Empirical Software Engineering (2023) 28:67
https://doi.org/10.1007/s10664-023-10305-y

Integrating human values in software development using a human values dashboard

Arif Nurwidyantoro[1,2] · Mojtaba Shahin[3] · Michel Chaudron[4] · Waqar Hussain[1,5] · Harsha Perera[5] · Rifat Ara Shams[5] · Jon Whittle[5]

Accepted: 6 February 2023 / Published online: 18 April 2023

Abstract

There is a growing awareness of the importance of human values in software systems. However, limited tools are available to support the integration of human values during software development. Most of these tools are focused on concepts related to specific, well-known human values (e.g., privacy, security) in software engineering. This paper aims to (partially) address this gap by developing a human values dashboard. We conducted a multi-stage study to design, implement and evaluate a human values dashboard. First, an exploratory study was conducted by interviewing 15 software practitioners to investigate the possibility of using a human values dashboard to help address human values in software development, its potential benefits, and required features. Second, we experimented with four Machine Learning approaches to detect the presence of human values in issue discussions. We used the best approach to develop a human values dashboard for software development. The dashboard displays whether any human values are present in each issue discussion. Finally, we interviewed ten different practitioners to investigate the usefulness of the dashboard in practice. This study found that the human values dashboard could help raise awareness, focus attention, and prioritise issues based on the presence of values. This study also identified two potential challenges to the adoption of the dashboard. First, the possible incorrect issues description that can mislead the automated values identification in the dashboard. Second, the lack of willingness of a company to adopt the dashboard.

Keywords Human values · GitHub · Issue discussion · Dashboard

This article belongs to the Topical Collection: *Special Issue on International Symposium on Empirical Software Engineering and Measurement (ESEM)*

Communicated by: Maria Teresa Baldassarre and Markos Kalinowski

✉ Arif Nurwidyantoro
arif.nurwidyantoro@monash.edu; arifn@ugm.ac.id

Extended author information available on the last page of the article.

1 Introduction

Human values such as inclusiveness, social justice, and privacy, or *'what people hold important in their life'* (Schwartz 2012; Rokeach 1973), have received increasing attention in the last few years in the software industry. Recent incidents have demonstrated people's awareness of their values and how they strongly react to the violation of their values in software. For instance, changes in WhatsApp's privacy policy in early 2021 have led millions of its users to migrate to alternative messaging apps (Best 2021). One of the reasons was fear of another privacy breach by WhatsApp's parent company, Facebook, which will have access to user data after the new policy comes into play (Best 2021). In this case, the trust of Facebook users did not appear to have recovered after the infamous Cambridge Analytica case in 2015 (Confessore 2018). Another example is that digital and human rights groups protested the use of facial recognition systems in justice systems that introduce racial bias (Schapiro and Bacchi 2020). This bias was suspected to come from the use of datasets in the recognition systems that underrepresent minorities (Schapiro and Bacchi 2020). The bias caused people of colour to be more likely to be detected as offenders and increased fear of unfair treatments (Schapiro and Bacchi 2020). These events are aligned with a characteristic of human values where people feel threatened when their values are jeopardised (Schwartz 2012). To avoid these situations, considering human values, i.e., being attentive to the implication of human values, in an application or software is necessary because it could influence the acceptance of users (Wang et al. 2013; Harris et al. 2016; Fu et al. 2013). In this paper, we used the terms *human values dashboard* and *values dashboard* interchangeably.

Addressing human values is difficult because of their subjective nature (Winter et al. 2018) and their definition depends on the context in which they are applied (Kujala and Väänänen-Vainio-Mattila 2009; Mougouei et al. 2018). Several solutions have been proposed to support practitioners in addressing values in software. These solutions are commonly in the form of frameworks, techniques, practises, and guidelines, such as Value-based Requirements Engineering (Thew and Sutcliffe 2018), Value Sensitive Design (Friedman et al. 2008; Friedman et al. 2013), or Continual Value(s) Assessment (Perera et al. 2020). However, these solutions aim to consider values at a specific phase of software development, such as requirements or design phase, or satisfy a specific type of practitioner (e.g. designer). We argued that providing a *human values dashboard* can bridge this gap and help practitioners effectively and efficiently understand and handle values in the software development lifecycle.

In software development, dashboards are commonly used to support decision making (Ivanov et al. 2018a, b), promote awareness within a project (Treude and Storey 2010; Baysal et al. 2013), and monitor development activities (Leite et al. 2015). It is common for a software development dashboard to use software developmeent artefacts as its source, because these artefacts capture software development process. For instance, Leite et al. developed a dashboard that used commit history to detect unusual events (Leite et al. 2015). Several other dashboards have also been developed using artefacts from software repositories (GitHub 2021a,b; Cauldron 2021; Mautic 2022). Recent studies also suggested that human values, security (Fischer et al. 2017; Viega et al. 2002; Pletea et al. 2014; Alqahtani and Rilling 2017), privacy (Kim et al. 2012; Li et al. 2015; Gibler et al. 2012; Naseri et al. 2019; Kuznetsov et al. 2016; Slavin et al. 2016; Sharma et al. 2014), or energy efficiency (Bao et al. 2016; Pereira et al. 2017), can be found in software development artefacts. Although these works did not specifically address security, privacy, or energy efficiency as values, they show the possibility of discovering values in the artefacts. A dashboard is suitable for our purpose because it allows information to be visually displayed at *'facilitate*

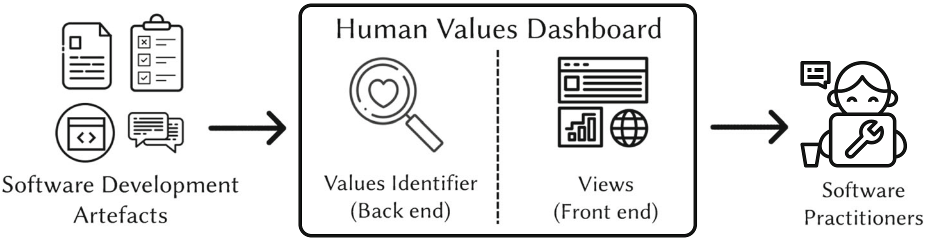
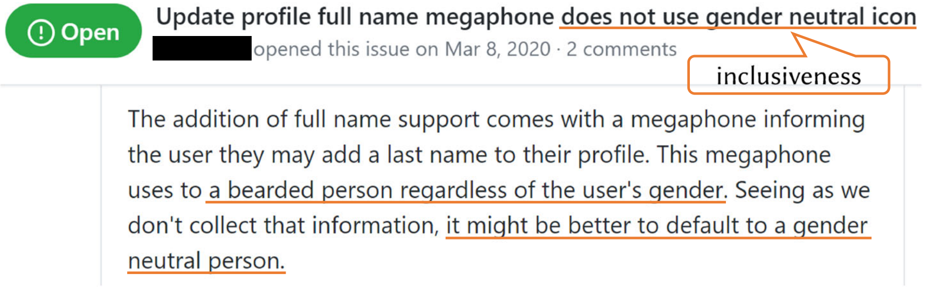


Fig. 1 Proposed human values dashboard

understanding' (Wexler et al. 2017). We believe that a dashboard can help clarify the less known and abstract concept of values (Mougouei et al. 2018; Perera et al. 2020) for software practitioners.

Figure 1 presents our vision of a human values dashboard that uses software development artefacts as its data source and displays the values identified in the artefacts to support practitioners in addressing those values in the software. To this end, we propose a human values dashboard consisting of a back end and a front end. The back end of the dashboard provides functionality to identify values from software development artefacts. The identification of values could be done manually (e.g. by the development team) or using an automated approach. The back end is necessary because these artefacts naturally do not have values identified in them yet. For example, Fig. 2 shows a user of an open source application expresses his/her opinion of *inclusiveness* to be present in the application in an issue discussion. Based on this example, we define a human value can be identified in a software development artefact if *there is a notion of that value in the artefact*. This argument is supported by a recent work that discovered human values in issue discussions as an example of software development artefacts (Nurwidyantoro et al. 2021b). The front end of the dashboard displays values identified from various artefacts in different views (for different roles).

This study to design, implement, and evaluate a human values dashboard consists of three stages, shown in Fig. 3. First, we conducted an exploratory study by developing a prototype of the dashboard and interviewed software practitioners to obtain their perceptions and what is necessary for such a dashboard (i.e., exploratory stage). Second, we developed a human values dashboard as a proof-of-concept based on the findings of the exploratory study (i.e.,



Update profile full name megaphone does not use gender neutral icon

[redacted] opened this issue on Mar 8, 2020 · 2 comments

The addition of full name support comes with a megaphone informing the user they may add a last name to their profile. This megaphone uses to a bearded person regardless of the user's gender. Seeing as we don't collect that information, it might be better to default to a gender neutral person.

Fig. 2 Example of values identified in an issue discussion

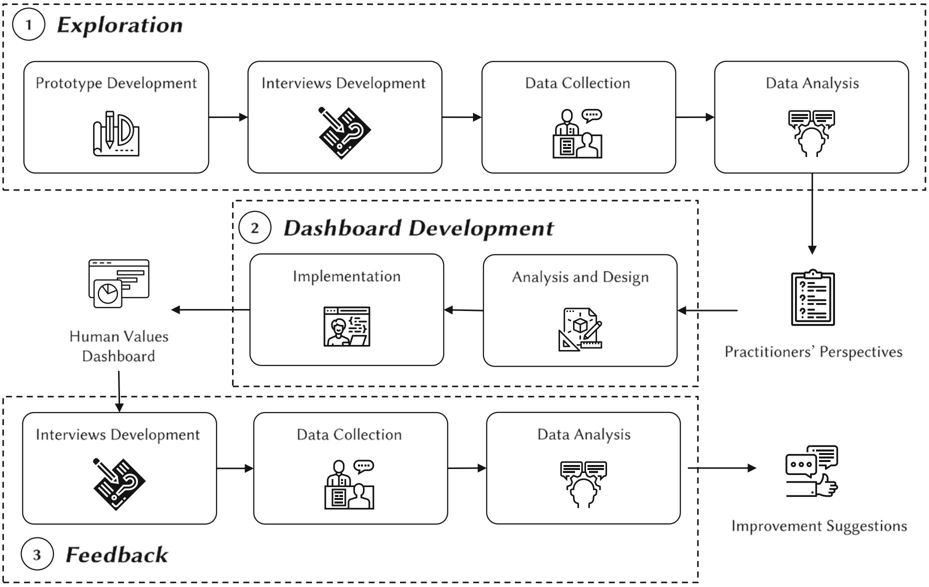


Fig. 3 Methodology of this study

dashboard development stage). Finally, we presented the human values dashboard to software practitioners and asked for their feedback and suggestions to improve it (i.e., feedback stage).

Our results reveal that the participants agree that human values are important to be considered in software, e.g., developing an application that is aligned with its users' cultural background. Participants also believe that a values dashboard can benefit various software development roles (e.g. *project manager*, *developer*, *tester*), primarily to raise awareness of values and support values-based decision making in project management (e.g. prioritising issues). Our participants also acknowledge that software development artefacts are suitable as a source for the dashboard. Among those artefacts, requirements documents and issue discussions are deemed to be the most suitable. This study also identified two potential challenges for the adoption of the dashboard. First, the possible incorrect issues description that can mislead the automated values identification in the dashboard. Second, the lack of willingness of a company to adopt the dashboard.

We have previously reported the design and findings of the exploratory stage of our research at the ESEM conference in 2021 (Nurwidyantoro et al. 2021a). This paper significantly extends the previously published work by adding the design and development of the human values dashboard and its components. We also conducted 10 further interviews (i.e., feedback stage) to see practitioners' perspectives on the usefulness of the dashboard in practice.

The remaining sections of this paper are organised as follows. Section 2 summarises the background of this study. The next three sections describe the methodological details of each stage in this study, namely, the exploration stage, the dashboard development stage, and the feedback stage. Section 3 describes the exploration stage to understand the potential benefits and what is necessary for a human values dashboard. Section 4 explains the development of the human values dashboard. Section 5 describes the feedback stage to obtain practitioners' perceptions and suggestions toward the dashboard. Section 6 discusses the findings and potential future directions. Section 7 describes the threats to the validity of this study. Finally, Section 8 concludes the paper and proposes future work for this study.

2 Background

This section introduces the theoretical background and previous work related to this study. First, we describe the definition of human values and their model in social science. Second, we present related work on human values in software. Finally, we present previous work on the dashboard for software development.

2.1 Human Values

Human values, such as *achievement* or *benevolence*, are defined by Schwartz as '*things that people hold important in their life*' (Schwartz 2012). Meanwhile, Rokeach defined values as '*a belief that a particular way of doing something is personally or socially preferable to the opposite ways*' (Rokeach 1973). Studies in social sciences suggested a degree of relative importance between these values for each person (Rokeach 1973; Schwartz 2012; 2017). Because values are intertwined with feelings (Schwartz 2012), a threat to someone's values can upset that person. Otherwise, consideration of values will bring enjoyment for a person. For example, in a software engineering context, a user of an application who values inclusiveness can expect this value to be present in the application (Fig. 2).

Social sciences have proposed several models that identified human values and divided them into several categories (Rokeach 1973; Bird 1998; Cheng et al. 2010; Schwartz 2012; 2017). Among those models, Schwartz's model (Schwartz 2012) is considered the most complete as it covers the largest number of values compared to other models (Cheng et al. 2010). Schwartz's model, also known as, Schwartz's theory of basic values, categorises human values into 10 types based on their motivation. These types are self direction, universalism, benevolence, conformity, tradition, security, power, achievement, hedonism, and stimulation (Schwartz 2012). Schwartz also provides *'exemplary specific values that primarily represent each value type'* (Schwartz 1994) or *value items* (Schwartz 2012), such as *privacy*, *choosing own goals*, and *equality* for self direction values. This model is organised in a circular manner so that the supporting values are adjacent to each other, while the conflicting values are diametrically opposite to each other (Schwartz 2012) (see Fig. 4). For example, pursuing freedom could introduce conflict to the value of respecting tradition. In this study, we used Schwartz's model to introduce values to software practitioners during data collection.

2.2 Human values in SE

2.2.1 Solutions to integrate values in software

Several solutions have been proposed to integrate human values into software engineering. For example, Value-based Requirement Engineering (VBRE) was introduced for the requirements engineering stage to elicit values from users and stakeholders (Thew and Sutcliffe 2018). Another approach called Value Sensitive Design (VSD) (Friedman et al. 2008; Friedman et al. 2013) was proposed to integrate the consideration of values into the design

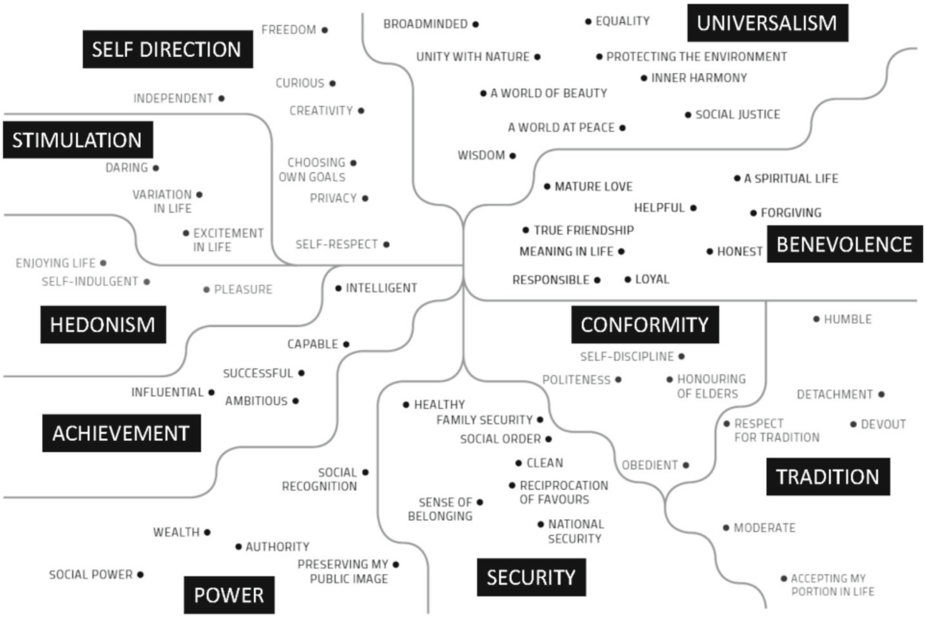


Fig. 4 The Schwartz models of basic human values (Schwartz 2012) taken from (Holmes et al. 2011)

process of a system. A framework called Continual Value(s) Assessment (CVA) was also proposed to extend a set of an application's functionalities based on an evaluation of value implications of the existing functionalities (Perera et al. 2020).

These solutions proposed values to be considered in specific stages of software development, especially in the early stages, such as requirements and design. We believe it is possible to support the integration of values in the later stages of the development (e.g. implementation). For example, Hussain et al. have identified several places to introduce values throughout the software development phases in the SAFe Agile framework (Hussain et al. 2022). In addition, these works proposed solutions as methods or frameworks. We argued that to be practical, a solution could also be in the form of a tool. Meanwhile, not so many studies have proposed a tool to support values in software. Our study addressed this gap by envisioning a dashboard as a solution. We believe that our idea of a human values dashboard has the potential to support various stages of software development by utilising artefacts generated during software development as its data source.

2.2.2 Human values in software development artefacts

Software development activities normally generate artefacts. For instance, requirements documents are written as a result of requirements-gathering activities. Development teams also discuss an issue report within repositories. These artefacts have been used in previous work to investigate human values. Recent studies mainly considered more familiar values in software engineering, such as *security*, *privacy*, and *energy efficiency*. For example, some studies have investigated the notion of security in source codes (Fischer et al. 2017; Viega et al. 2002) and issue discussions (Pletea et al. 2014; Alqahtani and Rilling 2017). Privacy has received a lot of attention through several investigations on various artefacts, such as source code and configuration files (Kim et al. 2012; Li et al. 2015; Gibler et al. 2012; Naseri et al. 2019; Kuznetsov et al. 2016), application programming interfaces (API) (Slavin et al. 2016), and project documentation (Sharma et al. 2014). Other studies focused on the energy efficiency of an application (Bao et al. 2016; Pereira et al. 2017). Although these solutions are related, they do not specifically consider security, privacy, or energy efficiency as values. As a complement to previous work, a recent study has demonstrated that human values as defined in the social sciences are present in the issue discussion artefact (Nurwidyantoro et al. 2021b). These studies support our idea that values are present in software development artefacts. Therefore, development artefacts can be suitable as a data source of a values dashboard.

2.3 Dashboard for software development

A dashboard is generally used to monitor progress (Wexler et al. 2017) and support decision making (Janes et al. 2013) in an organisation. In software development, recent studies have demonstrated the use of a dashboard to make decisions (Ivanov et al. 2018a, b) and to promote awareness of the software project to the development team (Treude and Storey 2010; Baysal et al. 2013). For example, Leite et al. proposed a dashboard to alert developers to unusual events in repositories (Leite et al. 2015). Another study used a dashboard to visualise concerns in the context of software evolution (Treude and Storey 2009). In practise, software projects use dashboards during development to monitor the development activities of a project (Cauldron 2021; Mautic 2022; GitHub 2021b; 2021a). In this study, our objective was to use the benefits of promoting awareness, in our case awareness of values, during software development for practitioners.

Our study proposes a human values dashboard to promote awareness of values in software development. In this regard, recent work has developed dashboards that include various indicators to support awareness about the software project, such as code quality and non-blocking code (Lòpez et al. 2021), project size, issue density, and productivity (Thiruvathukal et al. 2018). Unlike our work, these recent works focused more on the technical aspects of the software. For non-technical aspects, other work proposed dashboards, not in software contexts, for online discussions. These works proposed dashboards to visualise team dynamics (Vivian et al. 2015) and provide suggestions for inclusive meetings (Samrose and McDuf 2021). Unlike these previous works, our dashboard highlights human values, as a non-technical aspect, in software development.

In terms of evaluation of a dashboard, previous work primarily used interviews (e.g., (Ivanov et al. 2018b; Baysal et al. 2013; Leite et al. 2015; Samrose and McDuf 2021)). Some of these studies asked participants to interact with the dashboard before interviews (Ivanov et al. 2018b; Samrose and McDuf 2021). Other works used surveys or questionnaires to evaluate their dashboard (e.g., (Lòpez et al. 2021)) or in combination with interviews (e.g., (Treude and Storey 2009)). This study followed the approach of allowing participants to interact with the dashboard followed by interviews.

3 Exploration stage

This stage aims to explore whether our envisioned human values dashboard would be useful to support the consideration of human values during software development. To understand this, first, it is necessary to understand whether software practitioners consider human values important. Second, it is necessary to explore the possible benefits of that tool for different roles in software development. As the dashboard uses software development artefacts as its source, it is also important to understand which artefacts are considered by practitioners to be the most suitable. Finally, the dashboard was intended to help software practitioners in incorporating human values during software development. Therefore, it is also necessary to obtain requirements from practitioners for the dashboard. Based on these, the following research questions were developed:

RQ1 What are the perceptions of practitioners towards human values in software development?
RQ2 Who will benefit from and what is the benefit of a human values dashboard?
RQ3 Which artefacts are suitable for the dashboard?
RQ4 What is needed for a human values dashboard to be helpful in software development?

In this stage, we first developed a prototype of our visioned human values dashboard. The prototype in this stage was developed using static HTML that presents manually values-labelled issues in three different views. The labelling process was carried out by the authors following the methods presented in Nurwidyantoro et al. (2021b). Then we conducted interviews with 15 software practitioners (i.e., P01 - P015). The interview questions for this stage are available in Nurwidyantoro et al. (2022a). Finally, we analysed the interviews to address the research questions. We used the thematic analysis approach (Braun and Clarke 2012) to analyse the interviews in this stage and, later, in the feedback stage (Section 5). Table 1 shows the analysis of parts of interview transcripts to themes and sub-themes. The first and second examples are from the exploration stage interview and the third example is from the feedback stage interview. This stage of the study has been published in Nurwidyantoro et al. (2021a). The remainder of this section summarises the findings for the exploration stage.

3.1 Practitioners' Perceptions of Human Values (RQ1)

Participants indicated that human values are important in software. However, their understanding of human values is limited to those that are well known in software engineering, such as security or privacy. The participants found that other values, such as achievement or ability, are not easily understood to be translated into software engineering. However, they argued that they have considered some human values in their software development

Table 1 Examples of the analysis from interviews to a theme and a sub theme

#	Quote	Theme	Sub Theme
1	'I am **a bit unsure** about this area of the achievements and capable **means**'. (P01 - Developer)	Perspectives on human values	Unfamiliar values
2	'Some projects, accessibility will be the number one priority' (P08 - UI Designer)	Perspectives on human values	Relative importance
3	'sometimes the management or the [project] plan, and does not bother [with] that type of issues (P19 - Developer)	Challenges in adopting the Dashboard	Reasons against the Dashboard

activities. For example, an application is developed by following the cultural background of its users. In terms of the importance of human values, participants believed that it depends on the nature of the software being developed. However, in general, some values, such as security and privacy, are always more important regardless of the functionalities of the software.

3.2 Benefits of a Human Values Dashboard (RQ2)

The participants suggested that a human values dashboard can benefit all roles in software development in several stages of software development. A human values dashboard can be used, especially, to determine values-driven priorities in a project and to raise awareness of values within the software development team. For example, the project manager could use the dashboard to discuss the project values priorities with the product owner. Another example is that the dashboard could also help other roles, such as requirements engineers or developers, to be aware of the existence of values that need to be addressed in their tasks. In open source projects, information on the presence of human values could inform users of the project to assess whether they would like to use the application, i.e., the values present is aligned with their values.

3.3 Artefacts as the Datasource of the Dashboard (RQ3)

The participants proposed several artefacts, namely market research documents, requirements documents, design documents, features specification documents, issue discussions, and pull request discussions, as potential artefacts suitable for a human values dashboard. The participants chose these artefacts for the following reasons: (a) values can be potentially identified within these artefacts; and (b) these artefacts are used and referred to during software development. Additionally, among these artefacts, the participants identified the requirements documents and issue discussions as the most suitable sources for the dashboard.

3.4 Requirements for a Human Values Dashboard

The participants suggested six high-level requirements that are necessary for a human values dashboard, shown in Table 2. First, the dashboard should be able to identify the presence of human values within the artefacts automatically. Second, the dashboard should also be able to refer the identified values to the artefact source. The third and fourth requirements are about determining the values priority and displaying the artefacts based on the priority. The last two requirements are more related to the development of a project, where the updates and different views concerning values in the artefacts are presented in the dashboard.

4 Dashboard Development Stage

This stage involves developing a human values dashboard as a proof of concept. The results of the exploration stage were considered during the development of the dashboard. The dashboard development used the prototype views (Figure 4 in Nurwidyantoro et al. (2021a) as the basis. This stage began with designing the components and functionality of the dashboard. Subsequently, we evaluated machine learning techniques to automate the detection of human values as a component of the dashboard (see Table 2). The dashboard was then implemented and populated with an artefact of open-source projects hosted on GitHub.

4.1 Analysis and Design

The analysis considered the perspectives of the practitioners on what is required of a human values dashboard in the exploration stage (Table 2). The analysis of those high-level requirements is described as follows:

R1 **The identification of values in the dashboard shall be conducted automatically**. To address this requirement, a human values detector is used as a component in the back end of the dashboard. To support automatic detection, the human values detector utilised machine learning models. The experiments to determine which machine learning models used is presented in Section 4.2.

R2 **The dashboard should maintain the traceability between the identified values and their artefact source**. This requirement was addressed by storing the web page URLs of the artefacts in a database. These URLs would be displayed in the front end along with the artefacts. Using this approach, practitioners could use the URLs to refer to the actual location of the artefacts in the repository.

R3 **The dashboard shall allow the development team to determine the values priority of a project**. The machine learning model presented in Section 4.2 was used to detect the presence of human values in artefacts. At this time, this model is unable to detect the presence of specific values (e.g. privacy or inclusiveness). Due to this limitation, the dashboard will only display the presence of the general human values.

R4 **The dashboard shall display the artefacts based on the values priority determined in a project**. As explained in the previous requirement, the presence of any human values was assumed to be the priority. To address this requirement, the dashboard provided a filtering mechanism on the front end. This filtering allows the dashboard to display only the issues that had been identified that have values present. To inform the latest update on the artefacts, the dashboard displayed the date and time the artefacts were reported and closed. The dashboard also has notifications to inform users when the human values detector found human values in an artefact.

R5 **The dashboard shall provide different views for various roles to support addressing values in software development**. In the dashboard prototype (Nurwidyantoro et al. 2021a), the dashboard provided three views for various roles in software development. The development of the dashboard included these views with some adjustments based on the availability of the artefacts (i.e. issue discussions) and the capability of the human values detector (i.e. in detecting whether any human values were present).

Table 2 Proposed requirements for the dashboard

Requirements	
R1	The identification of values in the dashboard shall be conducted automatically.
R2	It should maintain the traceability between the identified values and their artefact source.
R3	It shall allow the development team to determine the values priority of a project.
R4	It shall display the artefacts based on the values priority determined in a project.
R5	It shall inform the latest update on the artefacts where values are identified in a project.
R6	It shall provide different views for various roles to support addressing values in software development.

After considering the high-level requirements from the practitioners in the exploration stage, this stage continued with designing the components of the human values dashboard. Similarly to the prototype, the human values dashboard was designed to have a back end and a front end. The back end provides an automated downloading of artefacts from project repositories and automated labelling of human values in the artefacts. The dashboard's front end provides three views similar to those on the previous prototype with some adjustments based on the practitioners' suggestions in the exploration stage. Figure 5 shows the components of the human values dashboard. The first component on the back end, **artefacts downloader**, allows development teams to specify repository URLs and download the corresponding artefacts. The downloaded artefacts are stored in the **database**. The **human values detector** could then be used to automatically detect the presence of human values in the downloaded artefacts. It uses **pre-trained models** from the human value detection experiments (Section 4.2) and stores the results in the database mentioned above. The **views** in the front end provides visualisations of the detection results and their corresponding artefacts.

To download artefacts, a software practitioner specifies the project and artefact they want to download. Then, the **artefacts downloader** connects to the project repository via the GitHub API and downloads the specified artefacts to the dashboard database. For this study, we chose issue discussions for the dashboard source. We made this decision for the reason that it is one of the artefacts suggested by our participants in the exploration stage and also supported by previous work (Fischer et al. 2017; Viega et al. 2002; Pletea et al. 2014; Alqahtani and Rilling 2017; Kim et al. 2012; Li et al. 2015; Gibler et al. 2012; Naseri et al. 2019; Kuznetsov et al. 2016; Slavin et al. 2016; Sharma et al. 2014; Bao et al. 2016; Pereira et al. 2017; Nurwidyantoro et al. 2021b). Thus, from this process, the database stores the project information, issues, and corresponding posts. To obtain the results of values detection in the issues, a practitioner runs the **human values detector** against the issues. Afterwards, the human values detector uses pre-trained models to detect the presence of human values in the downloaded issues and stores the results in the database. Issues and their detected values are then displayed in the front end.

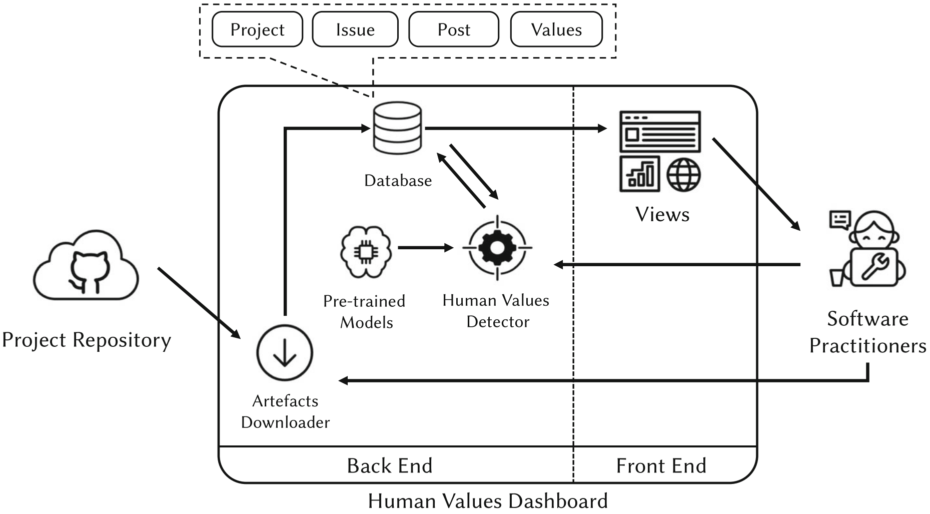


Fig. 5 The components and simplified flow of the human values dashboard

4.2 Automating the Detection of Human Values

To address the automated detection requirements of human values in the exploration stage (Section 3.4), we formulated the detection of human values as a classification problem of whether human values are present in software development artefacts. We used a data set of the presence of human values in issue discussion from our previous work (Nurwidyantoro et al. 2021b). This dataset consists of 1,097 issues manually labelled with the presence of values. The labelling of this data set followed the same concepts of human values. Thus, this dataset is suitable for for providing human values perspectives to the dashboard. Software practitioners identified the issue discussion as one of the appropriate artefacts for the human values dashboard (see Section 3.3). Finally, we evaluated well-known machine learning techniques to detect human values in the issue discussions. These techniques had been used to classify software development artefacts.

4.2.1 Methodology

To automate the detection of human values, we first preprocessed the dataset. Second, we extracted the classification features from the issues. Finally, we conducted experiments to evaluate four well-known machine learning techniques for detecting the presence of human values in issue discussions. These steps are described below.

Preprocessing Two preprocessing activities were performed, namely content abstraction and data cleansing. The content abstraction process *abstracted contents to their types* (Prana et al. 2019). For example, a mention detected in issue discussions was replaced with a ˆmentionˆ string. Table 3 shows the abstracted contents and their string abstraction found in the issues. The data cleansing process removed punctuations, numbers, source codes, stop words, and HTML tags from the dataset. The removal of stop words was done using the Natural Language Toolkit library (NLTK) (Bird et al. 2021).

Feature extraction Two statistical features and a sentiment feature were extracted from the preprocessed dataset. The two statistical features, namely BoW (bag of words) and TF-IDF (term frequency-inverse document frequency), have been used in previous studies for the classification of human values and their related concepts in software engineering (e.g. (Jha and Mahmoud 2019; Rezaei Nasab et al. 2021; Ishita et al. 2010; Ortu et al. 2016)). BoW represents each issue in terms and its number of occurrences in that unit (Schütze et al. 2008). Meanwhile, TF-IDF considers the importance of each term in the dataset by multiplying the frequency of a term t in an issue d by the inverse frequency of the issue

Table 3 Content types found in issues and their abstractions

Content Type	Description	Abstr. String
Mention	A reference to another contributor's username (GitHub 2021e) (e.g. @username)	mention
Issue number	A reference to a relevant issue number (GitHub 2021c) (e.g. #123)	issue
Commit	A reference to a relevant commit (GitHub 2021c) (e.g. a2c1423)	commit
Image	An image posted in the issue	img
URL	A link posted in the issue	url
Email	An email address posted in the issue	email

where that term is present (Schütze et al. 2008):

$$\text{TF-IDF}_{t,d} = tf_{t,d} \times \log \frac{N}{df_t},$$

where:

- $tf_{t,d}$ is the frequency of a term t in an issue d,
- N is the number of issues in a dataset, and
- df_t is the number of issues in the dataset that contains the term t.

In addition to those two features, we also extracted the sentiment feature from the issue discussion dataset. The sentiment feature was derived from the results of the sentiment analysis of the issues. This feature was suggested by Nurwidyantoro et al. (2021b) who also provided the dataset that we used. The sentiment analysis aims at *'analyse people's opinions, sentiments, and emotions towards entities (e.g. products)'* (Liu 2020). To determine the sentiment score of each issue, this study used SentiStrength (Thelwall et al. 2010) because it supports sentiment analysis in informal text communication (Thelwall et al. 2010). The SentiStrength tool[1] provides two sentiment strengths: positive and negative sentiments. The positive sentiment that resulted in this tool is scaled from 1 (less positive) to 5 (extremely positive). Meanwhile, negative sentiment is scaled from -1 (less negative) to -5 (extremely negative). This score resulting from the tool was used as the sentiment feature for this study.

Classification experiments This study formulated the detection of human values as a binary classification problem to identify whether there are human values present in issue discussions. We started with binary classification for the presence of any values rather than for specific values because the dataset contains a small number of cases for each value (see Nurwidyantoro et al. 2021b). Furthermore, the dataset is quite unbalanced, with the number of issues in which the values were identified being only one-third of the total issues in the dataset.

The experiments evaluated four well-known supervised learning methods, namely, support vector machines, random forest, multi-layer perceptron, and logistic regression. This study used these methods due to an earlier study on the identification of human values in text documents that reported that a deep learning approach performs less well in smaller datasets and *'achieve[s] good results in data-rich settings'* (Ishita et al. 2019). All of these methods have been used in previous studies to classify the content of GitHub repositories (Golzadeh et al. 2021; Arya et al. 2019; Fan et al. 2017; Eluri et al. 2019; Trockman et al. 2019; Munaiah et al. 2017; Kikas et al. 2016; Song and Chaparro 2020). The experiments used the implementation of these methods in the *scikit-learn* library (Pedregosa et al. 2011).

In the experiments, the performance of the classifiers were evaluated using ten fold cross validation. It is commonly used to evaluate classifiers performance, including in software engineering field (Ding et al. 2018; Ma et al. 2018). This technique splits the dataset into ten equal-sized parts. Then, a classifier is trained using nine parts of the dataset and evaluated using the remaining one. This training and evaluating process is repeated 10 times such that each part is evaluated once. The average and standard deviation of the results were then calculated for the final scores. For the performance measures, this study used precision, recall, F1, and Matthew's correlation coefficient (MCC). These first three metrics are considered as standards performance measures for classification problems (e.g. (Jha and Mahmoud 2019;

[1] http://sentistrength.wlv.ac.uk/

Ding et al. 2018; Ma et al. 2018)). The MCC was included due to recent studies that argue that it provides an unbiased measure of performance (Yao and Shepperd 2020; 2021).

The experiments considered three parameters, namely resampling technique, feature set, and classification method, explained as follow:

1. **Resampling technique**. This parameter was considered because of the unbalanced nature of the dataset. An unbalanced dataset can affect the performance of a classifier toward the majority class (Padurariu and Breaban 2019). In the experiments, we evaluated the performance of the classifier without using any resampling techniques and then compared it with the use of oversampling and undersampling techniques. An oversampling technique attempts to balance a dataset by generating new samples for the under-represented class (Mohammed et al. 2020). We used SMOTE (synthetic minority oversampling technique) (Chawla et al. 2002) as one of the prominent oversampling techniques used in classification experiments (e.g. (Arya et al. 2019; Beyer et al. 2020; Catolino et al. 2019)). In contrast to oversampling, an undersampling technique balances the dataset by selecting a subset of a class with the majority number of samples (Mohammed et al. 2020). It has been used in software engineering research as an alternative way to handle unbalanced dataset (e.g. (Biswas et al. 2019; Canedo et al. 2020)). In these experiments, we randomly selected a subset of samples using the RandomUnderSampler implementation in the *imbalanced-learn* library (Lemaître et al. 2017).
2. **Feature set**. This parameter investigated how the features influence the performance of the classifiers. In the experiments, we compared the performance of the classifiers using **BoW**, **TF-IDF**, and the combination of each statistical feature with the sentiment feature (i.e. **BoW+Sentiment** and **TF-IDF+Sentiment** features).
3. **Classification method**. We experimented with four classification methods, namely support vector machine (SVM), random forest (RF), multi-layer perceptrons (MLP), and logistic regression (LR). To obtain the best parameter for each classifier method (*hyper-parameter tuning*), the grid search process was used on a set of values for the methods' parameters. This approach has been used in previous work (e.g. (Golzadeh et al. 2021; Arya et al. 2019; Song and Chaparro 2020)) for classification experiments. The arguments and their values used in the experiments are shown in Table 4. We then selected the best results for each classification method and compared them with each other.

4.2.2 Experiments Results

The experiments used the F1 score, which provides *'the balance between precision and recall'* (Arya et al. 2019) as the primary metrics to determine the performance of the classifier. The remaining metrics were used to provide different perspectives on the classification results. This approach is considered common in classification studies, including those in software engineering (e.g. (Arya et al. 2019; Fan et al. 2017; Prana et al. 2019)). Table 5 shows the best performance of each classification method.

The best performance of the SVM method was demonstrated using the undersampling technique and the TF-IDF feature with kernel parameter radial basis function. The BoW with sentiment features and the oversampling technique performed best for the RF method. The hyper-parameter setup for this performer used the entropy information gain and 1,000 decision trees for the RF method. The best F1 performers for these two methods had the same F1 score (0.619). However, the RF method offered better precision but slightly lower recall than the SVM method.

For the MLP method, Table 5 shows that the best performance used the undersampling technique and TF-IDF feature. The best hyper-parameter setup for this method was using the hyperbolic tan function (*tanh*) for the activation function and the L-BFGS (Limited-memory Broyden–Fletcher–Goldfarb-Shanno) solver. Similarly, for the LR method, the best performer was using the undersampling method and the TF-IDF but with sentiment feature with the stochastic average gradient (SAG) solver. The F1 scores for the MLP and LR were very close (0.001 difference). However, the precision of the MLP was slightly better than that of the LR method. Conversely, the recall of the MLP was slightly lower than that of the LR method.

Comparing the precision for all methods, Table 5 shows that the RF method was the best. As for the recall, the LR method had the best score. The MCC score was aligned with the performance rate of the F1 scores. The MCC score for the RF method was higher than that of the SVM method, although the F1 scores were the same. This condition means the RF method offered a better overall prediction than the SVM method. Nevertheless, the MLP method was still the best performer among all these methods, with the highest F1 and MCC scores.

Table 6 shows the average values for the confusion matrix of all 10 folds in the top performer of the MLP classifier mentioned in Table 5. The **Total Actual** column of this table shows the imbalanced nature of the testing set (i.e. 37 values issues : 73 no values issues). This means the undersampling was only applied in the training set. The confusion matrix shows that the MLP classifier correctly identified the majority of the issues where values were found (i.e. 27 out of 37 issues). However, this classifier had lower performance in detecting issues where values were not found (i.e. 53 out of 73 issues). The classifier incorrectly identified 20 issues to have values (i.e. false positives). Meanwhile, only 10 issues were incorrectly identified to have no values found (i.e. false negatives). This results in a higher recall (0.74) and a lower precision (0.58). The complete comparison and results of these experiments can be found in Nurwidyantoro (2022). We used the best classifier in the human values dashboard, i.e. multi-layer perceptrons with the undersampling classification model.

Table 4 Arguments for the classification methods

Method	Arguments	Description	Values
SVM	kernel	Kernel function for the SVM algorithm	(polynomial, rbf, sigmoid)
RF	max_depth	The maximum depth of the tree	(4, 5, 6, 8, 100)
	criterion	The function to measure the quality of the decision split	(gini, entropy)
	n_estimators	The number of trees in the random forest	(10, 100, 1000)
MLP	activation	Activation function for the hidden layer	(identity, logistic, tanh, relu)
	solver	The solver function for the weight optimisation	(lbfgs, sgd, adam)
LR	solver	The algorithm to use in the optimisation problem	(newton-cg, lbfgs, liblinear, sag, saga)

Table 5 The best performance of each classification method

Method	Imb. Handling	Feature	Precision	Recall	F1	MCC
SVM	Undersampling	TF-IDF	0.575±0.051	0.676±0.059	0.619±0.038	0.407±0.064
RF	Oversampling	BoW+Sentiment	0.637±0.063	0.610±0.085	0.619±0.058	0.438±0.078
MLP	Undersampling	TF-IDF	0.582±0.043	0.741±0.062	0.650±0.032	0.451±0.053
LR	Undersampling	TF-IDF+Sentiment	0.570±0.045	0.757±0.047	0.649±0.035	0.445±0.059

Table 6 The average values of confusion matrix of all folds in the MLP (Precision=0.58, Recall=0.74, F1=0.65, MCC=0.53)

		Predicted		Total Actual
		Values found	Values not found	
Actual	Values found	**27**	**10**	37
	Values not found	**20**	**53**	73
Total Predicted		47	63	110

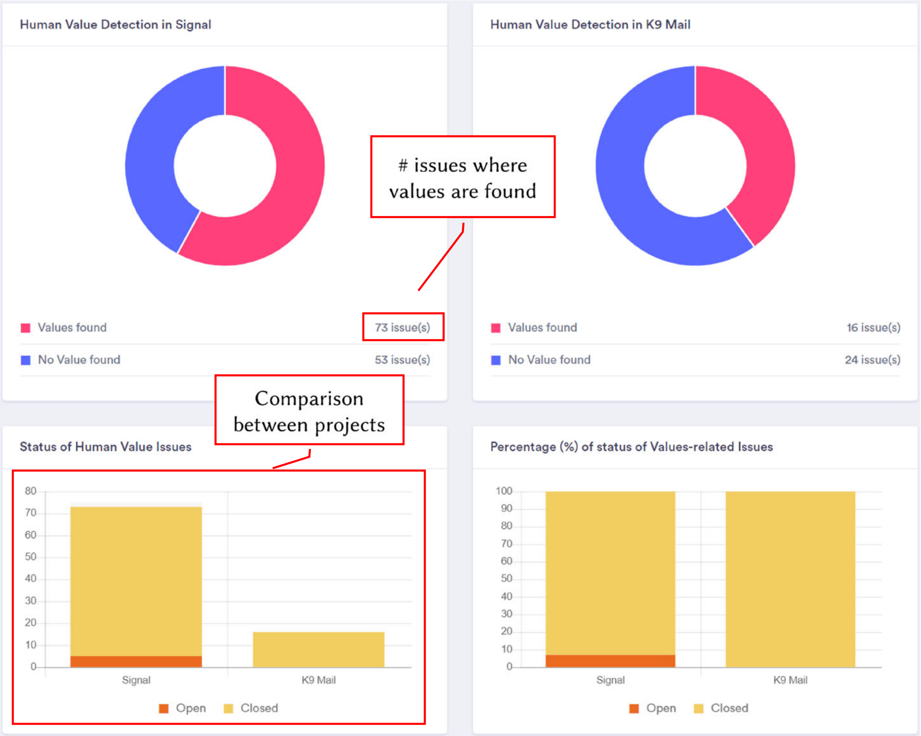
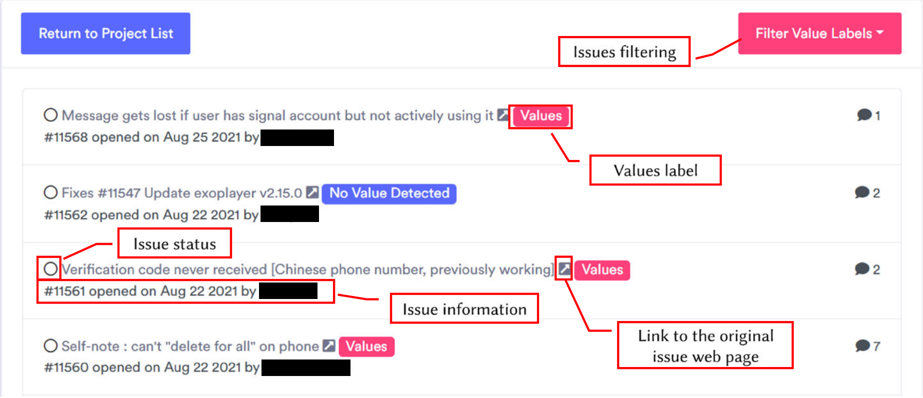

Human Value Detection in Signal
Values found
73 issue(s)
No Value found
53 issue(s)
Human Value Detection in K9 Mail
Values found
16 issue(s)
No Value found
24 issue(s)
# issues where values are found
Comparison between projects
Status of Human Value Issues
80
70
60
50
40
30
20
10
0
Signal
K9 Mail
Open
Closed
Percentage (%) of status of Values-related Issues
100
90
80
70
60
50
40
30
20
10
0
Signal
K9 Mail
Open
Closed

Fig. 6 Dashboard summarised overview (OV). (Boxes in the red outline are not part of the dashboard)



○ Message gets lost if user has signal account but not actively using it Values 1
#11568 opened on Aug 25 2021 by [redacted]

○ Fixes #11547 Update exoplayer v2.15.0 No Value Detected 2
#11562 opened on Aug 22 2021 by [redacted]

○ Verification code never received [Chinese phone number, previously working] Values 2
#11561 opened on Aug 22 2021 by [redacted]

○ Self-note : can't "delete for all" on phone Values 7
#11560 opened on Aug 22 2021 by [redacted]

Fig. 7 Dashboard values-labelled list (LI). (Boxes in the red outline are not part of the dashboard)

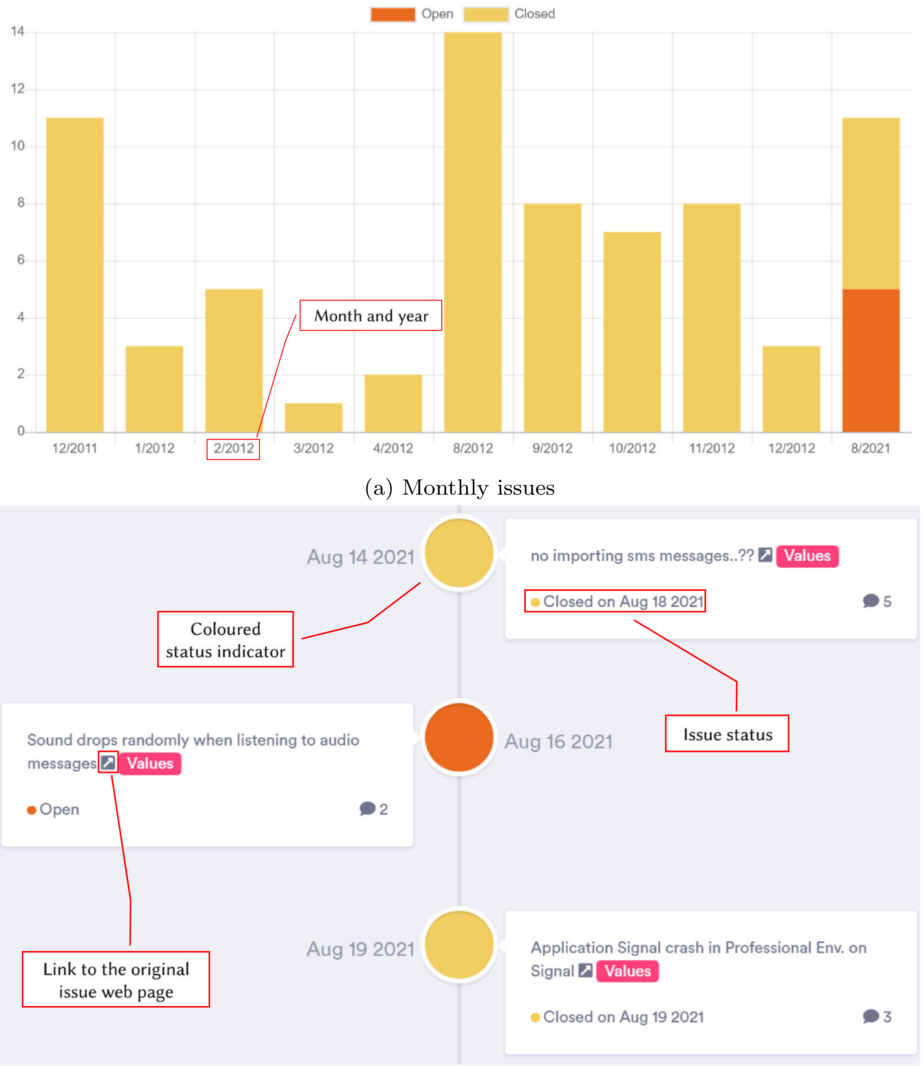
4.3 Dashboard Implementation

The dashboard was implemented using Flask[2], a web framework written in Python. A Python-based framework was chosen to facilitate the integration of the human values detector into the back end. The implementation of the dashboard focused on the use of issue discussions, as suggested by the empirical findings of a previous study (Nurwidyantoro et al. 2021b) and practitioners in the exploration stage (Section 3.3).

Front-end views were developed using the Chart.js library[3]. The three views proposed in the exploration stage were retained because participants in the exploration stage considered those views useful for various roles in software development. We made some adjustments to the three views in the implementation because of the limitation of the automated human values identification (Section 4.2). This limitation only allowed us to display whether human values are present or not in the issues, without specifying which specific values, such as security or inclusiveness, are present. The adjustments made are explained below.

1. **Summarised values overview (OV)**. This view displays the number of issues where human values were present and not present in a pie chart. To allow for comparisons between projects, this view provides two of these pie charts side by side. This view also displays the number of issues where values are detected based on the status of the issues (i.e., open or closed). This view aims to provide insight into the number of values-labelled issues that need to be addressed. Figure 6 shows the OV implemented on the dashboard.
2. **Values-labelled list (LI)**. This view shows a list of issues similar to how issues are displayed on GitHub. This view displays a label as the result of the human values detector indicating the presence of human values in a particular issue. An issue is labelled with either the ‘Values’ label if the human values detector finds human values in that issue or the ‘No Value’ label if the human values detector does not find human values in that issue. This view also includes information on when the issue was opened, by whom, whether it is open or closed, and the number of posts. There is a filtering capability for issues and a link to the original webpage of the issue, as suggested in the exploration

[2] https://flask.palletsprojects.com/en/2.0.x/
[3] https://www.chartjs.org/



no importing sms messages..?? Values

Closed on Aug 18 2021

Sound drops randomly when listening to audio messages Values

Open

Application Signal crash in Professional Env. on Signal Values

Closed on Aug 19 2021

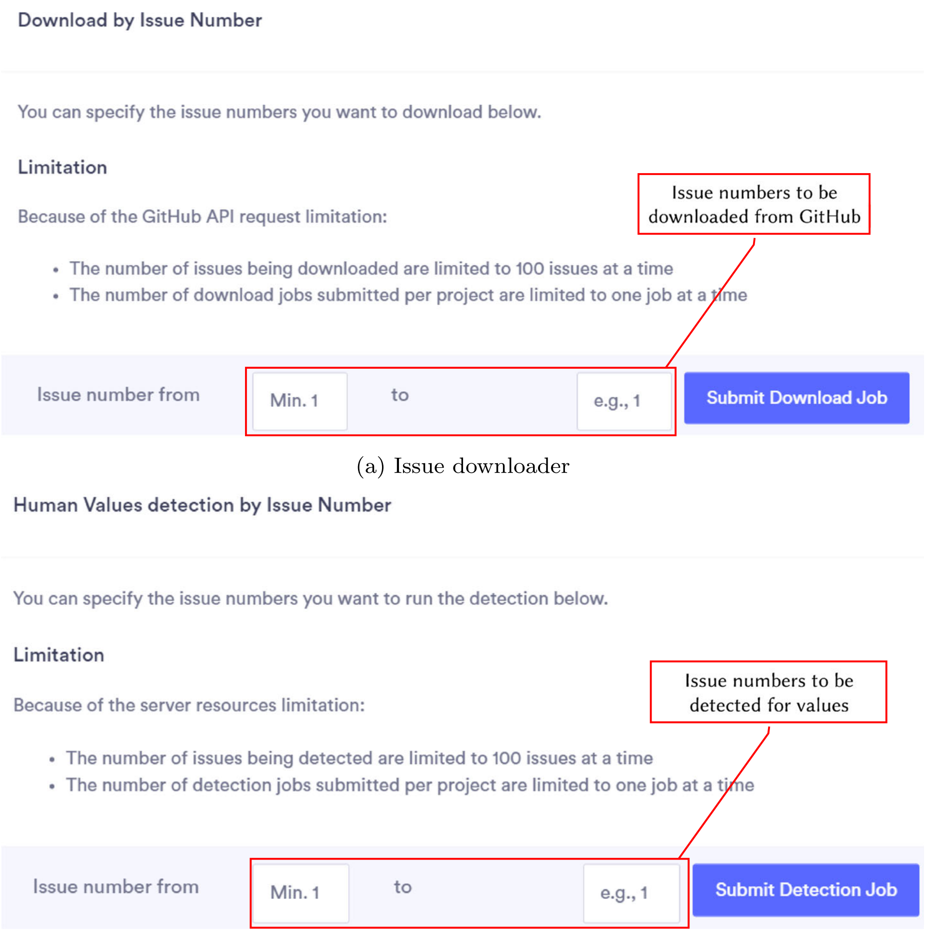
(a) Monthly issues

(b) Timeline view of the issues

Fig. 8 Dashboard timeline (TM) (Boxes in the red outline are not part of the dashboard)

stage (Section 3.4). The filtering feature allows software practitioners to view issues in which human values are identified. Figure 7 shows this view on the dashboard.

3. **Values-labelled timeline (TM)**. This view shows the issues chronologically according to the date the issues were opened. This view shows a bar graph showing the monthly number of open and closed issues where values are present (Fig. 8a). At the bottom, this view presents a timeline of issues where values are present, with two different colours to indicate whether the values are open (orange) or closed (yellow) (Fig. 8b). Figure 8 shows both visualisations in the timeline view.



Download by Issue Number

You can specify the issue numbers you want to download below.

Limitation

Because of the GitHub API request limitation:

- The number of issues being downloaded are limited to 100 issues at a time
- The number of download jobs submitted per project are limited to one job at a time

(a) Issue downloader

Human Values detection by Issue Number

You can specify the issue numbers you want to run the detection below.

Limitation

Because of the server resources limitation:

- The number of issues being detected are limited to 100 issues at a time
- The number of detection jobs submitted per project are limited to one job at a time

(b) Issue detector

Fig. 9 User interfaces for the dashboard's back end

On the back end, there are two interfaces used for the artefacts downloader and human values detector components. The first interface allows the development team to specify a range of issue numbers that they want to download from the project repository (Fig. 9a). The issue downloader was implemented using the GitHub API and the github3.py[4] library. The second interface enables the development team to specify a range of issue numbers to be detected by the human values detector (Fig. 9b). This range of issues is then used as a parameter to run either the download or detection as a background task on the server. Figure 9 shows those interfaces for the dashboard back end. Two open source projects, Signal Android and K9 Mail, were used as examples in the feedback stage. This human values dashboard is available online[5].

[4] https://github3.readthedocs.io/

[5] https://arifn.github.io/showcases/

5 Feedback Stage

This stage involved presenting the human values dashboard developed in the previous stage to gather feedback from the software practitioners. It focused on obtaining feedback from the practitioners and determining whether the human values dashboard is useful. Additionally, because the human values dashboard used an automated technique to detect human values (i.e. a human values detector – Figure 5), it is also necessary to understand the practitioners' opinions on the performance of the detector. We believe this is necessary regardless of the performance of the current classifier to know the acceptable level of performance to practitioners. Therefore, for this feedback stage, the following sub-research questions were defined:

RQ5 To what extent do practitioners find the human values dashboard useful?
RQ6 How do practitioners perceive the performance of the automated human values detection?

In addition to the answers to these sub-research questions, the participants' suggestions were also collected to improve the dashboard in the future.

5.1 Interview Guide Development

An interview guide was developed for the feedback interview to obtain the practitioners' feedback and suggestions on the human values dashboard. This semi-structured interview consisted of two parts. The first part of the interview asked for the demographic information of the participants, such as their roles and experiences. The second part of the interview started with an introduction on human values concepts and a demonstration of the values dashboard. Then, the second part continued by asking the practitioners' opinions regarding the dashboard and its usage. This part also asked questions related to the human values detector, e.g. a component to automatically detect the presence of values. This interview guide was discussed with the supervisory team and other group members, resulting in several suggestions. Several adjustments were made to the interview guide by incorporating these suggestions.

5.2 Data Collection

Participant selection criteria. The selection criteria used in this stage were similar to those used in the exploration stage. This interview sought for practitioners who had been involved in a software development project and were familiar with artefacts from software repositories. This stage involved a new set of participants, i.e. practitioners who had not been involved in the exploration stage. This choice was made to investigate whether different practitioners viewed the human values dashboard as acceptable.

Participant recruitment. The recruitment of the participants was done by inviting contributors of open-source projects hosted on GitHub via email. The email addresses of these contributors were made available by them on their GitHub pages. Interested participants were asked to reply to the email invitation. An invitation to participate was also published on the group web page and LinkedIn. In addition, our colleagues were asked to broadcast the invitation to their networks. Interested practitioners were asked to inform us of their emails through our colleagues or fill out an online form on the group web page. These candidates were then contacted via email to request their consent and arrange an interview session.

Table 7 Profile of the participants

Code	Role	Experience (years)	Location
P16	Developer	3	Asia
P17	Project Manager	16	Asia
P18	Developer	4	Asia
P19	Developer	8	Asia
P20	Developer	5	Asia
P21	Developer	6	Australia
P22	Software Architect	10	Asia
P23	Developer	12	Asia
P24	System Analyst	16	Europe
P25	Developer	6	Asia

Profile of the participants. Table 7 shows the profiles of the participants for the feedback interview. Please note that the participants for this stage are different from the participants in the exploration stage. Participants mostly had developer roles. Most of them had less than 10 years of experience in software development; 4 had 10 or more years of experience. The participants were mostly located in Asia, with one participant located in Europe and another in Australia.

Interview protocol. Before the interview session, participants were asked to read the explanatory statement and complete the interview consent form. The informed consent document is available in Nurwidyantoro et al. (2022b). All interview sessions were conducted in English. We ensured that all participants have adequate English proficiency. The interview consisted of two parts. The first part focused on obtaining the professional backgrounds of the participants. The second part started by explaining human values' concepts and the study. The participants then were given the link to access the dashboard developed in Section 4.3. The interviewer then demonstrated the dashboard and provided 10-15 minutes for the participants to interact and evaluate the dashboard and its contents. Then, the participants were asked for their perspectives on the usefulness of the dashboard (e.g. *'Would the dashboard be useful for you in software development? How?'*) and on the performance of the values detection (e.g. *'At what level is the dashboard accuracy tolerable for you? Why?'*). Before asking these questions, we explained and discussed the background of each question to ensure that the participants understood. This second part of the interview also asked for their suggestions and feedback for the dashboard (e.g. *'Does the information provided in the dashboard prototype sufficient to help you?'*). The interview questions for this study are available in Nurwidyantoro et al. (2022a).

The interviews in this stage were recorded using a video conference system with the permission of the participants. Similar to the exploration phase, the number of interviews had not been set in advance. Recruitment and interviews were conducted in parallel with data analysis until data saturation was reached (Beitin 2012; Ournani et al. 2020). The convergence of answers and ideas became apparent in the data analysis after 10 interviews. The mean duration of the interviews was 30 minutes and 49 seconds. Professional transcription services transcribed all the audio recordings of the interviews.

5.3 Data Analysis

The interview data was analysed using the thematic analysis approach (Braun and Clarke 2012). Similar to the data analysis in the exploratory stage of this work and other previous studies (Tomasdottir et al. 2017; Tómasdóttir et al. 2020), the first author performed a large portion of the analysis, which was followed by reviews and discussions with the other authors. In this analysis process, the supervisory team was also consulted in the event of doubts or difficulties. The first author started to familiarise himself with the interview data by reading the transcriptions and listening to the audio recordings. Then, the first author generated codes and themes from the analysis of the transcriptions. Subsequently, the first author had several discussions with the supervisory team to review the identified codes and themes and determine their relations. The first author then assigned a name and definition to each theme. The resulting themes were presented to the other authors for feedback. The themes were then adjusted by incorporating that feedback.

5.4 Results

This section presents the results of the feedback stage. First, this section describes the usefulness of the human values dashboard and the challenges of deploying it in a company. Second, this section presents the practitioners' perceptions of the human values detector. Finally, this section lists suggestions from the practitioners to improve the dashboard further.

5.4.1 Usefulness of the Dashboard (RQ3.5)

To understand the practitioners' perspectives on the extent to which the human values dashboard could be helpful, the interview started with presenting and providing the dashboard to the participants to explore. Then, the interview asked the participants about the usefulness of the dashboard to support their development activities. The analysis of the interviews suggested that the practitioners agreed that the dashboard could be useful for them. Some participants argued that there would be some potential challenges for the dashboard to be implemented in their company. These findings are described below.

The human values dashboard was considered useful. The participants agreed that the human values dashboard could be useful to support them in software development activities. Identifying the values present in issues would help the practitioners focus their attention on the issues, which, in turn, would ensure these issues were addressed. A developer mentioned:

> *'... Developers will pay their attention to that one [the **LI** view]. So, if we make sure that we have covered all possible scenarios in the issue list to take down [address] that human values in those tickets.'*. (P16–Developer)

In addition to focusing the development team's attention, the participants believed that the dashboard's values labels would provide human values perspectives in addition to the well-known technical perspectives. A developer mentioned:

> *'When we look at an issue right now, so we do not think about any values aspect, like human values normally. We just think about it from a technical side usually. This would be helpful to understand there is another aspect for the ticket there.'*. (P20–Developer)

The human values perspectives would subsequently help them prioritise their tasks. This usefulness would be apparent if there were a substantial backlog of issues:

> ‘*... Especially when there is a huge backlog of issues, I think it is very hard to kind of prioritise and a lot of issues get lost in the backlog, and we file it during one time and then it kind of gets lost and then it does not come up or it just that. So, if there is some sort lot of, let us say, a subjective value, let us say morals assigned to an issue. I think it would help to kind of prioritise it.*’. (P22–Software Architect)

Some participants believed that the dashboard could also inform the team’s performance. The dashboard summarised overview (Fig. 6) could also be used to compare the progress between projects. A developer mentioned:

> ‘*I have a company, and I am running several projects. Okay. So, I can measure the team performance by this tool easier, and also the complexity of the project I can understand from this.*’.(P19–Developer)

A project manager suggested that linking values-labelled issue posts to their contributors could help identify values champions. The participant referred to a values champion as ‘*anyone who aligns themselves with human values in the organisation*’. The participant mentioned:

> ‘*In this dashboard, you can see who is the champion of these values or maybe what is the level of “do not do evil” in the discussion, inside the repository and the issue tracker.*’. (P17–Project Manager)

Potential challenges in adopting the dashboard. When the dashboard was presented, some participants reacted by suggesting potential challenges in adopting the dashboard in their environment. A developer mentioned that a contributor might not describe the issue correctly and that this could influence the result of the human values detector:

> ‘*Because, in my experience, I have gone through some issues that may be the QA developers, ... I mean, QA when raising these issues, but they are not correctly describing the issue in the field.*’. (P16–Developer)

Another challenge concerned the willingness of a company to use the dashboard. The participants suggested some reasons that could hinder the use of the dashboard in a company. First, a company may not be familiar with the concept of human values. This situation could lead to a lack of awareness of human values in the company and the company tending to focus on the financial aspects of the business:

> ‘*Although it has some significant impact while I am developing something or not, but sometimes the management or the [project] plan, and does not bother [with] that type of issues or that type of thing. They only think about money and business.*’.(P19–Developer)

Even if a company is aware of human values, it must decide how to address conflicting values from different users. An additional effort may be necessary to determine what needs to be done:

> ‘*These are two issues that we need to prioritise. Are (users from) China our main priority or (users from) [the] US our main priority? The domain is specific. So, how can I prioritise these two issues by these two (users)? Is it possible?*’. (P19–Developer)

Second, a corporation could argue that the consideration of human values is not required because it is unregulated. A project manager suggested that a company itself is in a position to decide whether it wants to support the consideration of values:

> *'This is [an] area where the company, right now within the US or maybe international law is not compulsory. It is more like the company does assessments on their intentions, on their diversity, and so on, as a public campaign, but not regulated.'*. (P17–Project Manager)

The practitioners in the study considered the dashboard to be useful for focusing attention and prioritising issues. However, there are some potential challenges, such as the willingness and extra efforts required from a company in adopting the dashboard.

5.4.2 Perceptions of the Performance of the Human Values Detector (RQ3.6)

The use of an automated approach to identify human values in a dashboard has the possibility of leading to inaccuracies. This interview stage used the term 'accuracy' to simplify the communication with the participants regarding the correct or incorrect identification of values. To understand how the practitioners perceive the automated human values detector, the interview probed the extent to which the performance of the detector was tolerable to the participants.

The analysis of the interviews indicated that the practitioners understood the possibility of inaccuracies occurring in the identification of human values. However, the level of tolerance to accuracy varied among practitioners. One practitioner preferred to have 90% accuracy to trust the identification results:

> *'To have that kind of level of trust, I think at least 90% accuracy is needed. Less than 90%, usually we do not trust the tools, we do not put any action point on the tools.'*. (P17–Project Manager)

Meanwhile, another practitioner considered 50% accuracy to still be tolerable:

> *'This is a machine learning thing, so there will be some issues. It cannot give an exact solution, so I think 50 is enough and it will develop after some time.'*. (P18–Developer)

The analysis of the interviews also discovered that all participants preferred to have false positives on the detector than false negatives. This finding meant that it was acceptable to have the human values detector identify that an issue had values present even though that might not be correct. All participants agreed among themselves that false positives were better than missing critical issues because the detector was unable to detect the presence of the values. A developer mentioned:

> *'It says there is no value, but actually there is a value. We can neglect this since it notifies that this has no value, and we neglect it without further investigating the issue.'*. (P17–Developer)

Table 8 Suggestions for the overall dashboard

No.	Suggestion	Quotation
G1	Specific human values detection	*'If you can show these (values) categories in the dashboard, I guess, it would be helpful.'*. (P20 - Developer)
G2	Issue management	*'These are my plan[s] to address the issue. So, if we can enrich this system with our own way of planning, how to handle them, that would be good.'*. (P21 - Developer)
G3	Colour customisation for values labels	*'It is in there using the red or the blue, I think I would suggest a value that would be blue and no value, but I do not know; we tend to look to red as problem.'*. (P24 - System Analyst)
G4	Progress of an issue	*'I think the most important thing for us, in progress section, because we need to plan our delivery. So, by seeing this, we can predict when can we deliver this or not.'*. (P19 - Developer)
G5	Customisation for prioritised values	*'I think there should be some sort of weight to our value because privacy may not be that important to some application[s].'*. (P22 - Software Architect)
G6	Ranking of issues based on additional criteria	*'So, I think, basically, some sort of ranking for this. I do not know how urgent or popular this issue is in this view.'*. (P21 - Developer)
G7	Indication of values violation	*'And of course, as our earlier discussion, the violation of value that is also pulled in. It has value but in a negative way. It has violation of value, not just it has value. That's also important.'*. (P17 - Project Manager)

> Practitioners have different level of tolerance for inaccuracies in the identification of values. However, false positives were preferred over false negatives.

5.4.3 Suggestions for Improving the Dashboard

To obtain feedback, each view in the dashboard (Figs. 6, 7, and 8) was demonstrated to the participants. Then, the participants were probed for suggestions to improve the dashboard. The first author, as the main analyst of the interview, collected the participants' feedback and suggestions on each view and the overall dashboard.

Table 8 shows the feedback from the participants on the dashboard. In **G1**, the participants wanted to have the dashboard display which specific values were detected in the issues. This would provide a development team with an opportunity to address issues based on their values priorities. One participant also wanted to have functionality, such as a to-do list or a planner, to manage the issues that a developer want to address (**G2**). Some of these functions are provided in GitHub (2021d). In **G3**, it was found that each team or practitioner had a preference on the label colour. A colour customisation feature could be developed to address this suggestion. Practitioners also suggested that the dashboard display the progress of each issue (**G4**). This information could help them to plan or predict application delivery. Additionally, they wanted the dashboard to allow them to specify which values they wanted to prioritise (**G5**). This suggestion could be addressed by having the human values detector detect the presence of specific values (e.g. privacy or longevity). In **G6**, the practitioners suggested having additional criteria for ranking the issues on top of the presence of

Table 9 Suggestions for the summarised overview in the dashboard

No.	Suggestion	Quotation
OV1	Provide additional categorisation (e.g. issue type)	*'I mean, for the QA person, we can add some categorising, (such) as user experience, user interfaces side, and we can add such things to one category.'.* (P16 - Developer)
OV2	Reporting	*'Maybe you have this on your future plans, maybe we can add some reporting here.'.* (P16 - Developer)

the human values. The urgency level or the popularity of an issue could be the indicators for this ranking. In **G7**, a practitioner proposed a suggestion to indicate not only the presence of human values but also the values violations in an issue. The practitioner stated that this indication would be helpful to prioritise the issues.

Table 9 lists the practitioners' suggestions for the summarised overview (OV) of the dashboard. In **OV1**, the participants suggested that the dashboard should have additional categorisation based on the type of issues. An example of this categorisation could be based on the types of roles in the team that could address the issues, such as UI issues for the UI designer. A practitioner also suggested that the OV have a reporting functionality to support the decision-makers of a software project (**OV2**).

For the list view of the dashboard, the practitioners had some suggestions, which are listed in Table 10. For example, a practitioner suggested that the issues' assignees be displayed in the dashboard (**LI1**). The practitioner mentioned that this information would help filter out issues that still need someone to work on them. In **LI2**, a practitioner suggested including the topics of the issues. This information would provide the development team with a quick summary of what all the issues are about. Related to the accuracy of the human values detector (Section 5.4.2), a participant mentioned that it would be helpful if the confidence level of detection is displayed on each issue (**LI3**). This level of confidence is the

Table 10 Suggestions for the list view in the dashboard

No.	Suggestion	Quotation
LI1	Assignees' information (available in GitHub)	*'I think can it have something like assigned to kind of thing there? ... I think for me it might make more sense to look into it if it is not assigned or if nobody is looking into it.'.* (P25 - Developer)
LI2	Topic of the issues (e.g. word cloud)	*'So, I have just one suggestion, which is to cluster the issues. ... But I think let's say if I want to know what are these all value issues are discussing. They might be basically three or four topics, right. So, if I want, you know your points, you know word clouds, right?'.* (P21 - Developer)
LI3	Display of the detection confidence level	*'To include the rank of the classification as a values discussion in the view here so that you can filter depending on these. ... So, if value is one, let's say, for example, it might be 90% prediction or maybe 80%, or maybe even 51%, right. So, if you show this number here, ... I will be able to give a priority to the ones that are highly rank and then just leave those 51% to look at later.'.* (P21 - Developer)
LI4	Suggestions for solving the issues	*'In your automated tool, using (the) automated tool, can you search on Google about those issues? It is better [if] you can display some more information about this.'.* (P18 - Developer)

Table 11 Suggestions for the timeline view in the dashboard

No.	Suggestion	Quotation
TM1	Customisable time range	*'Something like last one month or last few months kind of thing. So that it gives me a real idea on how things are looking at real. So you can always change, you can always select all to view everything. But so as a user, when I see it, if it gave me the latest data or latest strain over the week or the month, I think it would be useful.'.* (P25 - Developer)
TM2	Duration of time that had passed since an issue was first reported	*'And it's hard to figure out which was created when, and ... even it to take the immediate action or not? Since issues (were) created yesterday, (it) might wait for one or two or weeks, but if it is already one month, then you might want to take action and look into it, at least try to solve it.'.* (P25 - Developer)
TM3	Duration needed to fix the issues	*'Duration. Yeah, yeah, yeah. From open to close. What I am telling (you) is so, if I am an administrator or someone who manages everything, what I want to look is on average, how much time is it going?'.* (P25 - Developer)

prediction score from the classifier in classifying an issue, e.g., whether the issue has values or not. This suggestion could help practitioners prioritise issues with a higher confidence level of detection. To further help practitioners in addressing issues, one participant also requested that the dashboard search and display relevant solutions from search engines (e.g., Google) or questions-and-answer forums (e.g., Stack Overflow) for each issue (**LI4**).

In the timeline view of the dashboard, the participants provided several suggestions related to the time perspective, as shown in Table 11. First, the practitioners requested a customisable time range (**TM1**). The practitioners felt that this customisation would highlight recent issues depending on the frequency of issues in a project. The remaining two suggestions were related to the duration of time that had passed since an issue was first reported (**TM2**) and the duration needed for an issue to be completed (**TM3**). The former suggestion (**TM2**) would highlight an issue that had not been addressed for a period of time, while the latter (**TM3**) would provide analytics of issue completion to the project managers.

> Sixteen suggestions to improve the human values dashboard were collected. These suggestions are not only for each view of the dashboard but also for the overall dashboard.

6 Discussions

This section highlights and discusses the findings of the exploratory and feedback stages. First, this section discusses the findings on the awareness of values. Second, it discusses the possibility of using software artefacts as the source to identify values for the dashboard. Third, this section discusses the possibility of providing a human values dashboard for users. Finally, it discusses the limitations and challenges of the dashboard.

6.1 Awareness of Values

The analysis in the exploration stage showed that software practitioners are familiar with only a limited set of values, such as *security* and *accessibility* (Section 3.1). This finding strengthened the findings of Perera et al. (2020), which highlighted that only a few values have been discussed in recent academic publications on software engineering. This lack of awareness was also found in the feedback stage as a potential challenge to the adoption of the human values dashboard in a company (Section 5.4.1). One possible reason for this stems from the fact that there is a lack of understanding of these values in the software engineering context. Furthermore, participants in the exploration stage thought that the values they were familiar were important and believed that they had already considered these values during software development. The other values that they were not familiar with became 'nice to have' in an application. It could be argued that the values that participants are familiar with are similar to non-functional requirements that are related to the quality properties or characteristics of software (Glinz 2007; Mairiza et al. 2010; Barn 2016). This might also be the reason why practitioners are not familiar with some other values. This argument makes sense for us because, based on our previous work (Nurwidyantoro et al. 2021b), we believe that human values have a much broader sense that includes non-functional requirements. These findings also showed the need to increase awareness of values not only of practitioners but also of companies. A possible solution could be to provide a contextualised software engineering definition for each of these values (Mougouei et al. 2018; Perera et al. 2020), as presented in Nurwidyantoro et al. (2021b). Furthermore, as suggested in the findings, a tool, such as a human values dashboard, could be used to introduce and increase the awareness of values of companies and their development teams. These findings were in line with a previous study that suggested that a dashboard has the benefit of increasing awareness (Treude and Storey 2009).

6.2 Artefacts as the Source for the Dashboard

The software practitioners in the exploration stage suggested that requirements documents and issue discussions are considered more suitable for the mining of human values. This paper focused on one of these artefacts, namely issue discussions, as the source for the dashboard. Future work could extend this research by investigating the presence of human values in requirements documents. If such a study could find the presence of values in requirements documents, then it could be followed by incorporating requirements documents as another artefact source in the human values dashboard.

Based on the results of the exploration stage, a human values dashboard was developed. This dashboard labels the presence of human values in issue discussions. The feedback stage found that the participants agreed that the dashboard could be helpful in focussing their attention and prioritising issues. Nevertheless, it is still possible to enhance the developed human values dashboard by adding other artefacts. Therefore, future research could investigate the presence of human values in other artefacts to incorporate them into the dashboard. Some suggestions by the participants could be acomplished by integrating the dashboard with existing software repositories. More studies could be conducted to investigate to what extent this integration is possible.

6.3 A Human Values Dashboard for Users

The exploration stage results suggested that one of a human values dashboard's main benefits is promoting the awareness of values. This awareness of values could trigger discussions among stakeholders on what values must be considered in an application. Then, as suggested by the findings in the feedback stage, the development team could focus on the prioritised values and ensure these values are addressed during development. This study focused on software practitioners involved in software development. It did not include end users of an application as one of the stakeholders in software development. Application users are indirectly involved in application development by providing feedback. Giving them access to a human values dashboard would help users evaluate the values of an application (Kujala and Väänänen-Vainio-Mattila 2009), which in turn could guide them to choose their preferred application (Wang et al. 2013; Harris et al. 2016; Fu et al. 2013). However, to understand the dashboard's usefulness for users, a future study involving users needs to be carried out.

6.4 Limitations and Challenges of the Dashboard

The human values dashboard (Fig. 5) may have several limitations. First, the dashboard depends on the availability of artefacts (e.g. issue discussions). A project may not have all the artefacts mentioned in Section 3.3 depending on how it is managed. Second, an automated approach has been chosen to identify the presence of values because it can reduce manual efforts. Identification using automated approaches has accuracy limitations. This performance limitations also happened in prior studies on the detection of human values in text documents. These studies initially reported low performance (F1 score of 0.45 (Ishita et al. 2010)), but a series of studies later in the following years (Takayama et al. 2013; Takayama et al. 2014; Takayama et al. 2015) resulting in better performance (F1 score of 0.74 (Takayama et al. 2016)). These recent works demonstrated that classifying human values is not a trivial task. Abstract concepts of human values may contribute to this challenge. Regarding the accuracy, although the tolerance level for inaccuracies varied between participants, here the inaccuracies were understandable by the participants. The findings also found that practitioners preferred false positives to false negatives. This means that the classification methods evaluated in Section 4.2.2 could be considered to be tolerable by practitioners. Furthermore, evaluation metrics that emphasise false positives, such as the F2 score (Jha and Mahmoud 2019), could be used to evaluate the performance of the automated human values detector. In this case, the F2 score for each classifier mentioned in Table 5 are 0.65 for the support vector machine, 0.61 for the random forest, 0.70 for the multi-layer perceptrons, and 0.71 for the logistic regression classifiers. These F2 results show that the logistic regression classifiers now performed better, although not significant, than the multi-layer perceptrons. In addition, in the feedback stage, the practitioners (Section 5.4.3) suggested displaying the confidence level for the detection of human values. This information could help practitioners prioritise issues with a higher level of confidence. Third, the automated approach used in the dashboard at this point is only capable of detecting if any human values are present, without specifying which values. This limitation was due to the limited number of cases wherein specific values were discovered (see Nurwidyantoro et al. (2021b)). The unbalanced nature of the dataset was also found in previous studies of human values analysis in text documents (e.g. (Ishita et al. 2010; Takayama et al. 2013; Takayama et al. 2014)). Future work could expand the datasets for specific values by targeting specific types of applications. For example, the hedonism value could potentially be discovered in issue discussions of computer games. Furthermore, the human values detection could be improved in the future by using emerging approaches such as deep learning techniques. However, the datasets may need to be expanded to cater to such approaches. Alternatively, other methods that do not require large datasets, such as keyword-based or rule-based approaches, could also be used. Another way to improve automated detection could be to ask practitioners as users of the dashboard to add or correct the labels on the artefacts. These additions and corrections could then be incorporated as feedback to retrain the classification model to improve the identification over time.

Despite these limitations, the findings showed that the dashboard would be beneficial for software development. However, the practitioners highlighted two potential challenges in adopting the dashboard. The first challenge was an unclear or incorrect description of an issue provided by the reporter. One way to address this challenge is to provide issue reporting guidelines. Additionally, practitioners could ask for clarification in a post on that particular issue. The second challenge was related to the willingness of a company to adopt the dashboard. To address this challenge, essential efforts must be made to increase awareness of human values. Providing regulations and standards (e.g. GDPR Wolford 2021) is one potential way to increase this awareness.

7 Threats to Validity

This section discusses the potential threats arising from the research method and the findings. This section uses the following validation criteria, which are considered suitable for qualitative research (Guba 1981; Stol et al. 2014; Cruzes and Dybå 2011):

Credibility: Possible threats to the credibility of this study could arise from the procedures used to collect data, develop interview questions, or select participants. Although the collected data was only from one source (interviews), the initial step of examining the literature before developing the dashboard prototype could increase the plausibility of our findings. To mitigate the threats of the interview questions, open-ended questions were used, and follow-up questions tailored to each participant's responses were asked. The use of issue discussions in the prototype of the exploration stage's interviews may have introduced bias into the participants' responses. This threat was mitigated by probing the participants to consider the possibilities of using other artefacts as the dashboard's source.

To reduce the possible threats resulting from the selection of participants, this study relied on the criteria for the recruitment of participants. The list of participants consisted of software practitioners with diverse roles, experiences, and work locations. Therefore, the participants had the right competencies to provide insight for the study. To mitigate the uneven number of participants in each role, the interviews also asked them to share their opinions from other roles' perspectives. This approach allowed for cross-validation of the findings across different roles.

Confirmability: A possible threat to the confirmability of this study might have been introduced by the definitions of human values, which have not specifically developed for software engineering. To mitigate this, some examples were provided to the participants to describe what a value could possibly mean in software engineering contexts (e.g. '*A user who values privacy may not choose an application with a bad privacy reputation*').

Participants were also allowed to reflect and translate values into contextualised software engineering definitions based on their experiences. The data analysis could have introduced another possible threat to the confirmability, as it was carried out primarily by the present author. This threat was mitigated by having other authors review and validate the codes/themes in several discussions.

Transferability: This study accepted that the findings cannot be generalised to all software organisations and practitioners. Different results might have been discovered if another group of participants had been included. However, this threat was reduced by involving a reasonable number of participants with various development roles and work locations. Furthermore, the data reached saturation during the parallel work of interviews and data analysis. This study also accepted that the relative importance of some specific values to others cannot be generalised because the entire list of values was not presented to the participants. This threat was mitigated by concluding that some values are more important than others. We also accepted that the results of the classification experiments were limited to the dataset that we used. Different results might have been obtained if other datasets had been used for the experiments. This study also realised that the use of only open source repositories made the results cannot be generalised to other projects.

8 Conclusion and Future Work

This study envisioned a values-driven dashboard and investigated whether it would help software practitioners address values during software development. This study consists of three stages, namely, exploration, development, and feedback stages. The exploration stage was conducted by providing a prototype of the dashboard and interviewing 10 software practitioners. This stage found that the participants acknowledged that a human values dashboard would be beneficial to them. The dashboard could raise awareness of values among development teams and inform values-based decision-making in project management. Supporting the idea of using artefacts as the dashboard source, practitioners suggested requirements documents and issue discussions as the most suitable artefacts for values identification in the dashboard. This stage of the study also received suggestions as a set of requirements to develop the envisioned dashboard.

In the development stage, a human values dashboard was developed as a proof-of-concept based on the requirements suggested in the previous stage. Then feedback interviews were conducted with 10 other practitioners to obtain their opinions on the dashboard. This study found that the human values dashboard could help focus attention and prioritise issues, in line with the findings from the exploration stage. Practitioners also suggested several potential challenges, such as a possible unclear or incorrect description of an issue by the reporter and the lack of willingness due to extra efforts required to deploy the dashboard in a company. Regarding the performance of the human values detector, the practitioners had different levels of tolerance, but all agreed that false positives were preferable to false negatives. Participants also made 16 suggestions to improve the dashboard.

The suggestions of practitioners and the results of this study could further improve the human values dashboard. Future studies could extend the dashboard to include other development artefacts suggested by interview participants. This direction has the potential to create a more comprehensive dashboard that covers the software development life cycle. Alternatively, because some suggestions can be integrated with an existing software repository, a future study could explore to what extent this integration is possible. We also realised that the performance of automated human values detection is quite low. Therefore, future studies could focus their efforts on improving the performance of the classifier, such as evaluating other classification approaches such as deep learning or newer classification features such as word embedding (Wang et al. 2021) or newer data imbalance handling approaches, such as transfer learning (Al-Stouhi and Reddy 2016). More work is also necessary to develop the classifier for specific values, such as face or hedonism. In terms of dashboard evaluation, future studies could evaluate it in a company setting using additional approaches, such as an observational study or a controlled experiment. These approaches could complement interviews to obtain a comprehensive evaluation of the dashboard in real world settings.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Data Availability The datasets used for the development of the classifier is available publicly at https://github.com/ovislabmonash/values-issues-dataset. The datasets of interviews analysed during the current study are not publicly available due to protect the individual privacy of the participants.

Declarations

Conflict of Interests The authors declared that they have no conflict of interest.

References

Al-Stouhi S, Reddy CK (2016) Transfer learning for class imbalance problems with inadequate data. Knowl Inf Syst 48(1):201–228

Alqahtani SS, Rilling J (2017) An ontology-based approach to automate tagging of software artifacts. In: Proceedings of the 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pp 169–174, https://doi.org/10.1109/ESEM.2017.25

Arya D, Wang W, Guo JL, Cheng J (2019) Analysis and detection of information types of open-source software issue discussions. In: Proceedings of the IEEE/ACM 41st International Conference on Software Engineering, pp 454–464, https://doi.org/10.1109/ICSE.2019.00058

Bao L, Lo D, Xia X, Wang X, Tian C (2016) How android app developers manage power consumption? In: Proceedings of the 13th International Conference on Mining Software Repositories, ACM, pp 37–48, https://doi.org/10.1145/2901739.2901748

Barn BS (2016) Do you own a volkswagen? values as non-functional requirements. In: Proceedings of the Joint 6th International Working Conference on Human-Centred Software Engineering and 8th International Working Conference on Human Error, Safety, and System Development, https://doi.org/10.1007/978-3-319-44902-9_10

Baysal O, Holmes R, Godfrey MW (2013) Developer dashboards: the need for qualitative analytics. IEEE Softw 30(4):46–52. https://doi.org/10.1109/MS.2013.66

Beitin BK (2012) Interview and sampling. In: The SAGE handbook of interview research: the complexity of the craft. Sage Thousand Oaks, CA, pp 243–254

Best S (2021) Whatsapp loses millions of users to rivals Telegram and Signal amid fears of increased data sharing with Facebook. https://www.dailymail.co.uk/sciencetech/article-9183553/Whatsapp-loses-MILLIONS-users-rivals-telegram-signal-ahead-privacy-policy-update.html. Accessed: 28 Apr 2021

Beyer S, Macho C, Di penta M, Pinzger M (2020) What kind of questions do developers ask on Stack Overflow? A comparison of automated approaches to classify posts into question categories. Empir Softw Eng 25(3):2258–2301. https://doi.org/10.1007/s10664-019-09758-x

Bird S, Klein E, Loper E (2021) Natural language processing with python. https://nltk.org/book. Accessed: 26 Nov 2021

Bird W (1998) The nature of managerial moral standards. J Bus Ethics 6(1)

Biswas E, Vijay-Shanker K, Pollock L (2019) Exploring word embedding techniques to improve sentiment analysis of software engineering texts. In: Proceedings of the ACM/IEEE 16th International Working Conference on Mining Software Repositories, pp 68–78, https://doi.org/10.1109/MSR.2019.00020

Braun V, Clarke V (2012) Thematic analysis. In: APA Handbook of research methods in psychology: vol. 2. Research designs, pp 57–71, https://doi.org/10.1037/13620-004

Canedo ED, Bonifácio R, Okimoto MV, Serebrenik A, Pinto G, Monteiro E (2020) Work practices and perceptions from women core developers in OSS communities. In: Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, https://doi.org/10.1145/3382494.3410682

Catolino G, Palomba F, Zaidman A, Ferrucci F (2019) Not all bugs are the same: understanding, characterizing, and classifying bug types. J Syst Softw 152:165–181. https://doi.org/10.1016/j.jss.2019.03.002

Cauldron (2021) Level up software development analytics. https://cauldron.io/explore. Accessed 18 Apr 2021

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: Synthetic Minority over-sampling technique. J Artif Intell Res 16:321–357. https://doi.org/10.1613/jair.953

Cheng AS, Fleischmann KR, Wang P, Ishita E, Oard DW (2010) Values of stakeholders in the net neutrality debate: applying content analysis to telecommunications policy. In: 2010 43Rd hawaii international conference on system sciences, https://doi.org/10.1109/HICSS.2010.434

Confessore N (2018) Cambridge Analytica and Facebook: The scandal and the fallout so far. https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html. Accessed: 28 Apr 2021

Cruzes DS, Dybå T (2011) Recommended steps for thematic synthesis in software engineering. In: Proceedings of the International Symposium on Empirical Software Engineering and Measurement, IEEE, pp 275–284, https://doi.org/10.1109/esem.2011.36

Ding J, Sun H, Wang X, Liu X (2018) Entity-level sentiment analysis of issue comments. In: Proceedings of the IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering, vol. 18, ACM, pp 7–13, https://doi.org/10.1145/3194932.3194935

Eluri VK, Sarkani S, Mazzuchi TA (2019) Open-source software survivability prediction using multi layer perceptron. EPiC Ser Comput 64:148–157. https://doi.org/10.29007/cmc6

Fan Q, Yu Y, Yin G, Wang T, Wang H (2017) Where is the road for issue reports classification based on text mining? In: Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp 121–130, https://doi.org/10.1109/ESEM.2017.19

Fischer F, Bottinger K, Xiao H, Stransky C, Acar Y, Backes M, Fahl S (2017) Stack overflow considered harmful? the impact of copy & paste on Android application security. In: Proceedings of the IEEE Symposium on Security and Privacy, pp 121–136, https://doi.org/10.1109/SP.2017.31

Friedman B, Kahn PH, Borning A, Huldtgren A (2013) Value sensitive design and information systems. In: Early engagement and new technologies: opening up the laboratory, pp 55–95, https://doi.org/10.1007/978-94-007-7844-3_4

Friedman B, Kahn Jr. PH, Borning A (2008) Value sensitive design and information systems. In: The handbook of information and computer ethics, pp 69–101

Fu B, Lin J, Liy L, Faloutsos C, Hong J, Sadeh N (2013) 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, pp 1276–1284, https://doi.org/10.1145/2487575.2488202

Gibler C, Crussell J, Erickson J, Chen H (2012) Androidleaks: automatically detecting potential privacy leaks in Android applications on a large scale. In: Proceedings of the International Conference on Trust and Trustworthy Computing, vol. 7344 LNCS, pp 291–307, https://doi.org/10.1007/978-3-642-30921-2_17

GitHub (2021a) About your organization dashboard. https://docs.github.com/en/organizations/collaborating-with-groups-in-organizations/about-your-organization-dashboard. Accessed 18 Apr 2021

GitHub (2021b) About your personal dashboard. https://docs.github.com/en/github/setting-up-and-managing-your-github-user-account/about-your-organizations/about-your-organization-dashboard. GitHub. Accessed 18 Apr 2021

GitHub (2021c) Autolinked references and urls. https://docs.github.com/en/github/writing-on-github/working-with-advanced-formatting/autolinked-references-and-urls. Accessed: 23 Sept 2021

GitHub (2021d) Github issues - project planning for developers. https://github.com/features/issues/. Accessed: 1 Oct 2021

GitHub (2021e) Mentioning people and teams. https://docs.github.com/en/github/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax#mentioning-people-and-teams . Accessed: 26 Nov 2021

Glinz M (2007) On non-functional requirements. In: Proceedings of the 15th IEEE International Requirements Engineering Conference, pp 21–26, https://doi.org/10.1109/RE.2007.45

Golzadeh M, Decan A, Legay D, Mens T (2021) A ground-truth dataset and classification model for detecting bots in GitHub issue and PR comments. J Syst Softw :175

Guba EG (1981) Criteria for assessing the trustworthiness of naturalistic inquiries. Educ Commun Technol J 29(2):75–91. https://doi.org/10.1007/BF02766777

Harris MA, Brookshire R, Chin AG (2016) Identifying factors influencing consumers' intent to install mobile applications. Int J Inf Manag 36(3):441–450. https://doi.org/10.1016/j.ijinfomgt.2016.02.004

Holmes T, Blackmore E, Hawkins R, Wakeford T (2011) The common cause handbook public interest research center

Hussain W, Shahin M, Hoda R, Whittle J, Perera H, Nurwidyantoro A, Shams RA, Oliver G (2022) How can human values be addressed in agile methods? A case study on SAFe IEEE Transactions on Software Engineering **Early access**

Ishita E, Fukuda S, Oga T, Oard DW, Fleischmann KR, Tomiura Y, Cheng AS (2019) Toward three-stage automation of annotation for human values. In: Information in contemporary society, pp 188–199, https://doi.org/10.1007/978-3-030-15742-5_18

Ishita E, Oard DW, Fleischmann KR, Templeton TC (2010) Investigating multi-label classification for human values. In: Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem

Ivanov V, Pischulin V, Rogers A, Succi G, Yi J, Zorin V (2018a) Design and validation of precooked developer dashboards. In: ESEC/FSE 2018 - Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ACM, pp 821–826, https://doi.org/10.1145/3236024.3275530

Ivanov V, Rogers A, Succi G, Yi J, Zorin V (2018b) Precooked developer dashboards: What to show and how to use. In: International Conference on Software Engineering: Companion, pp 402–403, https://doi.org/10.1016/j.asoc.2012.02.004

Janes A, Sillitti A, Succi G (2013) Effective dashboard design. Cutter IT J 26(1):17–24

Jha N, Mahmoud A (2019) Mining non-functional requirements from App store reviews. Empir Softw Eng 24(6):3659–3695. https://doi.org/10.1007/s10664-019-09716-7

Kikas R, Dumas M, Pfahl D (2016) Using dynamic and contextual features to predict issue lifetime in GitHub projects. In: Proceedings of the ACM/IEEE 13th Working Conference on Mining Software Repositories, pp 291–302, https://doi.org/10.1145/2901739.2901751

Kim S, Cho JI, Myeong HW, Lee DH (2012) A study on static analysis model of mobile application for privacy protection. In: Computer science and convergence, Springer, pp 529–540

Kujala S, Våånånen-Vainio-Mattila K (2009) Value of information systems and products: understanding the users' perspective and values. JITTA J Inf Technol Theory Appl 9(4):23

Kuznetsov K, Avdiienko V, Gorla A, Zeller A (2016) Checking app user interfaces against app descriptions. In: Proceedings of the International Workshop on App Market Analytics, pp 1–7, https://doi.org/10.1145/2993259.2993265

Leite L, Treude C, Filho FF (2015) UE dashboard: Awareness of unusual events in commit histories. In: 2015 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2015 - Proceedings, pp 978–981, https://doi.org/10.1145/2786805.2803184

Lemaître G, Nogueira F, Aridas CK (2017) Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. J Mach Learn Res 18:559–563

Li L, Bartel A, Bissyandé TF, Klein J, Traon YL, Arzt S, Rasthofer S, Bodden E, Octeau D, Mcdaniel P (2015) iccTA: Detecting inter-component privacy leaks in Android apps. In: Proceedings of the International Conference on Software Engineering, pp 280–291, https://doi.org/10.1109/ICSE.2015.48

Liu B (2020) Sentiment analysis, Mining Opinions, Sentiments, and Emotions. Cambridge University Press

Lȯpez L, Manzano M, Gȯmez C, Oriol M, Farrė C, Franch X, Martinez-Fernȧndez S, Vollmer AM (2021) qaSD: A quality-aware strategic dashboard for supporting decision makers in agile software development. Sci Comput Program 102568:202. https://doi.org/10.1016/j.scico.2020.102568

Ma Y, Fakhoury S, Christensen M, Arnaoudova V, Zogaan W, Mirakhorli M (2018) Automatic classification of software artifacts in open-source applications. In: Proceedings of the IEEE/ACM 15th International Conference on Mining Software Repositories, pp 414–425, https://doi.org/10.1145/3196398.3196446

Mairiza D, Zowghi D, Nurmuliani N (2010) An investigation into the notion of non-functional requirements. In: Proceedings of the ACM Symposium on Applied Computing, pp 311–317, https://doi.org/10.1145/1774088.1774153

Mautic (2022) Mautic community dashboard. https://dashboard.mautic.org/. Accessed 18 Apr 2021

Mohammed R, Rawashdeh J, Abdullah M (2020) Machine learning with oversampling and undersampling techniques: Overview study and experimental results. In: Proceedings of the 11th International Conference on Information and Communication Systems, pp 243–248, https://doi.org/10.1109/ICICS49469.2020.239556

Mougouei D, Perera H, Hussain W, Shams R, Whittle J (2018) 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, pp 780–784, https://doi.org/10.1145/3236024.3264843

Munaiah N, Kroh S, Cabrey C, Nagappan M (2017) Curating github for engineered software projects. Empir Softw Eng 22(6):3219–3253. https://doi.org/10.1007/s10664-017-9512-6

Naseri M, Borges NP, Zeller A, Rouvoy R (2019) Accessileaks: Investigating privacy leaks exposed by the Android accessibility service. In: Proceedings on Privacy Enhancing Technologies, pp 291–305, https://doi.org/10.2478/popets-2019-0031

Nurwidyantoro A (2022) On the presence of human values in software development artefacts: An evaluation of GitHub's issue discussions. Ph.D thesis

Nurwidyantoro A, Shahin M, Chaudron M, Hussain W, Perera H, Shams R, Whittle J (2022) Human Values Dashboard Feedback Questions. https://doi.org/10.6084/m9.figshare.19601938.v1. https://figshare.com/articles/online_resource/Human_Values_Dashboard_Feedback_Questions/19601938

Nurwidyantoro A, Shahin M, Chaudron M, Hussain W, Perera H, Shams R, Whittle J (2022) Human Values Dashboard Informed Consent. https://doi.org/10.6084/m9.figshare.21256467.v1. https://figshare.com/articles/online_resource/Human_Values_Dashboard_-_Informed_Consent/21256467

Nurwidyantoro A, Shahin M, Chaudron M, Hussain W, Perera H, Shams RA, Whittle J (2021a) Towards a human values dashboard for software development: an exploratory study. In: Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pp 1–12

Nurwidyantoro A, Shahin M, Chaudron MR, Hussain W, Shams R, Perera H, Oliver G, Whittle J (2021b) Human values in software development artefacts: A case study on issue discussions in three android applications. Information and Software Technology p 106731. https://doi.org/10.1016/j.infsof.2021.106731 . https://www.sciencedirect.com/science/article/pii/S0950584921001828

Ortu M, Murgia A, Destefanis G, Tourani P, Tonelli R, Marchesi M, Adams B (2016) The emotional side of software developers in JIRA. In: Proceedings of the 13th International Conference on Mining Software Repositories, pp 480–483, https://doi.org/10.1145/2901739.2903505

Ournani Z, Rouvoy R, Rust P, Penhoat J (2020) On reducing the energy consumption of software: From hurdles to requirements. In: Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp 1–12, https://doi.org/10.1145/3382494.3410678

Padurariu C, Breaban ME (2019) Dealing with data imbalance in text classification. Proc Comput Sci 159:736–745. https://doi.org/10.1016/j.procs.2019.09.229

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830

Pereira R, Carcao T, Couto M, Cunha J, Fernandes JP, Saraiva J (2017) Helping programmers improve the energy efficiency of source code. In: Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017, pp 238–240, https://doi.org/10.1109/ICSE-C.2017.80

Perera H, Hussain W, Whittle J, Nurwidyantoro A, Mougouei D, Shams RA, Oliver G (2020) 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, ICSE '20, pp 409–420, https://doi.org/10.1145/3377811.3380393

Perera H, Mussbacher G, Hussain W, Ara Shams R, Nurwidyantoro A, Whittle J (2020) Continual Human Value Analysis in Software Development: A Goal Model Based Approach. In: Proceedings of the IEEE International Conference on Requirements Engineering, pp 192–203, https://doi.org/10.1109/RE48521.2020.00030

Pletea D, Vasilescu B, Serebrenik A (2014) Security and emotion: sentiment analysis of security discussions on GitHub. In: Proceedings of the 11th Working Conference on Mining Software Repositories, pp 348–351, https://doi.org/10.1145/2597073.2597117

Prana GAA, Treude C, Thung F, Atapattu T, Lo D (2019) Categorizing the content of GitHub README files. Empir Softw Eng 24(3):1296–1327. https://doi.org/10.1007/s10664-018-9660-3

Rezaei Nasab A, Shahin M, Liang P, Basiri ME, Hoseyni Raviz SA, Khalajzadeh H, Waseem M, Naseri A (2021) Automated identification of security discussions in microservices systems: Industrial surveys and experiments. J Syst Softw 181

Rokeach M (1973) The Nature of Human Values Free press

Samrose S, McDuf D (2021) Meetingcoach: an intelligent dashboard for supporting efective and inclusive meetings. In: Conference on human factors in computing systems - proceedings, https://doi.org/10.1145/3411764.3445615

Schapiro AA, Bacchi U (2020) U.S. protests fuel calls for ban on racially biased facial recognition tools. https://www.reuters.com/article/us-usa-protests-tech-trfn-idUSKBN23b3b5. Accessed: 28 Apr 2021

Schütze H, Manning CD, Raghavan P (2008) Introduction to information retrieval cambridge university press cambridge

Schwartz SH (1994) Are there universal aspects in the structure and contents of human values? J Soc Issues 50(4):19–45. https://doi.org/10.1111/j.1540-4560.1994.tb01196.x

Schwartz SH (2012) An overview of the Schwartz Theory of Basic Values. Online Read Psychol Cult 2(1):12–13. https://doi.org/10.9707/2307-0919.1116

Schwartz SH (2017) The refined theory of basic values. In: Values and behavior: Taking a cross cultural perspective, pp 51–72, https://doi.org/10.1007/978-3-319-56352-7_3

Sharma VS, Ramnani RR, Sengupta S (2014) A framework for identifying and analyzing non-functional requirements from text. In: Proceedings of the 4th International Workshop on Twin Peaks of Requirements and Architecture, pp 1–8, https://doi.org/10.1145/2593861.2593862

Slavin R, Wang X, Hosseini MB, Hester J, Krishnan R, Bhatia J, Breaux TD, Niu J (2016) Toward a framework for detecting privacy policy violations in Android application code. In: Proceedings of the 38th IEEE International Conference on Software Engineering, pp 25–36, https://doi.org/10.1145/2884781.2884855

Song Y, Chaparro O (2020) BEE: A tool for structuring and analyzing bug reports. In: Proceedings of the 28th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp 1551–1555, https://doi.org/10.1145/3368089.3417928

Stol KJ, Avgeriou P, Babar MA, Lucas Y, Fitzgerald B (2014) Key factors for adopting inner source. ACM Transactions on Software Engineering and Methodology 23(2)

Takayama Y, Tomiura Y, Fleischmann KR, Cheng AS, Oard DW, Ishita E (2015) Automatic dictionary extraction and content analysis associated with human values. Inf Eng Expr 1(4):107–118. https://doi.org/10.52731/iee.v1.i4.34

Takayama Y, Tomiura Y, Fleischmann KR, Cheng AS, Oard DW, Ishita E (2016) An automatic dictionary extraction and annotation method using simulated annealing for detecting human values. In: Proceedings of the 2015 IIAI 4th International Congress on Advanced Applied Informatics, pp 177–182, https://doi.org/10.1109/IIAI-AAI.2015.268

Takayama Y, Tomiura Y, Ishita E, Oard DW, Fleischmann KR, Cheng AS (2014) A word-scale probabilistic latent variable model for detecting human values. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp 1489–1498, https://doi.org/10.1145/2661829.2661966

Takayama Y, Tomiura Y, Ishita E, Wang Z, Oard DW, Fleischmann KR, Cheng AS (2013) Improving automatic sentence-level annotation of human values using augmented feature vectors. In: Proceedings of the Conference of the Pacific Association for Computational Linguistics (PACLING)

Thelwall M, Buckley K, Paltoglou G, Cai D, Kappas A (2010) Sentiment strength detection in short informal text. J Am Soc Inf Sci Technol 61(12):2544–2558. https://doi.org/10.1002/asi.21416

Thew S, Sutcliffe A (2018) Value-based requirements engineering: method and experience. Requir Eng 23(4):443–464. https://doi.org/10.1007/s00766-017-0273-y

Thiruvathukal GK, Shilpika HNJ, Läufer K (2018) Metrics dashboard: A hosted platform for software quality metrics

Tómasdóttir KF, Aniche M, Van Deursen A (2020) The adoption of JavaScript linters in practice: A case study on ESLint. IEEE Trans Softw Eng 46(8):863–891. https://doi.org/10.1109/TSE.2018.2871058

Tomasdottir KF, Aniche M, Van Deursen A (2017) Why and how JavaScript developers use linters. In: Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, APA, pp 578–589, https://doi.org/10.1109/ASE.2017.8115668

Treude C, Storey MA (2009) Concernlines: A timeline view of co-occurring concerns. In: Proceedings of the 31st International Conference on Software Engineering, pp 575–578

Treude C, Storey MA (2010) Awareness 2.0: Staying aware of projects, developers and tasks using dashboards and feeds. In: Proceedings - International Conference on Software Engineering, pp 365–374. https://doi.org/10.1145/1806799.1806854. http://www.facebook.com/

Trockman A, Van Tonder R, Vasilescu B (2019) Striking gold in software repositories? An econometric study of cryptocurrencies on GitHub. In: Proceedings of the ACM/IEEE 16th International Working Conference on Mining Software Repositories, vol. 2019-May, pp 181–185, https://doi.org/10.1109/MSR.2019.00036

Viega J, Bloch JT, Kohno T, McGraw G (2002) Token-based scanning of source code for security problems. ACM Trans Inf Syst Secur 5(3):238–261. https://doi.org/10.1145/545186.545188

Vivian R, Tarmazdi H, Falkner K, Falkner N, Szabo C (2015) The Development of a Dashboard Tool for Visualising Online Teamwork Discussions. In: Proceedings - International Conference on Software Engineering, vol. 2, pp 380–388. https://doi.org/10.1109/ICSE.2015.170. https://www.researchgate.net/publication/277022554

Wang HY, Liao C, Yang LH (2013) What affects mobile application use? the roles of consumption values. International Journal of Marketing Studies 5(2)

Wang J, Zhang X, Chen L (2021) How well do pre-trained contextual language representations recommend labels for github issues? Knowl-Based Syst 107476:232

Wexler S, Shaffer J, Cotgreave A (2017) The big book of dashboards: Visualizing your data using real world business scenarios

Winter E, Forshaw S, Ferrario MA (2018) Measuring human values in software engineering. In: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp 8–11, https://doi.org/10.1145/3239235.3267427

Wolford B (2021) What is GDPR, the EU's new data protection law? https://gdpr.eu/what-is-gdpr/. Accessed: 1 Oct 2021

Yao J, Shepperd M (2020) Assessing software defection prediction performance: Why using the Matthews correlation coefficient matters. In: Proceedings of the Evaluation and Assessment in Software Engineering, pp 120–129, https://doi.org/10.1145/3383219.3383232

Yao J, Shepperd M (2021) The impact of using biased performance metrics on software defect prediction research. Inf Softw Technol 139:106664. https://doi.org/10.1016/j.infsof.2021.106664

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Arif Nurwidyantoro has completed his PhD at Monash University and is currently a Lecturer at Universitas Gadjah Mada, Indonesia. His research interests include data analytics and software engineering. His current research focuses on analyzing software repositories to support software development.

Mojtaba Shahin is a Software Engineering Lecturer at RMIT University, Australia. Previously, he was a Research Fellow at Monash University. His research interests reside in Empirical Software Engineering, Human and Social Aspects of Software Engineering, Software Architecture, and Secure Software Engineering. He has published over 45 papers in premier software engineering journals and conferences, including TSE, EMSE, JSS, ICSE, and MSR. He received an ACM SIGSOFT Distinguished Paper Award (MSR 2022). He completed his PhD study at the University of Adelaide, Australia.

Michel Chaudron is a full professor of Software Engineering at the Department of Computer Science of Eindhoven University of Technology, The Netherlands. Previously he was a professor at the joint dept of Computer Science and Engineering that is part of both Chalmers and Gothenburg University in Sweden. His interests are in Software Architecture, Software Design and Software Modeling, particularly in empirical studies and applications of AI to the aforementioned topics. He is one of the creators of the Lindholmen Dataset of UML models. He is on the editorial board of the Software Quality Journal and the steering committee of the Euromicro Software Engineering and Advanced Applications Conference (SEAA) since 2015.

Waqar Hussain is a Senior Research Scientist at CSIRO's Data61. His research interests include ethical artificial intelligence (AI), human-centric software engineering. His current research is focused on operationalization and evaluation of fairness in AI based systems.

Harsha Perera is a postdoctoral research fellow at CSIRO's Data61. He received his PhD from Monash University, Australia in 2022. In his thesis, he introduced a framework to operationalise human values in requirements engineering. His current research interests include risk assessment and requirements engineering for Responsible AI.

Rifat Ara Shams is a Postdoctoral Fellow at CSIRO's Data61, the digital technologies and data science arm of Australia's national science agency. She is working on Diversity and Inclusion in AI and Requirements Engineering for Responsible AI. She completed her PhD from Monash University, Australia, on operationalizing human values in mobile applications.

Jon Whittle is the director of CSIRO's Data61, the digital technologies and data science arm of Australia's national science agency. He is also an adjunct (full) professor with the Faculty of Information Technology, Monash University, Melbourne. His research interests include the intersection of software engineering and human-computer interaction. He is best known for his work in model-driven development, aspect-oriented modelling, digital technologies for social good, and values in software.

Affiliations

Arif Nurwidyantoro[1,2] · Mojtaba Shahin[3] · Michel Chaudron[4] · Waqar Hussain[1,5] · Harsha Perera[5] · Rifat Ara Shams[5] · Jon Whittle[5]

Mojtaba Shahin
mojtaba.shahin@rmit.edu.au

Michel Chaudron
m.r.v.chaudron@tue.nl

Waqar Hussain
waqar.hussain@monash.edu; waqar.hussain@data61.csiro.au

Harsha Perera
harsha.perera@data61.csiro.au

Rifat Ara Shams
rifat.shams@csiro.au

Jon Whittle
jon.whittle@data61.csiro.au

[1] Department of Software Systems and Cybersecurity, Faculty of IT, Monash University, 3800, Clayton, Australia

[2] Department of Computer Science and Electronics, Universitas Gadjah Mada, 55281, Yogyakarta, Indonesia

[3] School of Computing Technologies, RMIT University, 3000, Melbourne, Australia

[4] Department of Mathematics and Computer Science, Eindhoven University of Technology, 5600, Eindhoven, Netherlands

[5] CSIRO's Data61, 3168, Clayton, Australia