AnnotAISE: Web-Based Data Annotation Platform For Software Engineering Research

João Paulo Lopes
PUC-Rio
Rio de Janeiro, Brazil
jlopes@aise.inf.puc-rio.br

Bruno T. Fernandes
PUC-Rio
Rio de Janeiro, Brazil
btardin@aise.inf.puc-rio.br

Beatriz Ritter
PUC-Rio
Rio de Janeiro, Brazil
bia@aise.inf.puc-rio.br

Daniel Coutinho
PUC-Rio
Rio de Janeiro, Brazil
dcoutinho@inf.puc-rio.br

Robbie Carvalho
PUC-Rio
Rio de Janeiro, Brazil
robbie@aise.inf.puc-rio.br

Alessandro Garcia
PUC-Rio
Rio de Janeiro, Brazil
afgarcia@inf.puc-rio.br

Juliana Alves Pereira
PUC-Rio
Rio de Janeiro, Brazil
juliana@inf.puc-rio.br

Abstract

Empirical software engineering studies frequently rely on manual annotation to transform raw software artifacts into structured data for analysis and evaluation. However, organizing annotation campaigns can be challenging, particularly when multiple evaluators are involved and studies require controlled item distribution, disagreement management, progress monitoring, and traceable results. Existing annotation platforms are often either too general-purpose, too domain-specific, closed-source, or focused on workflows outside empirical software engineering research. Moreover, these studies frequently require richer annotation schemes involving multiple interrelated questions per instance, which is not well supported by many existing tools. To address these challenges, we present AnnotAISE, an open-source web platform designed to support structured annotation workflows for empirical software engineering studies. The platform enables customizable multi-question annotation forms, annotator coordination, disagreement resolution, progress tracking, and analysis-oriented result export. We evaluated the platform through a TAM-based user study involving 28 participants, including annotators and administrators. The results indicate positive perceptions regarding usability, usefulness, and intention to use, suggesting that AnnotAISE can help support more reliable and manageable annotation-based research workflows.
Demo video: https://youtu.be/MLSml5Pz65Q

Keywords

Empirical Software Engineering, Data Labeling, Annotation Tool

1 Introduction

In empirical software engineering studies, researchers often need to transform raw software artifacts into data that can be analyzed and acted upon [1, 27]. For example, a study analyzing a set of version control revisions may need to perform a manual annotation process to label the type of change performed between revisions [10, 12]. Several other tasks, such as requirement classification [9], sentiment analysis [13, 17], and code smell identification [18, 28] can require such manual annotation. Additionally, as large language models are massively scaling the amount of generated data that needs to be validated [1, 14], this type of task is also increasingly more in demand. In all of these examples, human evaluators with domain expertise must be involved.

The labels assigned by annotators in these processes often serve as the basis for subsequent research activities (such as the aforementioned examples). Therefore, these annotation processes must be rigorously conducted, avoiding mistakes that could compromise the quality of the data and, consequently, the results of the subsequent study [11, 13, 28]. Examples of common mistakes include uneven item distribution, ad-hoc annotation criteria over multiple evaluators, loss of response traceability, or difficulty in identifying disagreements [11, 18]. Thus, utilizing a dedicated tool for this workflow could reduce the operational effort required from researchers. Indirectly, utilizing a tool focused on increasing the rigor of the annotation process could also potentially improve the reliability, reproducibility, and methodological quality of the subsequent empirical studies.

As such, prior work has proposed or studied annotation tools for different settings. However, several of these tools are either unavailable, closed-source (thus limiting extensibility for specific tasks), or only offer paid access [2, 4, 15, 20, 21, 26]. In terms of feature set, they are often either too general-purpose or too domain-specific, focusing not only on annotations but on miscellaneous tasks such as active learning, computer vision, or qualitative coding [11, 21, 23].

While these workflows are valuable in their respective contexts, empirical software engineering studies frequently require task-specific customization. For instance, studies analyzing code review comments or other development-related data [1, 22] often require syntax highlighting that supports smoothly alternating between regular text and source code. Moreover, these platforms often assume annotations consist of assigning a single label to each data point, whereas empirical software engineering studies frequently require richer annotation schemes involving multiple interrelated questions per instance. This requirement makes both generic form tools and existing data labeling platforms inadequate for many research-oriented workflows.

To address these gaps, we propose AnnotAISE, a web-based platform designed to create and execute structured annotation campaigns over datasets. The tool allows researchers to import their datasets and automatically transform each data record into an independent annotation form instance. To achieve this, the tool supports building customized form templates based on characteristics of that dataset. By using these templates, annotators are presented with forms that automatically change while they annotate the different data records that make up the dataset. AnnotAISE supports several

data formats when building these forms, such as unformatted text, markdown, code snippets, URLs, images, and video.

Beyond form creation, AnnotAISE supports the operational control of the annotation. The platform allows researchers to define strategies for data distribution, and supports configurable multi-annotator labeling. It also lets researchers track annotation progress, can be used to collect and store participant background information, supports easy distribution of annotation guidelines and guides, and has features to simplify the resolution of disagreements that occur between evaluators. At the end of the process, AnnotAISE automatically calculates descriptive statistics for the collected data, including aggregated views of responses and agreement information among annotators. It also supports exporting these results.

To evaluate AnnotAISE, we conducted a user study based on the Technology Acceptance Model (TAM) [6]. The aim is to investigate participants' perceptions regarding the platform's perceived ease of use, perceived usefulness, and intention to use. The evaluation considers two user profiles involved in the annotation process: administrator and annotator. The former is responsible for creating and managing annotation campaigns. The latter is in charge of evaluating the items distributed by the platform. This design allows us to assess both the experience of configuring and controlling an annotation campaign and the practical experience of performing annotation tasks. In total, the evaluation involved 28 participants, including 21 annotators and 7 administrators.

The results indicate that AnnotAISE was positively perceived by both participant groups according to the TAM-based evaluation conducted using a five-point Likert scale. Annotators reported high usability scores for core workflow activities, such as navigating between annotation instances ($\mu = 4.76/5$), accessing assigned tasks ($\mu = 4.67/5$), and completing annotation forms ($\mu = 4.38/5$). Administrators also evaluated the platform favorably, particularly regarding exporting annotated datasets ($\mu = 4.86/5$), configuring question structures ($\mu = 4.86/5$), assigning annotators to tasks ($\mu = 4.71/5$), and monitoring responses through the dashboard ($\mu = 4.71/5$). Participants from both groups also expressed strong intention to use the platform again in future projects, suggesting positive acceptance of the tool in practical research workflows.

The main contribution of this work is AnnotAISE, an open-source platform specifically designed to support rigorous annotation workflows in empirical software engineering research. By providing structured support for customizable multi-question annotation forms, annotator coordination, disagreement resolution, progress tracking, and result aggregation, the platform aims to reduce the operational effort required from researchers while improving the reliability, reproducibility, and methodological quality of annotation-based studies. Furthermore, the TAM-based evaluation involving both annotators and administrators provides initial evidence that the platform is usable, useful, and well accepted by users in practical annotation scenarios.

2 Related Work

Data annotation is a key activity in the construction of reliable datasets for machine learning, empirical analysis, and human-in-the-loop evaluation [27]. In software engineering, labeled datasets support studies on code smells, defect prediction, code review analysis, and the evaluation of generated software artifacts [1, 14, 24, 28]. Prior work has also explored annotation support in different settings, including text classification with active learning [23], computer vision labeling [21], and qualitative coding in empirical software engineering [5, 11]. As such, several tools have been proposed and utilized for these tasks. The remainder of this section discusses these tools and compares AnnotAISE to them. Table 1 summarizes this comparison.

General-purpose data labeling platforms. Label Studio [15] is a widely used open-source platform for data labeling, supporting different data modalities such as text, images, audio, video, time series, and multimodal data. Related academic tools, such as Label Sleuth, support text classification through iterative labeling and active learning [23]. In software engineering, studies such as ToxiSpanSE show that manual annotation is also used to construct domain-specific datasets, for instance by annotating toxic spans in code review comments [22]. However, these platforms are often centered on generic labeling workflows with isolated labels per instance. In contrast, empirical software engineering studies frequently require transforming structured CSV records into richer multi-question annotation instances. AnnotAISE specifically targets these research-oriented workflows.

Computer vision annotation platforms. Several mature platforms focus on computer vision annotation. Roboflow supports dataset upload, annotation, preprocessing, training, and deployment for visual datasets [19, 20]. Supervisely provides resources for image, video, LiDAR, DICOM, collaboration, quality assurance, and AI-assisted visual annotation [25, 26]. Clarifai also provides an AI-oriented platform for visual and multimodal workflows [4]. These tools are commonly discussed in terms of image annotation, object-level labeling, automation, quality assurance, and integration with machine learning pipelines [21]. Although these platforms provide advanced support for visual annotation and machine learning workflows, they are less aligned with structured textual annotation campaigns common in empirical software engineering studies.

Qualitative analysis tools. Another related category is qualitative data analysis software, such as ATLAS.ti. These tools support the organization, coding, and analysis of qualitative materials such as interviews, documents, transcripts, and multimedia sources [2]. In empirical software engineering, they have been used to support qualitative coding, reliability analysis, inter-coder agreement assessment, and studies on topics such as code review [5, 11]. While these tools support qualitative coding and reliability analysis, AnnotAISE focuses more directly on operational annotation workflows, including item distribution, multi-annotator management, disagreement handling, progress monitoring, and structured data export.

3 AnnotAISE Overview

AnnotAISE is a web-based platform for running structured annotation tasks over tabular datasets. Using it, researchers can design a custom form template, import the dataset to be labeled, invite annotators, and collect their responses. The platform handles the operational side of the study: assigning work, tracking progress, resolving disagreements and giving structured outputs. As such, researchers can focus on study design and faster analysis. In terms of technology, AnnotAISE utilized a stack composed of a REST API built using Python (namely, the Django framework) and a React web interface built using Typescript. To illustrate the usage of the

Table 1: Comparison between AnnotAISE and related annotation tools.

Tool	Main focus	Typical data	Position in relation to AnnotAISE
Label Studio	General-purpose data labeling	Text, image, audio, video, time series, and multimodal data	Broad and flexible labeling platform, but less focused on structured research-oriented annotation workflows with customizable multi-question forms.
Roboflow	Computer vision dataset management and annotation	Images, segmentation, object detection, and classification datasets	Primarily designed for computer vision and machine learning workflows rather than annotation studies involving structured research data and configurable forms.
Supervisely	Computer vision annotation and AI workflows	Images, videos, LiDAR, DICOM, and other visual datasets	Focused on visual AI pipelines and dataset management, rather than annotation campaigns for empirical research studies.
Clarifai	AI platform for labeling and model development	Visual and multimodal AI data	Focused on AI-oriented workflows and model development, rather than controlled annotation workflows for empirical studies.
ATLAS.ti	Qualitative data analysis and coding	Documents, interviews, transcripts, and multimedia qualitative data	Supports qualitative coding and interpretive analysis, but is less focused on operational annotation workflows involving multiple annotators and structured result aggregation.
AnnotAISE	Research-oriented annotation workflows	CSV datasets containing annotation instances. Fields may hold text, code snippets, or URLs pointing to images, audio, video, and PDF files	Focused on customizable annotation forms, annotator coordination, disagreement management, progress monitoring, and analysis-oriented result export.

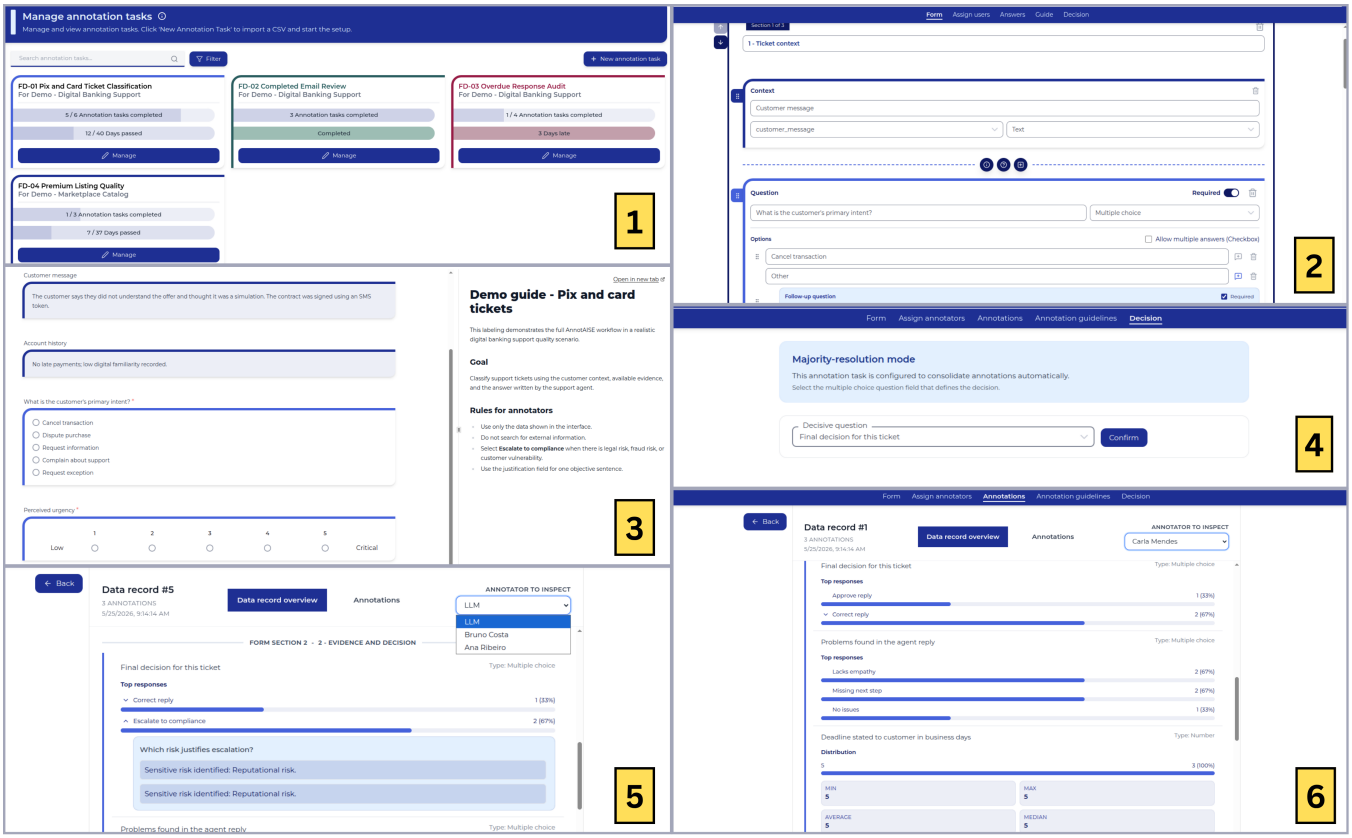

Figure 1: AnnotAISE screenshots

tool and its feature set, the remainder of this section walks through the platform's workflow in steps.

Step 1: Annotation Task Setup. An annotation task begins with the upload of a CSV file that contains the dataset to be evaluated. AnnotAISE splits it into discrete units: each row of the dataset is stored as a separate *data record*, which becomes the smallest piece of data that the platform manages independently. This row-level segmentation is what allows the platform to distribute work to annotators one piece at a time. The structure of the original file (i.e., headers) is stored and made available in the form builder, so that it can be referenced when designing the form. Additional data records can be added to an existing annotation task at any time by uploading a supplementary dataset that follows the same schema, which is useful when a need arises to label more data than originally planned. With the dataset in place, the researcher builds the annotation form through a visual editor, as shown in Figure 1 2. A form is organized into sections, each containing an ordered sequence of fields of two kinds: *data fields* and *question fields*.

Data fields are read-only and display content from the dataset to give annotators the material they need to annotate. Each data field is bound to one column of the imported dataset, and at annotation time the system renders the corresponding cell value for the current data record. A data field can be configured to display its content as text, number, image, date, category, code, audio, video, or PDF,

so that the web interface presents each piece of information to the annotator in the most appropriate format. On the other hand, question fields are input fields. Four types are available: (i) *text*, for free-form responses; (ii) *number*, for numeric input; (iii) *value range*, defined by a start value, end value, and step size; and (iv) *multiple choice*, with an ordered list of options. Multiple-choice questions may allow single or multiple selections and support follow-up questions: an option can optionally trigger a nested question that is presented only when that option is selected. Any question field can be marked as required.

Beyond the main form, researchers may enable an optional *background questionnaire*: a separate form with a set of questions that annotators answer once, before starting the task. This is useful for collecting demographic information or self-reported expertise, which can later support correlation analyses between annotators' backgrounds and their responses. Researchers may also provide an *annotation guideline*: a free-text document with full Markdown support. The guideline serves as a reference for annotators throughout the task, offering instructions, labeling criteria, and examples to promote consistency across responses.

Step 2: Distribution and Agreement Strategy Configuration. Once an annotation task is configured, `AnnotAISE` must decide which annotators evaluate which data records and how to reconcile their answers when they disagree. The platform offers configurable strategies for both. The default strategy is *on-demand distribution*. Data records are not pre-assigned; instead, each annotator requests the next available data record whenever they are ready to work, and the platform picks one they have not yet seen. The data record is temporarily reserved for that annotator, and if they leave it idle for too long the reservation expires and the data record returns to the pool, to prevent locking records for long periods. This lets annotators contribute at their own pace.

A key configuration of this strategy is the number of *annotations per record*, which specifies how many independent annotations each data record must receive before it is considered complete. With annotations per record set to one, each data record is labeled by a single annotator and closed as soon as it is submitted. With higher values, the platform ensures that several annotators evaluate the same data record, prioritizing records that already have partial annotations so that work converges toward completion rather than spreading thin across the dataset. Once a data record has gathered the required number of annotations, it is marked as completed. When every record is complete, the annotation task as a whole is closed automatically.

For studies where each annotator must work on a predetermined set of data records, for example, multi-phase designs in which annotators revisit their own earlier responses, `AnnotAISE` offers an alternative: *pre-assigned distribution* strategy. In this mode, the imported dataset includes a column identifying the intended annotator of each row, and the platform routes data records accordingly.

When more than one annotator labels the same data records, disagreements are expected, and `AnnotAISE` provides two optional mechanisms for resolving them automatically. The first is a *majority-resolution mode*. The researcher designates one multiple-choice question as the *decisive question* (Figure 1 4), and the platform tracks how annotators answer it on each data record. A data record is finalized only when one option achieves a strict majority, that is, when it has been selected more times than any other option. If a tie persists after the configured number of annotations has been reached, the data record stays open and additional annotators are recruited from the pool until the tie is broken.

The second mechanism is an *LLM-assisted tiebreaker*. When human annotators reach a tie on the decisive question, the platform can trigger a local decision round through the use of multiple LLMs. For this call, `AnnotAISE` builds a prompt containing the annotation guideline, the data record, the decisive question text, and the list of valid multiple-choice options. The prompt instructs the model to select exactly one option and return only the exact text of one valid alternative, without any additional explanation. The answer is only counted (to avoid hallucinations) if it can be matched to one of the valid options. We utilize multiple LLMs for this task. Currently, it queries four local models [1] through Ollama. The most-voted valid option is used as the final LLM decision. If the models themselves produce a tie, no final decision is assigned. When the item includes code context, the platform instead uses a specialized pool of three code-oriented models [2]. When a winner exists, the platform records an answer and marks the final decision as LLM-generated.

Step 3: Annotator Workflow. From the annotator's perspective, the workflow is designed to be self-contained: once they open an assigned annotation task, everything they need to evaluate the data and record their judgments is available in a single page. If the task has a background questionnaire enabled, the annotator completes it before starting the annotation task. This is a one-time step that can be used to gather contextual information about the annotator, such as experience level, domain expertise, or demographic data. This data can be used later by researchers to track and interpret variation across annotators.

After that, the annotator requests their first data record. The platform returns the rendered form containing the form instance (i.e., the template with each data field filled with data from the record). Each data field is presented according to its configured type, so that text and code appear inline, images are displayed directly, audio and video are played through embedded players, and PDFs are rendered within the page. Annotators therefore evaluate the source material without leaving the platform or switching to external tools. Throughout the task, the annotation guideline is available at any moment through a dedicated tab in the interface, as shown in Figure 1 3. Annotators can consult it before starting or mid-task to review the labeling criteria, examples, and instructions defined by the researcher. Once the annotator submits their responses, the current data record is released and the platform serves the next one. The annotator repeats this cycle, request, evaluate, submit, until no further records are available to them.

Step 4: Output and Analysis. Once annotations have been collected, `AnnotAISE` offers several complementary ways for researchers to make sense of the results, supporting both quick overviews and detailed inspection. The first is the *answers dashboard*, accessible from a dedicated management view that lists all collected annotations for a task (Figure 1 1). For each question, the dashboard shows how the responses are distributed — for example, how

[1] llama3.1:8b, llama3.2:3b, qwen2.5:7b, and mistral-nemo:12b
[2] qwen3-coder:30b, qwen2.5-coder:32b, and deepseek-coder-v2:16b

frequently each class appears in a classification task — giving researchers an immediate overview of the labeled dataset (Figure 1 6). When the task uses the LLM-assisted tiebreaker, the dashboard distinguishes the responses produced by the LLM from those submitted by annotators, allowing researchers to inspect the model's decisions separately (Figure 1 5). When a task is configured with multiple annotations per record, the dashboard also presents an *inter-annotator agreement summary*: for every multiple-choice question and each of its options, it reports the number of records on which at least k annotators independently selected the same option, where k is a threshold chosen by the researcher. This helps researchers spot questions where annotators converge or diverge and provides a first signal of annotation quality.

Additionally, the dashboard supports a more granular, per-record view. Researchers can drill into any record to inspect every annotation submitted for it side by side, review the original field values, and compare how annotators handled the same input. Background questionnaire responses are accessible per annotator as well, allowing researchers to cross-reference annotator profiles with their labeling behavior. Finally, a *CSV export* of all collected annotations is also available. The exported file contains one row per annotation, with columns for question and fields, the annotator ID, and an identifier for the record.

4 Evaluation

To evaluate the AnnotAISE platform, a user study was conducted based on the Technology Acceptance Model (TAM) [6]. TAM is a widely adopted framework for assessing user perception of technology, focusing on three core dimensions: perceived ease of use, perceived usefulness, and intention to use. These dimensions provide a structured basis for understanding how users accept and interact with a new system. Our evaluation is guided by the following research questions:

RQ1. How do users perceive the ease of use of AnnotAISE when performing labeling tasks across different user roles?

RQ2. How do users perceive the usefulness of AnnotAISE in practical annotation workflows involving different data types?

RQ3. To what extent do users across different user roles intend to use AnnotAISE in their future annotation work?

Because AnnotAISE supports two distinct user roles, we developed two separate questionnaires: one targeting administrators, who create and manage labeling projects, and another targeting annotators, who complete assigned labeling tasks. This distinction captures the different experiences and interaction patterns associated with each role, while allowing us to address the RQs from both perspectives. Each questionnaire was structured into four sections: informed consent, participant characterization, TAM-based statements, and general feedback. Participants rated TAM statements on a five-point Likert scale (1: Strongly Disagree to 5: Strongly Agree). Each statement was followed by an optional open-ended field for comments. The final section included an overall usefulness rating and a free-text field for suggestions and critiques. Respondents were recruited from ongoing real-world annotation projects using AnnotAISE and completed the online Google Forms voluntarily after interacting with the system. For example, Oliveira et al. [7, 8] used AnnotAISE to annotate code smells, while Canuto et al. [3] used it to annotate sentiments in agile meeting discussions.

Table 2: Structure of the questionnaire for Annotators

ID	Section	Question
		Evaluation of the AnnotAISE Tool
Q1	TAM	Finding and accessing the annotation tasks assigned to me was easy and intuitive.
Q2	TAM	Recording my responses in the annotation form was easy and intuitive.
Q3	TAM	Navigating between the annotation items was easy and intuitive.
Q4	TAM	AnnotAISE was useful for completing the annotation tasks assigned to me.
Q5	TAM	The platform provided the necessary features for me to perform the annotation tasks smoothly.
Q6	TAM	The in-platform guideline viewing feature was useful. (if applicable)
Q7	TAM	I would use AnnotAISE in the future for annotation tasks.
Q8	TAM	I would use AnnotAISE in the future if invited to participate in a new annotation project.
		Participant Characterization
Q9	Charact.	What is your highest level of education?
Q10	Charact.	What is your gender?
Q11	Charact.	What is your level of familiarity with annotation tasks?
Q12	Charact.	Have you participated in any annotation project before AnnotAISE?
Q13	Charact.	If yes, which platforms or tools did you use?
Q14	Charact.	If yes, how would you rate your experience in that/those project(s)?
Q15	Charact.	Please comment on your previous experience with annotation tasks.
Q16	Charact.	If you have a basis for comparison, briefly describe how you evaluate the differences between AnnotAISE and other annotation tools or methods you have previously used.
		Thank you!
Q17	Final	Any further comments or suggestions about the tool?

Table 3: Structure of the questionnaire for Administrators

ID	Section	Question
		Evaluation of the AnnotAISE Tool
Q1	TAM	Creating and configuring a project in AnnotAISE was easy and intuitive.
Q2	TAM	Importing data via CSV was easy and intuitive.
Q3	TAM	Configuring data fields and questions of the annotation form was easy and intuitive.
Q4	TAM	Inviting and assigning users to an annotation task was easy and intuitive.
Q5	TAM	Inspecting and analyzing the responses on the dashboard was easy and intuitive.
Q6	TAM	Exporting the annotated data was easy and intuitive.
Q7	TAM	AnnotAISE improved my productivity in creating and configuring data annotation projects.
Q8	TAM	AnnotAISE made the task distribution process among annotators more efficient.
Q9	TAM	The data field options (text, image, code, audio, video, PDF) were useful in meeting my project's needs.
Q10	TAM	The question type options (text, number, multiple choice, numerical range) were useful for structuring my project's annotation form.
Q11	TAM	The LLM tie-breaking feature was useful for resolving disagreements between annotators.
Q12	TAM	The responses dashboard was useful for tracking the progress and quality of the annotation tasks.
Q13	TAM	The CSV data export feature was useful for analyzing the results outside the platform.
Q14	TAM	I would use AnnotAISE in the future to configure and manage annotation projects for my team.
Q15	TAM	I would use AnnotAISE in the future to analyze annotation results.
		Participant Characterization
Q16	Charact.	What is your highest level of education?
Q17	Charact.	What is your gender?
Q18	Charact.	What is your level of experience in creating and managing annotation projects?
Q19	Charact.	Have you used any of these platforms to organize or label data before AnnotAISE?
Q20	Charact.	If yes, how would you describe your experience using those platforms?
Q21	Charact.	How do you evaluate the differences between AnnotAISE and the tools you have previously used?
Q22	Charact.	Briefly describe the annotation task you performed in AnnotAISE.
		Thank you!
Q23	Final	Any further comments or suggestions about the tool?

5 Results

This section presents the findings of our evaluation study, structured around the RQs (Section 4). We combine quantitative analysis of the Likert-scale responses across questionnaires with qualitative insights derived from the open-ended feedback fields included after each TAM statement. Table 4 summarizes the average Likert-scale scores for all of the TAM questions presented in Tables 2 and 3.

Overview of the participant's characterization: A total of 28 participants took part in the study: 21 annotators and 7 project

Table 4: Average Likert-scale scores for AnnotAISE usability evaluation.

Question ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
Administrators	4.14	3.00	3.43	4.71	4.00	4.86	4.57	4.71	4.57	4.86	3.00	4.71	4.86	4.71	4.14
Annotators	4.67	4.38	4.76	4.67	4.67	4.10	4.52	4.71	–	–	–	–	–	–	–

administrators. The annotator group was composed mostly of undergrad students (76.2%, Q9), with the remaining participants in higher degree programs. Most identified as male (85.7%, Q10), and only 28.6% (Q12) reported any prior experience with annotation projects—mainly using general-purpose tools rather than dedicated annotation platforms. Consistently, 42.9% had no familiarity with annotation tasks before this study, 38.1% considered themselves beginners, and the remaining 19% had higher experience (Q11). This profile aligns well with the target audience of AnnotAISE, designed to support both first-time and experienced annotators.

The administrator group was smaller but academically more advanced: 42.9% were enrolled in a master's program (Q16) and the others were distributed across undergrad and doctoral levels. The majority also identified as male (85.7%, Q17), and 85.7% described themselves as beginners in managing annotation projects (Q18). Notably, no administrator had previously used any dedicated annotation platform (Q19), which positioned all of them as first-time users. The configured projects covered a representative range of use cases—including smell classification, image-based annotation, structured annotation via CSV import, and numerical scoring tasks — exercising the platform across diverse scenarios.

RQ1 — Perceived Ease of Use. Annotators rated AnnotAISE as easy to use across all items, led by navigation between instances (Q3, $\mu = 4.76$), finding assigned tasks (Q1, $\mu = 4.67$), and submitting responses (Q2, $\mu = 4.38$), showing that the core workflow flows smoothly even for first-time users. Administrators rated recurring operations highly— exporting data (Q6, $\mu = 4.86$) and assigning annotators (Q4, $\mu = 4.71$)—but the initial setup emerged as the main friction: CSV import (Q2, $\mu = 3.00$) and configuring data fields and questions (Q3, $\mu = 3.43$) were the lowest-rated items. This points to project setup and data ingestion as clear targets for a more guided import flow and a simpler configuration interface.

RQ2 — Perceived Usefulness. Both groups found AnnotAISE useful. Annotators agreed it helps them complete tasks (Q4, $\mu = 4.67$) and provides the needed resources (Q5, $\mu = 4.67$); the optional guideline viewer was lower but still positive (Q6, $\mu = 4.10$). Administrators valued its flexibility, especially the variety of question types (Q10, $\mu = 4.86$) and CSV export (Q13, $\mu = 4.86$), followed by the dashboard (Q12, $\mu = 4.71$) and data-field support (Q9, $\mu = 4.57$). The exception was the LLM tie-breaking feature (Q11, $\mu = 3.00$): as disagreement resolution was relevant to only a subset of projects, we treat this as inconclusive rather than negative and plan a dedicated evaluation with higher-disagreement settings.

RQ3 — Behavioral Intention to Use. Participants in both roles expressed a clear intention to reuse AnnotAISE. Annotators would use it for future tasks (Q7, $\mu = 4.52$) and join a new project built on it (Q8, $\mu = 4.71$). Administrators would use it to configure and manage future projects (Q14, $\mu = 4.71$); intention to use it specifically for analyzing results was lower (Q15, $\mu = 4.14$), reinforcing result analysis as a refinement direction. Since respondents came from an ongoing project already using AnnotAISE, these responses reflect an intention to continue using it.

6 Conclusion and Future Work

This paper presented AnnotAISE, a web-based platform designed to support collaborative data annotation in empirical research. The platform provides a structured environment for multi-annotator workflows, incorporating majority voting mechanisms and LLM-assisted conflict resolution to facilitate the generation of high-quality annotated data from subjective datasets. A usability evaluation demonstrated that AnnotAISE performs effectively for both administrators and annotators across diverse scenarios, confirming its practical viability in research settings.

Regarding future work, two primary directions are planned. First, we acknowledge that the current evaluation assesses perceived usability rather than the quality of the resulting annotations. We therefore plan a follow-up study measuring annotation quality directly, through inter-annotator agreement, agreement with an expert-defined gold standard, and annotation time per instance, and comparing AnnotAISE with existing annotation tools on the same dataset and annotation guideline. This will allow us to assess whether the coordination and disagreement resolution mechanisms provided by the platform translate into measurable gains in reliability and reproducibility, beyond the usability results reported here. Second, we plan to evolve the current LLM-assisted tiebreaking mechanism — presently limited to text and code contexts — to support multimodal inputs, including image, video, and other context types already available on the platform. This will require integrating multimodal language models capable of processing and reasoning over non-textual data, enabling consistent and justified conflict resolution across a wider range of annotation tasks.

Artifact Availability

The artifacts are available at https://doi.org/10.5281/zenodo.21462965 [16] under the MIT license.

Acknowledgments

This research was partially funded by the Brazilian funding agencies CAPES/COFECUB (Grant 88881.879016/2023-01 and Ma1036/24), CNPq (Grant 406089/2025-6), FAPESP (Grant 2023/00811-0), and FAPEMIG (Grant APQ-01488-24). We also acknowledge the Brazilian company Stone[3] for their financial support.

References

[1] Toufique Ahmed, Premkumar Devanbu, Christoph Treude, and Michael Pradel. 2025. Can LLMs Replace Manual Annotation of Software Engineering Artifacts?. In *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*. IEEE, 526–538. doi:10.1109/MSR66628.2025.00086

[2] ATLAS.ti. 2026. ATLAS.ti: Qualitative Data Analysis Software. https://atlasti.com/. Accessed: 2026-04-24.

[3] https://www.stone.com.br/

[3] Theo Canuto, Eduardo Sardenberg, Paulo Mann, Matheus Utino, Daniel Coutinho, Anderson Uchôa, and Juliana Alves Pereira. 2026. Emotion Recognition in Agile Software Meetings: A Comparative Study of ML, DL, and Text-based LLM Approaches. In *Proceedings of the 19th International Conference on Cooperative and Human Aspects of Software Engineering*. 220–232.

[4] Clarifai. 2026. Clarifai Documentation. https://docs.clarifai.com/. Accessed: 2026-04-24.

[5] Atacílio Costa Cunha, Tayana Conte, and Bruno Gadelha. 2021. What really matters in code review? A study about challenges and opportunities related to code review in industry. In *Proceedings of the XX Brazilian Symposium on Software Quality*. 1–10.

[6] Fred D. Davis. 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly* 13, 3 (1989), 319–340.

[7] Igor Soares de Oliveira, Alexandre Andrade, Rayssa Athayde, Eduardo Nadai, and Juliana Alves Pereira. 2026. Scylla: A Unified Tool for Code Smell Classification in Python. In *Anais do Simpósio Brasileiro de Engenharia de Software (SBES)* (São Paulo/SP). SBC, São Paulo, SP, Brasil, 1–7.

[8] Igor Soares de Oliveira, Alexandre Andrade, Saymon Souza, Eduardo Figueiredo, and Juliana Alves Pereira. 2026. A Multi-Approach Evaluation of Python Code Smell Detection Using Heuristics, Learning Models, and Language Models. In *Anais do Simpósio Brasileiro de Engenharia de Software* (São Paulo/SP). SBC, São Paulo, SP, Brasil, 1–12.

[9] Edna Dias Canedo and Bruno Cordeiro Mendes. 2020. Software requirements classification using machine learning algorithms. *Entropy* 22, 9 (2020), 1057.

[10] Isabella Ferreira, Cleidson Souza, and Eduardo Figueiredo. 2022. Characterizing Commits in Open-Source Software. In *Proceedings of the XXXVI Brazilian Symposium on Software Engineering*. Association for Computing Machinery.

[11] Ángel González-Prieto, Jorge Perez, Jessica Diaz, and Daniel López-Fernández. 2023. Reliability in software engineering qualitative research through Inter-Coder Agreement. *Journal of Systems and Software* 202 (2023), 111707.

[12] Tjaša Heričko and Boštjan Šumak. 2023. Commit classification into software maintenance activities: A systematic literature review. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 1646–1651.

[13] Marc Herrmann, Martin Obaidi, Larissa Chazette, and Jil Klünder. 2022. On the subjectivity of emotions in software projects: How reliable are pre-labeled data sets for sentiment analysis? *Journal of Systems and Software* 193 (2022), 111448.

[14] Xing Hu, Ge Li, Xin Xia, David Lo, Shuai Lu, and Zhi Jin. 2022. Correlating Automated and Human Evaluation of Code Documentation Generation Quality. *ACM Transactions on Software Engineering and Methodology* (2022). doi:10.1145/3502853

[15] HumanSignal. 2026. Label Studio Documentation. https://labelstud.io/guide/. Accessed: 2026-04-24.

[16] João Paulo Lopes, Bruno Tardin Fernandes, Beatriz Ritter, Daniel Coutinho, Robbie Carvalho, Alessandro F. Garcia, and Juliana Alves Pereira. 2026. *AnnotAISE: Web-Based Data Annotation Platform For Software Engineering Research*. doi:10.5281/zenodo.21462965

[17] Nicole Novielli, Daniela Girardi, and Filippo Lanubile. 2018. A benchmark study on sentiment analysis for software engineering research. In *Proceedings of the 15th International Conference on Mining Software Repositories*. 364–375.

[18] Simona Prokić, Nikola Luburić, Jelena Slivka, and Aleksandar Kovačević. 2024. Prescriptive Procedure for Manual Code Smell Annotation. *Science of Computer Programming* 238 (2024), 103168. doi:10.1016/j.scico.2024.103168

[19] Roboflow. 2026. Roboflow Annotate. https://docs.roboflow.com/annotate/annotation-tools. Accessed: 2026-04-24.

[20] Roboflow. 2026. Roboflow Documentation. https://docs.roboflow.com/. Accessed: 2026-04-24.

[21] Christoph Sager, Christian Janiesch, and Patrick Zschech. 2021. A Survey of Image Labelling for Computer Vision Applications. *Journal of Business Analytics* 4, 2 (2021), 91–110. doi:10.1080/2573234X.2021.1908861

[22] Jaydeb Sarker, Sayma Sultana, Steven R. Wilson, and Amiangshu Bosu. 2023. ToxiSpanSE: An Explainable Toxicity Detection in Code Review Comments. In *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. IEEE, 1–12. doi:10.1109/ESEM56168.2023.10304855

[23] Eyal Shnarch, Alon Halfon, Ariel Gera, Marina Danilevsky, Yannis Katsis, Leshem Choshen, Martin Santillan Cooper, Dina Epelboim, Zheng Zhang, Dakuo Wang, et al. 2022. Label sleuth: From unlabeled text to a classifier in a few hours. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 159–168.

[24] Lili Song, Xiangxin Lu, Taiyue Wang, Xin Xia, David Lo, and John Grundy. 2023. A Practical Human Labeling Method for Online Just-in-Time Software Defect Prediction. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. Association for Computing Machinery.

[25] Supervisely. 2026. Supervisely: Computer Vision Platform. https://supervisely.com/. Accessed: 2026-04-24.

[26] Supervisely. 2026. What is Supervisely. https://docs.supervisely.com/. Accessed: 2026-04-24.

[27] Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large Language Models for Data Annotation and Synthesis: A Survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 930–957. https://aclanthology.org/2024.emnlp-main.54/

[28] Morteza Zakeri-Nasrabadi, Saeed Parsa, Ehsan Esmaili, and Fabio Palomba. 2023. A Systematic Literature Review on the Code Smells Datasets and Validation Mechanisms. *Comput. Surveys* 55, 13s, Article 304 (2023), 48 pages. doi:10.1145/3596908