

Advancing the Study of Human Values in Software Engineering

Emily Winter, Stephen Forshaw, Lucy Hunt, Maria Angela Ferrario
School of Computing and Communications, Lancaster University, UK
e.winter;s.w.forshaw;l.hunt1;m.a.ferrario@lancaster.ac.uk

ABSTRACT

This paper makes the case for the study of human values in Software Engineering (SE) as a highly important emerging area of research with significant societal implications. We offer two key principles in order to advance this research agenda: firstly, the significance of values as distinguished from, though connected to, ethics; and secondly, the need for clear theoretical frameworks for values study. We provide the emerging findings from an initial study (N=12 participants) using a Values Q-Sort tool that was designed in accordance with these two principles. We conclude with discussion around lessons learnt, ongoing challenges, and future directions.

CCS CONCEPTS

• **Social and professional topics** → **Computing profession**; • **Software and its engineering**;

KEYWORDS

human values, mixed methods, qualitative study, software industry

ACM Reference Format:

Emily Winter, Stephen Forshaw, Lucy Hunt, Maria Angela Ferrario. 2019. Advancing the Study of Human Values in Software Engineering. In . ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION- WHY VALUES MATTER

The study of human factors in software engineering (SE) is a growing research area. Contributions to this field include studies of: motivation and satisfaction [9]; managing radical software engineers [6]; and successful software engineering practice [21] [28]. Such research recognises that SE is not just about technical skills and competence but is an inherently human endeavour, shaped by such elements as workplace cultures and norms [20], emotions [26] and teamwork [21]. There has also been a recent emergence of research that considers individual values, of which *fairness* in software systems has been the most prominent, especially as applied in studies of algorithmic bias [13]. This is important work, but does not consider the *range* of values driving SE practices currently, and, most importantly, their interrelationships. Our paper responds to this gap by laying out two important principles for values study, and providing the emerging results from a pilot study that used a tool designed to respond to these principles.

In order to understand the cultures in which software is produced and the decisions that are made in SE, we need to understand software engineers' values. Values play a significant role in influencing decision-making and behaviors and these actions impact upon software products. This is significant given the considerable impact of software systems on contemporary, placing software engineers

in a position of particular societal responsibility. Recent scandals around data breaches, for example, demonstrate what can occur when this societal responsibility is not taken sufficiently seriously [8].

The core motivation of our research is twofold. Firstly, we aim to understand the values that are driving software production, primarily, at this stage, through the creation of tools and methods for the systematic study of values in SE. Secondly, through conducting this research, we aim to contribute to a more values-aware software industry, in which values can be articulated, and reflected and deliberated upon. We intend to contribute to on-going and emerging efforts aimed at promoting a deep cultural shift in the SE community, by fostering a practice-based and reflective awareness of software systems' social impact. Specifically, we aim to encourage fellow researchers to target these efforts to parts of our community where technology is still considered neutral and computing professionals 'just engineers'.

This concern around the unexpected consequences and social impacts of software systems represents a different motivation to many studies of human factors in SE that are primarily concerned with improving software engineers' working practices and fostering more productive, efficient workplaces. Whilst this is important, particularly as software systems become more complex, our own research is underpinned by wider societal concern, and by the aim to develop tools and techniques that can help address these challenges in everyday SE practice.

2 RELATED WORK AND PRINCIPLES FOR ADVANCING THE STUDY OF HUMAN VALUES IN SE

There is a small amount of emerging work in the study of human values in SE. For example, Mougouei et al. lay out a roadmap for operationalising human values in SE, which focuses on "(i) establishing practical definitions for human values, (ii) integrating values into software design, and (iii) measuring values in the software development life cycle" [25]. This is a useful contribution. However, our starting point is different, hoping first to understand the values driving SE and encourage values reflection before attempting to operationalise human values. Against claims that values are 'mushy' [24], we offer two principles for robust and systematic values study.

2.1 Values as Distinct from Ethics

Firstly, the study of human values should be distinguished from the study of computing ethics, though the two are clearly connected. The study of computing ethics is well established and includes studies within SE [34]. The recently updated ACM Code of Ethics [1] and the ongoing IEEE Ethics in Action initiative [17] are testament to the expertise within this field. However, ethics are usually fairly abstract [10], though the updated ACM Code of Ethics has made

welcome efforts to incorporate more practical guidance [1]. More significantly, ethics do not fully cover what an individual, group, or wider society believe to be important.

Values are criteria that people use “to select and justify actions and evaluate people (including the self) and events” [29]. Such guidelines may include achievement, prestige and wealth, values that do not have clear ethical import but still influence decision-making processes and actions. For example, Google employees’ letter demanding that their CEO withdraw Google from Project Maven not only appealed to their employer’s “moral and ethical responsibility”, but also to the potential “damage to the Google brand and its ability to compete for talent” [16]. Market value and public image are not conventionally thought of as having ethical import, yet are values that Google employees perceived to be of importance to their organization and used to drive action.

A variety of human values - both ones that would be commonly thought of as “ethical”, such as valuing social justice when working with disadvantaged others, and those that would not be, such as valuing financial success - impact upon the production of software. While the influential field of Values Sensitive Design, for example, focuses predominantly on values with ‘ethical import’ [11] [12], to not consider “non-ethical” values seriously fails to account for a significant part of the picture of SE, and the complex decision-making processes and values trade-offs that may result.

2.2 Theoretical Frameworks for the Systematic Study of Human Values

Secondly, values should be studied within a context of clear and robust theoretical frameworks. Shilton et al. have argued that within social computing “there is a lack of precision in how the construct of values is defined, applied, and investigated” [32]. We do not currently consider it necessary to propose a new theory of values; instead, we examined existing frameworks [3], and adopted an established, empirically-grounded one as our starting point. Social psychology is a discipline with a rich and long-standing tradition of values study. In particular, Shalom Schwartz’s universal values model [29] [31] has been very influential. Schwartz’s values model is based on survey research in 82 countries [30] - it identifies a series of distinct values that are recognised across cultures and are “structured in similar ways across culturally diverse groups” [30]. The major contribution of Schwartz’s work is not just the list of values he offers, but the patterns of relationships. This pattern consists of a circle centred around two oppositional axes: self-enhancement vs. self-transcendence and openness vs. conservation. (See Figure 1.) Schwartz explains, “the closer any two values in either direction around the circle, the more similar their underlying motivations; the more distant, the more antagonistic their motivations” [30]. Values thus operate in a universal, specifically organized way, though of course “individuals and groups differ substantially in the relative importance they attribute to the values” [30]. What the model suggests, however, is that if an individual values tradition (on the conservation axis), they are likely to also value conformity, but not to value highly hedonism, self-direction and stimulation (on the opposite- openness- axis). Within computing, Schwartz has been used in studies on technology end-users’ needs, though with little

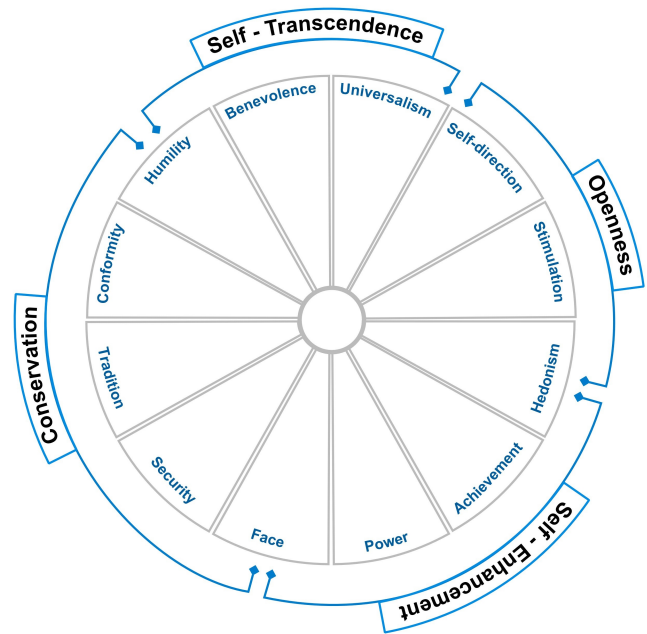


Figure 1: Schwartz’s Universal Values Model

emphasis on the relational nature of his model [19] [27], and in document analysis of industry IT for sustainability [18].

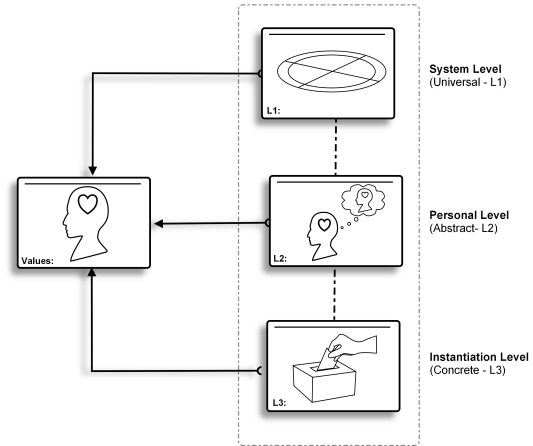


Figure 2: Three Levels of Values

In contrast to such work, our research builds on but also moves beyond Schwartz. We do this by drawing on research by social psychologists Maio et al. that considers values as mental representations that can be studied at three interconnected levels [22]: the system, or universal level - the level at which Schwartz’s model operates (L1); the personal, or abstract level (L2); and the instantiation, or concrete behavioral level (L3) (see Figure 2). Our work adopts this framework as a starting point for investigating values in a systematic way that takes into account their complexity.

Table 1: VQ-S Statements, associated Schwartz values and associated 2018 Code of Ethics statement

N	Q-Sort statement (it is important to me...)	Schwartz value	ACM Code Statement
1	to be given the freedom to produce new ideas, inventions and creative works	Self-direction: autonomy of thought	1.5
2	that the software I develop is robustly and usably secure	Security: societal	2.9
3	to enjoy the process of developing software	Hedonism	No associated statement
4	that I do not annoy or upset anyone in the course of my work	Conformity: interpersonal	No associated statement
5	that the public good is the central concern of all professional computing work	Universalism: concern	3.1
6	that the software I develop influences the end user	Power: over resources	No associated statement
7	that I credit fully the work of others and refrain from taking undue credit	Humility	ACM and IEEE-CS 1999 7.03
8	that I identify and address any environmental issues in my work	Universalism: nature	ACM and IEEE-CS 1999 3.03
9	that my work is respected	Face (i.e. public image)	No associated statement
10	that I am allowed to take risks when developing software	Stimulation	No associated statement
11	to improve public awareness and understanding of software	Benevolence: care	2.7
12	that the software I develop is commercially successful	Power: resources	No associated statement
13	that my workplace promotes my physical safety and psychological well-being	Security: personal	3.3
14	that I do not discriminate against others when developing software	Universalism: tolerance	1.4
15	that I know and apply industry rules when developing software	Conformity: rules	2.3
16	that I make own decisions when developing software	Self-direction: autonomy of action	No associated statement
17	that I personally achieve high quality in software design and production	Achievement	2.1
18	to uphold, promote and respect the principles of my industry	Tradition	4.1
19	to be an honest and trustworthy colleague	Benevolence: dependability	1.3

Considering the three levels of values in an SE context, L1 refers to the patterns of values relationships that software engineers hold. For example, a software engineer that highly values working autonomously (self-direction) would, according to Schwartz’s model also value taking risks (stimulation) but be less likely to highly value following industry rules (conformity). At L2, we can expect software engineers to have different interpretations of what achieving high quality software (achievement) looks like; for example, code that does the job vs. ‘elegant’ code. At L3, values are instantiated through particular actions and behaviors. A search engine company concerned with privacy, for example, may ensure that their system never logs user queries.

3 BUILDING NEW TOOLS TO STUDY HUMAN VALUES IN SE: THE VALUES Q-SORT

Using Schwartz and Maio as our theoretical framework, we have developed a series of tools to measure values in SE and elicit values narratives at the three levels. One of these is the Q-Sort, the method we have used most so far. The Q-Sort involves asking participants to sort a series of statements onto a grid according to their level of agreement with each statement. Q methodology seemed to correspond well with Maio et al.’s three levels. By asking participants to rank statements according to their level of agreement with them, the Q-Sort demonstrates values patterns and relationships (L1), while the accompanying interview provides insight as to how participants interpret these values (L2). In addition, we asked participants to fill in the Q-Sort for a specific project or product on which they were working in order to ground the exercise and encourage examples of values at the instantiation or behavioural level (L3). We also designed the Q-Sort statements using Schwartz’s values types as a framework (see below).

The Q-Sort is an established mixed method that was developed in the 1930s by the psychologist and physicist William Stephenson [33]. It was designed for the systematic study of subjectivity by providing structure to subjective opinions [35]. One significant benefit of the Q-Sort is that it yields both qualitative and quantitative data. The sort is accompanied by a semi-structured interview,

while the results of multiple sorts can be statistically analyzed. The Q-Sort also produces statistically valid results with smaller samples than traditional surveys and is hence regarded as “a robust tool for investigating human subjectivity” [5].

The Q-Sort differs from surveys insofar as participants’ responses are not free but have to be considered in relation to each other, forcing trade-offs and decisions between statements. In addition, the emphasis of the analysis is different: “In Q research, subjects and variables are inverted. Thus, the ‘subjects’ of a Q study are the Q statements and the ‘variables’ are the people- more specifically, their Q sorts. Q researchers look for patterns across the variables (e.g. people’s Q sorts) for each subject (e.g. Q statement)” [36]. Q-Sorts use q-statistics based on factor rotation and work best with samples that are smaller than the number of statements used in the sort, the reverse of traditional survey research that requires a greater number of participants than there are survey items [35].

3.1 Designing the Values Q-Sort

We designed the Values Q-Sort to be relatable for software industry professionals and respond to the difficulty posed by Miller and Larson that language used to articulate human values is less precise than technical language [24]. The systematic nature of the Q-Sort exercise offered an attractive option for engaging software engineers in a discussion about values in an accessible and time-efficient way.

Our starting point was Schwartz’s universal values model, following our principle of the importance of robust values theory. We piloted with two people an already existing Q-Sort with statements derived from Schwartz’s 57-item values survey. The pilot found that 57 statements were too cognitively overwhelming and time-consuming for participants. In response we turned to the latest available draft (the third) of the ACM Code of Ethics [1], and dual-coded its principles according to the latest version of Schwartz’s values model, which identifies 19 distinct values types and sub-types. The rationale for this was twofold: firstly, to utilize language generated by computing professionals; and secondly, to

Table 2: Values Q-Sort Participants

Name	Sex	Years in SE	Sector	Job Title
Alex	M	10-15	Public-Research	Senior RA
Dan	M	10-15	Public-IT support	IT Dev
Donna	F	10-15	Public-Media	SE
Joshua	M	5-10	Public-Media	Senior SE
Casper	M	5-10	Private-Freelance	IT Dev
John	M	>30	Private-Industry	Consultant
Richard	M	<5	Private-Industry	SE
Sean	M	10-15	Public-Media	Senior SE
Ali	M	10-15	Public-Media	Senior SE
Mike	M	10-15	Public-Research	Snr Lecturer
WW	W	15-20	Public-Research	Lecturer
Jesse	M	10-15	Private-Freelance	IT Dev

iteratively examine and reflect upon, through both the V-QS design process and results analysis, the relationship between values and ethics. The dual-coding produced 80 per cent agreement, and remaining discrepancies were discussed by the two researchers. Reference was also made to the 1992 IEEE-ACM CS Code of Ethics [15] as an additional source of statements.

We chose the most appropriate Code of Ethics principle for each value type as the basis of a Q-Sort statement. All value types (though not all sub-types) were represented in the Code except four ('Hedonism', 'Stimulation', 'Power', and 'Face'). In such cases, additional statements were developed with reference to Schwartz's values [31]. The resulting statements went through a second piloting cycle with four computing researchers at another institution before being finalized in their current form (Table 1).

3.2 Values Q-Sort Participants

We have carried out Q-Sorts and accompanying semi-structured interviews with 12 software practitioners, from a variety of sectors and with varying levels of expertise (see Table 2). This was a purposeful sample, chosen to provide a proof of concept for the tool's applicability in SE across sectors, ranging from industry to public sector to freelancers. This number of participants corresponds to advice that there should be fewer participants than Q-Sort statements [35]. Interviews ranged in length from 45 minutes to 1.5 hours.

This research received full ethical clearance from the authors' university. At the beginning of the interview, informed consent was obtained and participants were asked about their role, career trajectory, and current projects. Upon completion, questions were asked about the sort, focusing particularly on participants' top- and bottom-ranked statements. Finally, feedback about the tool was collected. From the 12 interviews, we were able to perform Q statistics on the data.

3.3 Values Q-Sort Quantitative Results

The Q-Sort provides quantitative and qualitative data. The quantitative results from our 12-participant study are reported more fully in [37], and so are only briefly summarised here, before moving on to the insight provided by the qualitative data. The qualitative

results complement the quantitative data, but also caution against generalisation, demonstrating a more complex picture.

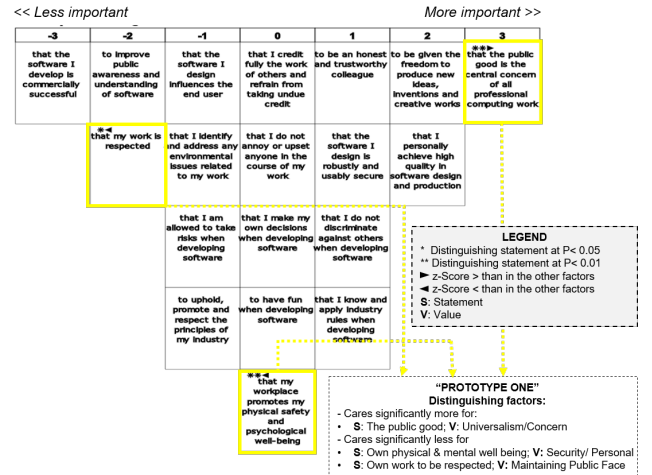
The Q-Sort data from the 12 participants were inputted into an online Q-analysis program that performs factor analysis to identify "patterns of similarity in the Q-Sort configurations produced" [35]. Factor extraction allows for the emergence of statistically significant patterns and is produced through centroid factor analysis. From our data, three factors emerged as statistically significant, with Eigenvalues greater than one. These were rotated using varimax rotation, which statistically positions the factors so that they cover the maximum amount of variance and ensures that each participant loads onto only one factor [35]. The three rotated factors each represent an abstract type of software practitioner. From analysis of the composite Q-Sorts produced for each factor (see Figure 3), we call these ideal-type software engineers the Intrinsically-driven Altruist (Factor 1), the Autonomous, Nonconforming Risk-taker (Factor 2), and the Extrinsically-driven Fun-lover (Factor 3).

Key features of the three factors

The Intrinsically-driven Altruist. Three participants loaded on to this factor (P2, P10, P12). S5 (public good) was ranked higher than in other factors. S13 (physical safety and psychological wellbeing) and S9 (work being respected) were ranked lower than in other factors.

The Autonomous, Nonconforming Risk-taker. Two participants loaded on to this factor (P1 and P5). S10 (risk-taking), S6 (influencing the end-user), S8 (environmental issues) and S12 (commercial success) were ranked higher than in other factors. S19 (honest and trustworthy) and S17 (achieving high quality) were ranked lower than in other factors.

The Extrinsically-driven Fun-lover. Three participants loaded on to this factor (P4, P8, P11). S3 (enjoyment), S9 (work being respected) and S18 (industry principles) were ranked higher than in other factors. P1 (freedom of thought) and P5 (public good) were ranked lower than in other factors.

**Figure 3: Composite Q-Sort for Factor 1- The Intrinsically-driven Altruist**

3.4 Values Q-Sort: Qualitative Results

Whilst the factors reveal statistically constructed types of values perception and their relationships to each other (thus offering insight at L1), the statistics alone don't provide insight into what these values actually mean to people (L2), and how they are enacted (L3). For this, we need to look at the qualitative data generated by the semi-structured interviews. The interviews were qualitatively analyzed by manually extracting from each interview what had been said about each of the 19 Q-Sort statements, creating what we call "values slices".

Within each values slice, corroborating information was included, such as the factor the participant loaded on to (where this was the case) and how each participant ranked the statement. How each participant interpreted, reflected upon and reacted to the statement was then thematically analyzed using Atlas T.I. [2]. Space does not permit a full exploration of the qualitative results, the 12 interviews having produced almost 100,000 words of transcribed material. Instead, we consider emerging key themes in relation to N5 (Universalism- public good) and N9 (Face- respect), distinguishing statements (i.e. statistically significant) for each of the three factors. This brief discussion is not exhaustive but rather intended to be indicative of the capacity of the Values Q-Sort to provide data at L2 and L3, and to demonstrate the complexity of values when they are studied at these three levels, defying any simplistic or generalised interpretation of the statistics.

Universalism- public good. The public good statement is interesting to examine because the "public good" is a central concept of the ACM Code of Ethics from which the statements were constructed. It was also a statement that exhibited almost all possible rankings, from -2 to +3 (-3 and +3 representing the "extremes" on the Q-Sort grid). It also demonstrated highly varied interpretations.

Whose public good? Participants whose responses fell into this theme felt confused about what the public good might actually constitute and to whose public good the statement might refer. Jesse, for example, commented, "who decides what is public good, right? [...] what is public good? Because each individual might...what might be good for someone might not be good for others", while Richard stated "the thing about public good is different people have different ideas of public good". Personal attitudes to this value were complicated by their relationship with the perceived attitudes of others.

Public good as not wasting resources. This cluster of respondents considered the public good very concretely in the context of being paid with public sector money, and thus not wasting resources (temporal or material). This was expressed particularly strikingly by Ali, who stated: "I need to deliver. I need to work. [...] I shouldn't just be selfish and just misuse trust that the organization has given me [...] I should think about the people who are paying [my] salary". Mike also held a view of the need not to waste public funds, but interpreted the implications of this more broadly: "we are publically funded, we are public servants, even if we are not called public servants [...] it is at the expense of public funds and therefore I want to use it in a way that is beneficial to society". From similar perspectives (L2), Ali and Mike came to different conclusions about the correct instantiation (L3) of this value.

"Bad" software vs. neutral software. Thirdly, some participants drew a distinction between software development that was clearly negative or controversial (e.g. the on-line gambling industry) and other forms of software development that were seen as neutral. Sean, for example, commented "there's a big difference between something being like specifically against the public good and something just being sort of fairly neutral, middle of the road". Other participants had a different viewpoint. Richard, for example, commented: "what really kind of grinds my gears is when people say 'I'm just an engineer'". Richard's comment is interesting because it not only confirms the perception that some developers lack values awareness in software production, but also suggests that some computing practitioners contest the "technology neutrality" view.

This early, indicative discussion highlights the complexity in articulating values and the variety of personal meanings ascribed to them. Both Richard and Ali placed S5 in the +2 column of their grid (L1); however, their understandings of it were very different (L2). Richard was conscious that other people would have different conceptions of the public good to him, but ranked S5 highly based on his own definition and his disinclination to be involved in anything that harmed this, i.e. gambling. Whilst the statement has clear ethical implications, Ali interpreted these narrowly in terms of the instantiation (L3) that he identified (not wasting money).

Work being respected. Almost directly opposite S5 (universalism- public good) in the Schwartz circular model (L1) is S9 (face- public image), which is situated at the intersection of "self-enhancement" and "conservation". It thus represents an interesting value to compare with S5.

Respect for the work or the person? Joshua considered respect for the person and respect for the work as highly interlinked. He explains: "I think that the work that I do is a reflection of myself almost and [...] so I suppose if my work is respected than I'm respected to some extent". Richard, however, distinguished between these two elements of respect: "it's still important to respect the work they do, but respect the person first and foremost, because the person's physical safety and their psychological wellbeing [...] that to me is much more important than respecting the work they do". Here, Richard makes an explicit values trade-off, expressing a preference for S13 (Security- personal). In so doing, he demonstrates a very different attitude (L2) to S9 than Joshua.

Signs of being respected. Several participants were able to identify signs of their work being respected. Joshua, for example, highlighted respect from his peers, clients, and software users, while John commented on the importance of people caring about the work and his suggestions being "treated seriously". Sean meanwhile highlighted two distinct domains in which respect for his work was manifested differently. From his team, he interpreted respect as "when they're coming to you and asking you questions, 'cause they know that you know your stuff". From a wider organizational perspective, the instantiation of such respect was manifested in being paid well and equivalent to people "doing the same level of work to the same quality". For Joshua and Sean, the concrete evidence (L3) of being respected was sought in different things. This also points to the fact that values are instantiated not just by personal behavior but also by behaviors of others.

However, WW was frustrated by the personal significance she placed upon external respect for her work: "that's something that

I am personally working on ‘cause if it is that I respect myself and I value what I do why is it that my external environment has that effect?”. WW ranked S9 at +1 and her Q-Sort loaded onto the same factor as Sean and Joshua. However, in her frustration at her placement of S9, she demonstrates a very different attitude (L2) to the value.

Effects of being respected. Sean found his work more enjoyable when he knew it was being respected, while both WW and Ali found that being respected gave them motivation to work. For Ali this had ethical implications: “if somebody respects my work I will be more motivated, I would be producing better software, so this is where business and ethics overlap”. Interestingly here, Ali identifies a value that lies on the self-enhancement/conservation segment of the Schwartz model as having an ethical dimension. The distinction between “ethical” and “non-ethical” is not then always clear-cut. “Ethical” values may not have clear ethical instantiations, whilst “non-ethical” values may have ethical implications. This strengthens the case for looking at values more broadly, and being conscious of the complex relationship values have with ethics.

To summarize, the VQ-S produces quantitative results that shed light on system-level values relationships (L1), while the qualitative data reveals the diverse meanings and interpretations associated with different values (L2) and hints at some examples of consequent instantiations and behaviors (L3).

4 DISCUSSION

4.1 Lessons Learnt

Values as distinguished from ethics. The results that emerged from mapping the ACM Code of Ethics statements onto Schwartz’s values types were illuminating. Several important value types and sub-types, such as self-direction (action) and stimulation, were missing and the Code of Ethics statements strongly mapped onto the self-transcendence and conservation segments of the Schwartz values spectrum. Self-transcendence includes the values of benevolence and universalism, while conservation includes the values of tradition, conformity and security.

Our research yields two useful findings that can inform the design of future values investigations. Firstly, it supports the idea that values should be distinguished from ethics, as focusing on ethics alone risks leaving out not only key values influencing technology outcomes, but also important values tensions and relationships. Secondly, ethics codes themselves may be strongly influenced not only by values with evident ethical import (such as universalism and benevolence), but also by values that are not necessarily inherently ethical, such as conformity. In the case of the ACM Code of Ethics, conformity features highly as a way of encouraging adherence to standards of ethical best practice. However, in other contexts, conformity might have negative effects, such as curtailing software practitioners’ independence and creative endeavor.

Theoretical frameworks for values study Our approach considered values as mental representations that can be studied at three interconnected levels: the system, or universal level (L1), as advanced by Schwartz; the personal, or abstract level (L2); and the instantiation, or concrete behavioral level (L3) [30]. This framework proved helpful in providing a clear and explicit structure for

examining values. Used with this framework in mind, the Values Q-Sort worked well at illuminating “inter-subjective orderings of beliefs that are shared among people” [36]. Q-methodology seems to be well suited to studying human values as understood through the lens of a relational values model as advocated by Schwartz. The Q-Sort reflects the relational nature of values by asking participants to consider the statements together and make trade-offs. Furthermore, the three factors extracted and discussed seem to support the values patterns observed in Schwartz’s model. For example, Factor 1, “The Intrinsically-driven Altruist” demonstrated a preference towards values connected to universalism, and less preference towards oppositional values, such as power, as Schwartz’s model would suggest.

The Schwartz values model seemed to be a comprehensive framework for our study, though when asked if there were any missing values, some participants did identify examples, such as learning or ‘avoiding harm’. In addition, applying Schwartz’s values to a specific sector of life - work and, in the context of our study, a specific software project - may put to test a relational values model designed to apply to life generally. As Sean says, “I also put the public good thing a bit on the less important side, but I would actually say that in terms of my sort of life outside work, doing things for the public good and being environmentally conscious are actually probably much more of a focus”. One direction for future work may be to explore the degree to which the Schwartz relational model is maintained when applied to specific life domains.

The importance of better articulating values. The implications of the varying interpretations of the public good discussed above are important. The ACM Code of Ethics states that computing professionals “should reflect upon the wider impacts of their work, consistently supporting the public good” [1], but deliberately does not define what the public good is. The confusion participants expressed around the concept of the public good, and their different interpretations of its implications, demonstrate that a Code of Ethics is not enough without opportunities for individuals, software teams, and companies to deliberate upon some of its central tenets. The Values Q-Sort offers one tool for encouraging such reflection.

4.2 Ongoing Challenges

Values studies within industry need careful reframing. Our participants were a purposeful sample, ranging from academic software practitioners to public sector software developers to corporate software consultants, all of whom were actively involved in software creation. However, we encountered some challenges with access to industry. Based on our experience, the word “values” comes loaded with meanings and does not seem to encourage corporate-level participation in a research project. In addition, the equation of values with ethics seems to permeate industry too; for example, industry partners put us in touch with employees who they had identified as having particular interest in computing ethics. Getting access to software practitioners who would identify as “just an engineer” is highly important, but was also rather difficult, our own experience being that it was far easier to engage with freelance and research software developers. This raises questions as to how we describe our research to industry.

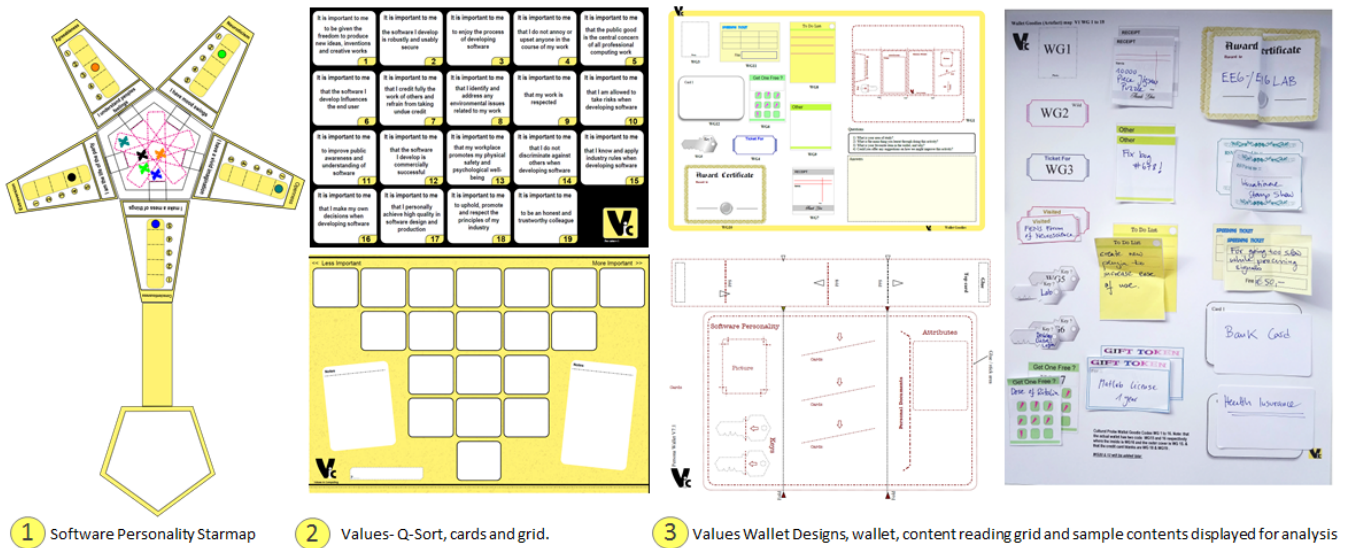


Figure 4: Values in SE Tools

The complexity of studying values at the instantiation level (L3). Further work is required to more deeply investigate values at each of the three levels, particularly at the instantiation level (L3). Research has found that the same value can often be expressed in, and identified with, different behaviors or instantiations. A cross-country study, for example, found that people of different nationalities held different ideas about sustainable behavior [21]. For example, Brazilian participants were more likely than British participants to see saving water as an example behavior that would stem from valuing care for the environment.

Whilst we encouraged software practitioners participating in our study to situate their reflections within a specific project, discussion often became more generic and abstract, representing one difficulty that we encountered. In addition, the Q-Sort interviews only give us insight into espoused attitudes and behaviors, and don't provide indications of actual behaviors. The difficulties of capturing and understanding behaviors have been acknowledged [22]. Ethnographic research offers one way forward, but even then, it is hard to know which values are being enacted and instantiated through behaviors and practices.

Further complexity of values. The Values Q-Sort interviews revealed complex relationships between personal values, the values of others (e.g. peers), perceived organizational values, and perceived societal values. Future research should further explore the interactions between these different dimensions. Another possibility is that values may be strongly influenced by different SE roles, such as business analysts, product owners, testers, etc. Our research has not taken into account potential differences in values priorities according to role, though other research has found that different SE roles were associated with different personality traits [38].

While the Q-Sort helps encourage discussion around values, there are some limits. Maio and Olson suggest that values "may be widely shared, rarely questioned, and, therefore, relatively bereft of cognitive support" and that values thus "function like truisms"

[23]. They contend further that "values are supported primarily by affective information (feelings about values) and, secondarily, by behavioral information (recollections of value-affirming behavior). That is, values are important in large part because people attach strong feelings to their values" [23]. This creates a certain amount of difficulty for studying human values, as they may not lend themselves easily to verbal articulation. The Q-Sort interviews confirmed this to some degree, as participants sometimes stated that they had not really thought about why a value was important to them or struggled to explain their reaction to a particular statement. Dan, for example, when faced with N9, explained "I don't really care if the work is respected. I don't know how to- let's see, I've never really introspected that one".

5 ON-GOING AND FUTURE DIRECTIONS

5.1 Digitisation of existing tools

There is potential for the tools, including the Values Q-Sort and the Values wallet, to be digitised, for increased scalability and easy replicability. However, digitisation comes with several caveats, particularly in relation to the Q-Sort. The identification of 'types' enabled by the Q-Sort is something that could be easily misused. The Values Q-Sort can be easily replicated, but results from specific studies should not be generalised to software engineers as a whole.

5.2 Beyond Qualitative and Quantitative

One of the strengths of the Q-Sort is that it yields both quantitative and qualitative data. However, the structured nature of the Q-Sort also marks a limitation, as it can bound conversation. In response, and to be open to values as they emerge through the research process [4], we have been developing a series of tools that are less structured and more provocative and unbounded. Influenced by design thinking techniques, such as Gaver's 'cultural probes' [14] and speculative design [7], we call these 'values probes' (see Figure

4). These have been designed to elicit a greater range of discussion and perspectives. Some of these tools are in an advanced stage of development and have been used with a variety of participants ($N > 80$) in teaching and research contexts. The Software Personality Starmap and the Values Wallet (see Figure 4) have also been piloted with a small number of researchers and developers. By explicitly anthropomorphizing digital systems, starmaps and values wallets invite participants to project the values perceived to be driving software systems design onto the artefacts. Further pilots are planned in research and education, and we are currently adapting the tools for computing professionals' use.

6 CONCLUSION

In this paper, we laid out two principles to advance the study of human values in software production. Firstly, we made the case for distinguishing between values and ethics. Secondly, we argued for a stronger theoretical framework for understanding values. We gave emerging results from a Values Q-Sort tool designed to respond to these principles. From our initial pilot study, the Q-Sort was largely a successful tool for this purpose. However, values remain a complicated thing to study, especially due to their operation at three levels and their variety of subjective interpretations. It is this complex nature though that demands advances, both theoretical and empirical, in the study of the role of human values within software production.

ACKNOWLEDGMENTS

This work was partly funded by an Engineering and Physical Sciences Research Council grant: number EP/R009600/1.

REFERENCES

- [1] ACM. 2018. ACM Code of Ethics and Professional Conduct. Retrieved September 11, 2018 from <https://www.acm.org/code-of-ethics>
- [2] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.
- [3] An-Shou Cheng and Kenneth R. Fleischmann. 2010. Developing a meta-inventory of human values. In *Proceedings of the American Society for Information Science and Technology* 47, 1, 1–10.
- [4] Christopher A. Le Dantec, Erika Shehan Poole, and Susan P. Wyche. 2009. Values as lived experience: evolving value sensitive design in support of value discovery. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. 1141–1150. <https://doi.org/10.1145/1518701.1518875>
- [5] Martin P. Davoren, Mary Cronin, Ivan J. Perry, and Karl O'Connor. 2016. Alcohol consumption among university students: a typology of consumption to aid the tailoring of effective public health policy. *BMJ Open* 6 (2016), 1–9.
- [6] Kevin C. Desouza and Yukika Awazu. 2005. Managing radical software engineers: between order and chaos. *Proceedings of the 2005 workshop on Human and social factors of software engineering (HSSE 2005)* (2005), 1–5. <https://doi.org/10.1145/1082983.1083005>
- [7] Anthony Dunne and Fiona Raby. 2013. *Speculative everything: design, fiction, and social dreaming*. MIT press.
- [8] Maria Angela Ferrario, William Simm, Stephen Forshaw, Adrian Gradinar, Marcia Tavares Smith, and Ian C. Smith. 2016. Values-first SE: research principles in practice. In *Proceedings of the 38th International Conference on Software Engineering (ICSE 2016)*. 553–562.
- [9] Cesar Franca, Fabio Q. B. Da Silva, and Helen Sharp. 2018. Motivation and Satisfaction of Software Engineers. *IEEE Transactions on Software Engineering* (2018).
- [10] Christopher Frauenberger, Marjo Rauhalu, and Geraldine Fitzpatrick. 2017. Interaction ethics. *Interacting with Computers* 29, 2 (2017), 220–236.
- [11] Batya Friedman. 1996. Value-sensitive design. *Interactions* 3, 6 (1996), 16–23.
- [12] Batya Friedman, David G. Hendry, and Alan Borning. 2017. A Survey of Value Sensitive Design Methods. *Foundations and Trends in Human-Computer Interaction* 11, 23 (2017), 63–125.
- [13] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. 2017. Fairness testing: testing software for discrimination. In *Proc. of the 2017 11th Joint Meeting on Foundations of Software Engineering*. ACM, 498–510.
- [14] Bill Gaver, Tony Dunne, and Elena Pacenti. 1999. Design: cultural probes. *Interactions* 6, 1 (1999), 21–29. <http://doi.acm.org/10.1145/291224.291235>
- [15] Don W. Gotterbarn, Amy Bruckman, Catherine Flick, Keith Miller, and Marty J. Wolf. 2018. ACM code of ethics: a guide for positive action. *Commun. ACM* 61, 1 (2018), 121–128. <https://doi.org/10.1145/3173016>
- [16] The Guardian. 2018. We work for Google. Our employer shouldn't be in the business of war. Retrieved 2 February, 2019 from <https://www.theguardian.com/commentisfree/2018/apr/04/google-ceo-drones-ai-war-surveillance>
- [17] IEEE. 2018. The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. Retrieved September 11, 2018 from <https://ethicsinaction.ieee.org/>
- [18] Bran Knowles, Lynne Blair, Stuart Walker, Paul Coulton, Lisa Thomas, and Louise Mullagh. 2014. Patterns of persuasion for sustainability. In *Proceedings of the 2014 conference on Designing interactive systems*. 1035–1044.
- [19] Sven H. Koch, Romyana Proynova, Barbara Paech, and Thomas Wetter. 2013. How to approximate users' values while preserving privacy: experiences with using attitudes towards work tasks as proxies for personal value elicitation. *Ethics Inf Technol* 15 (2013), 45–61.
- [20] Per Lenberg and Robert Feldt. 2018. Psychological Safety and Norm Clarity in Software Engineering Teams. In *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '18)*. ACM, New York, NY, USA, 79–86. <http://doi.acm.org/10.1145/3195836.3195847>
- [21] Paul Luo Li, Andrew J. Ko, and Jiamin Zhu. 2015. What makes a great software engineer? *Proceedings of the 37th International Conference on Software Engineering (ICSE 2015)* (2015), 700–710.
- [22] Gregory R. Maio. 2010. Mental Representations of Social Values. In *Advances in Experimental Social Psychology* (Vol 42). Academic Press, 1–43.
- [23] Gregory R. Maio and James M. Olson. 1998. Values as Truisms: Evidence and Implications. *Journal of Personality and Social Psychology* 74, 2 (1998), 294–311.
- [24] Keith W. Miller and David K. Larson. 2005. Agile software development: human values and culture. *IEEE Technology and Society Magazine* 24, 4 (2005), 36–42.
- [25] Davoud Mougouei, Harsha Perera, Waqar Hussain, Rifat Shams, and Jon Whittle. 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 (ESEC/FSE 2018)*. ACM, New York, NY, USA, 780–784. <http://doi.acm.org/10.1145/3236024.3264843>
- [26] Marco et al. Ortu. 2016. The Emotional Side of Software Developers in JIRA. In *Proceedings of the 13th International Conference on Mining Software Repositories (MSR '16)*. ACM, New York, NY, USA, 480–483. <http://doi.acm.org/10.1145/2901739.2903505>
- [27] Timo Partala and Sari Kujala. 2016. Exploring the Role of Ten Universal Values in Using Products and Services. *Interacting with Computers* 28, 3 (2016), 311–331.
- [28] Paul Ralph and Paul Kelly. 2014. The dimensions of software engineering success. In *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*. 24–35.
- [29] Shalom H. Schwartz. 1992. Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries. In *Advances in experimental social psychology* (Vol 25). Academic Press, 1–65.
- [30] Shalom H. Schwartz. 2012. An Overview of the Schwartz Theory of Basic Values. *Online Readings in Psychology and Culture* 2, 1 (2012), 1–20. <https://doi.org/10.9707/2307-0919.1116>
- [31] Shalom H. Schwartz et al. 2012. Refining the theory of basic individual values. *Journal of personality and social psychology* 103, 4 (2012), 663–688.
- [32] Katie Shilton, Jes A. Koepfler, and Kenneth R. Fleischmann. 2014. How to see values in social computing: methods for studying values dimensions. In *Proceedings of the 17th ACM conference on Computer supported cooperative work social computing (CSCW14)*. ACM, New York, NY, USA, 426–435.
- [33] William Stephenson. 1993. Introduction to Q-methodology. *Operant Subjectivity* 17, 1 (1993), 1–13.
- [34] Jeroen Van den Hoven, Gert-Jan Lokhorst, and Ibo Van de Poel. 2012. Engineering and the Problem of Moral Overload. *Science and Engineering Ethics* 18, 1 (01 Mar 2012), 143–155. <https://doi.org/10.1007/s11948-011-9277-z>
- [35] Simon Watts and Paul Stenner. 2012. *Doing Q methodological research: Theory, method interpretation*. Sage.
- [36] Thomas Weblar, Stentor Danielson, and Seth Tuler. 2009. Using Q method to reveal social perspectives in environmental research. *Greenfield MA: Social and Environmental Research Institute* 54 (2009), 1–45.
- [37] Emily Winter, Steve Forshaw, and Maria Angela Ferrario. 2018. Measuring Human Values in Software Engineering. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '18)*. ACM, New York, NY, USA, Article 48, 4 pages. <http://doi.acm.org/10.1145/3239235.3267427>
- [38] Murat Yilmaz, Rory V. O'Connor, Ricardo Colomo-Palacios, and Paul Clarke. 2017. An examination of personality traits and how they impact on software development teams. *Information and Software Technology* 86 (2017), 101–122.