

Human values in software development artefacts: A case study on issue discussions in three Android applications

Arif Nurwidyanoro^{a,*}, Mojtaba Shahin^a, Michel R.V. Chaudron^b, Waqar Hussain^a, Rifat Shams^a, Harsha Perera^a, Gillian Oliver^a, Jon Whittle^c

^a Department of Software Systems and Cybersecurity, Faculty of IT, Monash University, 3800, Clayton, Australia

^b Department of Mathematics and Computer Science, Eindhoven University of Technology, 5600, Eindhoven, Netherlands

^c CSIRO's Data61, 3168, Clayton, Australia

ARTICLE INFO

Keywords:

Human values
Issue tracker
Software artefacts
Software repositories

ABSTRACT

Context: Human values such as inclusion, privacy, and accessibility need to be considered during software development to attract and maintain users. However, little effort has been made to study human values consideration in software development, particularly in software development artefacts.

Objective: Issue discussion is potentially a rich source for human values analysis because it is a common place for users and developers to share and communicate their concerns. This paper aims to investigate the extent to which human values are discussed and whether the presence of values differs across projects.

Method: We carried out a case study to discover human values in 1,097 issues collected from three Android projects: Signal, K-9, and Focus.

Results: We identified 20 value themes and proposed a contextualised software engineering description for each of them. The analysis shows that privacy, freedom, usability, and efficiency were the prevalent value themes in the issue discussions of these three projects. Meanwhile, Self-direction - Action and Security - Personal are the common prevalent human values found in the projects. Moreover, we found that a statement of values from the apps and their functionalities could contribute to the presence of values.

Conclusion: The results suggest that human values are present in software development artefacts, for which automated tools can be developed to extract and classify human values from them.

1. Introduction

As of September 2020, Google Play Store, one of the largest mobile application stores for Android, has more than 3 million applications [1]. To stand out in the crowd, mobile application developers need to consider socio-technical factors besides the technical functionalities of their apps. For example, research in marketing sciences suggests that some factors such as emotional and epistemic values (e.g., curiosity, novelty, desire for knowledge) have a substantial effect on the intention to use an application [2]. Other research indicated that security and familiarity significantly impact users' trust towards installing a mobile application [3]. Another study found that, among others, attractiveness, stability, cost, accuracy, and compatibility are the main factors that influence users to favour a mobile application [4]. The factors influencing users' decision to install and use mobile applications are similar to those formulated by social scientists as **human**

values. Human values, or 'something that people consider important in life' [5], such as *privacy* or *hedonism*, are served as 'criteria or standards used by a person to make a decision' [6]. In the context of mobile applications, a privacy-aware user may decide not to install or use an application if the application had a history of a privacy breach. The recent Facebook-Cambridge Analytica case is an example of how a breach of a human value (e.g., *privacy*) could affect users' decision on using an application. In this particular case, a survey showed that 40% of the respondents said they are either stopped or used Facebook less than before [7].

To attract and maintain users, mobile application development communities must pay attention to human values. However, Mougouei et al. [8] argue it is difficult to incorporate human values into software development. This difficulty stems from, for example, the lack of understanding and contextualised software engineering definitions

* Corresponding author.

E-mail addresses: Arif.Nurwidyanoro@monash.edu (A. Nurwidyanoro), Mojtaba.Shahin@monash.edu (M. Shahin), m.r.v.chaudron@tue.nl (M.R.V. Chaudron), Waqar.Hussain@monash.edu (W. Hussain), Rifat.Shams@monash.edu (R. Shams), Harsha.Perera@monash.edu (H. Perera), Gillian.Oliver@monash.edu (G. Oliver), Jon.Whittle@data61.csiro.au (J. Whittle).

<https://doi.org/10.1016/j.infsof.2021.106731>

Received 31 December 2020; Received in revised form 5 September 2021; Accepted 7 September 2021

Available online 22 September 2021

0950-5849/© 2021 Elsevier B.V. All rights reserved.

for human values. For this reason, we believe previous work on human values in software chose to use existing human value models from social sciences instead [9–14]. Unlike previous work, this study attempted to better understand human values in software by analysing a software development artefact as a place where human values can be discovered.

There is an idea from the literature that suggests technological artefacts could embody human values [15]. In software engineering, this argument is supported in previous studies that used software development artefacts, such as issue trackers or commit messages, to investigate some values, such as *privacy* [16,17], *security* [18–21], *accessibility* [22], or *energy efficiency* [23,24]. While their focuses could be related to human values, those works did not specifically discuss them as values. Furthermore, these studies are still limited to a specific set of values known in software engineering, leaving other remaining human values open for further investigation. These previous studies demonstrated that it is possible to discover human values in development artefacts. We chose issue discussion (or **Issues** in GitHub) among other development artefacts because through Issues, users and developers can communicate their feedback, problems, and concerns that they may have during software development¹ [25,26].

In this paper, we report a case study that collected 1,097 issue discussions from three open-source Android applications, Signal, K-9, and Focus, to measure the presence of values during software development and to propose contextualised software engineering descriptions for those values. We purposely selected these applications based on several criteria, including having a statement supporting some specific values, to increase the chance of discovering values. We then conducted a qualitative analysis to discover the presence of values in the issue discussions based on the definition that we developed in our pilot study. Based on the qualitative analysis of 1,097 issues consisting of 5,615 posts, 20 value themes were found. We also propose contextualised software engineering description for each value theme. Further, we observed that *privacy*, *usability*, *efficiency*, and *freedom* are commonly prevalent in all projects. However, our analysis reveals that the values statement and functionalities of the apps play a role in the presence of other value themes, particularly *privacy*, *pleasure*, and *sense of belonging* across projects. Our mapping of the value themes to Schwartz's refined theory of basic values revealed seven out of 19 values can be found. Among them, Self-direction - Action and Security - Personal are commonly prevalent across projects.

Our main contributions in this paper are: (1) We identify 20 value themes in the issue discussions of three messaging apps and their mapping into Schwartz's values, (2) We provide a contextualised software engineering description for each of the 20 discovered value themes, (3) We create a dataset of 1,097 annotated issues consisting of 5,615 posts and publish it [27] to be used by others to replicate, validate, or expand our study and develop potential automated values detection tools.

The remainder of the paper is organised as follow. Section 2 presents the theoretical background of this study. We outline our research questions and methodology in Section 3. Section 4 reports the results of our study. The discussions and reflections on the findings are discussed in Section 5. Section 6 describes threats to validity. Section 7 summarises related work. Finally, Section 8 concludes the paper and presents potential direction for future work.

2. Background

2.1. Human values

Human values are the 'guiding principles of what people consider important in life' [5]. Values are personal beliefs and serve as criteria or standards to guide the selection or evaluation of actions or policies [6,28,29]. Human values are also known to be 'infused with

feeling' [6] such that a threat or support to a value could respectively upset or excite people. For example, in the SE context, an app's user in an issue discussion expressed his/her dislike when the app depends on Google Play Services. This particular user feels his *independence* being threatened by the possible changes of policies from the service provider, which is beyond the control of the app's developers.

Human values have been studied extensively in social sciences resulting in several proposed models in recent decades. Fig. 1 shows the renowned human values models called Schwartz's theory of basic values [6,30] and its refined version [31]. In the earlier model, Schwartz categorised ten types of values (i.e., *basic values*), such as benevolence, stimulation, and universalism, along with their 'exemplary specific values that primarily represent each value type' [32] or *value items* [6], such as privacy, freedom, and equality. In this model, human values are arranged in a circular manner such that supporting values are adjacent to each other while values that are conflicting are in the opposite direction [6]. For example, pursuing freedom could introduce conflict to the value of respecting tradition. Fig. 1(a) shows the Schwartz original theory of basic values that consists of ten basic values (highlighted in black) and each of their corresponding value items. In the newer version of the model (Fig. 1(b)), the definition of values was refined, resulting in 19 basic values arranged similarly.

In this study, we use Schwartz's model as our primary source to illustrate what human values are. We use Schwartz's model for the following reasons: Firstly, Schwartz's model has been universally assessed in 83 countries [31]. Secondly, this model covers most values compared to other existing human value models. [5]. Thirdly, this model has been used in different fields such as computer science [33], data mining [34], and software engineering [9]. These values in Schwartz's theory are not meant to limit the analysts in discovering values but to illustrate the landscape of values.

2.2. Human values in SE

In acknowledging the importance of human values, some previous works attempted to integrate values into software development. For instance, Value-based Requirements Engineering (VBRE) was proposed to guide human values elicitation from stakeholders in the requirements phase [36]. Another framework from Human-Computer Interaction (HCI) field, called Value Sensitive Design (VSD) focused on values with ethical import in the design phase by identifying potential harms to stakeholders [37,38]. These previous works considered a specific set of values (e.g., values with ethical import) and addressed a specific phase in software development phases (i.e., requirements and design). Another notable related work by Winter et al. proposed a tool called Values Q-Sort to measure Schwartz's values in SE [39]. Schwartz's values are considered broader than ethics that do not cover things that 'an individual, group, or wider society consider as important' [39], such as successful or social recognition. However, Winter et al.'s work was focused on characterising the values of software practitioners instead of the presence of values related to apps from users' perspectives. Ferrario et al. [9] proposed Values-first SE to address the broader Schwartz's values in the entire development phases. This framework maps requirements to Schwartz's values model to build a prototype. Then, a reflection on values is carried out from users' feedback on the prototype. Similar to Ferrario et al. [9], our paper considered Schwartz's values from feedback submitted by users and contributors in issue discussions from three open-source applications.

2.3. GitHub and issue tracker

GitHub is one of the most popular collaborative coding platforms, especially for open-source projects. GitHub not only serves as a code repository but also as a platform for communication between software developers amongst each other and between users and developers [40]. These collaborations are captured in many of its features, such as code

¹ <https://github.com/features>.

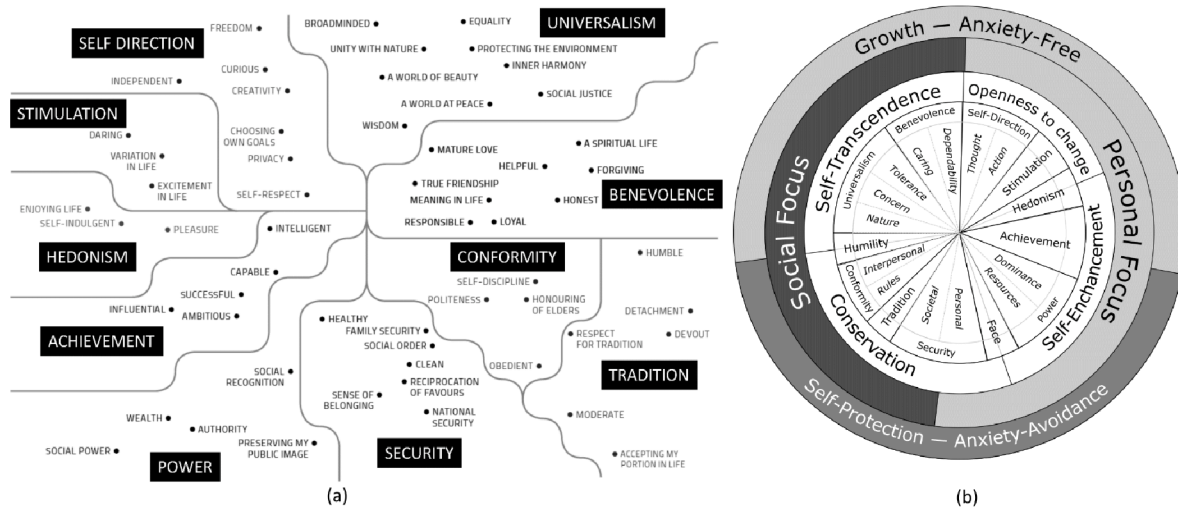


Fig. 1. The Schwartz models of basic human values: (a) the original model [32] taken from [35] and (b) the refined model taken from [31].

review and issue tracker, in the form of discussions (i.e. *discussion artefacts*). These discussions are accessible via a public API provided by GitHub. The data accessibility of GitHub, along with its popularity among developers, attracted researchers to use it as a primary source for software engineering studies [41].

The issue tracker in GitHub, or also known as *Issues*, is a place where users can report bugs and provide feedback to the project [40]. Issue tracker also allows project maintainers to organise tasks and discuss feature requests or enhancements with other contributors [25,26]. A *discussion* in the issue tracker (i.e. *issue discussion*) comprises several posts, initiated with the first post from a reporter (see Fig. 2). This post is then followed by subsequent posts from contributors who can be the project maintainers, developers, users, or the reporter itself. In issue discussions, it is easy to observe that a developer who contributes to the application usually also uses the app. Conversely, not all users identified in the discussion contribute to application development. Because issue discussions have been used as an investigation object for values in previous works [18,19], we believe that issue discussion is a suitable object of study to discover the extent human values are discussed in application development.

The focus of our study is on discussions between developers and/or users about how the application supports or violates end-users values. We referred to this as the *app-to-users* perspective (in Section 3.2.2). We did not focus on the social interaction between developers that happens in the issues (i.e., referred to as *contributor-to-contributor* in Section 3.2.2).

3. Methodology

Our overarching objective is to discover the presence of values in software development discussions. The following research questions guide our study:

RQ1 What values are discovered in issue discussions?

RQ2 Does the presence of values differ across projects? If so, how?

At the highest level, our case study consists of three phases: Data Collection, Pilot Study, and Main Study (see Fig. 3). In the data collection phase, we collected issues from each project's GitHub repository. The pilot study consists of four steps, namely issues sampling, issues allocation, pilot analysis, and discussion, to obtain insights for the main study. In the main study, we sampled issues from the collected dataset and conducted analyses to answer the research questions. The following subsections describe the methodology phases in detail.

3.1. Data collection

As a case study, we decided to investigate open source Android projects because of the following reasons:

- *Open source* - the software development process of these applications is publicly available in their repository.
- *Android applications* - as of 31 October 2020, the Android operating system has almost 73% of the mobile operating system market share worldwide [42].

We first purposively browsed for the applications under the communication category to select which projects should be investigated. We chose applications under the communication category because these applications tend to be used frequently and require more permissions than other categories [43,44]. Consequently, a discrepancy of behaviour in the apps (e.g., breach of personal information) will encourage users to express their concerns. Then, we reviewed each application to find applications that fit our criteria:

- the application has a substantial number of downloads (e.g., at least five million times)
- the application has a publicly available repository with a substantial number of contributors and active community
- the application has a statement of supporting human values (e.g., privacy, freedom, efficiency) in its homepage website to increase our chance of discovering values

In the end, we selected Signal Private Messenger, K-9 Mail, and Firefox Focus out of 10 applications that fit all of our criteria. We chose these three applications because we wanted to have one app for each type of application that we discovered in the communication category, namely, instant messenger, email client, and web browser.

We briefly introduced these three projects as follows:

1. *Signal Private Messenger* (or **Signal**) is a privacy-oriented messenger application supported by a non-profit organisation: The Signal Foundation. This application is determined to provide a private messenger to its users.² Since its first public beta release in 2010 [45], this application has evolved from only providing encrypted SMS (formerly known as TextSecure) into an integrated private communication application, supporting voice and video calls on top of text messaging function. In 2015, Signal Private Messenger phased out support for encrypted

² <https://signal.org>.

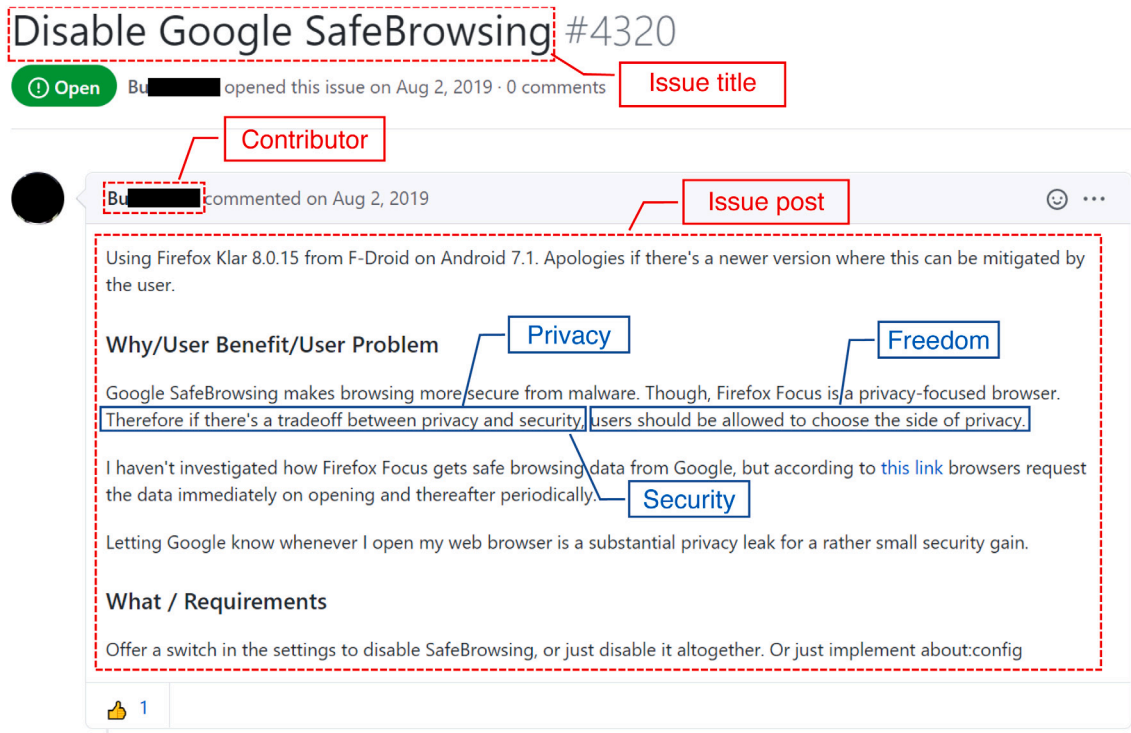


Fig. 2. An example of concern for values in the first post of an issue.

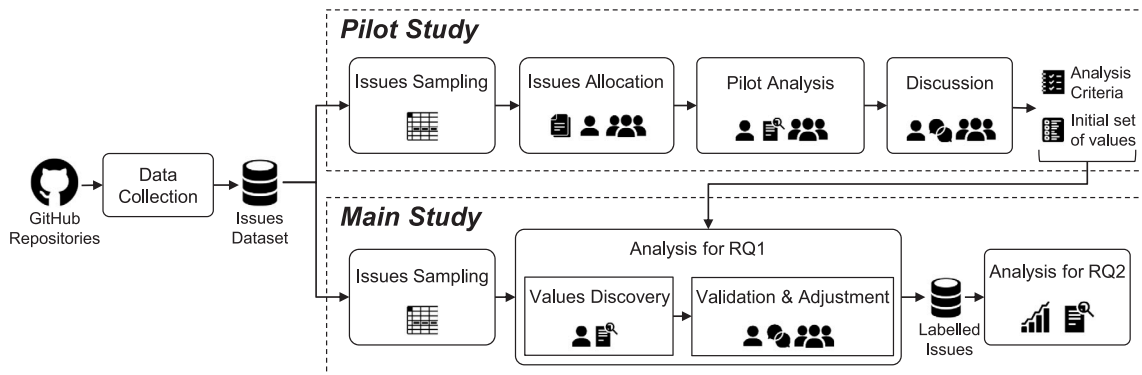


Fig. 3. Methodology.

- SMS [46] in favour of data-based communication and support for PC and iOS. As of 1 October 2020, this application has been downloaded more than ten million times in the Google Play Store. Fig. 4a shows the welcome message of the application having a statement about privacy.
2. *K-9 Mail* (or *K-9*) is an email application supporting PGP/MIME for private communication.³ K-9 Mail intends to provide seamless interaction for users to send and receive encrypted emails [47]. As of 1 October 2020, this application has been downloaded more than five million times from the Google Play Store. Fig. 4b shows the welcome message of the application showing a statement of efficiency and a list of supported features.
 3. *Firefox Focus* (or *Focus*) is a dedicated privacy browser application provided by Mozilla. This browser has functionalities, such as blocking advertisements and online trackers, resulting in better performance while maintaining the privacy and security of its users [48]. As of 1 October 2020, this application has

been downloaded more than five million times from the Google Play Store. Fig. 4c shows the onboarding page of the application showing a statement of privacy.

Among the available artefacts (e.g., code comments, README files), we decided to explore issue discussions in these projects to answer our research questions. Research [25,26] has shown that issues provide a place where users, contributors, and project maintainers can propose features, share their thoughts and knowledge, and comment on and criticise design decisions. We wrote a script utilising github3.py library⁴ to collect issues from these projects' repositories via the GitHub public API. We collected all of the 11,928 issues consisting of 62,526 posts from the Signal, K-9, and Focus repositories. Table 1 shows the number of collected issues and their corresponding number of posts. These three projects were used in both the pilot and the main study.

³ <https://k9mail.github.io/documentation/security/pgpmime.html>.

⁴ <https://github3.readthedocs.io>.

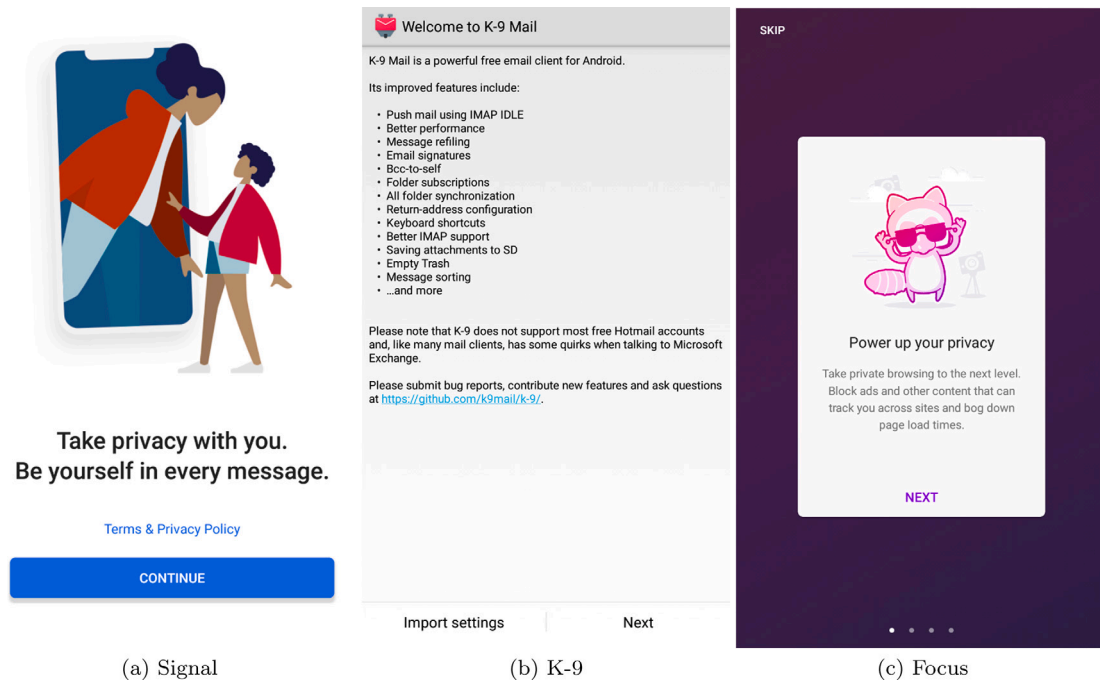


Fig. 4. Screenshots of the applications.

Table 1
The number of issues and posts collected.

Project	Repository URL	# Issues	# Posts
Signal	https://github.com/signalapp/Signal-Android	6,899	39,580
K-9	https://github.com/k9mail/k-9	2,499	11,663
Focus	https://github.com/mozilla-mobile/focus-android	2,530	11,283
Total		11,928	62,526

3.2. Pilot study

The pilot study was carried out to inform several key study design decisions: (i) identifying the criteria to determine whether human values are present and (ii) exploring the possibilities to discover themes specific to software engineering that may not yet be captured in the existing values models. This exploration was conducted because, to the best of our knowledge, there is no statement in prior works that the current human value models are exhaustive for software engineering fields. However, we still need to use the concepts and definitions of human values (i.e., Schwartz's theory) to ensure a similar understanding between the analysts involved. Fig. 5 shows how our pilot analysis was related to Schwartz's values. Initially, Schwartz's theory was used as a base for understanding human values. Then, we reflected on the descriptions we developed for value analysis with Schwartz's values.

3.2.1. Pilot process

The pilot study consists of four steps: issues sampling, allocating the analysts, pilot analysis, and discussion, described as follows:

Issues sampling: We started with Signal and K-9 as they have similar functionalities. We randomly selected 30 issues from these two projects such that each issue has 20 posts, hence comprising 600 posts in total.

Issues allocation: The pilot study involved six analysts (2 Females, 4 Males) ranging from Ph.D. students to an associate professor. All analysts had a reasonable understanding of human values before the study started, and all except one had a software engineering background. Because values are subjective concepts [49], there was a risk that the analysts interpreted human values and their definitions differently. We mitigated this risk by asking them to read seminal papers critically

(e.g., [6,31]) to ensure the analysts have a similar understanding about the concept of human values. The analysts then were divided into five pairs, in which a *primary analyst* (i.e., the first author) was paired with the five other analysts (i.e., *secondary analysts*). The decision to have a primary analyst was to overcome the limited availability of other analysts.

Pilot analysis: Each secondary analyst was asked to analyse six issues independently. In parallel, the primary analyst analysed all 30 issues. The analysts were asked to analyse the allocated issue discussions independently. The analysts used the open coding approach to allow themes to emerge from the issue discussions (i.e., *emerging themes*). The analysts are allowed to propose any theme as a value (i.e., *value theme* in Definition 1) without strictly following any human value models. For such cases, the analysts were required to describe their emerging values based on the phrases they found in the issue discussions. There was no limitation on the number of values the analysts could discover in a discussion. The analysts were given two weeks to finish their pilot analysis.

Discussion: Once the analysis process finished, the primary analyst held a meeting with each of the secondary analysts. These meetings mainly aimed to understand (i) the process and criteria employed by the analysts to discover values and (ii) the discovered value themes and their descriptions to develop an initial set of emerging values. Afterwards, the primary analyst discussed all the insights collected from these individual meetings with the other analysts to finalise the main study design decisions.

3.2.2. Pilot results

Our pilot study uncovered two perspectives on human values analysis. The first perspective concerns the (social) interaction between contributors (i.e., *contributor-to-contributor*). For instance, a contributor was *grateful* to another contributor who guided them to solve an issue: "Thanks for all the effort!". The second perspective concerns how the application supports or violates users' values (i.e., *app-to-user*). This analysis perspective could come from developers or end-users of the app. For example, users expressed their anger when a malfunction of the app costs him/her (i.e., *wealth* value): "This app is heavy on my data-bill". In another example, developers expressed their concerns about a possible privacy breach in the application: "It's bad enough that [the

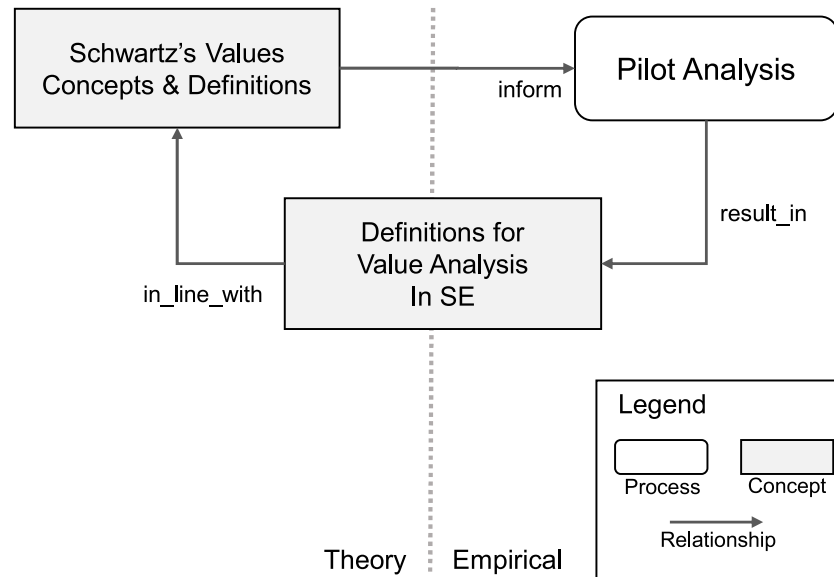


Fig. 5. The relation between the pilot analysis and Schwartz's theory.

app] uses phone numbers plus SMS for signup as opposed to usernames plus proof-of-work". Although the contributor-to-contributor perspective is also interesting, we only focused on the app-to-user perspective in our main study.

The pilot study led to the following observations:

- PO1 Some analysts were uncertain whether a theme they found in an issue discussion could be considered as a value (i.e., our objective). To address this, we considered a theme is a value theme if a phrase expresses the implications to users was found, resulting in dismay or appreciation of the app. This consideration is aligned with a human value's feature that explains 'when values are activated, they become infused with feeling such that people feel aroused when their values are threatened' [6]. Another literature also proposed that 'emotion is a source for values importance' [50]. For instance, users posted their displeasure for a malfunction in importing settings, preventing them from migrating the app to another device. We considered this issue has a value theme of *portability* because it fuels user dissatisfaction with the app.
- PO2 All analysts were comfortable using terminologies from Schwartz's models for some of their themes. We expected this to happen and decided to allow those terms to be used.
- PO3 In some cases, the analysts proposed a value with similar descriptions, but they adopted different terms to label them, such as *usability* and *convenience*. We calibrated those terms using the descriptions and examples provided by the analysts (see Table 2).
- PO4 Some emerging themes, such as *efficiency* and *accessibility* were relevant to SE. However, these value themes could not be easily mapped into Schwartz's values. This observation means either the analysts have different ways of thinking and require an extra inference process to map these themes to Schwartz's values, or the themes could not be mapped at all. For example, *accessibility* theme could be mapped to **Universalism - Concern** or **Benevolence - Caring** in Schwartz's refined theory. We believe this stems from the difference in levels of abstraction between the themes and the concept of human values. These themes are technical, while the concept of human values is abstract. We retain these themes and provide our suggestion of possible mapping to human values.

We calibrated the discovered values list by mapping each theme from all analysts based on their similarities and corresponding examples from the issue discussions. Then, we reached a consensus name

for those similar terms. Table 2 shows the themes discovered by the analysts, the mapping between the potentially similar themes, and also the proposed name for each value. For example, we found that there is a value term '*convenience*' from a secondary analyst (SA4), which was later defined into two different terms: *usability* and *efficiency*. Based on the discussions, we observed that, for some cases, '*convenience*' referred to how easy the app is for users. In another case, it referred to how quickly users can do tasks in the app.

We developed definitions to guide values discovery in the main study based on our discussion results. First, we defined how a theme found could be considered as a value theme:

Definition 1 (Value Theme). A theme that emerges from issue discussions can be considered a value theme if the theme captures a response of dismay or appreciation by a contributor towards the app.

As an example for Definition 1, a contributor reported that "... a mobile search (without server) using a single word query on 200 email takes *more than a minute* and k9 freezes and screen goes black, ... this is *extremely annoying*". In this example, we analysed that the contributor values *efficiency* such that the slow search annoyed this contributor.

Afterwards, we define when a value theme is present in issue discussions as follows:

Definition 2 (Value Theme Presence). A value theme is present when a theme from Definition 1 is discovered in issue discussions.

For example, contributors posted their opinion, "I *dislike* this because it is *reliant* on additional apps and resources". Based on Definition 2, we considered that value theme is present in this phrase because the value theme of *independence* found in the phrase.

Regarding the decision for the analysis method for the main study, we found that several value themes, such as *usability* and *efficiency* were discovered in issue discussions. These themes are not easily mapped to Schwartz's values in our discussions. For this reason, we decided to keep using open coding in the main study.

3.3. Main study

In the main study, we analysed issue discussions to discover human values using the definitions developed in the pilot study. Besides that, we also used themes found during the pilot study as an initial set of value themes (see Table 2). However, we did not limit our main study

Table 2
The mapping of values discovered in the pilot study.

PA	SA1	SA2	SA3	SA4	SA5	Proposed
Accordance	–	–	Conformity-Rules	–	–	Conformity
Pleasure	–	–	Pleasure	–	Pleasure	Pleasure
–	–	–	Universalism-Tolerance	–	–	Inclusiveness
Freedom	Freedom	–	–	Freedom	–	Freedom
–	–	–	–	–	Independence	Independence
Financial	–	–	–	–	Wealth	Wealth
Privacy	Privacy	–	–	–	Privacy	Privacy
Security	Security	–	Security	Security	Security	Security
Secrecy	–	–	–	–	–	Secrecy
Trust	–	–	–	–	–	Trust
–	–	–	–	–	Accuracy	Correctness
Compatibility	–	–	–	–	–	Compatibility
Financial	–	Sustainability	–	–	–	Longevity
Efficiency	–	–	–	Convenience	Intelligent	Efficiency
Usability	Usability	–	Usability	Convenience	–	Usability
Accessibility	–	–	Universalism-Concern	–	–	Accessibility
Transparency	Transparency	Transparency	–	–	–	Transparency
–	–	–	Humility	–	–	Humility
–	–	–	–	–	Healthy	Healthy

PA: Primary Analyst; SA#: Secondary Analyst.

Table 3
The number of issues for the main study.

Project	# Sampled issues	# Sampled posts
Signal	364	2,233
K-9	333	1,541
Focus	400	1,841
Total	1,097	5,615

only to this set and allowed new value themes to emerge from the issues. For the main study, we also included another app, *Focus*, which has different functionalities to Signal and K-9. We randomly sampled 1,097 issues consisting of 5,615 posts from these three projects. This number of issues corresponds to a sample that gives a 95% confidence level and 5% confidence interval. Table 3 shows the number of issues and posts analysed in the main study from each project.

3.3.1. Analysis for RQ1

To answer RQ1, i.e., *what values are discovered in issue discussions*, we conducted a qualitative analysis using open coding with constant comparisons. Using the constant comparison method, newly found themes are constantly compared to the previously found themes [51]. We used constant comparison to refine the descriptions and ensure the consistency of value themes that we found. To compare our findings with the existing theory, we then mapped the themes that we found in the empirical analysis with the theory of Schwartz's refined value model. Fig. 6 shows the relation between Schwartz's values and our main study. This picture shows that we mapped the themes that we found with Schwartz's refined theory of basic values.

During the qualitative analysis, there was a limitation in terms of the availability of the analysts. To overcome this, we divided the analysis process into two phases, namely values discovery followed by validation and adjustment. The values discovery phase was conducted by the primary analyst in the pilot study. In the next phase, three validators were employed to validate a subset of the primary analyst's analysis results. The details of each phase are explained as follows:

Values Discovery. In this phase, the primary analyst analysed issue discussions for value themes. The primary analyst was allowed to propose any value themes without having to strictly refer to the initial set of themes from the pilot study. For each value theme found, the analyst formulated a description from phrases found in the issue discussions. Similar to the pilot study, we did not limit the number of value themes to be found in an issue discussion. Then, the primary analyst attempted to map each theme found to human values from Schwartz's refined value model. This model was chosen because it has

a finer distinction of each value than the previous model [31]. It took approximately two months for the primary analyst to complete this phase.

Values Validation and Adjustment. Once the values discovery phase finished, a validation was conducted on the analysis result. The validators ranged from research fellows to a professor. All of them had backgrounds in software engineering. Two of the validators were involved in the pilot study. A new validator was introduced to address possible bias in the validation due to the previous involvement in the pilot study. The new validator was also asked to study literature on human values prior to the validation.

To obtain a significant sample for the validation, we randomly sampled the issues analysed in the previous step using a confidence level of 95% with a 5% margin of errors resulting in 349 issues consisting of 1,976 posts. The first, second, and third validators were allocated 150, 100, and 99 issues, respectively. This allocation scheme was based on the time availability of the validators. The validators were asked to review each assigned issue. They had to indicate whether they agreed or disagreed with the primary analyst's analysis and provide comments. The validators were given two weeks to complete the validation.

At the end of the validation process, the primary analyst held a meeting with each validator. Before the meeting, the agreement between the validators and the analyst was 72%. After the meeting, the agreement reached 89%. The primary sources of disagreements were as follows:

- *Refinements to the definition of the emerging values.* The validators proposed several refinements to the descriptions of existing values based on their findings. For example, a validator found a conversation in issue discussions, indicating that users still need help from a contributor to configure the app if they change their phone. A validator then proposed to broaden the description of *independence* by adding '*relying on someone else*' to cover the issue. These refinements were agreed upon by the analyst and the validators.
- *Different inference process in the analysis.* A validator proposed to include themes relevant to human values that are not *directly* relevant, i.e., requires some steps of inference. For example, a user reported that in Android Auto, the app displayed only the latest message from a contact, not the complete messages. The validator proposed a value of '*helpful*' based on the reasoning that displaying all messages from contact is more helpful to a user.

Once the validation finished, the primary analyst updated the final labelling for 349 issues. The primary analyst then revisited the remaining 748 issues and made adjustments based on the validation

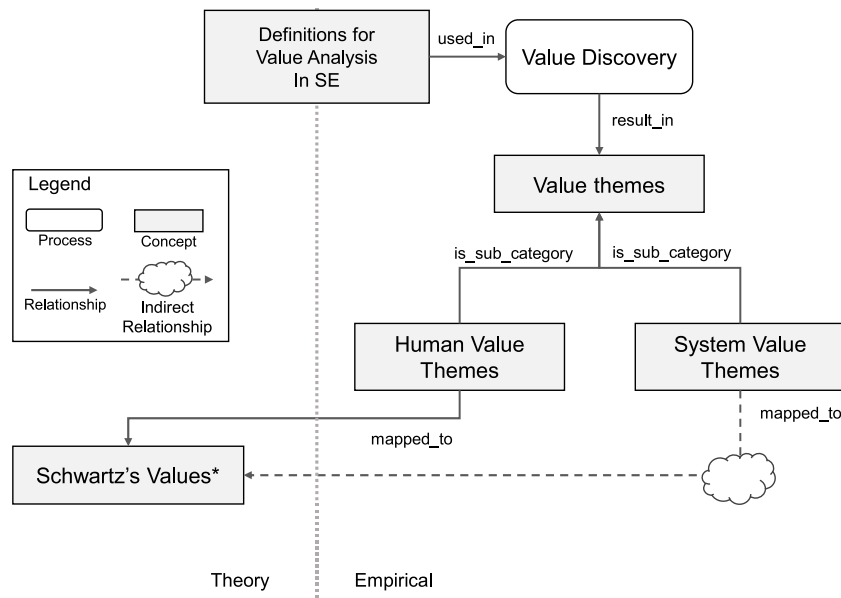


Fig. 6. The relation between the main study and Schwartz's values.

and adjustment phase insights. It took approximately three weeks for the primary analyst to adjust the dataset. We used this analysis result to answer the first research question.

3.3.2. Analysis for RQ2

For RQ2, we used the analysed dataset of qualitative analysis for RQ1. To answer the first part of RQ2, i.e., *does the presence of values differs across projects*, we conducted a quantitative analysis to obtain the number of issues where values are present in each project. Because the number of analysed issues differed for each project, we performed the analysis using the percentage of issues where values are present on each project instead of the number of issues.

To answer RQ2, we first identified and compared which values were more prevalent in each project. Afterwards, we compared the presence of values across projects. To this end, we identified which values have a substantial difference of presence across projects, followed by a qualitative analysis to comprehend the nature of their corresponding issue discussions and if they are different across projects.

4. Results

In this section, we describe the result of our analysis following the research questions:

RQ1 What values are discovered in issue discussions?

RQ2 Does the presence of values differ across projects? If so, how?

4.1. RQ1: values discovered

In the analysis of the three applications (Signal, K-9, and Focus), we discovered 20 value themes. Among them, 15 themes were the same value themes as identified in the pilot study. The remaining five themes emerged in the main study. We mapped these 20 themes to Schwartz's values as depicted in Fig. 6. We found ten of them could be mapped directly to Schwartz's values (i.e., *human value themes*). Meanwhile, the other ten themes were technical and required an extra step of inference to Schwartz's values or could not be mapped (see PO4 in Section 3.2.2). We referred to such themes as *system value themes*, as they are important for users when using the application (or system-human interaction).

4.1.1. Human value themes

We discovered ten themes that could be mapped to Schwartz values. Table 4 presents these themes, their description, and their mapping to Schwartz's refined theory of basic values [31]. The description of each theme is conceptualised from the issue discussion and represents how its corresponding human value could be manifested in software development. *Conformity* is about how an application follows the rules, regulations, guidelines, or standard practices. For example, we found a Signal's user who complained about the displayed text that did not align with the Android design guidelines. This theme could be mapped into **Conformity - Rules** in Schwartz. *Pleasure* promote the enjoyment of using the application. For instance, a contributor believed that emojis in Signal is amusing: "I do not see what you are complaining about, it [emojis] is a fun feature". Pleasure has a similar concept with **Hedonism** in Schwartz's. *Dignity*, which can be mapped to **Face**, is about maintaining honour and respect for users while using the application. For example, we found a user reported, "I have been in a huge fight with the Vodafone customer service, called them at least 10 times, have ruined their shop over here in Germany, write multiple heavy complaints to them and was really upset - until my girlfriend noticed that this Issue appeared since we have been using [the app]". *Inclusiveness* ensure anyone with diverse backgrounds could use the application. This theme could be mapped to **Universalism - Concern**. *Inclusiveness* in an application could manifest as user interface translation or other regional specific settings such as a right-to-left layout to support the Arabic language. *Sense of belonging* is about how users can connect and interact with their peers through the application. This theme could be mapped to **Security - Personal**. As an example for this theme, we found a user of Signal complained, "I have been invited to a group of friends and I have not received invite or messages".

The following two themes, *freedom* and *independence*, similarly could be mapped into **Self-direction - Action**, but they are somewhat different. Freedom is about providing options for users. The following post, "people are being forced to register with TextSecure [Signal] Message Server based in the USA", demonstrated the lack of options becomes a threat to freedom. On the other hand, independence emphasises not being influenced or dependent on any third parties. As an example, we found a contributor of K-9 dislike the idea to add a third-party spam filtering because "it is reliant on additional apps and resources". The value theme of *Wealth* could be mapped to **Power - Resources**. It relates to the monetary cost needed to use an application. For instance, a user

Table 4
Human value themes discovered in issue discussions.

No.	Value Theme	Human value*	Description
1	Conformity	Conformity - Rules	Follow rules, regulations, guidelines, or standard practices during the application development
2	Pleasure	Hedonism	Promote enjoyment and satisfaction while using the application
3	Dignity	Face	Maintain the honour and respect for users
4	Inclusiveness	Universalism - Concern	Facilitate different origins, languages, cultures, and level of knowledge in the application
5	Sense of belonging	Security - Personal	Connection and interaction between users
6	Freedom	Self-direction - Action	Options for user preferences
7	Independence	Self-direction - Action	Little or no reliance on any third parties in the use of an application
8	Wealth	Power - Resources	Monetary costs of using the application
9	Privacy	Self-direction - Action	Protection of personal information
10	Security	Security - Personal	Protection against threats and attacks from foreign entities

*Mapping result of the value theme to Schwartz's refined theory of basic values [31].

Table 5
System value themes discovered in issue discussions.

No.	Value Theme	Description
1	Trust	Compelling users that an application is harmless
2	Correctness	The application provides expected information and behaviour
3	Compatibility	Support a range of user devices and systems
4	Portability	Migratable to other devices
5	Reliability	Less occurrence of application's failures
6	Efficiency	Less time and effort when conducting tasks in an application
7	Energy Preservation	Less use of power or energy
8	Usability	Ease of use of the application
9	Accessibility	Ease of access in regards to information or functionalities in the application, especially for users with special needs
10	Longevity	Prolonged availability of the application in the future

was found complained about the way K-9 fetching email could result in “heavy on my data bill”. The last theme in this category, *privacy*, could be mapped into **Self-direction - Action**. Privacy, which is about protecting personal information, also becomes the centre of attention in software development lately. For example, K-9's contributors discussed the possibility of supporting network proxy to enhance users' privacy. *Security* concerns with protecting users against threats or attacks from foreign entities. This security description is more technical and slightly more narrow than a similar concept in Schwartz's theory. The theory defines security as “avoiding danger and feeling cared about” (i.e., personal security) and “social order and government stability” (i.e., societal security) [31]. In this case, we believed that the security theme that we found was more suitable to be mapped to **Security - Personal**.

Based on the mapping of these human value themes into Schwartz's refined model [31] in Table 4, we discovered seven out of 19 values. Two of those seven values, namely *Security - Personal* and *Self-direction - Action*, corresponded to more than one discovered theme. *Security - Personal* corresponded with security and sense of belonging. Meanwhile, *Self-direction - Action* is found to correspond with freedom, independence, and privacy. The remaining human value corresponded with one discovered theme each.

4.1.2. System value themes

We discovered ten themes that matter for users but require an extra inference to be mapped to Schwartz's values. We placed them into a different category and referred to them as system value themes. Table 5 presents the system value themes discovered along with their description.

Trust indicates how an application convinces users that it is harmless. For instance, contributors of K-9 raised their concern in the following post: “if one doesn't trust K-9 to behave, there's no hope”. *Correctness*, *compatibility*, *portability*, *reliability*, and *efficiency* are related to the quality attributes of software [52]. We found evidence that these

themes could affect users. *Correctness* aligns information or behaviour of the app with users' expectations. For instance, Signal's users reported that pressing their own message showed the recipient's number instead of their number. *Compatibility* relates to support of an app to various devices and internal systems. For example, some users of K-9 reported application crashes after they updated their systems. *Portability* allows users to migrate their data and configurations in the app from one device to another. For instance, the following comment in a Signal discussion reflects a case where users cannot change their device to use the app: “Currently there is no way to change the phone without losing existing data”. *Reliability* concerns the stability of the app during use. As an example, the lack of reliability encourage a user to report, “Unfortunately, that's all I have... :/ I tested a bit, and it seems to crash at random positions. Not the same post every time, not the same type of post”. *Efficiency* can be associated with the time and effort needed to perform tasks in the app. For example, a user of Signal pointed out how the low performance of the app could discourage users, “... group chat isn't even that fast, and this is discouraging my friends a lot. Energy preservation is similar to efficiency but in terms of energy usage. We separated energy preservation because we found many discussions demonstrating users' concerns about how the app drained their devices' power. For instance, a K-9 user reported how the app is “eating” his battery when there is no network connection.

Accessibility and *usability* have a similar characteristic in terms of how easy users can use the app. The difference is that *accessibility* is related to accessing information and features, especially for disabled users. The lack of accessibility support in the app made a K-9 user mentioned, “It probably also makes the app much harder or even impossible to use with accessibility services”. Meanwhile, *usability* concerned with the ease of use of the app. For instance, a K-9 user complained, “mail view should remember the scrolled position ... This is a tiny usability thing, but I think I would make a big difference”. Lastly, *longevity* ensures the future development and use of the app. For example, a contributor reminds the forum: “Be aware, that 95% of other users will have lost their backups because you removed support for external SD cards.”

As we have explained, these system value themes could not be easily mapped into Schwartz's values. Nevertheless, based on the empirical findings that we have, we attempted to map these themes to Schwartz's values. Table 6 shows our working mapping between system value themes and Schwartz's values. Our theme of *trust* could be loosely mapped to **Conformity - Interpersonal** with the similarity of objective in avoiding harm to others. The *compatibility* theme could be mapped to **Power - Resources** in terms of how devices (i.e., resources) are supported by the application. For *portability* theme, we could map it to **Self-direction - Action** in terms of how the application should allow users to migrate to other devices. *Portability* could also be mapped to **Power - Resources** in terms of its relation with devices as the resource. The themes of *efficiency* and *energy preservation* both could be mapped to **Power - Resources**. We thought both themes are related to

Table 6

The mapping between system value themes and human values in the context of the three projects.

No.	Value theme	Human values
1	Trust	Conformity - Interpersonal
2	Correctness	–
3	Compatibility	Power - Resources
4	Portability	Self-direction - Action, Power - Resources
5	Reliability	–
6	Efficiency	Power - Resources
7	Energy Preservation	Power - Resources
8	Usability	Benevolence - Caring
9	Accessibility	Benevolence - Caring, Universalism - Concern
10	Longevity	Security - Personal

resources management, namely time and energy. The theme of *usability* and *accessibility* could be mapped to **Benevolence - Caring** because they are about supporting users in using the application. Additionally, *accessibility* could also be mapped to **Universalism - Concern** because this theme also aimed to support users with special needs. Longevity could be mapped to **Security - Personal** because it is related to assurance to users that the application will be available in the future. We could not find any Schwartz's values that suitable for the *correctness* and *reliability* themes. These mapping between system value themes and human values are limited to the context of our three projects under study.

RQ1 Summary: We discovered 20 value themes in issue discussions and described each of them. The themes can be broadly classified into two groups:

- Ten value themes, including conformity, pleasure, dignity, inclusiveness, sense of belonging, freedom, independence, wealth, privacy, and security have a corresponding concept in Schwartz's refined theory of basic values
- Ten other value themes could not be directly mapped into Schwartz's values: trust, correctness, compatibility, portability, reliability, efficiency, energy preservation, usability, accessibility, and longevity

We also found seven out of 19 values from Schwartz's refined model corresponded to the discovered human value themes, namely Self-direction - Action, Security - Personal, Hedonism, Universalism - Concern, Power - Resources, Conformity - Rules, and Face.

4.2. RQ2: The presence of values across projects

4.2.1. Value themes across projects

As shown in Fig. 7(a), value themes were present in 33% of the sampled issues constructed from all three projects. If we break it down to each project, Signal has the most issues where value themes were present: 163 out of 364 issues or 45% (Fig. 7(b)). It is followed by K-9, in which 31% of issues (102 out of 333 issues) contained value themes (Fig. 7(c)). The project with the least presence of value themes is Focus with 25% or 102 out of 400 issues (Fig. 7(d)).

Table 7 shows the presence of value themes in each project. For human value themes, in Signal the top three most prevalent themes were *privacy*, *security*, and *freedom*. Aside from these three themes, the other remaining human value themes were relatively higher than other projects. The top three most prevalent human value themes in K-9 were similar to those in Signal. However, *sense of belonging* was not present in K-9. Meanwhile, Focus also has a similar top three most prevalent themes with Signal and K-9 except for *inclusiveness* which was more prevalent than *security*. Two human value themes, namely *sense of belonging* and *dignity* were not present in Focus at all.

Table 7

Percentage (%) of value theme occurrences.

Category	Value theme	Signal	K-9	Focus
Human Value Theme	Conformity	2.20	0.90	2.00
	Pleasure	3.57	0.30	0.25
	Dignity	1.37	0.90	0
	Inclusiveness	2.75	1.20	*3.00
	Sense of belonging	3.30	0	0
	Freedom	*6.32	*2.40	*4.00
	Independence	2.47	1.20	2.25
	Wealth	2.47	0.60	0.50
	Privacy	*9.07	*2.10	*8.75
	Security	*4.95	*6.61	2.25
System Value Theme	Trust	1.37	0.30	1.00
	Correctness	*2.20	0.90	0.50
	Compatibility	1.37	0.90	0.25
	Portability	1.10	0.30	0
	Reliability	0.27	0.30	0.50
	Efficiency	*7.42	*5.11	*3.75
	Energy Preservation	1.65	*2.10	0.75
	Usability	*13.19	*11.71	*8.75
	Accessibility	1.10	0.60	*1.75
	Longevity	1.10	0	0.25

*Denotes top three most prevalent themes of each category in each project.

For system value themes, Table 7 shows that in Signal, the top three most prevalent themes were *usability*, *efficiency*, and *correctness*. The top three most prevalent system value themes in K-9 were similar with the themes in Signal, except for *energy preservation* which was more prevalent than *correctness*. Longevity theme was not found in K-9. Focus also had similar prevalent system value themes with the other two projects except for *accessibility* which was more prevalent after *usability* and *efficiency*. Unlike the other two projects, the theme of *portability* was not found in Focus.

Fig. 8 makes it clear that the presence of *privacy* in Signal and Focus is substantially higher than in K-9. We observed that Signal and Focus contributors provided more attention to privacy-related functionalities provided by the apps, such as private messaging in Signal or private browsing in Focus. For example, a contributor of Focus reminded how an implemented functionality might not be appropriate by saying, "... something that feels less appropriate for a **privacy-centred app**". This finding is aligned with our expectations since both Signal and Focus have a strong statement of privacy.

Besides privacy, Fig. 8 shows that *pleasure* was also present substantially higher in Signal than in K-9 and Focus. We realised that users of Signal caused this as they demanded the application to be enjoyable to use. We observed many requests related to the app's look and feel (e.g., themes and colours) and support for individual's expressions (e.g., multimedia and emojis). For instance, Signal's user suggested having the functionality to colour incoming messages based on its authors for the following reason: "People love to customize their stuff, and many are [quite] **emotional** regarding colors".

Sense of belonging was only discovered in Signal. We found that all the discussions about this value came from the messaging functionality, especially group messaging, where it became personal to users. For example, a user expressed his/her dissatisfaction, "I'm just **dismayed** at the effort I put into pushing my whole family & all my friends to use it [Signal] & now I can't keep it working".

Similar to *sense of belonging*, some themes were also not discovered in K-9 (e.g., *longevity*) and Focus (e.g., *dignity*, *portability*, and *secrecy*). However, the presence of these themes was low (below 2%) in the other projects. The other remaining themes' presence was not substantially different (less than 2% difference) across the three projects.

4.2.2. Human values across projects

The presence of seven values from Schwartz's refined model based on the mapping of the human value themes are shown in Table 8. In

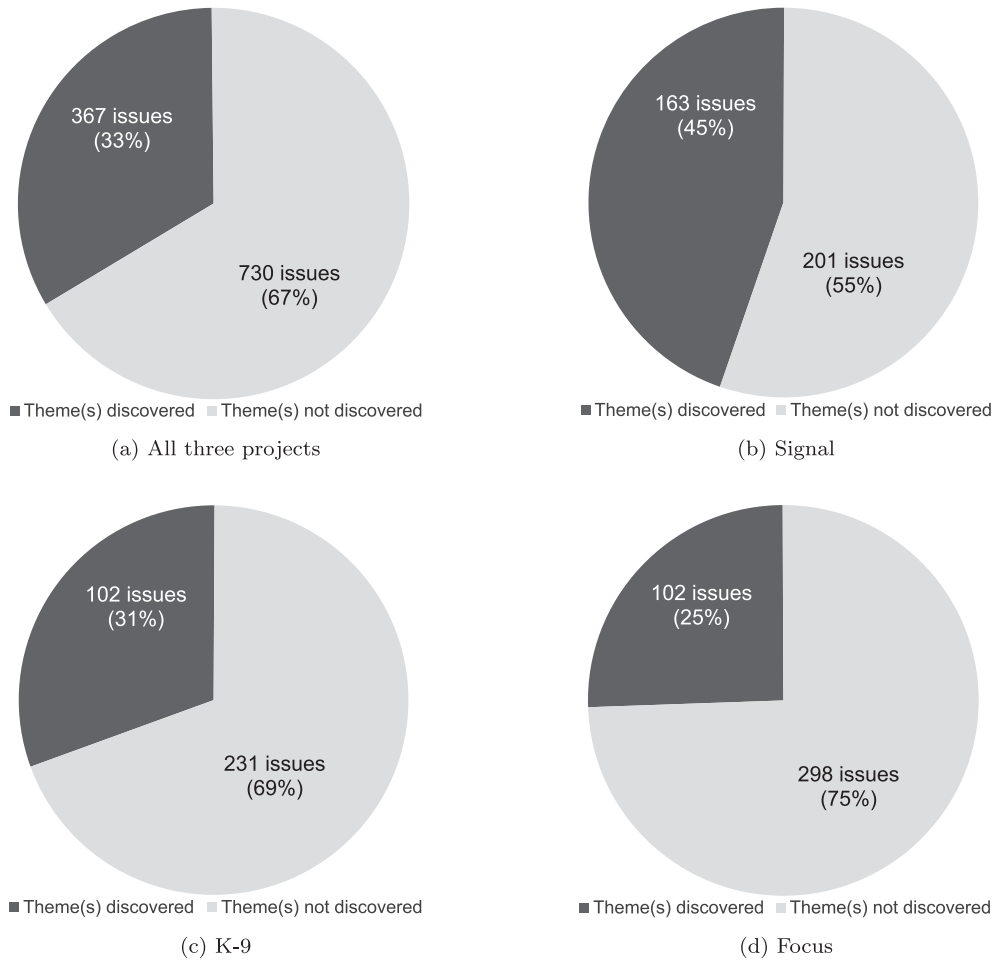


Fig. 7. The presence of values in issue discussions.

Table 8

Percentage (%) of human values occurrence.

No.	Human value	Signal	K-9	Focus
1	Self-direction - Action	14.56	4.20	11.00
2	Security - Personal	7.97	6.61	2.25
3	Hedonism	3.57	0.30	0.25
4	Universalism - Concern	2.75	1.20	3.00
5	Power - Resources	2.47	0.60	0.50
6	Conformity - Rules	2.20	0.90	2.00
7	Face	1.37	0.90	0

this section, we did not consider our working mapping between system value themes and human values in Table 6. Self-direction - Action is prevalent in both Signal and Focus. We found that the discussions in issues of Signal and Focus were mainly about privacy and the freedom-related functionalities provided in both applications. Meanwhile, the most prevalent value present in K-9 is Security - Personal. This is in line with the focus of K-9 as a secure mail client application.

The presence of other values in Signal is relatively high (more than 1%). Meanwhile, the values that present more than 1% of K-9 issues are Security - Personal, Self-direction - Action, and Universalism - Concern. In Focus, values that present in more than 1% of the issues are Self-direction - Action, Universalism - Concern, Security - Personal, and Conformity - Rules. The value of Face was not found in Focus. We found this is because Focus is somewhat more general and personal (e.g., not used to communicate with other people).

RQ2 Summary: Our analysis shows that almost one-third of the studied issues in Signal, K-9, and Focus apps include discussions where value themes can be found. The common prevalent human value themes across projects are privacy and freedom, while the common prevalent system value themes are usability and efficiency. However, the value themes of privacy, pleasure, and sense of belonging, were not evenly present across these apps, which can likely be attributed to the apps' values statement and functionalities. Based on the mapping of human value themes to a human value model, we found Self-direction - Action and Security - Personal as the most prevalent values in the three applications.

5. Discussion

5.1. Human values are present in issue discussions

The presence of values in design artefacts has been theorised in general. Our results contribute empirical evidence that human values are indeed found in software development repositories and, in particular, which values can be found in issue discussions. Our pilot study identified two distinctive perspectives of value themes from issue discussions: *contributor-to-contributor* and *app-to-user* (see Section 3.2.2). Our study focused on the *app-to-user* perspective by analysing project contributors' concerns and opinions as found in issue discussions. This perspective has allowed us some insights into how and to what extent project contributors concern about human values while developing software applications. Using this approach, we discovered 20 value themes

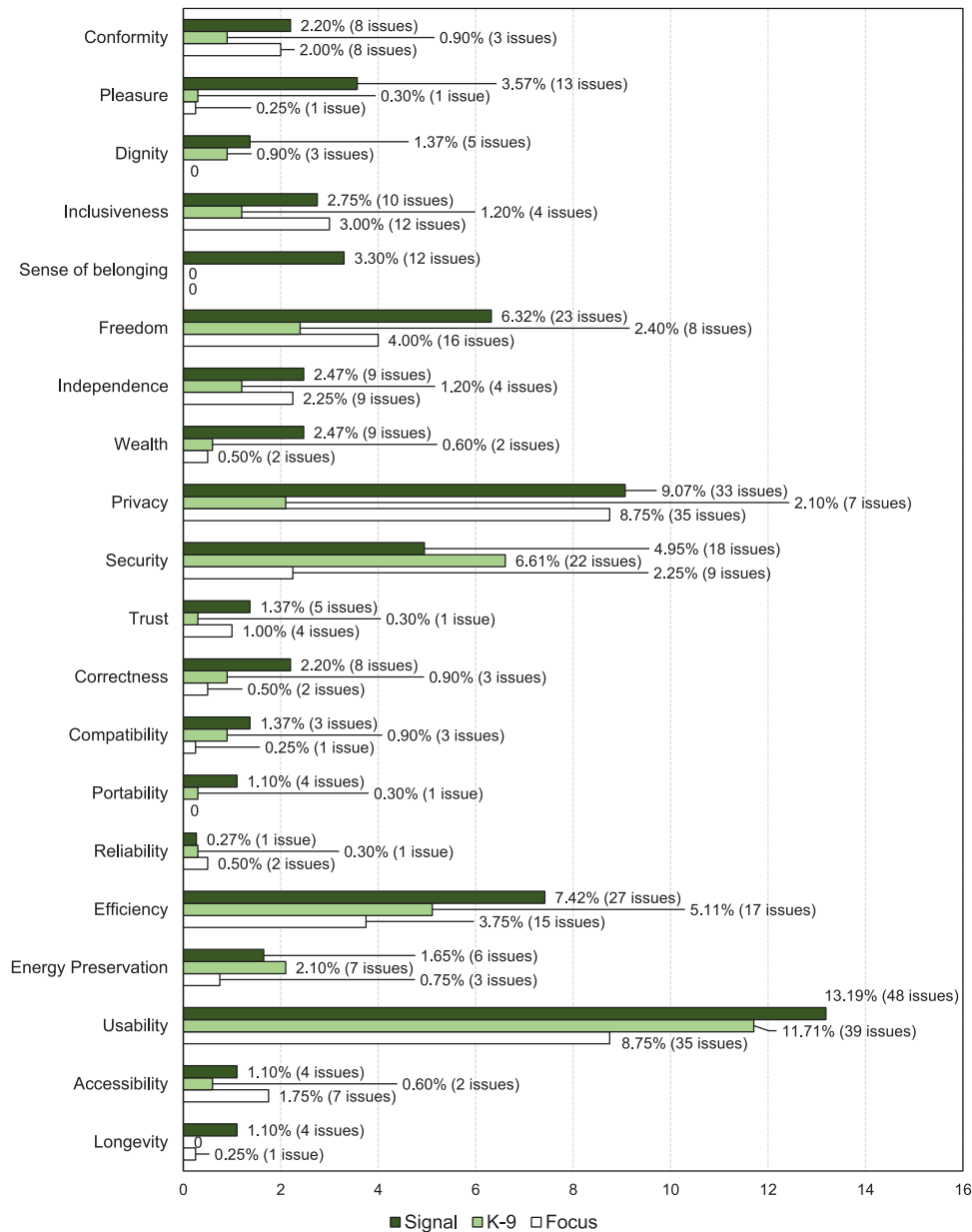


Fig. 8. Percentage (%) of values presence in the projects.

and formulated their descriptions based on a qualitative analysis of 5,615 posts from three projects. We found that the examples we found in issue discussions for the themes are similar to the concept of *value instantiation* [50,53,54]. Value instantiation is defined as “the representation of values in specific situations, issues, or behaviours” [50,54]. In our case, discussions in issues provide a representation of human values in software engineering contexts.

Among 20 value themes that we found, ten of them could be mapped into seven out of 19 values from Schwartz’s refined model. The other ten themes are technical such that mapping these themes to human values require an extra step of inference. Our attempt to mapping the system value themes to human values is presented in Table 6. These findings confirm the much earlier opinion that technological artefacts could contain human values [15]. Specifically, these findings are in line with previous works [16–24] that discovered values in software development artefacts. These results indicate that other artefacts, such as source codes, information about commits, README files, etc., might be potential places to find and discuss human values. We argue that further research is needed to explore such software

artefacts to understand the extent to human values may appear in these artefacts. Future research should also investigate the difficulty level of identifying human values in each software artefact.

5.2. The prevalence of human values in software engineering

The refined Schwartz’s model [31], as the refinement of the most widely used model in social sciences [6], includes 19 values. In this study, only seven of those 19 values are found in our case study’s issue discussions (Table 8). This result is in line with the previous work [11] which discovered that 60% of Schwartz’s value items (35 out of 58 value items) were not found in software engineering publications. This could mean some values are still under-represented in both research and practice, and thus further research is needed. For example, further research can investigate the reasons behind this under-representation (e.g., software practitioners are unfamiliar with these values). Another line of research can be conducted to investigate how to support those values in software engineering, e.g., provide a contextualised software engineering definition for these under-represented values.

5.3. Human values definitions for software engineering

To the best of our knowledge, there are no human value models developed exclusively for software engineering. Notable work in Human–Computer Interaction (HCI) field called Value Sensitive Design (VSD) defines a subset of values that have ethical imports [55] could be somehow transferred into SE. However, social sciences' definition of human values are much broader by including '*what an individual, group or wider society believe to be important*' [39], such as *public image* or *achievement*.

The lack of practical definitions for the broader human values in the SE context was argued by Mougouei et al. [8] as one reason why software engineering practices have not entirely successful in integrating human values in software. Previous works have attempted to address this issue using existing models from social sciences as a starting point. For example, some previous works chose to use Schwartz's model [8–12] because of its extensiveness in comparison with other models. Nevertheless, these models still need to be adapted for the SE context. This adaptation is necessary because of the possible '*differences in the meaning and interpretation of values in different contexts*' [11]. For instance, our findings of *security* in the software engineering context have a limited scope within the app, unlike the definition of security in Schwartz's model, which encompasses the whole society [6]. We contributed towards the lack of practical human values definitions for software engineering by proposing descriptions of human values from issue discussions. Although the descriptions can still be developed further, we argue that these initial descriptions can empower software practitioners to consider human values in their software engineering practices. For instance, if some users have concerns regarding the independence of an application, the application developers may respond by considering to what extent they want to use third-party services. We realise that some of the themes that we discovered, especially those that are well-known in SE, such as efficiency or usability, may already have defined in SE. In this case, future work can compare our descriptions from the empirical study of issue discussions with their definitions in SE literature. We also believe that future research is still necessary to investigate how these definitions can help incorporate human values in software engineering.

5.4. Human values and non-functional requirements

One may argue that the system value themes that we found (Section 4.1.2), such as *efficiency* and *usability*, are similar to non-functional requirements (NFRs). The NFRs are usually related to quality properties, characteristics, or attributes of software [52,56,57]. On the relation between NFRs and human values, Barn [57] conducted a meta-review of NFRs and found no reference for values. Barn concluded that '*either value are systematically ignored in the practice of NFRs elicitation or value may not be NFR*' [57]. Our definition of human values presence (Definition 2) and findings enable us to argue that values are similar to NFRs in terms of concerns a user or a contributor may have towards an application. However, we believe that they have a much broader sense, which includes non-technical requirements. In other words, we argue that NFRs can be seen as a subset of human values. In this sense, we agree with Barn's proposal of utilising NFR frameworks to help integrate values into software [57].

5.5. Factors that influence value presence

Our empirical study found that 33% of the discussions investigated in the three projects include at least one value theme. Among those, value themes appeared more in Signal, followed by K-9, then Focus. Our observation shows that at least two factors could explain these differences. Firstly, the functionality provided by an application could inspire discussions about a specific value theme. For instance, functionalities such as voice calls and group chat in Signal could trigger the

discussions of *pleasure* and *sense of belonging*. This finding is in line with a previous study [58] that argues different app categories have different types of NFRs. On the other hand, this also means that the list of value themes we discovered may not be exhaustive. There might be other value themes discussed during software development that we did not discover but present in other projects that have different functionalities. For instance, we did not find any themes that correspond to some values in Schwartz models (Fig. 1), such as *stimulation* or *achievement*, which may appear in different software categories, such as computer games or educational applications [59–61].

The second factor that we found is the emphasis on specific values (i.e., *values statement*) supported by an application. We observed that this statement instigates a higher expectation of the values from the app's users and contributors. For example, we discovered that Signal and Focus contributors frequently attempted to bring their privacy concerns in the discussions (see Section 4.2). These findings indicate that an app's functionalities and values statements can influence the values present in the discussions. We argue the need for further research to investigate the dedication and functionalities of an app as factors influencing the presence of human values.

5.6. Towards the development of an automated detection tool

The abundance of software development data in GitHub makes complete manual analysis of human values infeasible. Previous works [62, 63] have proposed automated techniques to analyse various aspects of software development from a vast amount of GitHub data. To the best of our knowledge, no automated tool exists to analyse the presence of value themes except for those well-known in SE in software development, such as *privacy* [64], *security* [18,19], or *energy preservation* [23]. This study can serve as an initial step to inform the future development of automated techniques. We argue that AI-based approaches (e.g., machine learning approaches) can be leveraged to detect issue discussions that include human values. Furthermore, such AI-based approaches can identify which human values are pointed out in issue discussions. For this purpose, the manually analysed discussions from this study [27] can be used in future research as a dataset to develop such automated approaches. Our criteria to discover value themes seem to correspond with the relation between human values and sentiment or emotion. For example, literature in social sciences believes that values are infused with feeling [6] and '*emotion is a source for values importance*' [50]. For this reason, one can suggest using sentiment analysis or the presence of emotions to identify the presence of a particular value. Future work could consider utilising sentiment analysis, as demonstrated in previous work [65–68] to better identify values in discussions.

An automation of values discovery in issue discussions can help stakeholders on the development side, such as product owners or developers, to gain insight into values that are important to users. For software practitioners, an automated tool can inform the value implications an issue can have. Thus, software practitioners can prioritise issues based on the values discovered. Applying the tool in the repositories will allow product owners to track the occurrence of values in issues across development phases. This application enables monitoring of whether values are discussed sufficiently early in development. From a software testing perspective, testers can pay more attention to those issues where values are discussed.

6. Threats to validity

A threat to **construct validity** comes from the definition of human values that have not been tested in software engineering. At the same time, there are also no theoretical considerations to believe that this definition does not transfer to SE. We mitigated this by not limiting the analysts only to consider Schwartz's basic values but also to provide descriptions of how the values look like in SE. All the persons involved in the analysis had a basic understanding of values theory. Moreover,

the values discovered in both the pilot and main study were validated and discussed to ensure there was no divergence of definitions across the analysts.

A threat to **internal validity** comes from the analysis process to discover values and the involvement of a single analyst in the main study. Although this can help increase the consistency of findings [69], the understanding of values and the subjectivity of the primary analyst may have affected the discovery of values. This issue was mitigated by performing a pilot study using an open coding approach used in other similar work [63,70] involving the primary analyst and five other analysts. In the main study, the primary analyst used the criteria and considered the initial values from the pilot study. Furthermore, three validators assessed and validated the primary analyst's results. Another possible threat to **internal validity** may come from the use of issue discussions as a single data source that may make the intention of an issue reporter not clear for the analysis. To mitigate this threat, we analysed the whole thread of the issue as the analysis unit. We argue that a thread of issue could provide a decent amount of information about contributors' concerns. The rationale behind this comes from the nature of the issue discussion, where contributors will ask for clarification from the reporter.

A threat to **external validity** arises from the limited number of projects we selected. All projects involved have a substantial number of issues, but we accept that the results may not be generalised to other projects. Due to our limitation, we only analysed a random portion of all the issues. We accept the possibility that the results may differ if we consider all the issues. Still, we mitigated this by using a statistical technique used in other similar work [71–73] to obtain a significant sample size.

7. Related work

7.1. Human values in discussion artefacts

Developer discussions have become an object study that may relate to some values, such as *security* and *energy preservation*. For example, Alqahtani and Rilling [18] propose an approach to discover and tag security concepts in open source issue reports. Another study investigated commit messages to understand how developers address energy consumption in Android applications [23]. Although related, these previous works may not discuss *security* and *energy preservation* as values. Even if we presume that these previous works considered values, only a limited number of values were discussed. In this study, we perceived those aspects as values and not limiting ourselves to specific values by including any human values found in developer discussions.

Emotions are considered as a motivational domain to *pleasure* value in the early version of the Schwartz values model [30]. Emotions are also considered as a source for values importance [50]. The presence of emotions has also become an object study in developer's discussions. For example, Ortu et al. [65] provided a manually labelled dataset of developer's emotions in issue comments. Another example, Werder and Brinkkemper [66] developed a tool to extract emotions from developer discussions in commits, pull requests, and issues from GitHub. Similarly, Ding et al. [67] developed a tool based to discover emotions in terms of developers' sentiment in issue comments. Another study investigated emotions identified in security discussions from commits and pull requests [68]. Those previous studies of emotions focused on developers' values in terms of how developers behave and interact during application development. The previous works' perspective is quite different from the perspective that we used. In this study, we used the *app-to-user* perspective that captures the users' concerns towards the application. Even though the perspective is different, the techniques presented in previous works on emotion detection have the potential to be utilised and extended to develop automated tools for the detection of human values.

7.2. Text analysis for human values

Analysis of human values has also been conducted on text documents. For example, Cheng et al. [74] analysed the Net neutrality debate to identify the values of each stakeholder involved. Their subsequent work [75] then emphasised how two values, namely *wealth* and *innovation*, are presents on the opposite side of the debate. Another study analysed human values in social media texts (i.e. Twitter) responding to several controversial topics and discovered how those values are expressed in this informal communications [76]. Similarly, Koepfler et al. [77] utilised frame analysis to discover human values in social media debates. Another recent work, Perera et al. [11] classified 1,350 publications in top SE venues to determine if a publication is addressing values or not. These previous works support the possibility to analyse text documents for human values. However, to the best of our knowledge, no study has analysed developer discussions as a means to understand human values consideration in software development.

7.3. Text analysis for non-functional requirements

In Section 5.4, we discussed that some values that we discovered in issue discussions are similar to non-functional requirements (NFRs). Regarding this, a number of studies were conducted to analyse and identify NFRs from software development artefacts. For example, Jha and Mahmoud [58] analysed user reviews of iOS applications and found 40% of the reviews have at least one type of NFRs. Another study was conducted by Zou et al. [78] discovered in Stack Overflow discussions that *usability* and *reliability* are the most unresolved and thus mostly discussed NFRs. These previous studies focused on app reviews and Stack Overflow discussions, leaving other artefacts, such as issue discussions, open for further investigations.

Some studies also developed automated techniques to identify NFRs from software development artefacts. For example, Tóth and Vidács [79] conducted experiments of automated NFRs classification using the PROMISE dataset. They discovered that Multinomial Naive Bayes gave the best performance among other classification methods implemented in the Scikit-Learn library. Abad et al. [80] discovered that implementing preprocessing techniques, such as POS, entity, and temporal tagging, could improve NFRs classifications' performance. They also proposed that the Binarised Naive Bayes method gave the best classification performance for the PROMISE dataset. Other studies developed automated techniques to classify NFRs on app reviews. For example, Lu et al. [81] proposed a feature extraction technique called augmented user reviews-bag of words (AUR-BoW) to improve NFRs classifications' performance. Another study by Jha et al. [58] proposed a dictionary-based classification technique to classify NFRs on app reviews. Although these automated techniques were developed for NFRs classification, these techniques can potentially be utilised to develop automated tools to detect human values.

8. Conclusions and future work

This paper performed a case study on 1,097 issue discussions collected from three Android projects to understand human values in software development artefacts. Our findings show that value themes could be found in issue discussions (33% of the inspected issues). We discovered 20 value themes comprising themes that directly corresponded to Schwartz's values (*human value themes*) and themes that could not be easily mapped to Schwartz's values (*system value themes*). The human value themes that we found could be mapped to seven out of 19 Schwartz's values. Additionally, we also propose a mapping between system value themes with Schwartz's refined values in the contexts of three applications. Finally, we proposed descriptions for each value theme to inform and empower software practitioners towards integrating human values during software development.

This study can also be considered as an initial study to identify factors that influence values in a software project. We found that the emphasis on certain values in a software project plays a role in the values' presence. Functionalities of the application may also contribute to the presence of some themes. For example, the sense of belonging theme was only found in an instant messaging application. Our study also contributes to a dataset [27] that has labelled values for 1,097 issues with 5,615 posts to develop automated value discovery tools.

The possibilities to find values in software development artefacts open up several directions for future work. For example, it opens up the possibilities to compare the level of attention to values on different projects. It also enables an opportunity for automation that allows larger-scale studies. Another direction can also investigate the presence of human values as part of the quality assurance of software.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to <https://doi.org/10.1016/j.infsof.2021.106731>.

References

- [1] Statista, Number of available applications in the Google Play Store from December 2009 to September 2020, 2020, <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>. (Accessed 05 December 2020).
- [2] H.-Y. Wang, C. Liao, L.-H. Yang, What affects mobile application use? The roles of consumption values, *Int. J. Mark. Stud.* 5 (2) (2013) <http://dx.doi.org/10.5539/ijms.v5n2p11>.
- [3] M.A. Harris, R. Brookshire, A.G. Chin, Identifying factors influencing consumers' intent to install mobile applications, *Int. J. Inf. Manage.* 36 (3) (2016) 441–450, <http://dx.doi.org/10.1016/j.jinfomgt.2016.02.004>.
- [4] B. Fu, J. Lin, L. Liy, C. Faloutsos, J. Hong, N. Sadeh, Why people hate your App - Making sense of user feedback in a mobile app store, in: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. Part F1288, 2013, pp. 1276–1284, <http://dx.doi.org/10.1145/2487575.2488202>.
- [5] A.S. Cheng, K.R. Fleischmann, Developing a meta-inventory of human values, in: *Proceedings of the ASIST Annual Meeting*, vol. 47, 2010, <http://dx.doi.org/10.1002/meet.14504701232>.
- [6] S.H. Schwartz, An overview of the Schwartz theory of basic values, *Online Read. Psychol. Cult.* 2 (1) (2012) 12–13, <http://dx.doi.org/10.9707/2307-0919.1116>.
- [7] H. Weisbaum, Trust in Facebook has dropped by 66 percent since the Cambridge Analytica scandal, 2018, <https://www.nbcnews.com/business/consumer/trust-facebook-has-dropped-51-percent-cambridge-analytica-scandal-n867011>. (Accessed 16 May 2019).
- [8] 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*, 2018, pp. 780–784, <http://dx.doi.org/10.1145/3236024.3264843>.
- [9] M.A. Ferrario, W. Simm, S. Forshaw, A. Gradinar, M.T. Smith, I. Smith, Values-first SE, in: *Proceedings of the 38th International Conference on Software Engineering Companion - ICSE '16*, 2016, pp. 553–562, <http://dx.doi.org/10.1145/2889160.2889219>.
- [10] 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.
- [11] 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: *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 409–420.
- [12] 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*, in: *ICSE-SEIS '20*, Association for Computing Machinery, New York, NY, USA, 2020, pp. 53–62, <http://dx.doi.org/10.1145/3377815.3381382>.
- [13] H. Perera, G. Mussbacher, W. Hussain, R. Ara Shams, A. Nurwidyantoro, J. Whittle, Continual human value analysis in software development: A goal model based approach, in: *2020 IEEE 28th International Requirements Engineering Conference (RE)*, 2020, pp. 192–203, <http://dx.doi.org/10.1109/RE48521.2020.00030>.
- [14] W. Hussain, H. Perera, J. Whittle, A. Nurwidyantoro, R. Hoda, R. Shams, G. Oliver, Human values in software engineering: Contrasting case studies of practice, *IEEE Trans. Softw. Eng.* 1 (5555) (2020) 1, <http://dx.doi.org/10.1109/TSE.2020.3038802>.
- [15] Langdon, Do artifacts have politics? *Tech. Rep.* 1, 1980, pp. 121–136.
- [16] S. Kim, J.I. Cho, H.W. Myeong, D.H. Lee, A study on static analysis model of mobile application for privacy protection, 2012, http://dx.doi.org/10.1007/978-94-007-2792-2_50.
- [17] C. Gibler, J. Crussell, J. Erickson, H. Chen, AndroidLeaks: Automatically detecting potential privacy leaks in android applications on a large scale, in: *International Conference on Trust and Trustworthy Computing*, in: *LNCS*, vol. 7344, 2012, pp. 291–307, http://dx.doi.org/10.1007/978-3-642-30921-2_17.
- [18] S.S. Alqahtani, J. Rilling, An ontology-based approach to automate tagging of software artifacts, in: *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2017, pp. 169–174, <http://dx.doi.org/10.1109/ESEM.2017.25>.
- [19] Y. Zhou, A. Sharma, Automated identification of security issues from commit messages and bug reports, in: *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, in: *ESEC/FSE 2017*, Association for Computing Machinery, New York, NY, USA, 2017, pp. 914–919, <http://dx.doi.org/10.1145/3106237.3117771>.
- [20] J. Viegas, J.T. Bloch, T. Kohno, G. McGraw, Token-based scanning of source code for security problems, *ACM Trans. Inf. Syst. Secur.* 5 (3) (2002) 238–261, <http://dx.doi.org/10.1145/545186.545188>.
- [21] F. Fischer, K. Bottinger, H. Xiao, C. Stransky, Y. Acar, M. Backes, S. Fahl, Stack overflow considered harmful? The impact of copy&paste on android application security, in: *Proceedings - IEEE Symposium on Security and Privacy*, 2017, pp. 121–136, <http://dx.doi.org/10.1109/SP.2017.31>.
- [22] M. Naseri, N.P. Borges, A. Zeller, R. Rouvoy, AccessiLeaks: investigating privacy leaks exposed by the android accessibility service, in: *Proceedings on Privacy Enhancing Technologies*, 2019, pp. 291–305, <http://dx.doi.org/10.2478/popets-2019-0031>.
- [23] L. Bao, D. Lo, X. Xia, X. Wang, C. Tian, How android app developers manage power consumption? in: *Proceedings of the 13th International Conference on Mining Software Repositories*, ACM, 2016, pp. 37–48, <http://dx.doi.org/10.1145/2901739.2901748>.
- [24] R. Pereira, T. Carcao, M. Couto, J. Cunha, J.P. Fernandes, J. Saraiva, Helping programmers improve the energy efficiency of source code, in: *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017*, 2017, pp. 238–240, <http://dx.doi.org/10.1109/ICSE-C.2017.80>.
- [25] L. Dabbish, C. Stuart, J. Tsay, J. Herbsleb, Social coding in GitHub: Transparency and collaboration in an open software repository, in: *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, 2012, pp. 1277–1286, <http://dx.doi.org/10.1145/2145204.2145396>.
- [26] T.F. Bissyande, D. Lo, L. Jiang, L. Reveillere, J. Klein, Y.L. Traon, Got issues? Who cares about it? A large scale investigation of issue trackers from GitHub, in: *2013 IEEE 24th International Symposium on Software Reliability Engineering, ISSRE 2013*, 2013, pp. 188–197, <http://dx.doi.org/10.1109/ISSRE.2013.6698918>.
- [27] Dataset of this study, <https://github.com/ovislabmonash/values-issues-dataset>. (Accessed 31 December 2020).
- [28] P.D. Hutcheon, Value theory: Towards conceptual clarification, *Br. J. Sociol.* 23 (2) (1972) 172–187.
- [29] M. Rokeach, *The Nature of Human Values*, Free Press, 1973.
- [30] S. Schwartz, W. Bilsky, Toward a theory of the universal content and structure of values: Extensions and cross-cultural replications, *J. Personal. Soc. Psychol.* 58 (5) (1990) 878–891.
- [31] S.H. Schwartz, The refined theory of basic values, in: *Values and Behavior: Taking a Cross Cultural Perspective*, 2017, pp. 51–72, http://dx.doi.org/10.1007/978-3-319-56352-7_3.
- [32] S.H. Schwartz, Are there universal aspects in the structure and contents of human values? *J. Social Issues* 50 (4) (1994) 19–45, <http://dx.doi.org/10.1111/j.1540-4560.1994.tb01196.x>.
- [33] J.A. Barceló, F.D.C. Bernal, R.D. Olmo, L. Mameli, F.M. Quesada, D. Poza, X. Vilà, Social interaction in hunter-gatherer societies: simulating the consequences of cooperation and social aggregation, *Soc. Sci. Comput. Rev.* 32 (3) (2014) 417–436.
- [34] M.S.H. Mukta, E.M. Khan, M.E. Ali, J. Mahmud, Predicting movie genre preferences from personality and values of social media users, in: *Eleventh International AAAI Conference on Web and Social Media*, 2017.
- [35] T. Holmes, E. Blackmore, R. Hawkins, T. Wakeford, *The Common Cause Handbook*, Public Interest Research Center, 2011.
- [36] S. Thew, A. Sutcliffe, Value-based requirements engineering: method and experience, *Requir. Eng.* 23 (4) (2018) 443–464, <http://dx.doi.org/10.1007/s00766-017-0273-y>.
- [37] B. Friedman, P.H. Kahn, Jr., A. Borning, Value sensitive design and information systems, in: *The Handbook of Information and Computer Ethics*, 2008, pp. 69–101.
- [38] B. Friedman, P.H. Kahn, A. Borning, A. Hultgren, Value sensitive design and information systems, in: *Early Engagement and New Technologies: Opening Up the Laboratory*, Springer, Dordrecht, 2013, pp. 55–95, http://dx.doi.org/10.1007/978-94-007-7844-3_4.

- [39] E. Winter, S. Forshaw, L. Hunt, M.A. Ferrario, Advancing the study of human values in software engineering, in: Proceedings - 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2019, 2019, pp. 19–26, <http://dx.doi.org/10.1109/CHASE.2019.00012>.
- [40] GitHub, About issues, URL <https://help.github.com/en/github/managing-your-work-on-github/about-issues>.
- [41] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D.M. German, D. Damian, The promises and perils of mining GitHub, in: Proceedings of the 11th Working Conference on Mining Software Repositories, 2014, pp. 92–101, <http://dx.doi.org/10.1145/2597073.2597074>.
- [42] StatCounter, 2020, URL <https://gs.statcounter.com/os-market-share/mobile/worldwide>.
- [43] M.H. Loorak, P.W. Fong, S. Carpendale, Papilio: Visualizing android application permissions, in: Computer Graphics Forum, vol. 33, Wiley Online Library, 2014, pp. 391–400.
- [44] K. Sokolova, C. Perez, M. Lemercier, Android application classification and anomaly detection with graph-based permission patterns, Decis. Support Syst. 93 (2017) 62–76.
- [45] O.W. Systems, Announcing the public beta, 2010, URL <https://web.archive.org/web/20100530011131/http://www.whispersys.com/updates.html>.
- [46] M. Marlinpike, Saying goodbye to encrypted SMS/mms, 2015, URL <https://signal.org/blog/goodbye-encrypted-sms/>.
- [47] V. Breitmoser, Openpgp considerations, part III: Autocrypt and encryption by default, 2018, URL <https://k9mail.github.io/2018/02/26/OpenPGP-Considerations-Part-III-Autocrypt.html>.
- [48] A. Wyman, et al., What is firefox focus? 2020, URL <http://mzl.la/1NDD2IB>.
- [49] E. Winter, S. Forshaw, M.A. Ferrario, Measuring human values in software engineering, in: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2018, pp. 8–11, <http://dx.doi.org/10.1145/3239235.3267427>.
- [50] G.R. Maio, Mental representations of social values, in: Advances in Experimental Social Psychology, vol. 42, Elsevier, 2010, pp. 1–43.
- [51] S. Adolph, W. Hall, P. Kruchten, A methodological leg to stand on: Lessons learned using grounded theory to study software development, in: Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds, 2008, pp. 13:166–13:178, <http://dx.doi.org/10.1007/s10664-010-9152-6>.
- [52] D. Mairiza, D. Zowghi, N. Nurmiliani, An investigation into the notion of non-functional requirements, in: Proceedings of the ACM Symposium on Applied Computing, 2010, pp. 311–317, <http://dx.doi.org/10.1145/1774088.1774153>.
- [53] P.H. Hanel, K.C. Vione, U. Hahn, G.R. Maio, Value instantiations: The missing link between values and behavior? in: Values and Behavior: Taking a Cross Cultural Perspective, 2017, pp. 175–190, http://dx.doi.org/10.1007/978-3-319-56352-7_8.
- [54] P.H. Hanel, G.R. Maio, A.K. Soares, K.C. Vione, G.L.H. 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 (MAY) (2018) 1–13, <http://dx.doi.org/10.3389/fpsyg.2018.00849>.
- [55] B. Friedman, P.H. Kahn, Jr., P.H. Kahn Jr., Human values, ethics, and design, in: The Human-Computer Interaction Handbook, 2003, pp. 1177–1209, URL <https://faculty.washington.edu/pkahn/articles/friedman03humanvalues.pdf>.
- [56] M. Glinz, On non-functional requirements, in: Proceedings - 15th IEEE International Requirements Engineering Conference, RE 2007, 2007, 21–26, <http://dx.doi.org/10.1109/RE.2007.45>.
- [57] B.S. Barn, Do you own a volkswagen? Values as non-functional requirements, in: Joint 6th International Working Conference on Human-Centred Software Engineering and 8th International Working Conference on Human Error, Safety, and System Development, 2016, http://dx.doi.org/10.1007/978-3-319-44902-9_10.
- [58] N. Jha, A. Mahmoud, Mining non-functional requirements from App store reviews, Empir. Softw. Eng. 24 (6) (2019) 3659–3695, <http://dx.doi.org/10.1007/s10664-019-09716-7>.
- [59] S. Aslan, O. Balci, GAMED: Digital educational game development methodology, Simulation 91 (4) (2015) 307–319, <http://dx.doi.org/10.1177/0037549715572673>.
- [60] H. Mahmoudi, M. Koushfar, J.A. Saribagloo, G. Pashavi, The effect of computer games on speed, attention and consistency of learning mathematics among students, Procedia - Soc. Behav. Sci. 176 (2015) 419–424, <http://dx.doi.org/10.1016/j.sbspro.2015.01.491>.
- [61] C.H. Tsai, C.H. Cheng, D.Y. Yeh, S.Y. Lin, Can learning motivation predict learning achievement? A case study of a mobile game-based English learning approach, Educ. Inf. Technol. 22 (5) (2017) 2159–2173, <http://dx.doi.org/10.1007/s10639-016-9542-5>.
- [62] Q. Huang, X. Xia, D. Lo, G.C. Murphy, Automating intention mining, IEEE Trans. Softw. Eng. (2018) <http://dx.doi.org/10.1109/TSE.2018.2876340>.
- [63] G. Viviani, C. Janik-Jones, M. Famelis, X. Xia, G.C. Murphy, What design topics do developers discuss? in: Proceedings of the 26th Conference on Program Comprehension, ACM, 2018, pp. 328–331, <http://dx.doi.org/10.1145/3196321.3196357>.
- [64] R. Slavin, X. Wang, M.B. Hosseini, J. Hester, R. Krishnan, J. Bhatia, T.D. Breaux, J. Niu, Toward a framework for detecting privacy policy violations in android application code, in: 38th IEEE International Conference on Software Engineering, 2016, pp. 25–36, <http://dx.doi.org/10.1145/2884781.2884855>.
- [65] M. Ortu, A. Murgia, G. Destefanis, P. Tourani, R. Tonelli, M. Marchesi, B. Adams, The emotional side of software developers in JIRA, in: Proceedings of the 13th International Conference on Mining Software Repositories, 2016, pp. 480–483, <http://dx.doi.org/10.1145/2901739.2903505>.
- [66] K. Werder, S. Brinkkemper, MEME – toward a method for emotions extraction from GitHub Karl, in: ACM/IEEE 3rd International Workshop on Emotion Awareness in Software Engineering, 2018, pp. 20–24, <http://dx.doi.org/10.2307/j.ctvt0023.22>.
- [67] J. Ding, H. Sun, X. Wang, X. Liu, Entity-level sentiment analysis of issue comments, in: IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering, vol. 18, ACM, 2018, pp. 7–13, <http://dx.doi.org/10.1145/3194932.3194935>.
- [68] D. Pletea, B. Vasilescu, A. Serebrenik, Security and emotion: sentiment analysis of security discussions on GitHub, in: Proceedings of the 11th Working Conference on Mining Software Repositories, 2014, pp. 348–351, <http://dx.doi.org/10.1145/2597073.2597117>.
- [69] K.F. Tomasdotir, M. Aniche, A. Van Deursen, Why and how JavaScript developers use linters, in: ASE 2017 - Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, APA, 2017, pp. 578–589, <http://dx.doi.org/10.1109/ASE.2017.8115668>.
- [70] G.A. Prana, C. Treude, F. Thung, T. Atapattu, D. Lo, Categorizing the content of GitHub README files, Empir. Softw. Eng. (2018) <http://dx.doi.org/10.1007/s10664-018-9660-3>, arXiv:1802.06997v2.
- [71] Y. Ma, S. Fakhoury, M. Christensen, V. Arnaoudova, W. Zogaan, M. Mirakhorli, Automatic classification of software artifacts in open-source applications, in: Proceedings - International Conference on Software Engineering, 2018, pp. 414–425, <http://dx.doi.org/10.1145/3196398.3196446>.
- [72] P. Chatterjee, K. Damevski, L. Pollock, V. Augustine, N.A. Kraft, Exploratory study of slack Q&A chats as a mining source for software engineering tools, in: IEEE International Working Conference on Mining Software Repositories, 2019-May, 2019, pp. 490–501, <http://dx.doi.org/10.1109/MSR.2019.00075>.
- [73] S. Wang, N. Phan, Y. Wang, Y. Zhao, Extracting API tips from developer question and answer websites, in: IEEE International Working Conference on Mining Software Repositories, 2019-May, IEEE, 2019, pp. 321–332, <http://dx.doi.org/10.1109/MSR.2019.00058>.
- [74] A.S. Cheng, K.R. Fleischmann, P. Wang, E. Ishita, D.W. Oard, Values of stakeholders in the net neutrality debate: Applying content analysis to telecommunications policy, in: 2010 43rd Hawaii International Conference on System Sciences, 2010, <http://dx.doi.org/10.1109/HICSS.2010.434>.
- [75] A.S. Cheng, K.R. Fleischmann, P. Wang, E. Ishita, D.W. Oard, The role of innovation and wealth in the net neutrality debate: A content analysis of human values in congressional and FCC hearings, J. Am. Soc. Inf. Sci. Technol. 63 (7) (2012) 1360–1373, <http://dx.doi.org/10.1002/asi.22646>.
- [76] J.A. Koepfler, K.R. Fleischmann, Classifying values in informal communication: Adapting the meta-inventory of human values for tweets, in: Proceedings of the American Society for Information Science and Technology, vol. 48, 2011, pp. 1–4, <http://dx.doi.org/10.1002/meet.2011.14504801116>.
- [77] J.A. Koepfler, T.C. Templeton, K.R. Fleischmann, Exploration of values and frames in social media texts related to the Homeless Hotspots debate, in: Proceedings of the American Society for Information Science and Technology, vol. 49, 2012, pp. 1–4, <http://dx.doi.org/10.1002/meet.14504901238>.
- [78] J. Zou, L. Xu, M. Yang, X. Zhang, D. Yang, Towards comprehending the non-functional requirements through developers' eyes: An exploration of stack overflow using topic analysis, Inf. Softw. Technol. 84 (2017) 19–32, <http://dx.doi.org/10.1016/j.infsof.2016.12.003>.
- [79] L. Tóth, L. Vidács, Study of various classifiers for identification and classification of non-functional requirements, in: O. Gervasi, B. Murgante, S. Misra, E. Stankova, C.M. Torre, A.M.A. Rocha, D. Tanian, B.O. Apduhan, E. Tarantino, Y. Ryu (Eds.), Computational Science and Its Applications – ICCSA 2018, Springer International Publishing, Cham, 2018, pp. 492–503.
- [80] Z.S.H. Abad, O. Karras, P. Ghazi, M. Glinz, G. Ruhe, K. Schneider, What Works Better? A study of classifying requirements, in: Proceedings - 2017 IEEE 25th International Requirements Engineering Conference, RE 2017, 2017, pp. 496–501, arXiv:1707.02358, <http://dx.doi.org/10.1109/RE.2017.36>.
- [81] M. Lu, P. Liang, Automatic classification of non-functional requirements from augmented app user reviews, in: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, 2017, pp. 344–353.