02236.x.

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, pages 107–114, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/143457.143468.

Ellis, C., Gibbs, S., and Rein, G. (1991). Groupware some issues and experiences. *Communications of the ACM*, 34(1):38–58. DOI: 10.1145/99977.99987.

Fuks, H., Raposo, A., Gerosa, M., Pimental, M., and Lucena, C. (2008). The 3c collaboration model. In *Encyclopedia of E-collaboration*, Pennsylvania, USA. IGI Global. DOI: 10.4018/978-1-59904-000-4.ch097.

George, J. F. (2003). Groupware. In Bidgoli, H., editor, *Encyclopedia of Information Systems*, pages 509–518. Elsevier, New York, NY, USA. DOI: 10.1016/B0-12-227240-4/00084-8.

Gross, T. (2013). Supporting effortless coordination: 25 years of awareness research. *Computer Supported Cooperative Work*, 22(4-6):425–474. DOI: 10.1007/s10606-013-9190-x.

Gutwin, C., Barjawi, M., and Pinelle, D. (2016). The emergence of high-speed interaction and coordination in a (formerly) turn-based groupware game. In *Proceedings of the 19th International Conference on Supporting Group Work*, GROUP '16, pages 277–286, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/2957276.2957315.

Gutwin, C. and Greenberg, S. (2002). A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work*, 11(3-4):411–446. DOI: 10.1023/a:1021271517844.

Holland, P. W., Dorans, N. J., and Petersen, N. S. (2006). 6 equating test scores. In Rao, C. and Sinharay, S., editors, *Psychometrics*, volume 26 of *Handbook of Statistics*, pages 169–203. Elsevier, Princeton, NJ, USA. DOI: 10.1016/S0169-7161(06)26006-1.

ISO/IEC 25.040 (2011). Systems and software engineering - systems and software quality requirements and evaluation (square) - evaluation process. Standard, International Organization for Standardization, Geneva, CH. Available at: `https://www.iso.org/standard/35765.html`.

Lopez, G. and Guerrero, L. A. (2017). Awareness supporting technologies used in collaborative systems: A systematic literature review. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, CSCW '17, pages 808–820, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/2998181.2998281.

Mantau, M. J. and Benitti, F. B. V. (2022a). Awareness support in collaborative system: Reviewing last 10 years of cscw research. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 564–569, Hangzhou, China. IEEE. DOI: 10.1109/cscwd54268.2022.9776091.

Mantau, M. J. and Benitti, F. B. V. (2022b). Towards an awareness taxonomy. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 495–500, Hangzhou, China. IEEE. DOI: 10.1109/cscwd54268.2022.9776129.

Mantau, M. J. and Benitti, F. B. V. (2023a). The awareness assessment model: measuring the awareness and collaboration support over the participant's perspective. In *Anais do XVIII Simpósio Brasileiro de Sistemas Colaborativos*, pages 30–43. SBC. DOI: 10.5753/sbsc.2023.229064.

Mantau, M. J. and Benitti, F. B. V. (2023b). The awareness assessment model repository. Data repository/Internal Documentation. Zenodo. DOI: 10.5281/zenodo.8298950.

Mantau, M. J. and Benitti, F. B. V. (2024). The awareness assessment model: measuring awareness and collaboration support over participant's perspective. *Universal Access in the Information Society*. DOI: 10.1007/s10209-024-01110-5.

Molina, A. I., Bravo, C., Gallardo, J., Lacave, C., and Redondo, M. A. (2025). Awareness support in collaborative programming tools: An evaluation based on programmer's perception and eye tracking. *Journal of Systems and Software*, 220:112276. DOI: 10.1016/j.jss.2024.112276.

Molina, A. I., Gallardo, J., Redondo, M. A., and Bravo, C. (2015). Assessing the awareness mechanisms of a collaborative programming support system. *Dyna*, 82(193):212–222. DOI: 10.15446/dyna.v82n193.53497.

Pasquali, L. (2020). *TRI-Teoria de resposta ao item: Teoria, procedimentos e aplicacoes*. Editora Appris, Curitiba, PR, BR. Available at: https://booklover.com.br/tri-teoria-de-resposta-ao-item-teoria-procedimentos-e-aplicacoes-luiz-pasquali/.

Rogers, Y., Sharp, H., and Preece, J. (2013). *Design de interação*. Bookman Editora, Porto Alegre, RS, BR. Available at: https://books.google.com.br/books/about/Design_de_Intera%C3%A7%C3%A3o_3ed.html?id=d_s4AgAAQBAJ&redir_esc=y.

Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika monograph supplement*. DOI: 10.1007/bf03372160.

Sansivieri, V., Wiberg, M., and Matteucci, M. (2017). A review of test equating methods with a special focus on irt-based approaches. *Statistica*, 77(4):329–352. DOI: 10.6092/issn.1973-2201/7066.

Schmidt, K. (2002). The problem with 'awareness': Introductory remarks on 'awareness in cscw'. *Computer Supported Cooperative Work (CSCW)*, 11(3):285–298. DOI: 10.1023/A:1021272909573.

Silva, L., Mendes, A. J., and Gomes, A. (2020). Computer-supported collaborative learning in programming education: A systematic literature review. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 1086–1095. DOI: 10.1109/EDUCON45650.2020.9125237.

Tenenberg, J., Roth, W.-M., and Socha, D. (2016). From i-awareness to we-awareness in cscw. *Computer Supported Cooperative Work (CSCW)*, 25(4):235–278. DOI: 10.1007/s10606-014-9215-0.

Thissen, D. and Wainer, H. (2001). *Test Scoring*. Routledge, New York, NY, USA. DOI: 10.4324/9781410604729.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, pages 1–10, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/2601248.2601268.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering: an introduction*. Springer Science & Business Media, New York, NY, USA. DOI: 10.1007/978-3-642-29044-2.