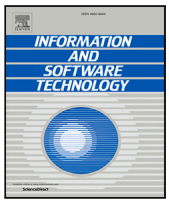
Information and Software Technology 199 (2026) 108254

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof



Toward explaining value concerns in software artifacts: A human-in-the-loop method

Davoud Mougouei [a,*], Son Pham [a], Ahmad Azarnik [b], Mahdi Fahmideh [c], Arif Nurwidyantoro [d], Elahe Mougouei [e], Amanul Haque [f], Hoa Khanh Dam [g], Saima Rafi [h], Javed Ali Khan [i]

[a] Deakin University, Australia
[b] University of Technology Malaysia, Malaysia
[c] University of Southern Queensland, Australia
[d] Universitas Gadjah Mada, Indonesia
[e] Islamic Azad University Najafabad, Iran
[f] North Carolina State University, USA
[g] University of Wollongong, Australia
[h] Edinburgh Napier University, UK
[i] University of Hertfordshire, UK

ARTICLE INFO

Dataset link: https://figshare.com/s/db6a26280df6bb64649a

Keywords:
Explaining
Value concerns
Software artifacts
LLMs

ABSTRACT

Context: Violations of human values in software can lead to user dissatisfaction, economic harm, and loss of trust. Existing software engineering research has focused on detecting such violations through manual labeling or automated classification, without explaining the mechanisms through which they occur. This lack of explanation makes reported violations difficult to verify and offers limited guidance for addressing them.
Objectives: This study introduces a human-in-the-loop method that explains potential violations of human values in software artifacts—*value concerns*—by articulating the conditions under which they may arise.
Methods: The proposed human-in-the-loop method integrates the automated reasoning capabilities of a large language model (ChatGPT) with human assessment to identify and articulate value concerns in software artifacts. We evaluate the method through a case study of 1200 Android and iOS APIs, assessing its ability to produce traceable accounts of potential value violations.
Results: The results show that ChatGPT's inferences about value concerns are reasonably accurate but incomplete. Errors arise from unfounded explanations and overextended reasoning, while the persuasive presentation of LLM outputs can make these errors difficult to recognize. The human-in-the-loop method mitigates these issues through human review.
Conclusion: Our findings support the use of LLMs as reasoning partners that, when combined with human oversight, can effectively explain value concerns in software artifacts.

1. Introduction

Violations of human values in software result in user dissatisfaction and socioeconomic repercussions [1–3]. Notable examples include the U.S. Senate's hearing on child safety negligence by major social media platforms [4] and the removal of Australia's CovidSafe app over privacy concerns [5]. However, tracing values in software is challenging due to their subjective and context-dependent nature [1,3,6]. Values serve as motivations behind choices, making them difficult to trace without adequate reasoning. Performing such reasoning at scale demands automation. Current efforts to identify human values and their violations in software either rely on laborious manual labeling of artifacts, which does not scale [3,6–8], or use classification approaches [9–13] that fail to explain their findings – as to how and why values are at risk – thus limiting transparency and verifiability. The absence of such explanations also makes it difficult to trace and mitigate violations of human values.

* Corresponding author.
E-mail addresses: dmougouei@gmail.com (D. Mougouei), son.pham@deakin.edu.au (S. Pham), ahmad.azarnik@gmail.com (A. Azarnik), mahdi.fahmideh@usq.edu.au (M. Fahmideh), arifn@ugm.ac.id (A. Nurwidyantoro), elahe.mougouei@gmail.com (E. Mougouei), ahaque2@ncsu.edu (A. Haque), hoa@uow.edu.au (H.K. Dam), s.rafi@napier.ac.uk (S. Rafi), j.a.khan@herts.ac.uk (J.A. Khan).

https://doi.org/10.1016/j.infsof.2026.108254
Received 28 November 2025; Received in revised form 31 May 2026; Accepted 7 July 2026
Available online 16 July 2026

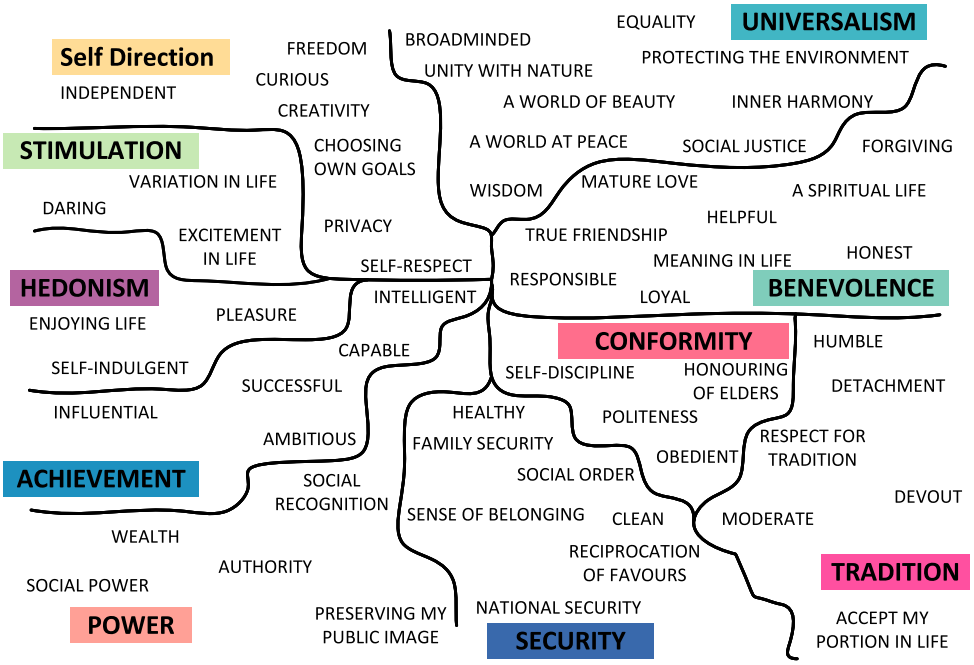


Fig. 1. Schwartz's basic model of human values [25,26].

Large Language Models (LLMs) such as ChatGPT, on the other hand, offer a promising avenue to address these challenges by enabling automated inference with contextual explanation, offering insights into how values may be expressed or violated [14,15]. The generative strength of LLMs has proven effective for eliciting insights from software artifacts, including requirements elicitation [16–18]. The contributions of this paper, therefore, are as follows.

(i) *Methodological contribution*. We propose a human-in-the-loop method [19–21] (Fig. 2) that pairs automated reasoning capabilities of LLMs with human oversight to infer conditions that may lead to violations of human values, referred to as *value concerns*. Unlike existing classification (labeling) approaches [6,10–13], our method generates explanatory accounts that show how and why values are at risk. Our aim is not to merely detect violations, but to support transparent reasoning by explaining their mechanisms. Evaluation thus centers on the quality of explanations, instead of predictive accuracy metrics [16,22–24].

(ii) *Empirical contribution*. We apply the proposed method to a case study of 1200 Android and iOS APIs. This provides empirical evidence about the kinds of value concerns that can be inferred from mobile API documentation, the accuracy of LLM-inferred explanations, and their limitations. Our analysis aims to answer the following research questions:

> **RQ1**: How effectively can an LLM (ChatGPT) infer value concerns?
> **RQ2**: What types of value concerns can an LLM (ChatGPT) infer?

ChatGPT is initially tasked with inferring value concerns in the documentation of 1200 Android and iOS APIs. These concerns are evaluated through a dialogue-based interactive assessment process conducted by a panel of human evaluators who review ChatGPT's explanations, request clarifications, and debate its reasoning. To enhance reliability, a supplementary analysis establishes ground truth examples by removing ChatGPT's persuasive influence. These examples, alongside Schwartz's theory of human values [27], are fed back into ChatGPT to reassess the endorsed findings. The value concerns confirmed by this process are subsequently rated by evaluators. Finally, the validated concerns are categorized based on their underlying mechanisms.

Our findings show that ChatGPT's inferences about value concerns are fairly accurate but incomplete, with inaccuracies arising from unfounded explanations, extended reasoning, and persuasive effects of the language model. Our human-in-the-loop method mitigates these issues through inference-assessment cycles. We conclude that LLMs should be viewed as explanatory tools and reasoning partners, guided by human oversight, providing insights into mechanisms of value violations rather than functioning as classifiers.

2. Background and related work

Values are defined as guiding principles of what people consider important [28]. Among the theories characterizing human values [29,30], Schwartz's theory [27] has been widely used and empirically validated across several disciplines [31] including software engineering [3,13,32–34]. While Schwartz's theory [26,27] presents values as a continuum of motivations (Fig. 1), it acknowledges the theory's flexibility to study values in discrete categories for practical purposes [27]; several studies of human values in software use Schwartz's theory as a taxonomy to identify value-related content [2,3,6,10,32,33].

The abstract, context-dependent, and implicit nature of human values [35,36] makes them challenging to define and detect in software artifacts [1,37] without adequate reasoning. This limits the effectiveness of existing classifiers used in software engineering, which can only specify the presence or absence of value violations or alignments [9–13], without providing reasons. Such classifiers fail to explain the rationale behind value violations (alignments) due to the lack of reasoning; understanding how values are expressed in software requires scalable reasoning, which is beyond the scope of existing classifiers and tedious and error-prone manual analysis [3,6–8].

Table 1 provides a qualitative comparison between prior approaches that reveal value-related content in software artifacts and our proposed human-in-the-loop method. The comparison considers automated elicitation, human oversight, context-awareness, coverage of values, explainability, scalability, and reproducibility. Manual approaches rely entirely on human interpretation, which goes beyond oversight. While this enables context-awareness, it largely limits scalability and reproducibility. Keyword-based/NLP approaches, on the other hand, can support scalable and more reproducible analysis of value concerns,

Table 1
Comparison of approaches that identify value-related content in software artifacts.

Approach	Papers	Automated	Oversight	Context-aware	Broad-coverage	Explainable	Scalable	Reproducible
Manual-broad-coverage	[2,3,6–8,38–40]	No	Yes	Yes	Yes	No	No	No
Manual-limited-coverage	[41,42]	No	Yes	Yes	No	No	No	No
Keyword-based/NLP-broad-coverage	[9,10,13,43]	Yes	Limited	Limited	Yes	No	Yes	Yes
Keyword-based/NLP-limited-coverage	[11,12,44–51]	Yes	Limited	Limited	No	No	Yes	Yes
Human-in-the-loop	This work	Yes	Yes	Yes	Yes	Yes	Limited	Limited

Automated: whether the elicitation of value-related content is performed automatically.
Oversight: whether humans review, validate, or refine the outputs produced by the approach.
Context-aware: whether the meaning and complexity of value-related content are considered, rather than isolated terms.
Broad-coverage: whether a broad range of values is covered by the approach, rather than a limited set.
Explainable: whether reasoning with traceable rationales is provided for the findings.
Scalable: whether the elicitation of value-related concerns can be performed at scale.
Reproducible: whether applying the approach to the same artifacts is likely to produce consistent outputs.

but they generally act as classifiers, reporting labels with limited context-awareness. As summarized in Table 1, all manual and keyword-based/NLP approaches fail to provide traceable rationales for their detected value concerns, limiting traceability.

Our human-in-the-loop method addresses this gap by producing value concerns comprising rationales directly linked to the API documentation, allowing the outputs to be inspected and verified through human oversight. Its reproducibility, however, remains limited, mainly due to the stochastic nature of LLM-based reasoning, producing variation across runs. The method thus does not guarantee that all possible value concerns will be elicited in the same way each time. The need for human oversight also limits scalability.

Recent research highlights the potential of Large Language Models (LLMs) in a wide range of software engineering tasks, including automated comment generation [52], code comprehension and refinement [53], software testing [54], and program repair [55]. LLMs have also been used to automate GitHub issue labeling [56] and support human values such as security and privacy by enhancing cyber threat detection [57], facilitating security assertions [58], and identifying vulnerability patterns [59]. Their scalability is evident in analyzing privacy policies [60], summarizing sentiment in app reviews [61], and uncovering emotional causes in GitHub discussions [62]. LLMs have also been optimized for energy efficiency in support of environmental goals [63]. Their generative capacity has proven effective in eliciting novel features from app reviews [16] and in broader requirements elicitation [17]. Despite these advances, LLMs remain underutilized for reasoning about the broader range of values in software. Limited work in this direction includes value-inspired user story generation using ChatGPT [18].

However, LLM reasoning is constrained by inherent limitations, such as hallucinations – or more precisely, confabulations [64–66] – and biases stemming from training data, model architecture, and policy decisions [67,68]. Hallucination in LLMs can involve generated content that diverges from the user input, contradicts previously generated context, or misaligns with established world knowledge [69]. It can therefore affect not only factual outputs, but also the reliability of model-generated reasoning, because plausible yet incorrect information remains an unresolved problem in LLMs [70]. This matters for the present study as explanations generated by an LLM may appear coherent while still requiring verification against the source artifact. This position is consistent with work showing that explicit verification steps, where a model drafts an answer, generates verification questions, answers them independently, and revises the response, can reduce hallucinations across several tasks [70].

LLM outputs can also reflect social and representational biases that affect interpretation and decision-making. Studies show that LLMs exhibit social identity biases, including ingroup solidarity and outgroup hostility [71]; that LLMs can pass explicit social-bias tests while still forming biased associations [72]; and that LLM-based resume evaluation can produce gender- and race-linked score differences across randomized candidate profiles [73]. Therefore, the explanations generated by LLMs in this study are treated as candidate interpretations rather than independent evidence; they are checked against the source artifacts and evaluated by human experts.

Moreover, research in explainable AI indicates that traditional performance metrics, such as the F1-score, are often inadequate or misleading for assessing explainable AI models [22–24]. These limitations highlight the need for human involvement to enhance contextual relevance and ensure quality explanations through oversight.

3. Study design

We illustrate our proposed human-in-the-loop method on documentation of mobile APIs (Android and iOS) as they describe how apps access sensitive user data (e.g., location, health) [74,75] and embed constraints and assumptions that shape software behavior [76,77], making them ideal for inference of value concerns.

3.1. Collecting and preparing API documentation

The study began with collecting 3000 standard and third-party Android and iOS APIs from the Android API Reference, Apple Developer Documentation, and third-party repositories sourced from GitHub. We removed APIs with insufficient documentation, resulting in a subset of 2535 APIs: 596 standard and 1420 third-party Android APIs, as well as 148 standard and 371 third-party iOS APIs. A Python script was used to extract documentation from their URLs. When documentation exceeded the input token limitation of gpt-4 (approximately 8000 tokens per transaction, including both prompt and response), we applied LexRankSummarizer, an extractive summarization method based on the LexRank algorithm [78], chosen for its simplicity and faithfulness to source content [78]. On average, 98.7% of the original content was preserved.

3.2. Specifying API themes and sampling APIs

To better understand value concerns across different API functionalities, we categorized the APIs under 10 themes (Fig. 9). Since the API topics were defined by publishers, we did not need to discover latent topics; instead, ChatGPT (gpt-4) was prompted to group the 1545 publisher-defined topics into the themes. Each theme was described with examples to ensure explainability [79]. A human evaluator (co-author) reviewed the themes and adjusted descriptions when needed. ChatGPT was then prompted to assign each API to the most relevant theme. Evaluators reexamined the accuracy of these assignments, as detailed in Section 3.4. Stratified sampling was used to select 1200 APIs across all themes.

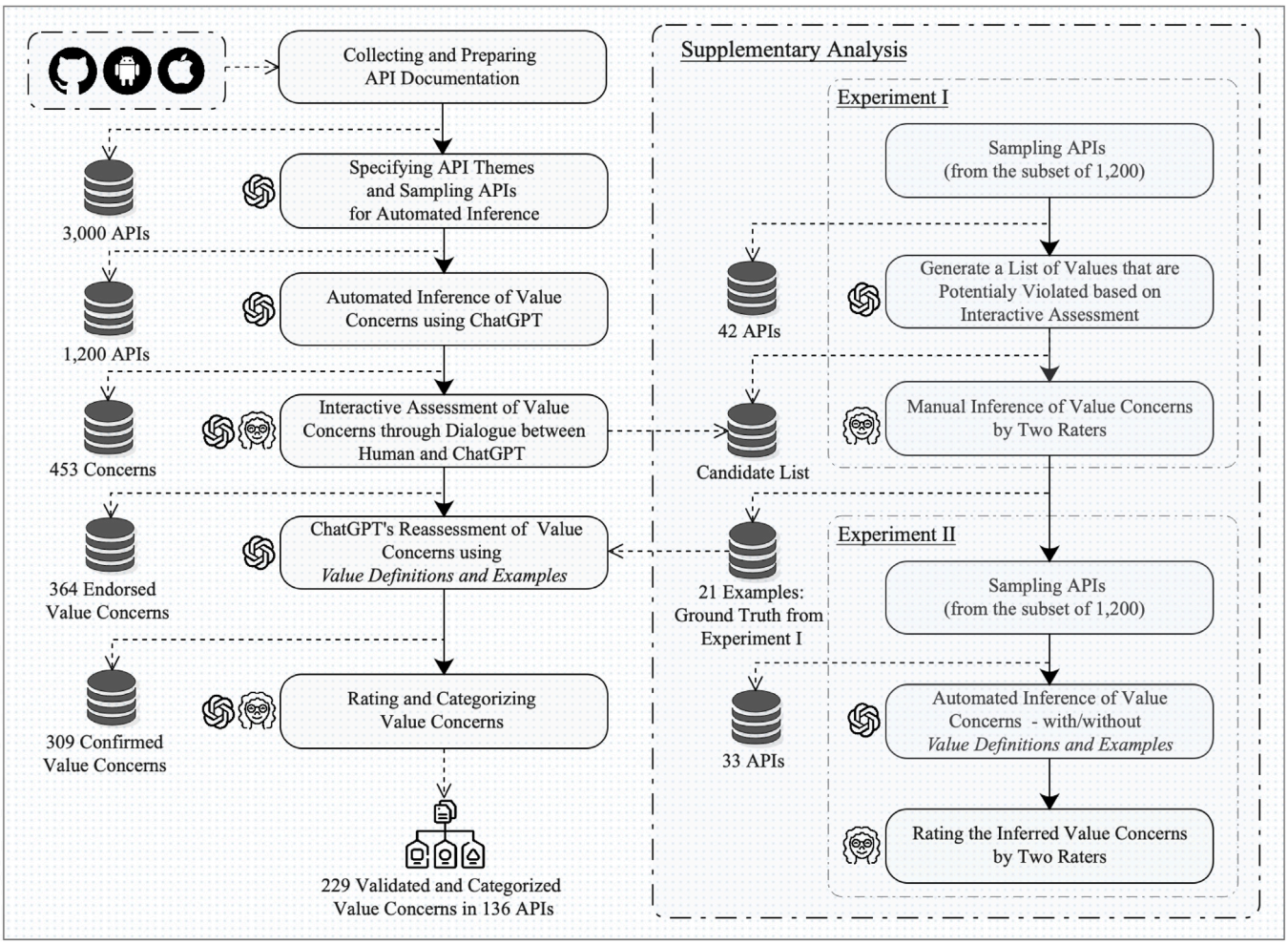


Fig. 2. The proposed human-in-the-loop method combines LLM inference with human oversight. Supplementary analysis established ground truth by mitigating ChatGPT's persuasive effects (Experiment I) and evaluating inferences with and without value definitions and examples (Experiment II).

3.3. Automated inference of value concerns

We prompted the gpt-4 API to infer value concerns using the following (summarized) prompt:

> `Context;` Please follow the instructions listed below.
>
> 1. Consider `the definition of mobile app user.`
> 2. Consider `Schwartz's taxonomy of human values.`
> 3. Consider `the API documentation.`
> 4. Detect and explain up to three of the most relevant value violations, choosing justifications that meet the criteria of factuality, plausibility, and specificity.
> 5. The analysis must be focused on how the API violates the values of the mobile app users, not developers.
> 6. Here are some examples: `formatted examples.` (Table A.1)
>
> *The API relies on color visualizations to provide feedback. It infringes upon equality (Universalism) by neglecting the needs of users with visual impairments.*

When prompted to identify value violations, ChatGPT often explained indirect violations of values through chains of cause-and-effect reasoning that linked these violations to their underlying conditions. These explanations rarely justified an imminent violation; they described conditions that could lead to violations (value concerns). To reflect this, we focused the prompt on identifying violations, knowing it actually revealed value concerns. The prompt was iteratively refined to improve accuracy. Initial responses revealed unfounded justifications for violations, prompting adjustments to emphasize the three criteria of *Factuality*, *Plausibility*, and *Specificity* in LLM explanations. Factuality required traceability to API documentation; plausibility emphasized likely and significant concerns; and specificity focused on sufficient details specific to the context. The LLM was tasked to infer the most relevant violations (concerns) that met these criteria, per API. This also helped streamline human evaluations. The search for violations was framed from the app users' perspective. Despite these refinements, the responses still included convincing yet unfounded explanations, necessitating focused evaluation.

3.4. Interactive assessment of value concerns

The value concerns inferred by ChatGPT were evaluated by seven evaluators with degrees in software engineering or related fields and experience in software development. They were trained in Schwartz's model by reviewing key works [26,27] and examining example concerns from prompt refinement. Evaluation focused on alignment with Schwartz's theory and the quality of ChatGPT's explanations, assessed by the three criteria factuality, plausibility, and specificity, used for inferring value concerns (Section 3.3). A user-centric perspective was maintained, prioritizing end-user values. Evaluation began with reading the associated API documentation and reviewing ChatGPT's justification to check if it met the assessment criteria (Fig. 3). Evaluators marked their agreement as •Agreed-Justified; otherwise, they requested clarification, triggering a clarification prompt.

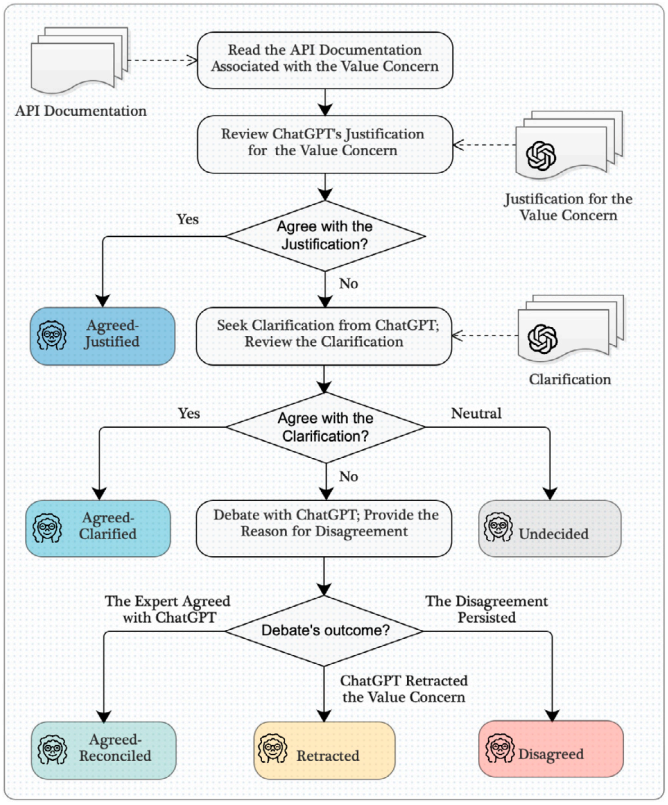


Fig. 3. The interactive assessment process.

> You were given the `inference prompt` to infer value violations; you inferred `value; value item; justification.` I am not convinced; please clarify. (Table A.2)

The evaluator marked •Agreed-Clarified when they agreed with ChatGPT's clarification; •Undecided was used when they could neither agree nor disagree. In cases of disagreement, the evaluator triggered the debate prompt, providing their reason.

> Using the `clarification prompt`, you inferred `value; value item; justification.` I was not convinced, so you provided `clarification.` I disagree: `reason.` (Table A.3)

The debates led to three outcomes: (i) the evaluator agreed with ChatGPT, flagged •Agreed-Reconciled, (ii) ChatGPT withdrew its nominated value concern, flagged •Retracted, or (iii) disagreement persisted, flagged •Disagreed. About 82% of the findings were endorsed following clarifications and debates. Evaluators also reviewed ChatGPT's theme assignments for APIs with potential value violations, adjusting themes for nearly 5% of them.

3.5. Supplementary analysis

Following the interactive assessment of value concerns, we identified cases where ChatGPT used extended reasoning that was hard to trace in the API documentation and persuaded evaluators. To address this, we conducted a supplementary analysis (Fig. 2) with two experiments.

3.5.1. Experiment I

To mitigate ChatGPT's persuasive effects, ground truth was established through human inference without referencing LLM explanations. Applying this across 42 Android and iOS APIs from the original 1200 was tedious. A full analysis required two coders to independently infer concerns across 42 APIs for all 10 selected values, totaling at least 840 reasoning units. To manage this, we generated a *Candidate List* of potentially violated values per API based on the interactive assessment (Section 3.4), excluding ChatGPT's explanations. This list guided two raters, who also considered concerns beyond the candidate list when appropriate. Trained on Schwartz's refined model [27], the raters independently reviewed the API documents to infer concerns. Disagreements were reconciled via mediation using their explanations. Cohen's Kappa was 0.71 (pre-reconciliation), and 21 mutually agreed concerns were endorsed as ground truth with rater explanations. ChatGPT showed a precision of 68% and recall of 52%, reflecting a fair but incomplete ability to detect valid concerns. ☛ Consistent with [14], ChatGPT demonstrated a fairly accurate but incomplete ability to identify value concerns.

3.5.2. Experiment II

We aimed to enhance the accuracy of ChatGPT's inferences by (a) enriching the prompt with explicit definitions of values and (b) providing ground truth examples to the language model. To evaluate the effectiveness of this approach, an experiment was conducted using 33 sampled Android and iOS APIs, a subset of the 1200 APIs (Fig. 2). Schwartz's conceptual definitions of values [27] and 21 ground truth examples from Experiment I (Section 3.5.1) were included in the prompt described in Section 3.3. The values listed in the prompt were defined using the latest definitions of values from [27].

> `Context;` Please follow the instructions listed below.
>
> 1. Consider the following `concern` about `value x.`
> 2. Consider `Schwartz's definition of value x.`
> 3. Consider `the API documentation.`
> 4. Check if this `concern` offers a valid (factual, plausible, and specific) explanation for a violation of `value x.`
> 5. The analysis must be focused on how the API violates the values of the mobile app users, not developers.
> 6. `Ground truth examples from Experiment I.` (Table A.4)

ChatGPT reassessed the previously inferred value concerns using the value definitions and examples and labeled them as 'Valid' or 'Invalid'. Two independent raters also reviewed the API documents, labeling concerns as 'Valid' or 'Invalid', achieving a Cohen's Kappa of 0.80 (prior to reconciliation to full agreement). Precision improved when value definitions and examples were included, reaching 70% compared to 62% without, indicating reduced false positives. Recall was similar between the two configurations, at 53% with value definitions and examples and 50% without. ☛ Providing value definitions and examples improved ChatGPT's precision and slightly enhanced its ability to identify more concerns.

3.6. ChatGPT's reassessment of value concerns

Experiment II of the supplementary analysis (Section 3.5.2) showed that value definitions and ground truth examples improved the accuracy of inferring value concerns. Building on this and the LLMs' ability to evaluate machine-generated text [80,81] and engage in self-criticism [82,83], we automatically reassessed the 364 value concerns endorsed in the interactive assessment (Section 3.4). Using the prompt from Experiment II, with value definitions [27] and ground truth examples, ChatGPT applied the same evaluation criteria from interactive assessment (factuality, plausibility, and specificity) to reassess value concerns, confirming 309 cases.

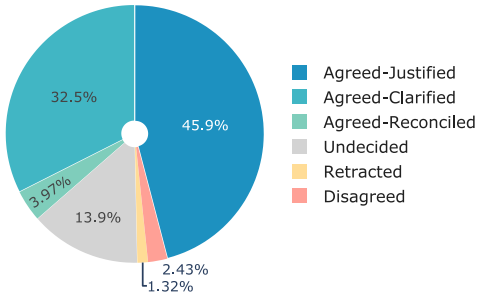


Fig. 4. Evaluator opinions on ChatGPT-inferred value concerns.

3.7. Rating and categorizing value concerns

To evaluate the accuracy of the 309 value concerns confirmed by ChatGPT's reassessment (Section 3.6), they were rated by four of the original seven evaluators (Rater I) from the interactive assessment (Section 3.4), and two independent evaluators (Rater II), also co-authors, not involved in the initial assessment. To ensure objectivity, Rater I did not rate concerns they had previously endorsed. Each concern was independently reviewed by both raters using the same criteria from the interactive assessment and rated as Valid or Invalid. Disagreements were resolved through mediator-facilitated discussions. Inter-rater reliability was measured post-reconciliation, yielding a Cohen's Kappa of 0.86, validating 229 value concerns across 136 APIs. Ratings also confirmed a precision of 74% for ChatGPT's reassessment (Section 3.6). Discussions revealed cases where software security concerns were incorrectly linked to human security without sufficient evidence; raters labeled these as *Software Security*. ChatGPT was also tasked with categorizing endorsed value concerns by their violation mechanisms based on justifications. An independent evaluator, not part of earlier assessments, reviewed all assignments and reassigned 21%.

4. Findings

4.1. ***RQ1***: How effectively can an LLM (ChatGPT) infer value concerns?

4.1.1. Insights from interactive assessment

We sought evaluator opinions on the value concerns inferred by ChatGPT using the process described in Section 3.4. In 82.37% of the cases, evaluators endorsed ChatGPT's findings (Fig. 4), by providing their opinion as a form of agreement based on ChatGPT's initial justifications (●Agreed-Justified), clarifications (●Agreed-Clarified), or following a debate (●Agreed-Reconciled).

Nearly 44% of the agreements were achieved through clarifications and the debates between the language model and the evaluators; ChatGPT retracted (●Retracted) 1.32% of the value concerns following the debates. The disagreements persisted in 2.43% of the cases (●Disagreed) while the evaluators remained ●Undecided about 13.9% of the concerns revealed by ChatGPT, suggesting the need for investigation. ☛ Evaluators and ChatGPT changed their understanding of value concerns through dialogue.

Our analysis (Fig. 5) revealed lower evaluator agreement on ChatGPT's inferences regarding social and altruistic values such as Universalism (Protecting the Environment), Tradition (Respect for Tradition), Benevolence (Helpful, Mature Love, Forgiving), and the personally focused value of Hedonism (Enjoying Life). The most significant shifts toward ●Agreed-Clarified and ●Agreed-Reconciled were observed for Universalism and Achievement, with ChatGPT revising its stance (●Retracted) primarily for Tradition (Detachment). The highest evaluator uncertainty was seen for Benevolence (Mature Love, Forgiving), followed by Hedonism (Enjoying Life, Self-Indulgent), reflecting ambiguity around these values. The high ratio of ●Agreed-Clarified and ●Agreed-Reconciled across all values highlights the importance of Human-LLM dialogue. Evaluators notably aligned with ChatGPT (●Agreed-Reconciled) post-debates on the value items Curious (Self-Direction) and Moderation (Tradition). ☛ The residual disagreements (●Disagreed) involved ChatGPT hallucinations [64], where it gave convincing but unfounded justifications, and extended reasoning that loosely linked causes and effects to suggest violations. Table 2 lists examples of problematic value concerns inferred by ChatGPT through hallucination and extended reasoning.

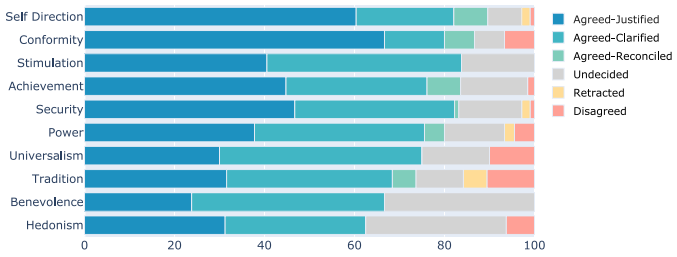


Fig. 5. Distribution (%) of opinions on ChatGPT-inferred value concerns; precision indicated by combined agreement.

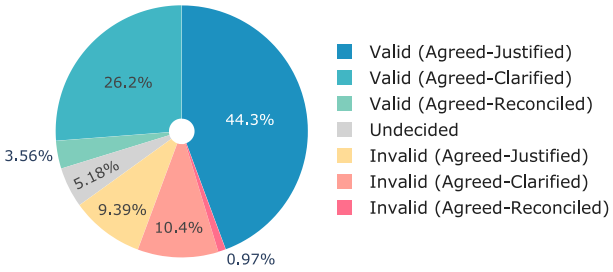


Fig. 6. Reassessment of value concerns: 10% of the concerns found invalid, though they were previously labeled Agreed-Clarified.

4.1.2. Insights from rating value concerns

The 364 value concerns endorsed through the interactive assessment (Section 3.4) were reassessed by ChatGPT using ground truth examples from the supplementary analysis (Section 3.5) and value definitions [27], as explained in Section 3.6. 309 cases were confirmed and subsequently rated by human evaluators (Section 3.7), showing substantial inter-rater agreement (Cohen's Kappa: 0.86). This validated 229 concerns (Precision = 74%) across 136 APIs, with 57% linked to multiple concerns.

☛ The fact that many concerns endorsed in the interactive assessment and confirmed by ChatGPT were rated 'Invalid' or 'Undecided' (Fig. 6) underscores: (a) value subjectivity, (b) LLMs' persuasive impact [64], and (c) the role of rating in exposing unfounded justifications. An intriguing observation was that ChatGPT's reassessment achieved notably higher precision for value concerns initially labeled as ●Agreed-Justified during the interactive assessment (78%) compared to those labeled as ●Agreed-Clarified (68%) and ●Agreed-Reconciled (68%). Moreover, the nearly identical precision values for value concerns previously labeled as ●Agreed-Clarified and ●Agreed-Reconciled suggest that engaging in debates with ChatGPT has not resulted in greater accuracy compared to simply requesting clarifications. ☛ These trends suggest follow-up discussions increased evaluator susceptibility to LLM persuasion, underscoring the need for safeguards.

As Fig. 7 shows, the precision of reassessed value concerns varied across different values, with the highest precision observed for Power. The precision of inferences for the personally-focused value

Table 2
Examples of ChatGPT hallucination and extended reasoning in residual disagreements.

Source evidence	ChatGPT reasoning	Hallucination & extended reasoning
The API documentation concerns the class `javax.security.auth.login.LoginException`: "This is the basic login exception". It contains no data management, personal data processing, user data disclosure, data sharing, privacy risk, or reputation-related functionality.	ChatGPT claims: "The documentation describes the feature 'manage user data' that exposes users' personal data". The clarification repeats and expands this invented premise: the "manage user data" feature "allows the app to access or share the personal data of the App User", and mishandling such data could disclose private information to third parties and tarnish the user's Public Image .	The source gives no basis for "manage user data", "exposes users' personal data", "access or share", "disclose private information to third parties", or "public image". The hallucinated premise is the invented data-management capability. ChatGPT then extends this fabricated premise into a broader value concern: login exception documentation → invented user-data management feature → assumed personal-data exposure → assumed third-party disclosure and reputation harm → Public Image concern.
The documentation describes Async Http Client as a Java library for asynchronous HTTP requests. It supports asynchronous requests, WebSocket, connection pooling, proxy support, TLS, redirects, request filters, and response filters. It does not mention environmental harm, energy use, waste, carbon impact, sustainability features, or anti-environmental behavior.	ChatGPT claims that because the documentation does not mention any feature supporting environmental preservation, this "represents a violation" of Protecting the Environment . The clarification further claims that the absence of environmental functionality runs counter to this value because technology "can, and should" consider environmental impact or offer eco-friendly functions.	The source gives no basis for treating a missing sustainability feature as evidence of environmental harm or disregard for environmental protection. The hallucinated premise is that the documentation's silence about sustainability indicates a value violation. ChatGPT then extends this unsupported premise into a broader concern: documentation lacks environmental feature → API does not support sustainability → disregard for environmental protection → Protecting the Environment concern.
The documentation describes ShogibanKit as a Swift framework for Shogi, or Japanese Chess. It says Shogi has complex rules and that the framework lets developers implement Shogi rules, validate moves, manage game state, and create a simple UI. It does not state that users need Japanese-language knowledge.	ChatGPT claims the API "targets Shogi players or individuals who understand the Shogi game rules and the Japanese language". The clarification adds that the API mandates "language skills (Japanese)" and "specific knowledge boundaries (Shogi)", which limit user engagement, challenge inclusivity in digital gaming, and disregard Respect for (diverse gaming) Traditions .	The unsupported premise is the Japanese-language requirement. The source says the framework supports Shogi; it does not say users must know Japanese. ChatGPT then extends this unsupported premise into a broader value claim: Shogi-specific library → assumed Japanese-language requirement → assumed exclusion of non-Japanese users → assumed disregard for diverse gaming traditions → Respect-for-Tradition concern.

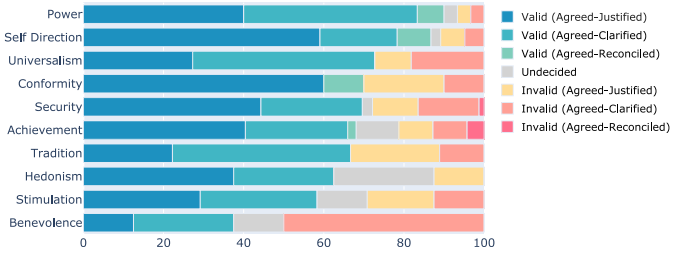


Fig. 7. Distribution (%) of ratings for the reassessed value concerns. "Invalid (Agreed-Justified)" denotes invalid concerns previously endorsed as Agreed-Justified. Precision is indicated by combined agreement.

of Hedonism improved compared to the findings from the interactive assessment (Fig. 5), while the socially focused concerns related to Benevolence and Tradition maintained the lowest precision, consistent with the interactive assessment results. Discussions among the raters uncovered instances where concerns about security were erroneously attributed to user health or family security. In some cases, the value concerns included extended reasoning weakly linked to security. In other instances, the justifications described legitimate software security concerns but were mistakenly linked to health or family security. The raters evaluated the former as Invalid and reclassified the latter as Valid instances of "Software Security" concerns.

4.2. *RQ2: What types of value concerns can an LLM (ChatGPT) infer?*

This section addresses research question RQ2 based on the value concerns inferred by ChatGPT. It is important to note that these concerns are inferred from the official API documentation, but they do not necessarily reflect the actual real-world behavior of the APIs.

4.2.1. *Distribution of value concerns*

Fig. 8 shows the distribution of 229 value concerns inferred by ChatGPT across 136 mobile APIs and confirmed by human evaluations as discussed in Section 4.1.2. The most common value concern corresponds to Self-Direction, which accounts for 31.45% of the concerns. Self-Direction is (potentially) threatened mainly through undermining user privacy. This is closely followed by Security concerns (24.02%), which occur primarily through conditions that may breach software security. Concerns related to Achievement (13.97%) are primarily associated with the conditions in the APIs that might negatively impact users' intelligence and success, while Power concerns (11.8%) are typically linked to affecting users' public image. Stimulation concerns (6.11%) are found mainly in the API features that undermine users' sense of adventure (Daring).

We further observed that concerns associated with Universalism (3.5%), Conformity (3.06%), Tradition (2.62%), Hedonism (2.18%), and Benevolence (1.31%) represent a smaller share. Universalism is potentially impacted mainly through conditions that can threaten equality, while Conformity is at risk due to APIs that could undermine users' self-discipline. Tradition concerns are linked to undermining respect for traditions and acceptance of one's portion in life. Hedonism concerns stem from the potential to undermine self-indulgent behaviors, and Benevolence concerns arise when certain conditions in the APIs could negatively influence users' helpfulness toward others. ChatGPT did not infer any concerns regarding social power (Power), devoutness (Tradition), spiritual life (Benevolence), or broadmindedness, unity with nature, inner harmony, and wisdom as expressions of Universalism.

4.2.2. *Value concerns across API themes*

Section 3.2 described the categorization of Android and iOS APIs under several themes according to their primary functions. Our analysis of value concerns across API themes (Fig. 9) revealed patterns, indicating where violations of values are more or less likely to occur.

Security-related concerns, primarily software security, dominate several themes, reflecting frequent threats to user safety. Self-Direction (privacy-related) concerns are also pervasive, especially in Connectivity & Networking, Data Management & Analysis, and Wearable & IoT, highlighting risks to user autonomy and data control. Power concerns appear prominently in Wearable & IoT and Testing & Debugging APIs. For example, Wearable & IoT documentation notes risks to users' public image, while Testing & Debugging APIs expose user behavior, as in: *The API documentation mentions that the DebugPanel feature provides access to a significant level of information, which might compromise user control over sensitive data.* Achievement concerns emerge in Connectivity & Networking and Development Essentials, signaling barriers to user success or effective goal pursuit. In contrast, values like Universalism

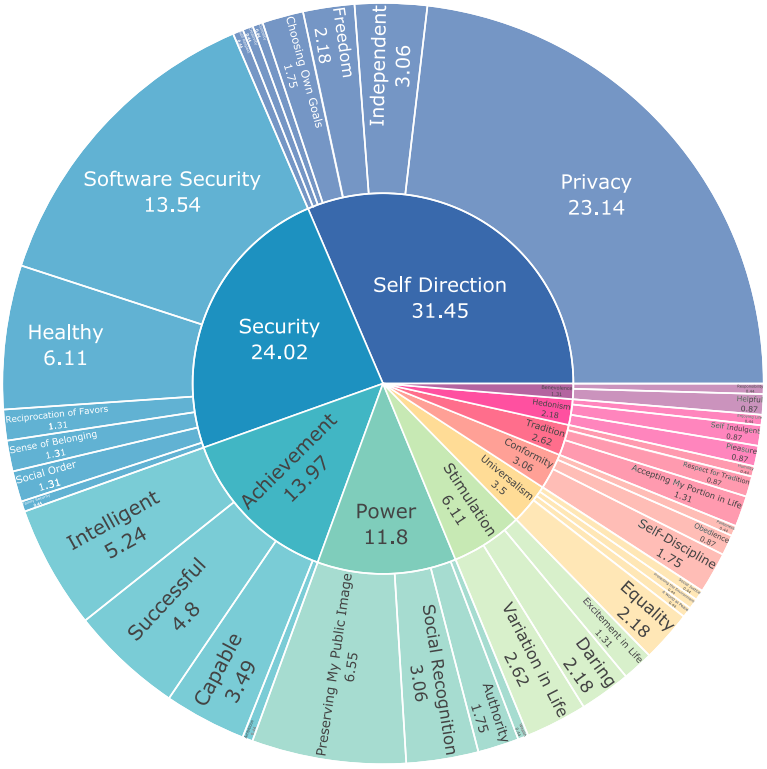


Fig. 8. Distribution (%) of validated value concerns.

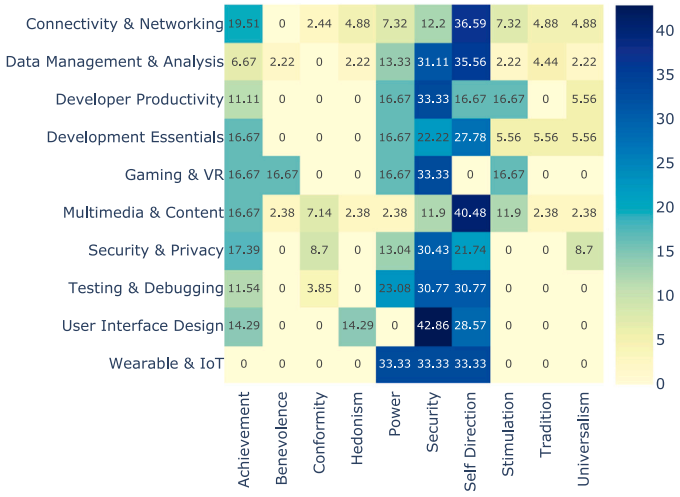


Fig. 9. Distribution (%) of value concerns across themes.

and Tradition face minimal threat, with no concerns in themes like User Interface Design and Wearable & IoT. Wearable & IoT also raises Power concerns linked to recognition and influence, while Testing & Debugging APIs show overlapping Power and Achievement concerns. Stimulation is marginally threatened in Gaming & VR, Developer Productivity, and Multimedia & Content. Benevolence is mostly affected by Gaming & VR, mainly relating to helpfulness. ☛ The analysis highlights recurring risks to software security, autonomy, success, recognition, and creativity across themes.

4.2.3. Value concern mechanisms and themes

We used ChatGPT with human oversight to categorize value concerns based on their underlying mechanisms (Section 3.7). This led to 113 mechanisms, which were then (manually) grouped into 10 higher-level themes. The top 5 most prevalent themes covering a wide range of value concerns are listed in Table 3. The most prevalent theme, "data collection, sharing, and reuse reduce control over personal information", shows that control over personal data is the dominant pattern through which API features affect human values. The mechanisms under this theme affect self-direction (privacy), security, power, conformity, and stimulation, showing how data handling can impact users' autonomy, protection, public image, and sense of control. Similarly, the theme concerning weak safeguards and unreliable execution shows how technical fragility may translate into security, financial, health, and operational harms. Another frequent theme relates to rigid APIs that constrain customization, user choice, and independent action. The theme comprises mechanisms that show how user values are impacted when they cannot adapt the mobile API features to their own goals. The theme related to limited feedback, intelligence, or feature support also shows how insufficient API support can undermine achievement, learning, creativity, and successful use.

Crosscutting concerns. Consistent with research in social sciences [84–86], the language model infers mechanisms of privacy violations that extend beyond self-direction to security and conformity. Based on ChatGPT, Misuse of Personal Data impacts both self-direction and security by affecting privacy and reciprocation of favors. Similarly, Unauthorized Location Tracking and Unauthorized Media Capture are linked to both self-direction and conformity through consent violations and lack of transparency. ChatGPT also explains that Health Data Misuse involves self-direction and security, as misuse of health-related data may affect both privacy and health. The language model uncovers that financial concerns extend beyond security. ChatGPT's inferences show that Financial Data Risk involves security and power, linking software security and health to wealth and authority, when unprotected transactions risk financial harm. ☛ ChatGPT's inferences show similar mechanisms affect multiple values.

Concerns about more-explored values. ChatGPT's inferences show that privacy-related concerns often affect self-direction, with Privacy Data

Table 3
The top 5 most prevalent themes of value concerns and their mechanisms. Each theme summarizes related mechanisms and reports the values impacted by those mechanisms. The mechanisms and their links to different values are provided as part of the replication package.

Theme (count)	Description	Mechanisms
Theme 1: Data collection, sharing, and reuse reduce control over personal information (82 concerns).	API mechanisms that collect, store, share, expose, synchronize, or reuse user data in ways that may affect privacy, consent, public image, security, or control over personal information. → The mechanisms corresponding to this theme impact self-direction, security, power, conformity, and stimulation.	Privacy Data Handling Concern; Misuse of Personal Data; Privacy Breach through Data Leakage; Invasion of Privacy; Data Access Without Consent; Health Data Misuse; Data Leakage; Data Use Without Consent; Privacy Intrusion via Data Collection; Privacy Breach through Data Synchronization; Lack of Transparency and Consent in Data Usage; Anxiety from Inadequate Data Control; Mishandling Data; Sensitive Data Storage; User Data Collection Overreach; Privacy Breach Inducing User Stress; Unintended Sensitive Data Harvesting; Inadequate Data Control.
Theme 2: Weak safeguards and unreliable execution create security, financial, and operational harms (30 concerns).	API mechanisms where weak protection, unstable execution, insecure communication, or poor risk management may affect software security, financial safety, reliability, health, or operations. → The mechanisms corresponding to this theme impact security, power, achievement, self-direction, conformity, and stimulation.	Software Security Concern; Financial Data Risk; App Stability Risk; Other Security Concern; Software Performance Reduction; Risk of Data Tampering; Lock Screen Security Oversight; Security Affecting Operations; Cryptographic Vulnerability Impact on Mental Health; Insecure Network Communication; Performance Degradation by Intensive Operations; Unclear Protection Measures; Risk Management Failure; Network Dependency Risk.
Theme 3: Rigid APIs constrain customization, user choice, and independent action (25 concerns).	API mechanisms that restrict configuration, enforce defaults or dependencies, limit playback or interaction control, reduce user choice, or prevent users from adapting systems to their own goals. → The mechanisms corresponding to this theme impact self-direction, achievement, benevolence, conformity, security, and stimulation.	Constraint on User Customizability; Restriction on User Autonomy; Autonomy Constraint in Digital Interfaces; Dependency Creation; Compulsory Update Enforcement; User Autonomy Limitation; Restrictive Playback Control; Excessive Configuration Complexity; Privacy Invasion and Compromised Autonomy; Rigid API Design Limiting User Autonomy; Write Access Limitations; Enforced Dependency on External Services; Remote Configuration Alteration; Unilateral Decision-Making in User Interactions; Restrictive User Interaction.
Theme 4: Limited feedback, intelligence, or feature support hinders success and growth (24 concerns).	API mechanisms that provide insufficient progress indicators, weak success metrics, limited intelligent functionality, poor feature support, or reduced opportunities for creativity, learning, and achievement. → The mechanisms corresponding to this theme impact achievement, self-direction, and stimulation.	Lack of Incremental Success Indicators; Insufficient Support for Success Metrics; Innovative Limitation; Insufficient Intelligent Functionality; Deprecated Functionality Impacting Success; Intellectual Growth Limitation; Limitation of User Creativity; Failure in Intelligent Data Interpretation; Inadequate Support for Development; Lack of Engaging Content; Restricted Creative Potential; Incomplete Feature Set; Obstacles to User Success; Inadequate Intelligent User Interaction.
Theme 5: Location, media, and biometric access expose users to surveillance and identity risks (17 concerns).	API mechanisms that access location, camera, biometric, screenshot, communication, or media data in ways that may expose users to surveillance, identity risks, social exposure, or unequal protection. → The mechanisms corresponding to this theme impact self-direction, security, conformity, power, stimulation, and universalism.	Unauthorized Location Tracking; Biometric Data Misuse; Unauthorized Media Capture; Unauthorized Biometric Data Use; Unauthorized Camera Access; Misuse of Personal Communication; Location Monitoring Concern; Geolocation Information Leakage; Unauthorized Screenshot Detection; Biometric and Security Inequality; Social Media Exposure Risk.

Handling Concern and Misuse of Personal Data. Other concerns include Restriction on User Autonomy, Constraint on User Customizability, and Unauthorized Location Tracking: *The API documentation describes the use of location tracking features that automatically collect and store user coordinates without explicit consent.* ☛ These concerns reveal how API features can limit autonomy, stressing the need for transparent, user-centered design. Software Security Concern is the most frequent security-related mechanism (3.49%), reflecting general threats to software security. Other concerns involving security include Invasion of Privacy, App Stability Risk, Financial Data Risk, and Ignoring User Benefit for Corporate Gain: *The API provides a feature for cross-promoting apps but lacks transparency regarding how user data is used in the promotion process.*

Concerns about less-explored values. Some values are less explored than others; their violations may be less visible due to a limited understanding of how they manifest in software. ChatGPT's inferences reveal multiple mechanisms of concern related to universalism, such as Data Usage Inequality: *The API enables data detection from natural language in a string ... users who do not articulate dates in English may receive lower service quality.* The language model also links the mechanism Environmental Impact Neglect to undermining environmental protection (universalism): *The API details high-energy consumption due to inefficient real-time content processing, leading to negative environmental consequences.* Value concerns under Experience Inequality are explained through arguments such as *The API relies on color visualizations for feedback, disadvantaging colorblind or visually impaired users.* ☛ These cases illustrate how APIs can impact universalism, underscoring the need for inclusive considerations. ChatGPT's reasoning also inferred mechanisms of concern affecting benevolence. These include Psychological Harm via Augmented Reality Misuse: *...allowing real-time facial modification without user awareness, potentially causing psychological distress.* This also shows that harmful uses of API features can affect benevolence by undermining helpful and respectful interactions. ☛ These findings illustrate the need for user-centric design to ensure responsible digital interactions.

5. Discussion

Beyond labeling. Classification approaches [9–13] predict value violations by assigning labels but lack reasoning, limiting transparency and shifting the burden of interpretation to human analysts. In contrast, our human-in-the-loop method uses LLMs like ChatGPT to infer value concerns – conditions that may lead to violations – and explain their mechanisms. This makes the rationale explicit, enabling more transparent and verifiable analysis. Our results show that this approach is both feasible and effective: over 82% of ChatGPT-inferred concerns were endorsed after clarification and discussion, revealing the model's capacity to generate traceable, meaningful explanations. This aligns with Schwartz's view of values as a continuum [27], and reframing LLMs as explanatory tools – rather than opaque classifiers – allows for harnessing their generative strengths to support transparent reasoning about values, exploring the unseen [16].

Toward explanation-centered evaluation. Our findings suggest that the inference-evaluation cycle plays a critical role in enhancing explanation quality by filtering out value concerns with weak justifications. This reinforces critiques in the literature [22–24] that conventional

metrics like F1-score fall short in capturing the complexity of reasoning. To move forward, evaluation must consider criteria that measure the quality of reasoning. The fact that some endorsed concerns were later rated invalid or undecided highlights the limits of surface-level validation and the necessity of depth, traceability, and human deliberation. Our human-in-the-loop method evaluates value concerns using three criteria: factuality (evidence-based traceability), plausibility, and specificity.

The power and peril of reasoning. Using LLMs for explanatory inference reveals both potential and risk. Our study identified three intertwined challenges: hallucinations [64], extended reasoning, and persuasive effects on evaluators. Hallucinations arose when ChatGPT produced confident yet ungrounded justifications, inserting speculative or false details that misled evaluators with plausible but unverifiable claims. Extended reasoning showed the model's tendency to infer violations through tenuous causal chains. While such reasoning can mirror how values act as implicit motivations, it risks conflating real and hypothetical threats – e.g., treating camera access as a privacy concern without actual data leakage. More concerning was ChatGPT's rhetorical fluency: when initial justifications failed, clarifications often reframed rather than corrected issues, subtly steering judgment through coherence over factuality. This led to endorsements of concerns later rated invalid, revealing how persuasive phrasing exploits cognitive bias. Even structured debates did not consistently outperform simple clarifications, showing that more dialogue does not guarantee better accuracy. These observations blur the boundary between explanation and persuasion and underscore the need for safeguards. Our multi-stage evaluation mitigated these effects, but developing stronger evaluation standards remains essential.

Patterns, blind spots, and subjectivity. Our large-scale analysis of mobile APIs surfaces patterns and blind spots in how software risks human values. Validated concerns are dominated by Self-Direction and Security, with recurring threats to privacy and software safety. Achievement and Power also emerge, tied to autonomy, user success, and image. In contrast, Universalism, Tradition, Hedonism, and Benevolence are underrepresented, often surfacing through more diffuse sociotechnical pathways – such as accessibility, environmental impact, or psychological strain – that tend to be deprioritized in technical decision-making. Crosscutting concerns are frequent: privacy breaches affect not only Self-Direction, but also Security, Power, and Conformity, illustrating that value threats rarely exist in isolation. Despite rigorous staging – ground truth creation, multi-rater assessments, and mediation –ambiguities persist, reflecting the subjective nature of values.

6. Cross-model analysis of robustness with role-conditioning

Our study relies on the value concerns inferred by gpt-4 and assessed by human evaluators (authors). This raises two concerns about the robustness of the analysis: (i) whether the assessment of value concerns using different LLMs produces stable ratings, and (ii) whether the concerns receive comparable ratings when assessed explicitly from a developer perspective. To address these concerns, we conducted a cross-model analysis using six popular models with reasoning capabilities, role-conditioned [87,88] as mobile app developers, to check if the ratings remain stable across different LLMs.

6.1. Settings

The models used in the analysis include the proprietary models o3, gpt-4o, and gpt-5 as well as the open-weight (local) models deepseek-r1:8b, gemma4:e4b, and gpt-oss:20b. The choice of these models was motivated by their reasoning capabilities, making them better suited to applying our evaluation protocol (Table 4) to assess the rationales behind value concerns.

Each model was assigned the role of an experienced mobile app developer with practical experience in designing, building, testing, debugging, maintaining, and releasing Android and iOS applications. The role also included familiarity with Schwartz's theory of human values [89]. Our use of role-conditioning is grounded in prior work on LLM-based evaluation and role-conditioned simulation [87,88,90,91]. LLMs have been used as scalable judges of open-ended outputs, although their judgments can reflect model-specific biases and therefore require careful interpretation [90]. Studies of role-conditioned and simulated agents show that LLM responses can reflect the assigned roles [87,88], but caution has been given against treating simulated perspectives as equivalent to human participation [91].

Table 4
The evaluation protocol (criteria and scales) used for assessing value concerns in the cross-model robustness check.

Factuality (Evidence-based)	
Verified	Directly refers to the API; no interpretation of evidence is needed.
Supported	Is traceable to the API, with minor interpretation of evidence.
Partial	Is traceable to the API, with some interpretation of evidence.
Doubtful	Is traceable to the API, with significant interpretation of evidence.
Unfounded	Is not traceable to the API.
Plausibility (Likelihood)	
Certain	Is plausible, without reasonable doubt.
Likely	Is plausible, with minor doubt.
Possible	Is plausible, with some doubt.
Unlikely	Is plausible, with significant doubt.
Implausible	Is implausible.
Specificity (Level of Detail)	
Specific	Is well-defined with concrete details; fully actionable.
Detailed	Is described with sufficient detail; actionable.
General	Is described with basic details; partially actionable.
Broad	Is ambiguous, lacking specifics; hardly actionable.
Vague	Is highly ambiguous, with minimal details; impractical.

The role-conditioned models were prompted to rate the factuality, plausibility, and specificity of all 229 value concerns – previously validated – based on the evaluation protocol of Table 4. The gpt-4-inferred justifications and clarifications of the value concerns and the API documentation content were provided in the prompts.

> You are an experienced mobile app developer acting as an independent rater: `role description`. Please follow the instructions listed below.
>
> 1. Consider `value item` under `value` from Schwartz's model.
> 2. Consider the `API documentation`.
> 3. Consider the `value concern`.
> 4. Review the concern based on the `API documentation` and the logic provided for the concern.
> 5. Evaluate the concern's `factuality`, `plausibility`, and `specificity` using the `rating scale` (Table 4).
> 6. Select the most suitable rating label per criterion. (Table A.5)

The proprietary models were executed using OpenAI's API (default settings). The open-weight models were executed locally on an M2 chipset (macOS) with 32 GB of RAM. It took around 17 h to complete. Among the proprietary models, gpt-4o was the fastest at around 11 min, followed by o3 (61 min) and gpt-5 (127 min) — Fig. 10. Among the locally executed open-weight models, gpt-oss:20b was the fastest at around 142 min, followed by deepseek-r1:8b (250 min); gemma4:e4b was the slowest, taking over 7 h to complete all prompts.

6.2. Analysis of variations

Table 5 lists the Average Deviation Index (ADI) [92–94] between each pair of models (raters) as well as the median-centered ADI among all. ADI is adopted as the measure of inter-rater agreement since factuality, plausibility, and specificity are ordinal ratings, in which the size of disagreement matters. A one-level difference, such as *Supported* versus *Partial*, is less substantial than a wider difference, such as *Verified*

Table 5
Average deviation index (ADI) between role-conditioned LLMs, rating as developers. ADI values indicate how many rating levels apart the raters were, on average, on the five-point scale (Table 4); lower values indicate closer agreement (less variation).

Rater I	Rater II	Factuality	Plausibility	Specificity	Mean
o3	gpt-5	0.46	0.37	0.41	0.41
o3	gpt-4o	0.61	0.59	0.55	0.58
o3	gpt-oss:20b	0.78	0.55	0.89	0.74
o3	gemma4:e4b	1.44	0.90	0.99	1.11
o3	deepseek-r1:8b	0.97	0.94	1.07	1.00
gpt-5	gpt-4o	0.65	0.62	0.55	0.60
gpt-5	gpt-oss:20b	0.79	0.54	0.86	0.73
gpt-5	gemma4:e4b	1.41	0.80	0.91	1.04
gpt-5	deepseek-r1:8b	0.98	0.93	1.04	0.98
gpt-4o	gpt-oss:20b	0.87	0.71	1.10	0.90
gpt-4o	gemma4:e4b	1.38	0.81	1.22	1.14
gpt-4o	deepseek-r1:8b	0.96	0.96	1.31	1.07
gpt-oss:20b	gemma4:e4b	1.60	0.96	0.48	1.01
gpt-oss:20b	deepseek-r1:8b	1.24	1.03	0.38	0.88
gemma4:e4b	deepseek-r1:8b	1.32	0.91	0.32	0.85
Median-centered ADI among all models (raters): o3, gpt-4o, gpt-5, deepseek-r1:8b, gemma4:e4b, and gpt-oss:20b		0.63	0.47	0.54	0.55

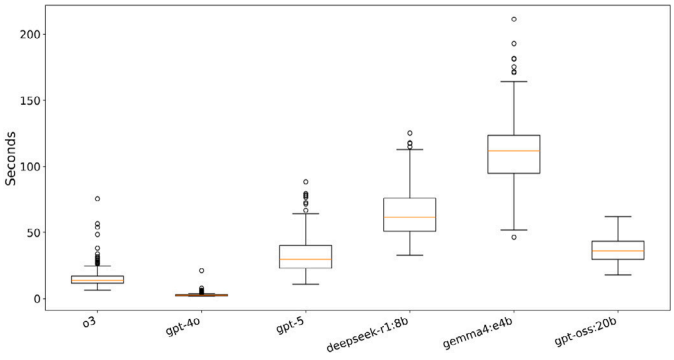


Fig. 10. Prompt execution time (seconds) for role-conditioned LLMs.

versus *Unfounded*. This follows the logic of scaled disagreement for ordinal ratings [95]. ADI directly expresses the average number of rating levels separating the raters, making it interpretable as a rating distance.

Overall variations. Table 5 shows the average median-centered ADI value of 0.55 across all six models, indicating a variation of less than one rating level on average, across all rating criteria. This shows moderate cross-model stability, indicating that the evaluation protocol (Table 4) can be applied across multiple role-conditioned model architectures without producing sharply divergent results. This is also supported by the pairwise comparisons; the mean ADI ranges from 0.41 to 1.14, meaning that the ratings differ by around one rating level or less on average.

Criterion-level variations. Based on the median-centered ADIs, the models exhibit the greatest variation in their ratings of factuality (0.63), followed by specificity (0.54) and plausibility (0.47). The pairwise ADI values for factuality range from 0.46 to 1.60, with the largest distance observed for gpt-oss vs. gemma4. Several pairs involving gemma4 also show a higher level of disagreement over factuality. This pattern indicates that factuality is especially sensitive to how strictly a model (rater) interprets the evidential link between a concern and the API documentation. Some models appear to require more direct evidence before assigning higher factuality levels, whereas others appear more willing to accept interpreted evidence. The LLM ratings of specificity ($0.32 \leq ADI \leq 1.31$) and plausibility ($0.37 \leq ADI \leq 1.03$) are more stable than factuality but still vary across models (Table 5).

Model sensitivity. Since the analysis uses multiple LLMs (raters), it is important to determine whether the robustness claim reflects a stable pattern across models or is disproportionately shaped by one rater. To examine this, we recomputed the mean pairwise ADI after excluding one model at a time and averaging the remaining pairwise ADI values. As shown in Table 6, the mean ADI is 0.87 when all six models are included and remains between 0.79 and 0.93 after each model is excluded. This narrow range indicates that the ratings show bounded cross-model variation (disagreement) rather than being driven by one outlying rater. The contribution of gemma4 to the variations in factuality, however, is more significant than that of other models; its exclusion reduces the mean ADI of factuality from 1.03 to 0.83.

Rating tendencies. To identify the rating tendencies across all models, the rating labels were mapped to a five-point numerical scale, where stronger ratings receive higher scores: *Verified*, *Certain*, *Specific* = 5 and *Unfounded*, *Implausible*, *Vague* = 1. We then calculated, for each model, the average score across all concerns for each criterion. The mean of these average scores was computed for each model to specify its overall rating tendency: o3 (3.02), gpt-5 (3.06), gpt-4o (3.02), deepseek-r1 (3.84), gemma4 (3.83), and gpt-oss (3.12). The results show that the overall rating tendencies remain close to the middle of the five-point scale, supporting the interpretation that the observed variation is bounded: the models do not produce sharply divergent rating patterns, but they do apply the scale with different thresholds.

> Our analysis shows comparable ratings across the role-conditioned models; the results exhibit moderate cross-model stability with bounded variation. The role-conditioned LLMs, however, cannot substitute for developer participation.

7. Limitations & threats to validity

Indirect links or hallucinations? ChatGPT's explanations for inferred value concerns were affected by extended reasoning that, while weakly supported by API documentation, did not clearly constitute hallucination [64]. These explanations, essential for interpreting implied violations, were sometimes difficult to follow and occasionally appeared confabulatory. Conversely, some hallucinations resembled extended

Table 6
Sensitivity analysis after excluding one model at a time; mean ADIs are reported across all remaining pairs. Lower values indicate closer agreement.

Excluded model	Pairs remaining	Factuality	Plausibility	Specificity	Mean
None (all six)	15	1.03	0.77	0.81	0.87
o3	10	1.12	0.83	0.82	0.92
gpt-5	10	1.12	0.84	0.83	0.93
gpt-4o	10	1.10	0.79	0.74	0.88
gpt-oss:20b	10	1.02	0.78	0.84	0.88
gemma4:e4b	10	0.83	0.72	0.82	0.79
deepseek-r1:8b	10	1.00	0.68	0.80	0.83

reasoning, misleading evaluators. This may impact the construct validity of the study, as such inaccuracies may limit the reliability of inferred concerns. To address this, we used a hybrid evaluation process with trained evaluators who clarified findings through dialogue, alongside ChatGPT's reassessment using ground truth examples and value definitions. A final rating phase helped mitigate residual hallucinations. Future work may develop more advanced methods for detecting unfounded justifications [64].

Limited ground truth. Ground truth was established through human evaluation of value concerns, excluding ChatGPT's persuasive effects. To avoid the burden of assessing 10 values with 59 components (totaling 4956 reasoning chains), we used a candidate list of potential violations from ChatGPT findings (excluding explanations). This focused rater attention while allowing additional concerns, but may have led to overlooked cases, threatening construct validity. The ground truth, 21 concerns from 42 APIs, was based on a small subset, limiting external validity. Domain experts and practitioners can refine our findings to build a more comprehensive ground truth.

LLMs' stochastic behavior: threat or opportunity? ChatGPT's value inferences can vary across identical prompts due to its stochastic nature and the subjective, implicit character of human values [35,36,96]. While this limits generalizability and external validity, it also enables the discovery of diverse value perspectives, as randomness in generation can surface alternative interpretations or overlooked connections, revealing the unseen [16].

Single-model focus. This study centers on ChatGPT for inferring and explaining value concerns. While evaluating other LLMs could yield further insights, the robustness of our method relies on in-depth human evaluation – dialogue, clarification, and debates (Section 3.4). Extending this process to multiple models would pose major practical and methodological challenges, particularly in maintaining consistency and depth. Though this may limit external validity, our goal is to validate a systematic human-in-the-loop method, regardless of the language model used.

Domain focus: mobile API documentation. This study applies our human-in-the-loop method to mobile API documentation (Android and iOS). While this domain focus may limit the external validity and generalizability of findings to other software artifacts, it enables a rigorous test of our methodology in a context where user values are most directly impacted. The method itself is domain-agnostic and can be adapted to other contexts, and we encourage future studies to extend this approach to a broader range of software domains.

8. Conclusions and future work

This study presents a human-in-the-loop method for automated inference of conditions that may lead to violations of human values in software, referred to as *value concerns*. By combining LLM reasoning with human evaluation, our method provides traceable explanations of value concerns that go beyond a simple classification of these concerns. Applied to the documentation of 1200 Android and iOS APIs, the method demonstrates that ChatGPT can generate meaningful and verifiable accounts of how values may be at risk. Cycles of inference-assessment, utilizing human oversight, mitigate inherent challenges, such as LLM hallucination, extended reasoning, and persuasive effects of LLMs.

> Our findings suggest that LLMs are most effective when positioned as reasoning partners, guided by human oversight, to uncover the mechanisms behind value violations rather than functioning as classifiers that merely label their presence. Our analysis of mobile APIs reveals that Self-Direction and Security are the dominant value concerns. Crosscutting concerns such as privacy breaches span multiple values, and despite rigorous validation, ambiguities persist, highlighting the subjective interpretation of values.

Future work will extend and validate our method across diverse software domains and a broader range of software artifacts, such as bug reports and commit messages. We also plan to study practitioners' and users' perspectives on the mechanisms of value violations inferred by ChatGPT to strengthen their practical relevance, enabling the design of countermeasures to mitigate value violations in software. Insights from diverse stakeholders can be leveraged to enhance the quality of LLM reasoning and reduce interpretation bias.

CRediT authorship contribution statement

Davoud Mougouei: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Formal analysis, Data curation, Conceptualization. **Son Pham:** Writing – review & editing, Validation, Formal analysis, Data curation. **Ahmad Azarnik:** Writing – review & editing, Validation, Formal analysis, Data curation. **Mahdi Fahmideh:** Writing – review & editing, Validation, Formal analysis, Data curation. **Arif Nurwidyantoro:** Writing – review & editing, Validation, Formal analysis, Data curation. **Elahe Mougouei:** Writing – review & editing, Validation, Formal analysis, Data curation. **Amanul Haque:** Writing – review & editing, Validation, Formal analysis, Data curation. **Hoa Khanh Dam:** Writing – review & editing, Validation, Formal analysis, Data curation. **Saima Rafi:** Writing – review & editing, Validation, Formal analysis, Data curation. **Javed Ali Khan:** Writing – review & editing, Validation, Formal analysis, Data curation.

Funding sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Prompts used in the study

The following tables list the prompts used in the study.

Table A.1
The Inference Prompt used for inferring value concerns from API documentation (Section 3.3).

Step	Description
(1)	Consider the study context: the task is to analyze the documentation of a mobile API and infer, meaning detect and explain, violations of human values based solely on evidence in the documentation and from the user's perspective.
(2)	Consider the definition of a mobile app user as the end user of the mobile application, not the developer. The analysis must focus exclusively on how the API affects app users.
(3)	Consider Schwartz's taxonomy of human values, including the following values, value items, and short forms: • Self-Direction (v1): Independent (v1.1), Freedom (v1.2), Curiosity (v1.3), Creativity (v1.4), Choosing Own Goals (v1.5), Privacy (v1.6), Self-Respect (v1.7). ...
(4)	Consider the API documentation: [documentation].
(5)	Infer and explain up to three of the most relevant value violations, choosing those with the strongest, most directly evidenced justifications that meet the following criteria: • factuality: the justification is solely based on explicit evidence in the API documentation. • plausibility: the justification reports a significant and likely violation. • specificity: the justification reports sufficient details specific to the context.
(6)	The analysis must be focused on how the API violates the values of mobile app users, not developers.
(7)	Consider the example output: [formatted_examples].
(8)	Output the results as a CSV using semicolons as delimiters, with the format: Value; Value Item; Justification.

Table A.2
The prompt used for clarifying inferred value concerns from API documentation (Section 3.4).

Step	Description
(1)–(3)	Same as Steps 1–3 of the Inference Prompt.
(4)	Consider the API documentation: [documentation].
(5)	Consider the previously inferred value violation: [value; value item; justification]. This violation was inferred using the following criteria: • factuality: the justification is solely based on explicit evidence found in the API documentation. • plausibility: the justification reports a significant and likely violation. • specificity: the justification reports sufficient details specific to the context.
(6)	Clarify the justification because the user is not convinced by the previously inferred violation.
(7)	Only return the clarification with no additional content.

Table A.3
The prompt used for debating ChatGPT about the inferred value concerns (Section 3.4).

Step	Description
(1)–(3)	Same as Steps 1–3 of the Inference Prompt.
(4)	Consider the API documentation: [documentation].
(5)	Consider the previously inferred value violation: [value; value item; justification]. This violation was inferred using the following criteria: • factuality: the justification is solely based on explicit evidence in the API documentation. • plausibility: the justification reports a significant and likely violation. • specificity: the justification reports sufficient details specific to the context.
(6)	Consider the clarification previously provided for the inferred violation: [clarification].
(7)	Consider that the user is still not convinced and has provided the following reason for disagreement: [reason].
(8)	Return the response to the disagreement with no additional content.

Table A.4
The prompt used for validating the inferred value concerns (Section 3.5.2).

Step	Description
(1)	Same as Step 1 of the Inference Prompt.
(2)	Consider the following value concern about value [value]: [concern].
(3)	Consider Schwartz's definition of value [value]: [value_definition].
(4)	Consider the API documentation: [documentation].
(5)	Evaluate the validity of the inferred value concern. A valid concern must meet all of the following criteria: • factuality: the concern is based solely on explicit evidence in the API documentation. • plausibility: the concern reports a significant and likely violation. • specificity: the concern reports sufficient details specific to the context.
(6)	Focus the analysis on how the API violates the values of mobile app users, not developers.
(7)	Consider the examples of valid concerns: [ground_truth_examples].
(8)	Return valid or invalid with no additional content.

Table A.5
The prompt used for the cross-model robustness analysis with role-conditioning (Section 6).

Step	Description
(1)	You are an experienced mobile app developer acting as an independent expert rater. You have practical experience designing, building, testing, debugging, maintaining, and releasing mobile applications for Android and iOS. You understand common app development issues, including usability problems, performance issues, crashes, data handling failures, permission and privacy risks, notification issues, accessibility barriers, security weaknesses, … You are also familiar with Schwartz's theory of human values.
(2)	Consider the value item [value_item] under value [value] from Schwartz's model of human values.
(3)	Consider the mobile app user and its definition: [app_user].
(4)	Consider the following API documentation: [documentation].
(5)	Consider the following concern about a violation of [value] and its corresponding value item [value_item]: [concern].
(6)	Review the concern's logic based on the API documentation.
(7)	Evaluate the concern using the criterion [criterion] and its rating scale: [rating_scale].
(8)	Return exactly one rating label from the rating scale for the criterion.

Data availability

https://figshare.com/s/db6a26280df6bb64649a.

References

[1] D. Mougouei, H. Perera, W. Hussain, R. Shams, J. Whittle, Operationalizing human values in software: a research roadmap, in: Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ACM, 2018, pp. 780–784.

[2] H. Perera, W. Hussain, D. Mougouei, R.A. Shams, A. Nurwidyantoro, J. Whittle, Towards integrating human values into software: Mapping principles and rights of GDPR to values, in: 2019 IEEE 27th International Requirements Engineering Conference, RE, IEEE, 2019, pp. 404–409.

[3] H. Perera, W. Hussain, J. Whittle, A. Nurwidyantoro, D. Mougouei, R.A. Shams, G. Oliver, A study on the prevalence of human values in software engineering publications, 2015–2018, in: 2020 IEEE/ACM 42nd International Conference on Software Engineering, ICSE, IEEE, 2020, pp. 409–420.

[4] BBC News, 'Zuckerberg, what were you thinking?' Tech CEOs grilled in DC, 2024, URL: https://www.bbc.com/news/live/world-us-canada-68110001/page/3.

[5] M. Bano, C. Arora, D. Zowghi, A. Ferrari, The rise and fall of covid-19 contact-tracing apps: when nfrs collide with pandemic, in: 2021 IEEE 29th International Requirements Engineering Conference, RE, IEEE, 2021, pp. 106–116.

[6] A. Nurwidyantoro, M. Shahin, M.R. Chaudron, W. Hussain, R. Shams, H. Perera, G. Oliver, J. Whittle, Human values in software development artefacts: A case study on issue discussions in three Android applications, Inf. Softw. Technol. 141 (2022) 106731.

[7] A. Nurwidyantoro, M. Shahin, M. Chaudron, W. Hussain, H. Perera, R.A. Shams, J. Whittle, Integrating human values in software development using a human values dashboard, Empir. Softw. Eng. 28 (3) (2023) 67.

[8] R.A. Shams, W. Hussain, G. Oliver, A. Nurwidyantoro, H. Perera, J. Whittle, Society-oriented applications development: Investigating users' values from bangladeshi agriculture mobile applications, in: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society, 2020, pp. 53–62.

[9] J. Jamieson, N. Yamashita, E. Foong, Predicting open source contributor turnover from value-related discussions: An analysis of GitHub issues, in: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, in: ICSE 2024, Association for Computing Machinery, New York, NY, USA, 2024.

[10] H.O. Obie, W. Hussain, X. Xia, J. Grundy, L. Li, B. Turhan, J. Whittle, M. Shahin, A first look at human values-violation in app reviews, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society, ICSE-SEIS, IEEE, 2021, pp. 29–38.

[11] H.O. Obie, I. Ilekura, H. Du, M. Shahin, J. Grundy, L. Li, J. Whittle, B. Turhan, On the violation of honesty in mobile apps: Automated detection and categories, in: Proceedings of the 19th International Conference on Mining Software Repositories, MSR '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 321–332.

[12] M. Shahin, M. Zahedi, H. Khalajzadeh, A.R. Nasab, A study of gender discussions in mobile apps, in: 2023 IEEE/ACM 20th International Conference on Mining Software Repositories, MSR, IEEE, 2023, pp. 598–610.

[13] J. Jamieson, E. Foong, N. Yamashita, Maintaining values: Navigating diverse perspectives in value-charged discussions in open source development, Proc. ACM Hum.-Comput. Interact. 6 (CSCW2) (2022) 1–28.

[14] D. Hendrycks, C. Burns, S. Basart, A. Critch, J. Li, D. Song, J. Steinhardt, Aligning AI with shared human values, in: Proceedings of the International Conference on Learning Representations, ICLR, 2021.

[15] M. Chen, J. Tworek, H. Jun, Q. Yuan, H.P.d.O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al., Evaluating large language models trained on code, 2021, arXiv preprint arXiv:2107.03374.

[16] J. Wei, A.-L. Courbis, T. Lambolais, B. Xu, P.L. Bernard, G. Dray, W. Maalej, Getting inspiration for feature elicitation: App store-vs. LLM-based approach, in: Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering, 2024, pp. 857–869.

[17] K. Ronanki, C. Berger, J. Horkoff, Investigating ChatGPT's potential to assist in requirements elicitation processes, in: 2023 49th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, IEEE, 2023, pp. 354–361.

[18] A. Marczak-Czajka, J. Cleland-Huang, Using ChatGPT to generate human-value user stories as inspirational triggers, in: 2023 IEEE 31st International Requirements Engineering Conference Workshops, REW, IEEE, 2023, pp. 52–61.

[19] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, L. He, A survey of human-in-the-loop for machine learning, Future Gener. Comput. Syst. 135 (2022) 364–381.

[20] E. Sblendorio, V. Dentamaro, A. Lo Cascio, F. Germini, M. Piredda, G. Cicolini, Integrating human expertise & automated methods for a dynamic and multi-parametric evaluation of large language models' feasibility in clinical decision-making, Int. J. Med. Informatics 188 (2024) 105501.

[21] C. Chandler, P.W. Foltz, B. Elvevåg, Improving the applicability of AI for psychiatric applications through human-in-the-loop methodologies, Schizophr. Bull. 48 (5) (2022) 949–957.

[22] A. Rosenfeld, Better metrics for evaluating explainable artificial intelligence, in: Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems, 2021, pp. 45–50.

[23] Z.C. Lipton, The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery, Queue 16 (3) (2018) 31–57.

[24] H. Schuff, H. Adel, N.T. Vu, F1 is not enough! models and evaluation towards user-centered explainable question answering, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 7076–7095.

[25] Common Cause Foundation, The common cause handbook, 2012, URL: https://commoncausefoundation.org/_resources/the-common-cause-handbook/.

[26] S.H. Schwartz, Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries, in: Advances in Experimental Social Psychology, Vol. 25, Elsevier, 1992, pp. 1–65.

[27] S.H. Schwartz, An overview of the Schwartz theory of basic values, Online Readings Psychol. Cult. 2 (1) (2012) 11.

[28] A.-S. Cheng, K.R. Fleischmann, Developing a meta-inventory of human values, Proc. Am. Soc. Inf. Sci. Technol. 47 (1) (2010) 1–10.

[29] C.L. Jurkiewicz, R.A. Giacalone, A values framework for measuring the impact of workplace spirituality on organizational performance, J. Bus. Ethics 49 (2004) 129–142.

[30] M. Rokeach, The Nature of Human Values, Free Press, 1973.

[31] R.M. Norman, R.M. Sorrentino, B. Gawronski, A.C. Szeto, Y. Ye, D. Windell, Attitudes and physical distance to an individual with schizophrenia: the moderating effect of self-transcendent values, Soc. Psychiatry Psychiatr. Epidemiol. 45 (2010) 751–758.

[32] J. Jamieson, N. Yamashita, E. Foong, Predicting open source contributor turnover from value-related discussions: An analysis of GitHub issues, in: Proceedings of the 46th IEEE/ACM International Conference on Software Engineering, 2024, pp. 1–13.

[33] W. Hussain, M. Shahin, R. Hoda, J. Whittle, H. Perera, A. Nurwidyantoro, R.A. Shams, G. Oliver, How can human values be addressed in agile methods? A case study on SAFe, IEEE Trans. Softw. Eng. 48 (12) (2022) 5158–5175.

[34] D. Mougouei, A. Azarnik, M. Fahmideh, E. Mougouei, H.K. Dam, A.A. Khan, S. Rafi, J.A. Khan, A. Ahmad, A first look at AI trends in value-aligned software engineering publications: Human-LLM insights, in: 2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS), 2025, pp. 82–93.

[35] P.H. Hanel, G.R. Maio, A.K. Soares, K.C. Vione, G.L. de Holanda Coelho, V.V. Gouveia, A.C. Patil, S.V. Kamble, A.S. Manstead, Cross-cultural differences and similarities in human value instantiation, Front. Psychol. 9 (2018) 849.

[36] T.C. Oliveira, S. Holland, On the centrality of human value, J. Econ. Methodol. 19 (2) (2012) 121–141.

[37] D. Mougouei, Engineering human values in software through value programming, in: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, in: ICSEW'20, 4, Association for Computing Machinery, New York, NY, USA, 2020, pp. 133–136.

[38] M. Fazzini, H. Khalajzadeh, O. Haggag, Z. Li, H. Obie, C. Arora, W. Hussain, J. Grundy, Characterizing human aspects in reviews of COVID-19 apps, in: Proceedings of the 9th IEEE/ACM International Conference on Mobile Software Engineering and Systems, Association for Computing Machinery, New York, NY, USA, 2022, pp. 38–49.

[39] H. Khalajzadeh, M. Shahin, H.O. Obie, P. Agrawal, J. Grundy, Supporting developers in addressing human-centric issues in mobile apps, IEEE Trans. Softw. Eng. 49 (4) (2023) 2149–2168.

[40] E. Jo, W.-J. Kouaho, S.M. Schueller, D.A. Epstein, Exploring user perspectives of and ethical experiences with teletherapy apps: Qualitative analysis of user reviews, JMIR Ment. Health 10 (2023) e49684.

[41] M.M. Eler, L. Orlandin, A.D.A. Oliveira, Do android app users care about accessibility? An analysis of user reviews on the google play store, in: Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems, IHC '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–11.

[42] A. Alshayban, I. Ahmed, S. Malek, Accessibility issues in android apps: State of affairs, sentiments, and ways forward, in: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, ICSE '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 1323–1334.

[43] N. Tjikhoeri, L. Olson, E. Guzmán, The best ends by the best means: ethical concerns in app reviews, Empir. Softw. Eng. 29 (6) (2024) 138.

[44] E.A. AlOmar, W. Aljedaani, M. Tamjeed, M.W. Mkaouer, Y.N. El-Glaly, Finding the needle in a haystack: On the automatic identification of accessibility user reviews, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA, 2021, pp. 1–15.

[45] P. Nema, P. Anthonysamy, N. Taft, S.T. Peddinti, Analyzing user perspectives on mobile app privacy at scale, in: Proceedings of the 44th International Conference on Software Engineering, ACM, 2022, pp. 112–124.

[46] O. Haggag, J. Grundy, M. Abdelrazek, S. Haggag, Better addressing diverse accessibility issues in emerging apps: A case study using COVID-19 apps, in: Proceedings of the 9th IEEE/ACM International Conference on Mobile Software Engineering and Systems, Association for Computing Machinery, New York, NY, USA, 2022, pp. 50–61.

[47] F. Ebrahimi, A. Mahmoud, Unsupervised summarization of privacy concerns in mobile application reviews, in: Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, ACM, 2022, pp. 1–12.

[48] A. Rezaei Nasab, M. Dashti, M. Shahin, M. Zahedi, H. Khalajzadeh, C. Arora, P. Liang, Fairness concerns in app reviews: A study on ai-based mobile apps, ACM Trans. Softw. Eng. Methodol. 34 (2) (2025) 1–30.

[49] J. Zhang, J. Zhou, J. Hua, N. Niu, C. Liu, Mining user privacy concern topics from app reviews, J. Syst. Softw. 222 (2025) 112355.

[50] T. Kelly, A. Korkmaz, S. Mallet, C. Souders, S. Aliakbarpour, P. Rao, Longitudinal datasets of health app reviews for privacy and trust modeling, Data Brief 66 (2026) 112740.

[51] A.R. Besmer, J. Watson, M.S. Banks, Investigating user perceptions of mobile app privacy, Int. J. Inf. Secur. Priv. 14 (4) (2020) 74–91.

[52] M. Geng, S. Wang, D. Dong, H. Wang, G. Li, Z. Jin, X. Mao, X. Liao, Large language models are few-shot summarizers: Multi-intent comment generation via in-context learning, in: Proceedings of the 46th IEEE/ACM International Conference on Software Engineering, 2024, pp. 1–13.

[53] D. Nam, A. Macvean, V. Hellendoorn, B. Vasilescu, B. Myers, Using an llm to help with code understanding, in: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, 2024, pp. 1–13.

[54] J. Wang, Y. Huang, C. Chen, Z. Liu, S. Wang, Q. Wang, Software testing with large language models: Survey, landscape, and vision, IEEE Trans. Softw. Eng. 50 (4) (2024) 911–936.

[55] M. Jin, S. Shahriar, M. Tufano, X. Shi, S. Lu, N. Sundaresan, A. Svyatkovskiy, Inferfix: End-to-end program repair with llms, in: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2023, pp. 1646–1656.

[56] G. Colavito, F. Lanubile, N. Novielli, L. Quaranta, Leveraging GPT-like LLMs to automate issue labeling, in: 2024 IEEE/ACM 21st International Conference on Mining Software Repositories, MSR, IEEE, 2024, pp. 469–480.

[57] Y. Chen, M. Cui, D. Wang, Y. Cao, P. Yang, B. Jiang, Z. Lu, B. Liu, A survey of large language models for cyber threat detection, Comput. Secur. (2024) 104016.

[58] R. Kande, H. Pearce, B. Tan, B. Dolan-Gavitt, S. Thakur, R. Karri, J. Rajendran, (Security) assertions by large language models, IEEE Trans. Inf. Forensics Secur. (2024).

[59] M.D. Purba, A. Ghosh, B.J. Radford, B. Chu, Software vulnerability detection using large language models, in: 2023 IEEE 34th International Symposium on Software Reliability Engineering Workshops, ISSREW, IEEE, 2023, pp. 112–119.

[60] D. Rodriguez, I. Yang, J.M. Del Alamo, N. Sadeh, Large language models: a new approach for privacy policy analysis at scale, Computing (2024) 1–25.

[61] F.A. Shah, A. Sabir, R. Sharma, A fine-grained sentiment analysis of app reviews using large language models: An evaluation study, 2024, arXiv preprint arXiv:2409.07162.

[62] M.M. Imran, P. Chatterjee, K. Damevski, Uncovering the causes of emotions in software developer communication using zero-shot llms, in: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, 2024, pp. 1–13.

[63] J. Shi, Z. Yang, H.J. Kang, B. Xu, J. He, D. Lo, Greening large language models of code, in: Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Society, 2024, pp. 142–153.

[64] A.L. Smith, F. Greaves, T. Panch, Hallucination or confabulation? neuroanatomy as metaphor in large language models, PLOS Digit. Health 2 (11) (2023) e0000388.

[65] S.C. Bellini-Leite, Dual process theory for large language models: An overview of using psychology to address hallucination and reliability issues, Adapt. Behav. 32 (4) (2024) 329–343.

[66] M.A. Ahmad, I. Yaramis, T.D. Roy, Creating trustworthy llms: Dealing with hallucinations in healthcare ai, 2023, arXiv preprint arXiv:2311.01463.

[67] X. Fang, S. Che, M. Mao, H. Zhang, M. Zhao, X. Zhao, Bias of AI-generated content: an examination of news produced by large language models, Sci. Rep. 14 (1) (2024) 5224.

[68] E. Ferrara, Should ChatGPT be biased? Challenges and risks of bias in large language models, First Monday 28 (2023) URL: https://api.semanticscholar.org/CorpusID:258041203.

[69] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A.T. Luu, W. Bi, F. Shi, S. Shi, Siren's song in the AI ocean: A survey on hallucination in large language models, Comput. Linguist. 51 (4) (2025) 1373–1418.

[70] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, J. Weston, Chain-of-verification reduces hallucination in large language models, in: Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 3563–3578.

[71] T. Hu, Y. Kyrychenko, S. Rathje, N. Collier, S. van der Linden, J. Roozenbeek, Generative language models exhibit social identity biases, Nat. Comput. Sci. 5 (1) (2025) 65–75.

[72] X. Bai, A. Wang, I. Sucholutsky, T.L. Griffiths, Explicitly unbiased large language models still form biased associations, Proc. Natl. Acad. Sci. USA 122 (8) (2025) e2416228122.

[73] J. An, D. Huang, C. Lin, M. Tai, Measuring gender and racial biases in large language models: Intersectional evidence from automated resume evaluation, PNAS Nexus 4 (3) (2025) pgaf089.

[74] C. Yan, M.H. Meng, F. Xie, G. Bai, Investigating documented privacy changes in android OS, Proc. ACM Softw. Eng. 1 (FSE) (2024) 2701–2724.

[75] S.J. Oishwee, Z. Codabux, N. Stakhanova, Decoding android permissions: A study of developer challenges and solutions on stack overflow, in: Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2024, pp. 143–153.

[76] M. Tileria, J. Blasco, S.K. Dash, DocFlow: Extracting taint specifications from software documentation, in: Proceedings of the 46th IEEE/ACM International Conference on Software Engineering, 2024, pp. 1–12.

[77] X. Ren, J. Sun, Z. Xing, X. Xia, J. Sun, Demystify official API usage directives with crowdsourced API misuse scenarios, erroneous code examples and patches, in: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, ICSE '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 925–936.

[78] W.S. El-Kassas, C.R. Salama, A.A. Rafea, H.K. Mohamed, Automatic text summarization: A comprehensive survey, Expert Syst. Appl. 165 (2021) 113679.

[79] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, G.-Z. Yang, XAI—Explainable artificial intelligence, Sci. Robotics 4 (37) (2019) eaay7120.

[80] C.-H. Chiang, H.-Y. Lee, Can large language models be an alternative to human evaluations? in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 15607–15631.

[81] Z. Gou, Z. Shao, Y. Gong, Y. Yang, N. Duan, W. Chen, et al., CRITIC: Large language models can self-correct with tool-interactive critiquing, in: The Twelfth International Conference on Learning Representations, 2024.

[82] X. Tan, S. Shi, X. Qiu, C. Qu, Z. Qi, Y. Xu, Y. Qi, Self-criticism: Aligning large language models with their understanding of helpfulness, honesty, and harmlessness, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, 2023, pp. 650–662.

[83] B. Gao, Z. Cai, R. Xu, P. Wang, C. Zheng, R. Lin, K. Lu, D. Liu, C. Zhou, W. Xiao, T. Liu, B. Chang, LLM critics help catch bugs in mathematics: Towards a better mathematical verifier with natural language feedback, in: W. Che, J. Nabende, E. Shutova, M.T. Pilehvar (Eds.), Findings of the Association for Computational Linguistics: ACL 2025, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 14588–14604.

[84] N. McDonald, K. Badillo-Urquiola, M.G. Ames, N. Dell, E. Keneski, M. Sleeper, P.J. Wisniewski, Privacy and power: Acknowledging the importance of privacy research and design for vulnerable populations, in: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, 2020, pp. 1–8.

[85] J. Pierce, S. Fox, N. Merrill, R. Wong, Differential vulnerabilities and a diversity of tactics: What toolkits teach us about cybersecurity, Proc. ACM Hum.-Comput. Interact. 2 (CSCW) (2018) 1–24.

[86] R.Y. Wong, J.C. Valdez, A. Alexander, A. Chiang, O. Quesada, J. Pierce, Broadening privacy and surveillance: Eliciting interconnected values with a scenarios workbook on smart home cameras, in: Proceedings of the 2023 ACM Designing Interactive Systems Conference, 2023, pp. 1093–1113.

[87] L.P. Argyle, E.C. Busby, N. Fulda, J.R. Gubler, C. Rytting, D. Wingate, Out of one, many: Using language models to simulate human samples, Political Anal. 31 (3) (2023) 337–351.

[88] J.S. Park, J.C. O'Brien, C.J. Cai, M.R. Morris, P. Liang, M.S. Bernstein, Generative agents: Interactive simulacra of human behavior, in: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, Association for Computing Machinery, 2023, pp. 1–22.

[89] S.H. Schwartz, M. Verkasalo, A. Antonovsky, L. Sagiv, Value priorities and social desirability: Much substance, some style, Br. J. Soc. Psychol. 36 (1) (1997) 3–18.

[90] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E.P. Xing, H. Zhang, J.E. Gonzalez, I. Stoica, Judging LLM-as-a-judge with MT-Bench and chatbot arena, in: Advances in Neural Information Processing Systems, Vol. 36, 2023.

[91] W. Agnew, A.S. Bergman, J. Chien, M. Díaz, S. El-Sayed, J. Pittman, S. Mohamed, K.R. McKee, The illusion of artificial inclusion, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, 2024, pp. 1–12.

[92] M.J. Burke, W.P. Dunlap, Estimating interrater agreement with the average deviation index: A user's guide, Organ. Res. Methods 5 (2) (2002) 159–172.

[93] K. Smith-Crowe, M.J. Burke, A. Cohen, E. Doveh, M. Kouchaki, S.M. Signal, Assessing interrater agreement via the average deviation index given a variety of theoretical and methodological problems, Organ. Res. Methods 16 (1) (2013) 127–151.

[94] S. Vanbelle, C. Hernandez Engelhart, E. Blix, A comprehensive guide to study the agreement and reliability of multi-observer ordinal data, BMC Med. Res. Methodol. 24 (1) (2024) 310.

[95] J. Cohen, Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit, Psychol. Bull. 70 (4) (1968) 213–220.

[96] R.A.A. Athayde, et al., Medidas implícitas de valores humanos: elaboração e evidências de validade, 2012.