

Applying Human Values Theory to Software Engineering Practice: Lessons and Implications

Maria Angela Ferrario¹, Member, IEEE and Emily Winter²

Abstract—The study of human values in software engineering (SE) is increasingly recognised as a fundamental human-centric issue of SE decision making. However, values studies in SE still face a number of issues, including the difficulty of eliciting values in a systematic and structured way, the challenges of measuring and tracking values over time, and the lack of practice-based understanding of values among software practitioners. This paper aims to help address these issues by: 1) outlining a research framework that supports a systematic approach to values elicitation, analysis, and understanding; 2) introducing tools and techniques that help elicit and measure values during SE decision making processes in a systematic way; and 3) applying such tools to a month-long research sprint co-designed with an industry partner and conducted with 27 software practitioners. The case study builds on lessons from an earlier pilot (12 participants) and combines in-situ observations with the use of two values-informed tools: the Values Q-Sort (V-QS), and the Values-Retro. The V-QS adapts instruments from values research to the SE context, the Values-Retro adapts existing SE techniques to values theory. We distil implications for research and practice in ten lessons learned.

Index Terms—Software engineering, human values in software practice, human-centric software engineering, case studies of practice, ethics, non-functional requirements, responsible innovations, agile retrospectives, mixed-methods

1 INTRODUCTION

“SOFTWARE is designed and built primarily to address human needs,” and many of the problems affecting SE, from costly project failures to life-threatening situations, have often been traced back to the lack of adequate consideration for human-centric issues in SE, including human values [1]. Although research on human values in SE has recently grown, with the case for values studies in SE made [2], their relevance for model driven SE [3] and requirements engineering [4], [5] highlighted, and examples of their use in practice examined [6], the field is still emergent and challenged by a number of issues: crucially “the lack of precision in how the construct of values is defined, applied, and investigated” [7]. We argue that, as a consequence of weakly defined values constructs, software practitioners can find it challenging to express “ideas about human values with language that is not as precise or articulate as the language routinely used to express technical ideas” [8]. Similarly, Hussain *et al.* report on the difficulty of eliciting and measuring values in SE projects, the lack of reliability of values-studies instruments to software practitioners, and the limited consideration for and understanding of how values work in SE practice [9].

- Maria Angela Ferrario is with Queen’s University Belfast, Belfast, Antrim BT9 5BN, U.K. E-mail: m.ferrario@qub.ac.uk.
- Emily Winter is with Lancaster University, Lancaster, Lancashire LA1 4YW, U.K. E-mail: e.winter@lancaster.ac.uk.

Manuscript received 13 May 2021; revised 13 April 2022; accepted 14 April 2022. Date of publication 26 April 2022; date of current version 15 March 2023. This work was supported by UKRI under Grant EP/R009600/1.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by Lancaster University’s Faculty of Science and Technology’s Ethics committee.

(Corresponding Author: Maria Angela Ferrario.)

Recommended for acceptance by Y. Dittrich.

Digital Object Identifier no. 10.1109/TSE.2022.3170087

The overarching *goal of our research* is to advance a systematic and SE-relevant approach to the understanding and the application of human values in SE practice. Our *objective* is three-pronged: first, to identify a scientifically robust research framework that can help systematically study how values work in SE; second, to develop new and adapt existing tools that, informed by this framework, help capture what software practitioners think and do when making values decisions; and third, to use these tools to help them reflect and articulate what specific values guide their practice, and how. Our previous publications outline the key principles of our research framework [10], introduce a set of tools designed according to the framework [11], and report on the initial results from a pilot with software practitioners from different organisations and sectors [12].

The *aim of this paper* is to report on the findings and lessons learned from an in-depth industry case study where we apply the identified framework and tools and address the challenges encountered during the pilot. The *key contribution* of this paper are ten lessons learned from the case study from which we draw ten corresponding implications for SE research and practice (Section 9). This paper hence centers on *three research questions*:

- 1) RQ1- How can SE research support a more systematic investigation of values in SE practice? What values theories and models should we draw from?
- 2) RQ2- What new tools can be developed and what existing SE techniques can be adapted (and how) to help elicit, articulate, and measure values in SE practice?
- 3) RQ3- What can be learned from applying the identified research framework (RQ1) and tools (RQ2) to SE practice? Specifically, what can we learn about software practitioners’ understanding and articulation of values, and their relation to SE practice?

RQ1 is an epistemic question about how values can be conceptualised so that they can be studied in a systematic and empirical way. To answer this question we draw from social and cognitive psychology studies. Specifically, we consider values as ‘mental constructs’ [13], draw from social psychologist Schwartz’s Universal Values theory [14], and combine it with cognitive and behavioural studies investigating the different interpretations and enactments of values [13], [15]. We have outlined this approach in previous work [2], [6], [10], [12]; here, we include a detailed motivation and explanation of the principles underpinning our proposed research framework (Section 3). *RQ2* is a question about methods: given the research framework identified in *RQ1*, what type of tools and techniques can be designed and used to investigate values in SE? We have developed and piloted a range of tools in previous work [11]; here, we address the challenges identified in the pilots and co-design a case study as a month-long research sprint with our industry partner. *RQ3* is about practice: what can we learn from applying the framework and tools to SE practice? This paper uses an industry case study to address *RQ3* by deploying tools and techniques (*RQ2*) designed according to the identified research framework (*RQ1*). The case study (27 participants) combines in-situ observations with the use of two values-informed tools: the Values Q-Sort (V-QS), and the Values-Retro. We designed these two tools by: 1) tailoring values studies’ instruments to the SE context (the V-QS), and 2) adapting existing SE techniques to more explicitly consider human values (the Values-Retro). The case study findings are reported as both quantitative measurements of SE practitioners’ values orientations, and rich narratives explaining the participants’ interpretations and enactments of values in SE practice. The rest of the paper is organised as follows: first, we review related work (Section 2) and outline our research framework and research strategy (Section 3); then, we introduce the case study design (Section 4) and the techniques used in the study (Section 5). We report on the findings from the V-QS statistical analysis (Section 6) and from the qualitative data elicited during the V-QS (Section 7); we complement these findings with highlight results from the Values Retros (Section 8). From this work, we draw ten lessons learned and their implications for SE research and practice (Section 9). We conclude with threats to validity, and future directions (Section 10, Section 11).

2 RELATED WORK

2.1 Values Studies in Technology Design

A large body of research exists looking into embedding values into technology design, including work focusing on computer ethics [16], [17], [18], and, in particular, Values Sensitive Design (VSD) [19], [20], [21]. Our previous work [2], [6], [10], [11], [12] - acknowledges VSD’s importance but also highlights how our work differs, and why such differences are important and relevant to SE. In summary, we argue that, first, VSD tends to focus on a selected list of values with “ethical import” (e.g., [19], [20], [21]); and second, that VSD values constructs can lack the precision required by SE practice [7]. First, by focusing on values with ethical import, VSD risks missing a wider range of values (see also [22] discussed in [11]) and, in turn, patterns of interaction between values. We find this problematic because investigating the relational

nature of values (e.g., alignments and oppositions as described in [23]) can offer useful insights into software development practices [11]. Section 3.5 further explains the importance of distinguishing Ethics from values studies. Second, we argue that the lack of precision (e.g., weak values constructs) can explain, at least in part, why the application of VSD tends to remain focused to the early stages of software development [24] and is considered more useful to end-users and stakeholders than to developers [25]. Our work can help improve the precision of values constructs, and, in turn, better support their inclusion in existing SE techniques and practice (e.g., user stories [11], [26], and retrospectives (Section 8)).

2.2 Values Studies in SE

The past few years has seen a rapidly emerging body of work in the study of human values in SE. Thew *et al.* study values in relation to emotions, soft goals, and requirements [27], [28]; Ferrario *et al.* place values at the centre of wider SE decision making processes [6]; Whittle *et al.* strengthen the case for studying values in SE [2]; and Hussain *et al.* study values in an industry context. However, in a number of cases [29] [30], values systems are used more as list of values for classification purposes rather than models according to which values relationships can be observed. Moreover, [29] and [30] carry out an analysis of values *a posteriori*. We express concerns about the limitation of values studies for which the ground truth is not readily accessible (Section 9). 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” [26]. Our work contributes to advancing this work, and provides it with a solid scientific framework.

2.3 Automated Techniques and Machine Intelligence

A growing number of researchers, such as Galhotra *et al.* [31], use automated SE techniques to test the value of ‘fairness’ in systems. Such data-driven techniques are easy to scale, but lack consideration of how fairness may relate to other values, and the real-life values of those who these studies wish to help. Machine Intelligence expert Stuart Russell offers an interesting perspective on how values may be enacted in intelligent systems design. First, he suggests to refer to values as ‘preferences’ [32], to avoid confusion with Ethics, thus linking to the ‘relative importance’ of values as described in Section 3.5. Russell then suggests that this could help “engineer systems that can learn or acquire values at run time” using inverse reinforcement learning (IRL), for instance, “in which a system infers the preferences of another rational or nearly rational actor by observing its behavior”. Though this approach may hold promises (and does further highlight the need for distinguishing values studies from Ethics), to simply abstract a value from observed behaviour risks missing the evaluative quality of values (see Section 3).

3 RESEARCH FRAMEWORK

We define human values as “guiding principles of what people consider important in life” [34] and, specifically, as the criteria that people use “to select and justify actions and

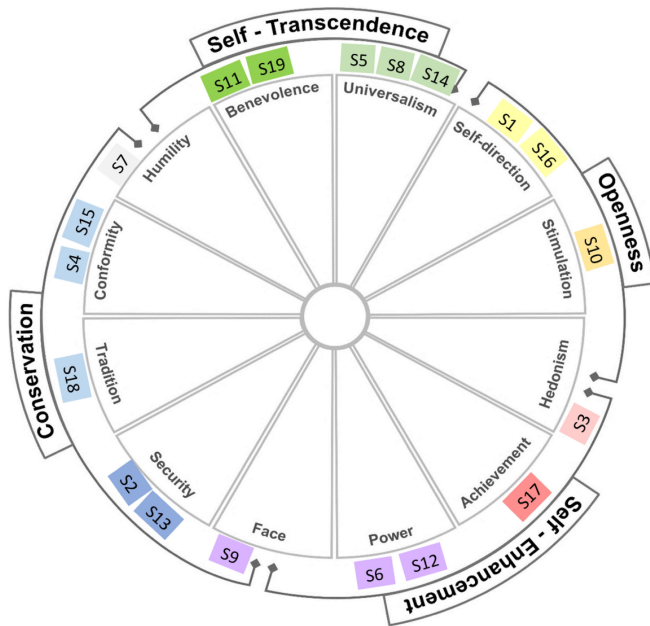


Fig. 1. Schwartz's values model, adapted from [33]. Each values statement (S#) is mapped onto one of Schwartz's values groups (e.g., Security, Self-Direction). For instance, the 'Security' values group includes two values: 'Personal Security' (S13) and 'Societal Security' (S2); we colour-code each value statement so that values belonging to the same group have the same colour. The full list of values statements is in Fig. 3.

evaluate people (including the self) and events" [14]. These criteria include prestige, creativity, pleasure, and power over resources (e.g., wealth): these values have ethical implications, but they cannot be directly equated with ethical principles. In focusing only on values that can be equated with ethical principles, we risk ignoring the role played by a wider range of values in influencing decision-making processes and actions [4], [6]. We discussed the distinction between values studies and Ethics in previous research [6], [12], and summarise key points in Section 3.5.

3.1 A Values Model

We do not consider it necessary to propose a new theory of values; instead, we have examined work on existing values theories [13], comparative studies of values frameworks [34] and models [15]. Based on the findings of this work, we adopt Schwartz's theory of Universal Values and its model [14], [33] as an established starting point. We draw on social psychology because it is a discipline with a long-standing tradition of empirical and systematic approaches to study values. Further, Schwartz's values model has been particularly influential and already adopted in computing and SE research [2], [6], [9], [23], [34]. Schwartz's values system is based on survey research in more than 80 countries across the world [35] and it identifies a series of distinct values that are "structured in similar ways across culturally diverse groups" [35]. Fig. 1 is a visualisation of Schwartz's values model in its latest form [33]. The model identifies 19 core values, and each value is defined in terms of its broad motivational goal. For instance, in [33] the value 'Personal Security' is defined with the statement "Safety in one's immediate environment," and is grouped under the 'Security' value group. Later in the paper, Fig. 3, we number and colour-code all Schwartz's values statements (S#) listed

in [33]. In Fig. 1, we use the same numbering and colour coding for ease of reference.

The major contribution of Schwartz's work, however, is not the list of values it offers, but the patterns of relationships that its extensive research has observed and quantified [13]. This pattern consists of a circle centred around two oppositional axes: self-enhancement versus self-transcendence and openness versus conservation (see Fig. 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"[35]. Values hence operate in a specifically organized way. What the model suggests is that if a person values tradition (on the conservation axis), they are likely to value conformity, but not to value highly self-direction and stimulation (on the opposite openness axis). Within computing, Schwartz has been used in studies on end-users' needs [36], [37], the prevalence of values in SE research [29], and SE practitioners' values [9], though with little emphasis on the relational nature of the model. In contrast, we explore the relational aspect of values by, for instance, noting if the statistical analysis of the V-QS reports values relationship patterns similar to those observed by Schwartz and others.

3.2 A Three-Level Values Study Approach

Our research builds and moves beyond Schwartz. We do so by drawing on research by Maio and colleagues [13], [15] for two main reasons: first, because much of their work builds on Schwartz's values theory and model, which we choose for the reasons stated above. Second, because, differently from Schwartz, they do not stop at the 'Universal' aspect of values, but they also investigate how values get translated into practice - a key focus of our research. To our knowledge, they are the first to do so in the 'Schwartz' tradition, upon which we build. Simply put, Maio's work considers values as mental constructs that can be studied at three interconnected levels [12], [13]: 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). Our work adopts this framework as a starting point for investigating values in a systematic way. 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 practitioner 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 industry rules (conformity). At L2, we can expect software practitioners to have different interpretations of what achieving high quality software (achievement) looks like; for example, code that does the job versus elegant code. At L3, values are instantiated through actions (e.g., a specific software system design decision). For example, a search engine company concerned with privacy (e.g., DuckDuckGo) may design their system to never log user queries. Our research shows that investigating these three levels can flag values tensions within SE teams and organisations [10], and between the resulting software systems and wider social needs and aspirations. Following this framework, we have designed, developed and used a number of tools and techniques, including the V-QS [11], which collects values

narratives and extracts statistical ‘types’, or factors, of values orientations (Section 6, Fig. 5). The analysis of these recurring values orientation patterns can model potential values tensions within and between software teams, and between organisations and the end-users of their software products. When “operationalising values” [26] it is crucial to articulate values at distinct levels to avoid overlooking the differences of understanding of the same value at different levels. Our ongoing research is one of the first to do so in SE, with tools specifically built for this purpose [10], [11], [12].

3.3 Five Values Features

Empirical research has found that values exhibit certain common characteristics or features [14], [33], namely: they are linked to affect (emotions); they transcend specific situations; they guide selection and evaluation of behaviour; they are ordered by relative importance; and the relevant importance of multiple values guides people’s actions. Our research finds that such features are recurrent and can be observed in the SE context. Below, we report a summary of the key values’ features and exemplify with quotes from our case study. We show in brackets the number and label of the V-QS value statement (S#) that elicited a participant’s response (Fig. 3 reports the full list of values statements).

#1 *Values are linked to affect-* When values are activated, particularly if an individual’s values are challenged, they often lead to the expression of emotion. For instance, in our study, we found that participants feel ‘angry’ when their work is not respected (S9 Face - Public Image) or ‘frustrated’ when they can’t be as creative as they would like at work (S10 Self Direction- Thought).

#2 *Values transcend specific actions and situations-* Values can be relevant in several contexts - the workplace or at home, with friends or team workers. For example, in our study we found participants who considered honesty ‘*very important from a personal perspective so... professional(ly) it’s the same*’ (S19 Benevolence - Dependability).

#3 *Values serve as standards or evaluation criteria-* Values guide the selection or evaluation of actions, people, and events. People may not act on the values that they hold important due to external circumstances (e.g., budget constraints), but they do still evaluate their actions against them, leading to emotional reactions. For instance, some participants felt ‘sad’ for not doing more for the environment (S8 Universalism - Nature).

#4 *Values are ordered by relative importance-* People’s values form an ordered system of priorities. For instance, a number of participants stated that, although it was important for a software product to be commercially successful (S12 Power - Resources), they valued positive social impact more (S5 Universalism - Concern).

#5 *Multiple values importance guides action-* Similarly, any attitude or behavior has implications for more than one value. For example, high quality and secure software (S2 Security - Societal) may come at the expense of exploring something new, more fun, and riskier (S10 Stimulation).

We have considered these features in the choice and design of our tools; the V-QS, for instance, requires participants to arrange a list of values statements in order of importance and explain their choice. The Values-Retro captures

how values change over the course of a project, and how these changes affect a team and its members’ emotions.

3.4 Values and Universal Human Requirements

Schwartz’s theory of Universal Values sustains that a possible reason for the observed presence of similar values across cultures is their grounding in fundamental human needs, or “universal requirements of human existence” [14], [35]. These requirements pertain to the biological needs of individuals (i.e., a safe and healthy work environment; S13 Security- Personal), the welfare of groups (i.e., to be a dependable and trustworthy colleague; S19 Benevolence - Dependable), and coordinated social interaction, (i.e., compliance with industry rules and standards; S15 Conformity - Rules). According to Schwartz, individuals cannot cope successfully with these requirements on their own. Rather, people need to articulate them, communicate with others about them, and gain cooperation in their pursuit. Simply put, values are “the socially desirable concepts” [14] used to represent fundamental human requirements and to express them in social interactions. This three-pronged consideration of human needs (of the individual, groups, and wider society) is embedded in Schwartz’s values model.

3.5 Values Studies as Distinct From Ethics

When investigating human values in SE, it is important to clarify the distinction between the study of values (drawing from cognitive and social psychology) and Ethics (as science that reflects on morals [38]). As argued previously, (e.g., [6], [12]), the two are linked but distinct areas of research, and their conflation can hinder a meaningful application of the two disciplines to SE practice. Simply put, values studies investigate *how* values work (their mechanics) and how they influence people’s actions and the evaluation of their actions. Ethics provide guidance on *what* values *should* guide people’s actions and evaluations. To use a metaphor, values studies offer a map and Ethics provides the compass. Values studies help build that map by investigating the different interpretations, instantiations, and relationships of values across individuals, organisations, and cultures. Values studies hence have a strong empirical orientation, while Ethics is normative in nature and lends itself to theoretical discourse and principled guidance. This paper takes an empirical stance, and focuses on the former, not the latter.

3.6 Research Strategy

Our research is based on empirical observations [39] and takes a two-pronged perspective: explanatory and exploratory [39], [40]; critical and reflective [41]. First, our research objective is to investigate how values work in SE practice, and to do so by using empirical, systematic, and widely applicable research methods. Our stance is explanatory and exploratory because we explain and describe the phenomenon without proving or disproving the proposed theoretical framework. Second, we use both qualitative and quantitative approaches. This, we argue, helps reflect on the limitations of using each method in isolation. For instance, only considering the V-QS quantitative analysis would risk ignoring that individuals with similar values orientations (L1) can interpret (L2) and act upon (L3) the same values differently

(Section 7). Vice versa, adopting solely an open-ended qualitative approach to values analysis would miss the systematic identification of patterns of relations as identified by the factor analysis (Section 6).

3.6.1 Choice of Approach and Data Collection Methods

This two-pronged perspective has guided the choice of this research's primary method (a single case study of exploratory nature [39], [42]) and the two main research techniques: the Q-sort (the V-QS), and the Values-Retro. The Q-sort, a mixed method technique, is particularly important because it simultaneously allows quantitative data collection and structured qualitative narrative elicitation. Within the case study, the Q-Sort has allowed three complimentary data analysis approaches: *Abductive quantitative approach*- we use statistical analysis to extract factors that can help *explain* participants' values orientations; Watts and Stenner (citing Pierce [43]) reflect on the abductive logic underpinning Q-methodology and state that abduction examines an observed phenomenon "in a pursuit of an explanation and new insight," by treating them "as clues pointing towards a potential explanation" [44]. *Abductive logic* is used for discovery, not for theory verification [44]. We are not interested in verifying Schwartz's values theory, but in seeing what can be discovered by adapting it and applying it to SE practice. *Deductive thematic analysis*- each Q-sort statement represents a value, helping to unambiguously identify when a value appears in the qualitative narrative elicited during the V-QS. *Inductive thematic analysis*- we use an interpretive and iterative approach for generating themes from the values narratives elicited by the V-QS [45].

4 CASE STUDY DESIGN

4.1 Choice of Organisation and Participants

Our case study organisation is the digital arm of a large membership-based organisation (4.5 million members); we refer to the entire organisation as *Share* and its digital arm as *Digital Share*. We approached Digital Share because they have an explicit interest in embedding values considerations into technical decisions, so there was clear overlap with our own research goals. We were also interested in Digital Share as a relatively new arm of the organisation, set up to facilitate a 'digital transformation'. Specifically, we anticipated that this would lead to engaging values discussions resulting from the interaction between older and newer ways of working. Our collaboration began with a kick-off meeting where we met with several key contacts, including the Head of Digital Technology, who expressed interest in exploring ways to support a 'values-based software engineering decision making process'. We arranged a taster session where we shared the tools developed, teased-out the research goals, and co-designed an intensive research sprint aimed to investigate the role that values play in Digital Share SE practice.

Digital Share assisted in choosing two teams where we could carry out our research. These were *Community* and *Net*. Net is responsible for web design and development, whereas Community is focused on digital services aimed at Share's membership base. Teams at Digital Share bring together a wide range of expertise and roles, including software

Name	P#	Team	Role	Exp.	Sex	V-Retro
Sarah	P1	C	SE	7	F	X
Sam	P2	C	QA/Testing	7	M	X
Matthew	P3	C	SE	40	M	
Jon	P4	C	SE	30	M	
Jimmy	P5	C	BA	7	M	
Will	P6	C	QA/Testing	9	M	X
Sophie	P7	C	Delivery Manager	10	F	X
Connor	P8	C	Interaction Designer	5	M	
Dave	P9	C	Digital Services Manager	19	M	
Rich	P10	N	Platform Engineer	9	M	
Laura	P11	N	Search Engine Optimisation	7	F	
Zoe	P12	N	SE	3	F	
Kayla*	P13	N	Content Designer	2	F	
Katherine	P14	N	Delivery Manager	12	F	
Bob*	P15	N	Content Designer	10	M	
Sally*	P16	N	Visual Designer	20	F	
Stuart	P17	N	Platform Engineer	40	M	
Ed	P18	N	Product Manager	5	M	X
Phil	P19	N	Product Manager	7	M	
Christian	P20	N	Analytics & Optimisation	13	M	
Carrie	P21	N	BA	11	F	
Ollie	P22	N	Principal Designer	10	M	
Martin	P23	N	Software Development	13	M	X
Rob	P24	N	Front-end Development	6	M	
Annette	P25	N	SE	>2	F	X-o
Alex	P26	C	SE	>2	M	X-o
Callum	P27	C	SE	>2	M	X-o

Fig. 2. Participants' demographics; experience (Exp.) is in years; P1-P24 conducted the V-QS; P25-27 only took part to a Values-Retro (V-o); (*) flags participants not included in the V-QS factor analysis.

engineers, front-end developers, business analysts, quality analysts, testers, user experience researchers, visual and content designers and data analysts. Software engineers are not siloed into back-end work but interact frequently and share working space with other roles and expertise. Within Community and Net, a key contact point helped facilitate interviews with members of the team. We interviewed 9 people in Community, and 15 people in Net. It was more challenging to recruit participants in Community, owing to the high proportion of contractors in this team (as opposed to Share employees). The Community team also had a reputation for being a more high-pressure working environment, which may have led to more disinclination to free up time to take part in the study.

To reflect the high integration of roles and expertise at Digital Share, we focused our interviews on software engineers but also interviewed people in different roles. Participants found the Q-Sort statements relatable, with the exception of three content and visual designers who could not reflect meaningfully on a number of V-QS statements (e.g., S2, S10, S15) because they specifically refer to 'developing software' in the first person. We remove these participants (P13, P15, P16) from the V-QS analysis. Fig. 2 shows participants' roles, years of experience, and their engagement in the study.

4.2 Context: Digital Share Work Environment

Digital Share was set up within the last decade to help transition Share, a large, historic organisation, into a 'digital transformation'. Digital Share has separate premises to Share, giving it space to do things differently and in an agile

and collaborative fashion. Digital Share is an informal, agile workplace, with daily sub-team stand-ups. The workplace is open plan and visually busy, with multiple physical Kanban boards and walls covered in sticky notes. Employees dress informally, and it was obvious when someone came over from another part of Share, as they would be much more smartly dressed. The atmosphere was one of openness with a lot of joking, knowledge sharing and asking questions. Digital Share has encountered several challenges, such as how its principles of agile, iterative digital transformation are integrated and embedded into other parts of the organisation. Another challenge is trying to shift the understanding of other parts of Share from seeing Digital Share as ‘kitchen-fitters’ (a more Waterfall approach) to seeing Digital Share as problem-solvers. Both Share and Digital Share are values-centered organisations. Share is a member-owned organisation, with a key focus on community and ethics. Whilst doing research at Digital Share, it was obvious that these values were widely-understood and strongly-shared, such that if an idea was proposed that didn’t feel in-keeping with these values, someone would often say ‘that’s not very Share’ or comment on the need to maintain the ‘Share difference’. Within Digital Share, there was also evidence of several values being held in high regard, such as teams having autonomy and being trusted to make decisions.

5 METHOD

5.1 The Values Q-Sort

One of our key imperatives was to use methods and tools that reflected the theoretical framework in use (based on Schwartz and Maio). One chosen method was the Q-Sort. The Q-Sort is a mixed-method that involves asking participants to sort a series of statements onto a grid (usually shaped as an approximate normal distribution) according to their level of agreement with each statement. Q-Sort exercises are accompanied by a semi-structured interview, and the results of multiple Q-Sorts can be statistically analysed. This 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), including the prevalent patterns within a group of participants, 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 encourage examples of values at the instantiation level (L3). We also designed the Q-Sort statements using Schwartz’s values model as a framework. The Q-Sort differs from surveys as participants’ responses are not free but have to be considered in relation to each other, forcing trade-offs and decisions between statements. 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 survey items [44].

5.1.1 Designing the Values Q-Sort Statements

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 [8]. 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 the original Schwartz’s values [14]. We piloted, with two people, an already existing Q-Sort with statements from Schwartz’s 57-item values survey. The pilot found that the wording was not specific enough for software engineers, and that 57 statements were cognitively overwhelming and time-consuming for participants. In response, we turned to the ACM Code of Ethics [46], and dual-coded its principles according to the latest version of Schwartz’s values model [33], which identifies 19 distinct values types and sub-types and is cognitively lighter. The rationale for choosing the ACM Code was to utilize language generated by computing professionals. Whilst not all values can be equated with ethical principles (Section 3.5), several can. We have leveraged on this overlap to construct values statements in a language that is relatable to software practitioners. The dual-coding produced 80 per cent agreement, and remaining discrepancies were discussed by the two researchers. Reference was made to the 1992 IEEE-ACM CS Code of Ethics [47] 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 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 [33] and the 57-item values survey previously piloted. The resulting statements went through a proof-of-concept cycle with four computing researchers at another institution before being piloted. The designs of both the V-QS grid and V-QS values statements cards are included in the supplemental material, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TSE.2022.3170087>.

5.2 Pilot Study

We piloted our V-QS using a purposeful sample before carrying out the present main case study. The pilot involved 12 participants chosen across sectors to probe relevance of the V-QS statements for software engineers and applicability in a range of organisations and domains. The pilot included an equal number of participants (four) from three sectors: private, public, and research. Feedback from the pilot study led to some rewording of the values statements (e.g., those liable to misinterpretation), shown in their final form in Fig. 3. The pilot also highlighted four key challenges [12], which needed addressing before the main study; we outline the challenges and the actions taken below.

5.2.1 Addressing Pilot Study Challenges

Conducting Values Studies Within Industry. During the pilot we noted that the word ‘values’ was often associated with ethics, and this seemed not to encourage corporate-level participation. This association made it also difficult to get access to software practitioners who would identify themselves as ‘just an engineer’, as we would be often referred to practitioners who were interested in ethics. *Action:* we co-designed

S#	Values Q-Sort statement (it is important to me...)	Schwartz value definition in terms of motivational goal	Schwartz value	ACM Code Statement
S1	to be given the freedom to produce new ideas, inventions & creative works	Freedom to cultivate one's own ideas and abilities	SELF-DIRECTION Thought	1.5
S2	the software I develop is robustly and usably secure	Safety and stability in the wider society	SECURITY Societal	2.9
S3	to enjoy the process of developing software	Pleasure and sense of gratification	HEDONISM	N/A
S4	that I do not annoy or upset anyone in the course of my work	Avoidance of upsetting or harming other people	CONFORMITY Interpersonal	N/A
S8	that the public good is the central concern of all professional computing work	Commitment to equality, justice, and protection of all people	UNIVERSALISM Concern	3.1
S6	that the software I develop influences the end user	Power through control of people	POWER Over People	N/A
S7	that I credit fully the work of others and refrain from taking undue credit	Recognizing one's insignificance in the larger scheme of things	HUMILITY	ACM99 7.03
S8	that I identify and address any environmental issues in my work	Preservation of the natural environment	UNIVERSALISM Nature	ACM99 3.03
S9	that my work is respected	Maintaining public image and avoiding humiliation	FACE Public Image	N/A
S10	that I am allowed to take risks when developing software	Excitement, novelty, and change	STIMULATION	N/A
S11	to improve public awareness and understanding of software	Devotion to welfare of in-group members	BENEVOLENCE Care	2.7
S12	that the software I develop is commercially successful	Power through control of material and social resources	POWER Resources	N/A
S13	that my workplace promotes my physical safety & psychological well-being	Safety in one's immediate environment	SECURITY Personal	3.3
S14	that I do not discriminate against others when developing software	Acceptance and understanding of those who are different	UNIVERSALISM Tolerance	1.4
S15	that I know and apply industry rules when developing software	Compliance with rules, laws and formal obligations	CONFORMITY Rules	2.3
S16	that I make own decisions when developing software	Freedom to determine one's own action	SELF-DIRECTION Action	N/A
S17	that I personally achieve high quality in software design and production	Success according to social standards	ACHIEVEMENT	2.1
S18	to uphold, promote and respect the principles of my industry	Maintaining & preserving cultural, family or religious traditions	TRADITION	4.1
S19	to be an honest and trustworthy colleague	Being a reliable and trustworthy member of the in-group	BENEVOLENCE Dependable	1.3

Fig. 3. The table shows the V-QS statements derived by mapping Schwartz's values definitions to the principles listed in the ACM Code of Ethics [46]. For instance, the value 'Societal Security' (S2) maps onto the Code Principle 2.9 'Design and implement systems that are robustly and usably secure'. Where no corresponding principle was found in [46], we referred to the previous version of the ACM Code [47]. For instance, there is no specific principle relating to 'Preservation of the Natural Environment'(S8) in the latest Code, but there is in its previous version [47]. When no corresponding statement was found in either Codes, we created a definition following similar language to that used in the Codes. For instance, there is no principle capturing the value of 'Power on Resources' (S12) e.g., 'wealth' [14], [33]; we phrased it as commercial success. For each statement, we use the colour code as in Fig. 1.

the main case study with an industry partner interested in better understanding how values work and with the specific intent to engage a less selective range of practitioners. Recruiting participants from two of the most active teams made it easier to engage with a diversity of practitioners.

Studying Values at the Instantiation Level (L3). The pilot study identified the need for a deeper investigation of values at each of the three levels, with instantiation being particularly problematic. The difficulties of capturing and understanding behaviors are well documented [8], [48], and ethnographic research was identified as one way forward. *Action:* selecting practitioners working on the same projects made it easier to ground the conversation on specific examples during the V-QS; in-situ observations helped contextualise the narrative elicited during the V-QS.

Complexity of Values. The pilot study revealed the complex relationships between personal values, the values of others, the perceived organizational values, and perceived societal values; the pilot called for further investigation into these different dimensions, particularly whether different values may be strongly influenced by different SE roles. *Action:* recruiting participants from two distinct teams allowed the inclusion of a greater variety of roles (Fig. 2).

Limits of Q-Sort- Maio and Olson contend 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" [48]. The V-QS helps discussion around values, but it is not specifically designed to capture affect. *Action:* complementary research activities were arranged alongside the Q-sort to explore the affective aspect of values. The two Values-Retros, for example, were designed to capture affect at critical points of software projects.

5.2.2 Q-Sort Interview Questions

The V-QS interviews started with introductory questions, asking participants about their current role, their background,

and their career trajectory. Participants were then guided through the Q-Sort exercise, having been asked to focus on a particular project. After the participant finished their Q-Sort, they were asked about their top and bottom ranked statements, and whether there were any other statements that they wanted to explain. We also asked if they would have filled out the V-QS differently in the past, whether they thought their colleagues would fill it out in the same way, and if there was anything important to their work that was not encapsulated by the statements. Finally, we asked participants how they found the exercise.

5.2.3 Conducting the Values Q-Sort Interviews

At Digital Share, 24 interviews were conducted and audio recorded, except for in the case of one participant who asked not to be recorded, as he felt he would be more relaxed and at ease without the Dictaphone. In this case, notes were taken instead. Interviews lasted between 40 minutes and almost one and a half hours. All interviews took place at the participants' workplace, occasionally in the cafe. All interviews were fully transcribed, either by the researcher or by a professional transcriber.

5.3 Observations

The researcher spent 11 days at the company. This allowed time for informal observations, such as observing stand-ups, and casual conversations with Net and Community team members over coffee. She was also invited to observe various meetings of varying levels of formality. These included: one Community team member giving an informal presentation to people from another team; a 'mobbing' session involving three software engineers working collectively on a coding problem (Community); a 'User journey' workshop that brought together the Net team with teams from the main Share organisation; a post-mortem for a difficult project involving Community and different Share teams; and a 'Vision' workshop where the Community team mapped out ideas (both realistic and future-thinking) for e-commerce

initiatives. The researcher wrote field notes both in situ and back home at the end of the day. The observations helped to contextualise both our findings.

5.4 Ethics

This research had the approval of Lancaster University's Faculty of Science and Technology's Ethics committee. Employees within the two teams were aware that the researchers were present. All employees had been informed about the research and been given the opportunity to ask questions and raise concerns. V-QS interview and Values-Retro participants were given an information sheet to read and a consent form to sign (included as supplemental material, available online). A couple of people within the Community team (contractors rather than organisation employees) declined to take part. All observations are anonymised. The researchers asked permission to take notes in meetings to which they were invited. When the researchers were asked to meetings that involved people from other parts of the organisation, they were introduced and people were invited to ask any questions that they might have.

6 FACTOR ANALYSIS: FINDINGS & DISCUSSION

We focus our findings and discussion around the Values Q-Sort, as it was the method we used most extensively, and yields both quantitative and qualitative data. The quantitative results from the V-QS from the two Teams are reported here - Community (C) and Net (N). Section 7 reports on insight provided by the qualitative data.

6.1 Method

The V-QS data from the two teams (C, N=9; N, N=12) were inputted separately into an online Q-analysis program [44]. Factor extraction allows for the emergence of statistically significant patterns and is produced through centroid factor analysis. Each factor is given an Eigenvalue (the sum of the square of each of the individual Q-Sort loading onto the factor) and a factor variance. High scoring Eigenvalues and factor variances indicate that a factor has "strength and potential explanatory power" [44]. The Eigenvalue should be greater than 1 (Kaiser-Guttman criterion); if it is less than 1, it accounts for less variance than an individual Q-Sort.

From this process, four factors (two for each Team) had Eigenvalues greater than 1 and were selected for the next step in the analysis: factor rotation. Factor rotation involves ensuring that each factor offers the most informative viewpoint and can be done in two ways: manual or varimax (automatic). We used varimax rotation, which statistically positions the factors so that they cover the maximum amount of variance and ensures that each Q-Sort has a high factor loading to only one factor [44]. Each of the four factor 'viewpoints' (CF1, CF2; NF1, NF2) represent an abstract type of software practitioner. Fig. 5, for example, introduces type 1 for Team Community as Factor CF1. By examining its distinguishing statements, we refer to CF1 as a "Socially-concerned and Considerate" type of software practitioner, as it ranks public good (S5) as its top value, and risk taking (S10), as the least important one. Descriptively naming 'types' is a commonly used convention in Q-methodology

and it makes factors more memorable. Accordingly, we call these 'types' of software practitioners as follows:

- 1) CF1- Socially-Concerned and Considerate.
- 2) CF2- Ambitious and non-Conformist.
- 3) NF1- Dependable and Considerate.
- 4) NF2- Market Conscious and Autonomous.

Next, we report a summary of the statistics for each factor computed at $p < 0.05$.¹ Convention suggests displaying the composite Q-Sort for each factor [44], but, due to space constraints, we here include only the CF1 composite Q-sort (Fig. 5) as an exemplar; all composite Q-sorts are included at a higher resolution in the supplemental material, available online. For each factor, we report participants who loaded on the factor, the most distinguishing statements (S_{ij}) at $p < 0.05$ (*) and at $p < 0.01$ (**), their position on the importance scale (from -3 least important to +3 most important), and whether the z-score of a statement is significantly higher or lower than in other factors (marked by two small black triangles). The z-score gives the overall rank of a statement in the list of all statements of that factor (see Fig. 4a). For instance, within the factor CF1 (Fig. 5), public good (S5) is a distinguishing statement at $p < 0.01$ (**), with an importance position of +3 (most important), a z-score higher than in any other factor, and ranked first (1st) out of all nineteen statements. As convention we will report it as (S5: +3, 1st). All significant statements are reported and colour coded as per Fig. 1. Participants who do not load on any factor tend to have similarities with more than one factor; as such their ranking choices are not significantly distinct and not reported.

6.2 Team Community, 2 Factors

6.2.1 CF1- Socially-Concerned and Considerate.

CF1 has an eigenvalue of 1.67 and explains 19% of the study variance. Four participants loaded onto this factor (Sarah, Matthew, Sam, Will – all at $p < 0.05$), two SEs and two QAs/software testers. This factor is characterised by a high concern for others, including putting the public good at the heart of work (S5: +3, 1st), not discriminating against others (S14: +2, 2nd), and being an honest and trustworthy colleague (S19: +2, 3 rd). By contrast, personal risk-taking (S10: -3, 19th), and autonomous decision-making (S16: -2, 18th) are considered less important. Notably, commercial success is also a distinguishing statement in that its z-score rank is lower than in all other factors (S12, -1, 13th), and so are both high quality code and respect for own work. Their importance is marginally higher than commercial success, but their rank is lower than in all other factors (S17, 0, 8th; S9, 0, 11th). This factor considers self-transcendent values significantly more important than any other factor, and self-determination and self-enhancement values significantly less important. This factor is also characterised by risk aversion (low ranking S10), potentially associated with the QA/software testing role of Sam and Will.

6.2.2 CF2- Ambitious and Non-Conformist.

CF2 has an Eigenvalue of 1.76 and explains 20% of the study variance. Two participants loaded onto this factor (Jon,

1. The p-value is a measure of the probability that an observed relationship could have occurred by chance. The lower the p-value, the greater statistical significance of the observed relationship.

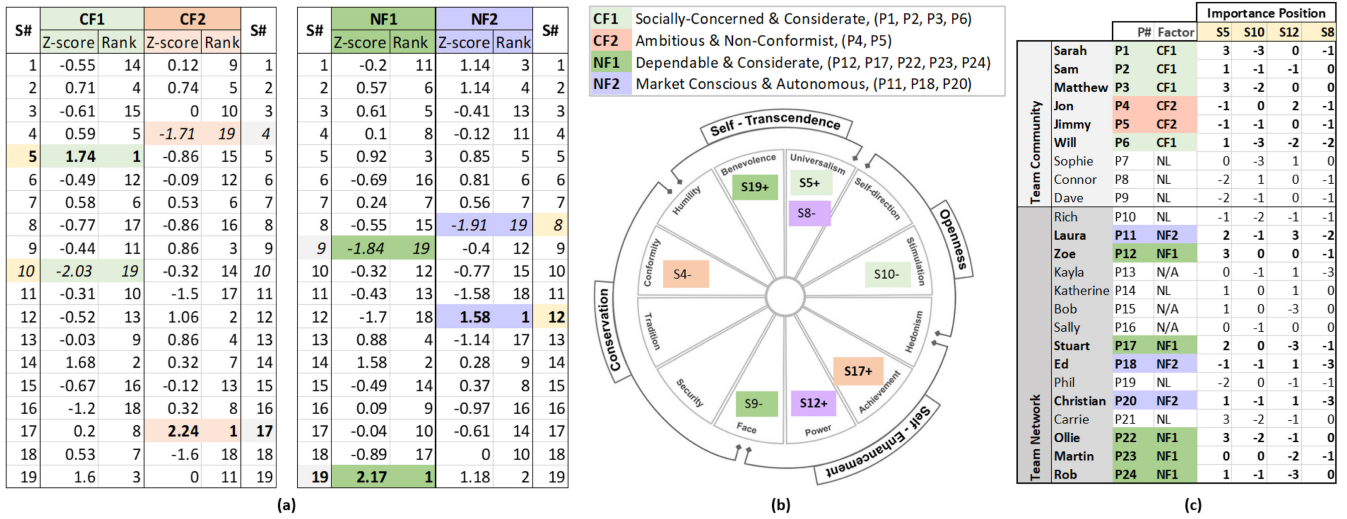


Fig. 4. (a) The four factors (CF1, CF2, NF1, NF2) extracted from the V-QS statistical analysis, together with the Z-score and ranks for each value statement (S#), the top and bottom ranked statement are highlighted (in yellow those selected for the qualitative analysis, Section 7); (b) the top and bottom statements are colour-coded with their factor's, and mapped onto the values model; (c) participants list, their factors, and importance score (from +3 to -3) of the value statements selected for the qualitative analysis (NL=P# not loaded on factor; N/A=P# not included in factor analysis).

Jimmy, both at $p < 0.05$), a software engineer and a BA. Personal achievement (S17, +3, 1st), commercially successful software (S12, +2, 2nd), and respect for own work (S9, +2, 3) are here considered important, and they rank significantly higher than in all factors suggesting the importance for this group of external validation for work. Risk taking (S10, -1, 14th), although receiving a marginally negative importance score (-1), is here ranked more highly than in any other factors. Conformity (both interpersonal and in terms of industry principles) is considered the least important value (S4, -3, 19th; S18, -2, 18th). Public good (S5, -1, 15th), trust and honesty at work (S19, 0, 11th), and not discriminating when developing software (S14, +1, 7th) are ranked lower than in any other factor.

6.3 Team Network, 2 Factors

6.3.1 NF1- Dependable and Considerate.

NF1 has an Eigenvalue of 3.59 and explains 30% of the study variance. Five participants loaded onto this factor, four software engineers/developers and an interaction designer (Zoe $p < 0.05$ only, Stuart, Ollie, Martin, Rob – last four also at $p < 0.01$). ‘Concern for others’ features as a high priority, particularly being honest and trustworthy with colleagues (S19, +3, 1st), and non-discriminating towards others (S14, +2, 2nd). Work-being respected and commercially successful (external validation) were low concerns (S9, -3, 19th; S12, -2, 18th), as was conformity to industry principles (S15, -1, 14th). Having fun at work and a safe and healthy workplace were also considered of some importance, and ranked higher than in any other factor. The high amount of software developers that loaded onto this factor suggests a certain values-alignment according to role.

6.3.2 NF2- Market Conscious and Autonomous.

NF1 has an Eigenvalue of 1.53 and explains 13% of the study variance. Three participants loaded onto this factor (Laura, Ed, Christian – all both at $p < 0.05$ and < 0.01), two people working in analytics and optimisation, and a product manager, again showing some potential values-

alignment according to role. This is the most difficult factor to provide with a simple summary. Developing software that is commercially successful and having freedom of thought are both rated as very important, and rank higher in this factor than in all any other factor (S12, +3, 1st; S1, +2, 3 rd). Being honest and trustworthy is also held as important, but it is ranked lower than in any other factor (S19, +2, 2nd). Care for environment is here positioned as the least important statement, and ranked lower than in any other factor (S8, -3, 19th). Overall, this factor seems to have a greater self-enhancement orientation than a self-transcendent one.

6.4 All-Teams: Overview & Discussion

Overall, out of the 21 participants, a greater number of practitioners (N=9), loaded onto the two factors (CF1 and NF1) that saw values related to Self-Transcendence (e.g., public good) as more important than values related to Self-Enhancement (e.g., commercial success). Specifically, the Self-Transcendence values which were statistically distinguishing here include: public good (S5- CF1, +3, 1st; NF1, +2, 5th); non discrimination (S14- F1, +2, 2nd; NF1, +2, 2nd); trust and honesty (S19- CF1, +2, 3 rd; NF1, +3, 1st).

Conversely, the other two factors, with a combined loading of five participants (N=5), favoured values linked to Self-Enhancement (S12), Personal Achievement (S17), and Self-determination (S1). For instance commercial success (S12) was top rated in NF2 (+3, 1st), and scored similarly high in CF2 (+2, 2nd), alongside values that are adjacent in the values model such as personal achievement (S17- CF2, +3, 1st), and independence of thought (S1- NF2, +2, 3 rd). These findings are important because the values orientations captured by the factor analysis seem to map onto the values relationships observed by previous empirical research from state of the art social psychology [13], [14], [15], and the results obtained in our pilot study [10], [12]. The mapping of job roles onto each factor is also note worthy, with software engineers, developers and QAs loading more onto factors oriented towards public concern, in-group trust and honesty, tolerance and non

Composite Q sort for Community Team Factor 1: CF1						
-3	-2	-1	0	1	2	3
*** that I am allowed to take risks when developing software S10	*** that I identify and address any environmental issues related to my work S12	*** that the software I develop is commercially successful S17	*** that I personally achieve high quality in software design S17	*** that the software I design is robustly and useably secure	*** that I do not discriminate against others when developing software S14	*** that the public good is the central concern of all professional computing work S5
	*** that I make my own decisions when developing software S16	*** to be given the freedom to produce new ideas, inventions and creative works	** that my workplace promotes my physical safety and S13 biological well-being	*** that I do not annoy or upset anyone in the course of my work S4	*** to be an honest and trustworthy colleague S19	
	to have fun when developing software	*** to improve public awareness and understanding of software S11	*** that I credit fully the work of others and refrain from taking undue credit			
	that I know and apply industry rules when developing software	*** that my work is respected S9	*** to uphold, promote and respect the principles of my industry S18			
		that the software I design influences the end user				

Fig. 5. CF1- Socially Concerned & Considerate, significant S# flagged.

discrimination, whilst other roles (e.g., product managers, analytics and optimisation) seem to be more oriented towards commercial success and personal achievement. Fig. 4b visualises these relationships by positioning, for each factor, the top (+) and bottom ranking (-) significant statements onto the values model.

7 QUALITATIVE DATA: FINDINGS & DISCUSSION

The factors are useful as they reveal statistically constructed types of values orientations and their relationships to each other (offering insight at L1). However, as stated in previous research [10], [12], and drawing on Q-methodology literature [44], 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 Q-sort semi-structured interviews. To summarize, the V-QS 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 provides some examples of consequent instantiations (L3).

7.1 Method

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" [12]. Corroborating information was included such as the factor the participant loaded on to (where this was the case) and how each participant rated the importance of the statement (from +3 to -3); this is summarised in Fig. 4. How each participant interpreted, reflected upon and reacted to the statement was then thematically analyzed [45]. Space and scope of the present research do not permit a full exploration of the qualitative results, the 24 interviews having produced just under 24.5 hours of recorded material and over 195,000 words of transcribed material. Of these, more than 72,000 were identified as either an interpretation or an instantiation of a specific values statement (about 37% of the total narrative elicited). Instead, we consider identified key themes in

relation to the top (1st) and bottom (19th) ranked values statements from two factors (CF1, NF2), one factor from each team.

The two steps followed for the selection and analysis were: **1)** from Team Community, we chose the factor *CF1- Socially Concerned and Considerate* because, similarly to NF1 in Team Net, it is characterised by Self-Transcending values orientations with *public good (S5)* and *taking risks (S10)* top and bottom ranked, respectively. **2)** From Team Net, we then chose *NF2- Market Conscious and Non Conformist* because, similarly to CF2 in Team Community, is characterised by Self-Enhancing values orientations with *commercial success (S12)* and *care for the environment (S8)* top and bottom ranked, respectively. During the thematic analysis, the themes were identified by Author 1 through three iterations. The iterative coding was conducted after a calibration exercise, where both Authors independently coded S5 qualitative data, discussed the rationale of their coding strategy (focusing on L2 and L3, often signalled by the interviewer's prompts). When reporting on the findings of the four values statements, we group the different values interpretations (L2) in themes - the thematic clusters identified during the thematic analysis. Each theme is reported in bold. For each theme, we then report their instantiations (L3). In addition, for each values statement, we include its narrative 'elicitation power' as the total number of words that participants used to discuss and articulate them. As convention and to help with cross referencing, we include the role of the named participant the first time they appear in this section. Raw data and thematic codes for the statements examined in this section (S5, S10, S12, S8) are included in the supplemental material, available online.

Whilst it was not always possible to capture a clear-cut distinction between values interpretations (L2) and instantiations (L3) for all our participants, there is evidence that our approach can help elicit and articulate values in a more structured and verifiable way - we provide examples in Section 9.3. Importantly, this discussion is not exhaustive but rather indicative of the capability of the V-QS to provide data at L2 and L3, and to demonstrate the complexity of values when they are studied at different levels, defying any simplistic interpretation of the statistics.

7.2 S5- Public Good

S5-Public Concern: It is important to me that the public good is the central concern of all professional computing work. S5 is a positively distinguishing statement in CF1 (+3, 1st) and negatively distinguishing in CF2 (-1, 15th). The V-QS generated nearly 9,000 words - or 45 min, at 100 words per minutes (11% of the total values narrative). This makes S5 the second top value in terms of narrative elicitation power after tolerance and non-discrimination with nearly 12% of the total (S14). The qualitative analysis identifies three broad themes for S5, and within each theme we report quotes with instantiations' examples.

The 'Right' User Experience. A number of participants linked public good to user experience. However, their views on what is 'right' for the end-user differ, even within members of the same team. For instance, in Team Net, *Christian* (analytics and optimisation) understands the public good in terms of a well structured and effortless user experience by '*getting them to find what they want quickly and easily*'. *Laura* (search engine optimisation) agrees that public good is about 'optimizing' user experience by leveraging users' data trails

to help structure and personalise their journeys; *'analyse that data once the user hits our website, we would then optimise that behaviour'*. Stuart's opinion (platform engineer) could not be more different; he thinks that end-users should be free to decide how to navigate the website without being too directed: *'you're guiding them down a route [...] and it's not necessarily what they would do [...] if people don't express themselves the way they want, then that's not in the public good'*.

Having impact- Participants also linked public good to the impact that the software they develop may have on people's lives and the wider society. This view varies from providing a positive impact to minimising harm. For software developer Martin, *'minimising harm to people'* is key, whilst BA Carrie emphasises that software should have at least *'neutral impact'* [...] and try to minimise any negative effects. SE Sarah, however, interprets public good as having a *'good influence on the world around me'*, a view more akin to S6-Power on People, in terms of the influence that software can have on people's lives.

A hindrance?- For some participants the public good cannot be relevant to all computing work. Martin wonders whether the public good might be actually *'at a detriment'* of the role of developers maintaining servers who are simply doing their job. However, BA Jimmy, has a different viewpoint from Martin, and see public good central to his work: *'A lot of what we were trying to do (in digital product research) was to think whether Digital Share could deliver products and services that would help solve some of society's problems.'*

7.3 S10- Stimulation / Taking Risks

S10-Stimulation: It is important to me that I am allowed to take risks when developing software. S10 is a negatively distinguishing statement in CF1 as it ranked significantly lower in CF1 (-3, 19th) than in CF2 (-1, 14th). With just over 3,300 words (4.6% of the total), its elicitation power is ranked 12th out of 19th. The analysis identifies five themes.

An expression of autonomy- Connor, an interaction designer, makes an explicit connection between S10 and the values of self-direction (S1, S16) and trust (S19) *'It ties in to freedom [...] I think trust as well is important. [...] if you're trusted or you feel trusted then (you can take risks)'*. SE Sarah also relates risk taking to self-direction (S1, S16), but differently to Connor, she sees it as something that she is not used to: *'(in my previous job) I was a resource, you do as you're told, and continues I don't see that (S10) as important because I lived without it for years'*. BA Carrie sees taking risk as linked to creative endeavour and for this reason she believes it should be limited to the first exploratory phases of software development. Christian (optimisation) enjoys the excitement of experimenting with something *'bold'* and to *'disrupt the journey a bit'*.

A balancing act- For a number of participants, taking risk is a balancing act between the risk taken and the value of the potential outcome, Rich for instance, states the importance of *'a really good balance between being able to take risks and try new things, versus [...] the value that that kind of work creates'*. QA Sam agrees, *'because although I like to take risks it doesn't come at the sacrifice of other things'*. Sam clarifies the need to carefully consider what is at stake, particularly in relation to data handling *'in GDPR terms we can't be that risky, we have to be careful with people's details'*. *Something that our organisation avoids-* A number of participants linked their

cautionary approach to risk to their organisation: *'we are generally a risk-averse business and that's rightly, because of all the history, the recent history [...] I probably couldn't tell you one particular big risk that I've taken. [...] you only measure yourself against other people'*, says principal designer Ollie who also notes that things are changing *'we are pushing the boundaries as well, we're doing some really good stuff. So I guess the vision is to create [...] a future-facing (company) with all the challenges that it comes with'*.

Something that can break things- For a number of participants, risk taking is something to be avoided, particularly for those who are then responsible to *'fix'* the broken systems. As platform engineer Rich puts it *'from a development point of view, it's us that fixes it when it goes down'*. QA/tester Sam, instead, quite likes taking risks, even when it means having to fix things after they break *'I quite like the idea of being risky, fix things and then there's fixing things on the fly, getting feedback from customers and bringing them into the process.'* Sarah instead explains why taking risks can be problematic *'might end up hurting someone or causing the company reputation or monetary damage in some way.'*

Something 'we' manage- Rich suggests an iterative approach to managing risk when implementing something new: *'if something is genuinely useful and genuinely is good, that involves a risk to implement it, then that will happen over time, and [...] if we approached this right and then, have you got obviously less of a risk.'* Sarah also sees agile as a way to manage risk *'I can't really think of any time where I would feel the need to take a risk, like it's not like I'm always playing it safe, I just feel like if we do take any risk, they're very small. Because [...] we're agile here we do really small releases'*.

7.4 S12- Commercial Success

S12- Power over Resources: It is important to me that the software I develop is commercially successful. S12 is a positively distinguishing statement for NF2 (+3, 1st), and negatively distinguishing for NF1 (-2, 18th). With just over 3,400 words (4.8% of the total) its elicitation power is ranked 11th out of 19th. The analysis identifies four themes.

A necessity- A number of participants view commercial success as a necessity. For instance, for Connor, commercial success is important *'because we're in a corporate world so that's how they measure success. It's the bottom line isn't it?'*. Christian also sees commercial success as needed for the company to stay alive (*'we need to make money, unfortunately'*); BA Jimmy adds a job security dimension stating *'if the business doesn't make money then I guess we all get made redundant possibly [...] Which is probably why number 12's higher than number 5 (public good)'*. However, a number of participants also stress that the company gives back its share of success to communities that need it most, as Sam notes *'they reinvest a lot of their profits so we want to make more money to give it back to people.'*

A measure of impact- Some see commercial success as an important way for assessing the impact of the systems developed. For instance, SE Jon states *'I've worked on far too much stuff that hasn't actually ever got used or actually delivered anything or improved anything'*. Sophie, sees measurable commercial benefit as the key rational in decision making: *'that's how I've been programmed the majority of my career [...] I've just always previously worked in an environment where you had to present a commercial benefit of doing something.'* Whilst this can

cause tensions (*'people get upset by it'*), she explains *'I want to have an understanding of why am I doing something, so whether it's to improve something for an end user, or improve the flow, or it's going to make us money or save us money'*.

Relative to other values- A number of participants view commercial success in relation to other values, and rate it as not as important. For instance, for Will, commercial success may signal a well designed system, yet it is not as important as writing high quality software (S17) *'I can write really, really good software, whether it's commercially successful or not kind of isn't really my problem [...] I like to write good software, I like to write it well, it's well-tested, all of that kind of stuff, the values that I like about software development, whether or not that's going to go and make a lot of money or something it's not, it's much less important'*. Sarah, instead, describes commercial success in relation to public good (S5) and she does not consider it as important: *'(I'd) much rather be doing good without having to think about how much money you make'*.

A source of tension- Delivery manager Katherine considers commercial success as her top priority, but she also sees it as a source of tensions: *'we're trying to do good but we're also trying to make money and which is the most important and what I do'*. She continues, *'There's lots of ways that you can encourage people to spend more on an e-commerce site and they talk about them as being dark patterns during the design of the organisation's e-commerce website people started to feel uncomfortable [...]'*, Laura (search engine optimisation) also reflects on the tension between end-users needs and business goals *'from a digital perspective it's all very user-centric, which is the most important thing. But then there will obviously be business centric needs that we need to fulfil'*.

7.5 S8- Care for the Environment

S8-Care for the Environment: It is important to me that I identify and address any environmental issues in my work. S8 was a negatively distinguishing statement in NF2 (-3, 19th), and was ranked significantly higher in NF1 (-1, 15th), though never given a great importance (i.e., more than +1) in any of the factors automatically extracted. With nearly 3,700 words (5.1% of the total) its elicitation power is ranked 6th out of 19th. Four broad themes have been identified.

Not a concern- Some of the participants did not see environmental impact as a matter of concern for what they do. *'Well there is no real environmental issues for analytics and optimisation'*, says Christian *'It doesn't really come on my radar'*. Similarly, Jon states *'you never really think of IT as having an environmental impact'*. Laura does not see environmental impact as a concern *'because all of our work is powered by Google, it's separate and wouldn't really impact'*.

A concern, but not considered- Some of the participants are concerned about the environment, but they often do not know where to start. Connor, for instance, says *'it's hard to think of the impact [...] I'm just not aware of it, but...we're making websites and apps [...], is there any damage? I don't know. I'd be interested in that side of things'*. Katherine also feels that is hard to consider, because it is such a big issue and it is *'out of our hands'*. SE Sarah and product manager Ed also agrees that this aspect is of concern; *'not considering it'* and *'not knowing what to do'* make Sarah feel *'sad'*, and Ed *'ashamed'*: *'especially, you know, like more recently in our family we've got a bit more, a lot more green focused'* says Ed, identifying family

practices (e.g., buying eco-friendly products), but finding it difficult to think of actions specific to software practice.

Less power consumption- A number of participants linked software-related environmental impact to power usage, with references to computing power, data centres energy consumption, and CO2 impact. Most of them find that the information about SE-related power consumption is either not there or hard to find; Ed, for instance, recalls *'I remember trying to find this information, and I couldn't, [...] it's really difficult to get those numbers'*. Others see the use of renewable energy as a good step forward. SE Zoe, for instance, thinks that *'it's quite important to see how data centres are using electricity, so are they just like pulling it from the grid or have they got renewables? A lot of them are powered by geothermic energy now, which is quite cool.'*

Part of wider organisational values and practice- Some of the participants connect environmental care to wider company practices, as Ollie notes that *'the business is pretty good at renewable energy, anyway; We are- I think we are completely neutral [...] We do use a lot of post-its [...] but also try to make sure that any correspondence you have is done on email or Slack, so we're not needlessly wasting paper'*. Stuart takes a wider perspective on the issue, reflecting on his career-change decision *'a lot of what I would broadly describe as environmental issues are kind of inherent in the way (this company) works'*. Zoe suggests additional ways for translating environmental values into practice: *'If I had any control over it, how we host all our stuff, I would be very careful about where it's hosted, because at the moment most of it's with AWS. And I don't think Amazon is a very good company in general.'*

8 WORK IN PROGRESS: AFFECT AND VALUES

When conducting the V-QS interviews and then analysing the qualitative data, we noted an abundance of affective information [48] covering a wide range of feelings: (e.g., 'happy,' 'frustrated,' 'sad,' 'guilty'); positive feelings tended to be associated with being able to act upon values participants consider important, and negative feelings were more often associated with having to align with values not perceived as important or not being able to enact a value considered so. In Section 5.2 we report on research that finds that values are "supported primarily by affective information" in large part because people attach strong feelings to values [48]. Specifically, the intuition is that affective information can signal the presence of potential values alignments or misalignments in a project. The V-QS helps facilitate discussion around values, but it is not intentionally designed to capture affect. To address this gap, we have adapted an existing SE technique (the agile retrospective) to capture values-related affect at critical points of software projects. To this end, we first developed a proof of concept, and then piloted this technique engaging with 9 software practitioners in total. We here report on emergent findings to illustrate future directions for this research.

8.1 Values-Retros: Capturing Affect

Two researchers facilitated two focus groups, which we call 'Values-Retros' as they were designed to be similar to the agile retrospective (which can include 'emotions seismogram' [49], but with a 'Values-twist'. The design of the Values-Retro

was as follows: first, the group was asked to draw a *timeline* for their current project on a wall, highlighting its key stages, hurdles, and milestones. Second, each participant was given a different colour pack of sticky notes and asked to write a *personal highlight* and stick it on to the relevant point on the timeline. Each participant then explained their highlight to the group. This was repeated with a *personal low-point*. The group were then asked, whether, as a team, they had experienced *collective highlights and low-points*. The next stage of the Values-Retro was that each participant was given a list of the simplified V-QS statements, and invited to place post-its on the timeline where any of these *values felt either affirmed or challenged*. Discussion was then facilitated around this. Finally, participants were asked whether there was anything they might do differently as a result of doing the retro. The Values-Retros were allocated an hour, but in both cases ran over the allocated time, as participants were keen to continue the conversations elicited.

The Values-Retro method was trialled as a proof-of-concept with a small team (3 members from Team Net), and then piloted with 6 members from Team Community. Being a novel and not fully evaluated method, we do not provide an in-depth analysis of its findings, but report research highlights of the pilot (Section 8.1.1) to illustrate its capability to capture the link between emotions and values, and trace values alignments and misalignments throughout a project.

8.1.1 Values-Retro Pilot: Research Highlights

We here report on the key highlights from the Values-Retro pilot. The session was 90 min long, and involved 6 Team Community members. The first part of the retro closely mirrored a sprint retrospective, including reflections on what went well, what could have been better, and the associated emotions ('emotions seismograms' as in [49]).

The 'values-twist' was then introduced by asking participants to label the project high-points and low-points with the values statements (slightly more concisely worded than in Fig. 3). In our session, this proved to be a very effective way to pinpoint values alignments and misalignments, and to associate them to specific phases of the project and to the moods of the the team and its members. In the end, the project was identified by the participants to be generally aligned to 7 values, namely: S1 (creativity), S3 (having fun), S4 (not annoying colleagues), S7 (give due credit), S16 (autonomy of action), S17 (achievement), S19 (honesty). Interestingly, some of the values' 'emotional responses' fluctuated during the project. For example, Sophie, at first, struggled with S9 (work being respected), but then "*found that once we started putting stuff on the wall, they (another team) now are respecting what we're doing. It's just that maybe we weren't communicating it in an effective manner. So I feel there's a lot more cross-team respect than there used to be.*"

S12 (commercial success) was the one value found in general misalignment with the project. Generally perceived as a "taboo," not talking about it made Sophie "upset" (resonating V-QS findings on S12's polarizing nature). Finally, asked about what actions the participants would take away as a result of the retro, there was a general consensus that achievement (S17) should be celebrated more, as Sarah puts

it "*it feels we struggle on things and then when we finally do it, it's a bit of an anti-climax, because we just carry on and go to the next thing and don't really celebrate or take a moment*".

In summary, we find that Values-Retros can encourage reflection on the role that values play in a project, and help better understand the relationship between values and emotions. Giving its potential, we encourage further use of the method, and include the Values-Retro plan and values statements as additional material.

9 LESSONS LEARNED AND IMPLICATIONS

This work addresses three challenges, each scoped by a research question (RQ). Previous sections describe how we address them and discuss emerging findings. Here, we identify ten lessons learned (#01-#10LL) and group them by RQs. For each lesson, we draw (→) a key implication for SE research (IMPr) or practice (IMPp), noting that, given the practice-oriented nature of our co-designed research sprint, the distinction between the two is not always clear-cut and some implications can apply to both (IMPr-p).

9.1 Conducting Values Studies in SE

RQ1- How can SE research support a more systematic investigation of values in SE practice? What values theories and models should we draw from? RQ1 stems from the need to identify a scientifically robust approach to studying human values in SE. We report on four key lessons.

01LL: Defining Values- Values-related expressions such as 'fair' [31], and 'for good' [50] are increasingly associated with software systems, but their understanding is often left open to interpretation and lacks precision, hence calls for "practical definitions" of values [26]. To address this issue, our work uses Schwartz's values definitions (Fig. 3), adapting their language to make it more relatable to SE practice (Section 5). Through both the pilot and main case study, we find that practitioners are able to relate these definitions to their practice. → *01IMPr-p: Less ambiguous and easier to enact definitions of human values should be sought and used in SE research and practice* [26]. Such definitions should be scientifically informed and can be drawn from (and form the basis of) professional codes and policies; our research offers working examples of such definitions (Fig. 3) and how to construct them (Section 5).

02LL: From Values to Practice- Having clearer and, where appropriate, shared values definitions may be useful, but it is not sufficient to ensure that identified values are consciously translated into practice. For example, Bender *et al.* urge significant budget allocation to data curation and documentation of training data for language models found to 'encode biased views harmful to marginalized populations' [51]. In this case, resource allocation can be a practical and measurable way to turn 'public good' (interpreted as, e.g., avoiding harm to the most vulnerable) into action at an organisational level ('budget allocation'). → *02IMPp: SE research and industry should provide concrete examples and the rationale of how values translate into practice.* This should include documenting how values change through individual SE projects life cycles as well as the organisational life-span. Our research has identified a framework and gives

examples of tools that can help do so at both organisational and project levels.

03LL: Examining Values Relationships- The V-QS quantitative analysis (Section 6) has revealed recurrent patterns of values orientations such as alignment and tensions between certain values pairs (i.e., commercial success and public good); similar patterns were observed in our pilot study [11]. Though validating Schwartz's values relationship model was not one of the objectives of this research, the findings are interesting because they bring to light potential values patterns to be further investigated. SE research exists that has started considering values as part of a model. However, we note a tendency to treat values models as classification taxonomies [29], [30], giving limited attention to values relationships. → *03IMPr: When carrying out values research in SE, consider using values models and methodologies that can help identify values relationships.* Our research uses Schwartz's model and applies q-statistics to observe emergent values relationships (e.g., alignments and misalignments) in a structured and systematic way.

04LL: Identifying Ground Truth- Our qualitative analysis has revealed a wide range of interpretations and instantiations for each value. Researchers attempting to abstract values from existing artifacts (e.g., systems functionalities, requirements) *a posteriori* [30] would need to carefully consider their ground truth, as researcher-led values identification exercises will be a few steps removed from the values decision making processes undertaken by the practitioners and stakeholders themselves. → *04IMPr: Given the potentially wide ranging values interpretations, values-classification exercises should carefully consider their ground truth.* In our research, the V-QS and the Values-Retro helped unambiguously identify when a value appears in discussion with practitioners (e.g., what values practitioners are actually trying to action).

9.2 Eliciting and Measuring Values in SE

RQ2- What new tools can be developed and what existing SE techniques can be adapted (and how) to help elicit, articulate, and measure values in SE practice? In the Introduction we stated that Miller and Larson argue that software practitioners find it challenging to express "ideas about human values with language that is not as precise or articulate as the language routinely used to express technical ideas [8]. Hussain *et al.* [9] report on the difficulty of measuring values in SE projects. We report on three lessons on these points.

05LL: Adopting Mixed-Methods for Values Capture- The Q-sort is a powerful mixed-method for both eliciting qualitative narratives and extracting quantitative insights into complex issues in which human subjectivity is involved. It identifies measurable patterns of thought by using a systematic procedure and an analytical process that is structured and applicable in a variety of contexts. Our findings show that the V-QS grounded in language generated by computing professionals (the ACM Code) can offer an entry point for encouraging values articulation in industry, whilst allowing the computation of values orientations. → *05IMPp: At the start of an SE project, values sorting tools and mixed-method techniques such as the V-QS should be used to carry out values baselines.* This can help draft a project or product values statement and the values 'type' of a project (as in Fig. 5).

Both the values statement and type, can then be included in the design documentation, and used as a touchstone at key points of the project to monitor values changes. We have piloted this with SE students with promising results [11].

06LL: Adapting Existing SE Techniques- One of our research objectives was to engage with SE industry to consider values at different stages of the software development process. The aim was to help software practitioners map and reflect upon their project values and how they develop over time. By adapting existing SE techniques (e.g., [9], [11], [26]), the Values-Retros have helped explore both the temporal and the affective dimensions of values in SE decision making process. Specifically, we find that the affect dimension complements values articulations since it signals values alignments and tensions. → *06IMPr-p: Further investigation into the link between values and affect [52]* For instance, the V-QS qualitative data analysis shows a recurrent sense of 'guilt' or sadness associated with statements where participants were willing to do something they feel they should (e.g., care for the environment) but do not because they lack resources or know-how or both. Our initial observations suggest that the intensity of affect seems to be related to the strength of values (mis)alignments. We have shared the plans for the Values Retros as additional material and we'd encourage their adaptation and further pilots.

07LL: Adapting the Values Q-Sort- The Q-sort, as a general mixed-method technique, is a flexible and adaptable instrument. The V-QS, and its adaptations, is a tool that can be used to elicit values orientations not only with practitioners, but also with end-users and stakeholders. In addition, it can be redesigned for use in a variety of domains. For instance, we have adapted our V-QS to map principles of beneficial AI [53] onto Schwartz's values models and ran workshops with researchers and students in higher and secondary education [11]. Finally, we note that the present research was completed pre-Covid and we were able to conduct the exercise face to face with the V-QS statements printed on physical cards. We have since piloted digital versions using collaborative platforms (e.g., Miro). New commercial online Q-Sort analysis platforms have now become available, increasing access and usability of the Q-Sort technique. → *07IMPr: The V-QS is a flexible instrument and can and should be adapted.* However, the V-QS relies on the elicitation of rich personal narratives and should be conducted by trained researchers. The Q-Sort statistical results are also context specific and cannot be over-generalized [44].

9.3 Articulating Values in SE

RQ3- What can be learned from applying the identified research framework (RQ1) and tools (RQ2) to SE practice? Specifically, what can we learn about software practitioners' understanding and articulation of values, and their relation to SE practice? Often functioning as truisms [48], values are difficult to articulate, and not only in SE practice. We report on three lessons.

08LL: Values Awareness- The Q-sort has helped elicit values narratives, and some participants demonstrated a deeply reflective, practice-relevant understanding of fundamental values. For instance, Stuart, a senior platform engineer (P17 in Fig. 2), linked public good (S5) to the respect of end-users' autonomy of action (S16), and specific software

design decisions. However, we observed different levels of values awareness and articulation ability with other participants. For example, in the case of care for the environment (S8), Christian admitted not having thought about it and dismissed its importance, whilst Ed considered it as a concern, but did not know how to address it in SE practice. →08IMPPr: *Working with software practitioners helps raise (self)awareness of the challenges associated with the elicitation and representation of values in SE.* Rather than a weakness, we encourage uncovering these challenges as an important step forward for the SE community. A key objective of our research is to support practitioners understand and articulate values in SE, including their own. This is about encouraging reflection in and on action, linking to Schon's work on the reflective practitioner [11]. Our paper illustrates how that reflection can be supported in a structured and systematic way.

09LL *Capturing and Representing Values-* Based on the findings of our first pilot [12], we anticipated a greater challenge with L3 articulation (the concrete enactment of a value) than L2 (abstract, interpretation). However, in this case study, it was L2 that proved more difficult to capture. We noted that when asked to explain what a participant meant by a value, they often found it easier to describe how they applied the value in their practice instead of giving a high level definition of it. This seems to confirm recent findings by social psychologists whose experiments have highlighted the importance of observing values informed behaviour in bridging the gap between abstract values descriptions and concrete actions [54], [55]. →09IMPPr: *Whilst it was not always possible to capture a clear-cut distinction between levels, there is evidence that our approach can help articulate values in a less ambiguous and more verifiable way.* For instance, if we apply levels L1, L2, L3 to Laura's² feedback on public good, we can have the following values construct: "The system will promote *public good* [VALUE OBJECT (L1)], *interpreted as 'effortless user experience'* [VALUE INTERPRETATION (L2)], and *enacted by 'analysing user data-logs patterns and then optimise that behaviour'* [VALUE INSTANTIATIONS (L3)]". This format is similar to requirements' standard constructs (e.g., "Subject, Action, Constraints" ISO/IEC/IEEE 29148:2018).

To summarise, in this paper we are not investigating if a value is 'correctly' interpreted, or if the way in which it is instantiated (i.e., implemented) is morally right or wrong (Section 3.5). We are interested in the 'anatomy' of a value, and in finding ways to express it with less ambiguity and more relevance to SE. Doing so, we argue, is a precondition for meaningfully applying Ethics to SE practice.

10LL: *Considering the Wider Context-* Digital Share has a declared interest in embedding values in their SE decision making process. Even when participants were not able to fully articulate SE-relevant definitions and instantiations of values, there was a general consensus that their company was genuinely striving to enact positive values, from re-investing income in local community projects, to encouraging mutual-support amongst teams and colleagues. A number of participants also stated that their decisions for leaving previous employment and joining Digital Share were based

on its consideration for equity and the provision of a supportive work environment. The search for values alternatives in the work place seems to be gaining momentum with software professionals prepared to leave their jobs when facing situations incompatible with their values [56]. →10IMPPr: *SE professionals occupy positions of increased responsibility* [17] Our research is responding to such need by creating a safe and informed space for the articulation and deliberation of human values in the context of SE industry, research, and education [11].

10 THREATS TO VALIDITY

10.1 External Validity

The objective of our research — and of Q-Sort methodology in general — is not to provide generalisable results, but an in-depth, systematic investigation of values in a specific, carefully sampled context. Our replication package allows for our study to be repeated in different contexts.

10.2 Internal Validity

We took several steps to ensure the validity of our research results. To address potential bias derived by the exploratory nature of our work, a number of strategies were deployed, following advice in [57]. First, we carried out triangulation of quantitative and qualitative data analysis to inform, complement, and critique our findings (Section 6, Section 7). Our findings were shared and discussed with our industry partners in both closed and open settings [57]. In addition, the research involved prolonged in-situ contact with participants, helping the researchers to gain a "reasonable understanding of the issues and phenomenon under study" [39]. Finally, where inductive thematic analysis was conducted, we report examples of the rich narrative elicited from our participants.

Given the nature of values research, one potential threat to the validity of our results is the social desirability factor; that is, whether participants were likely to emphasise the importance of values that they thought were desirable in the context of the interview. Due to the strong values orientation of Digital Share and the risk of participants feeling they needed to echo the company's values, we conducted most interviews in bookable meeting rooms at the company to provide privacy for participants and encourage openness. We also made it clear that, although aggregated results would be reported back, no individual results would be shared. It was more challenging to mitigate the risk of participants tailoring their responses to whatever they thought the researcher might want to hear. The researcher made it clear to participants that the Q-Sort was not a test, and tried to create a relaxed, informal atmosphere. Whilst we cannot fully control for the social desirability factor, the range of values orientations that emerged from our research suggests that social desirability did not play a significant role.

10.3 Construct Validity

We piloted the Q-Sort with 12 software engineers from different sectors before conducting our industry case study. Due to the fact that participants narrate their card sort, we were able to pick up on any of the Q-Sort statements that might have been understood differently to the meaning we

2. Laura's quotes are reported in pp 10 and 11.

intended. This enabled us to rephrase statements where necessary for better understanding.

Within the context of the case study, we used the Q-Sort with a range of people in different jobs related to software development. We found the constructs — represented in the Q-Sort statements — largely remained valid. However, for the three participants whose role was too far removed from software engineering (content and visual designers), statements explicitly referring to ‘developing software’ were less relevant. As a result, we removed these participants from our analysis - this is also explained in Section 4.

10.4 Potential of Q-Sort Misuse

Our work seeks to provide developers with a way to articulate their values and consider values trade-offs in their work. Our work aims to do so in a way that facilitates personal reflection, while providing developers with a vocabulary to consider aspects of their work that might otherwise seem very abstract. However, there are risks that the Q-Sort tool could be misused in software development contexts; for instance, as we state in Section 6, and in [10], [12], Q-Sort statistics should not be used alone nor should be the basis for generalisation. Further, the Q-Sort is one of the tools to be considered in wider values-study research and should be used alongside other activities, such as diversity training, user research, and full values audits of software development projects. The activities we describe should also serve as a support rather than a replacement for corporate social responsibility initiatives and wider debates about equity and justice issues related to computing [58] [59].

11 CONCLUSION

We first summarise how we address the research questions (RQs), and then outline future directions.

11.1 Research Questions Summary

RQ1- How can SE research support a more systematic investigation of values in SE practice? What values theories and models should we draw from? We propose a research framework (Section 3) that draws from social psychologist Schwartz’s Universal Values Theory [14], and, following cognitive psychologist Maio, considers values as ‘mental constructs’ that can be studied at different levels: system, interpretations, and instantiations [13].

RQ2- What new tools can be developed and what existing SE techniques can be adapted (and how) to help elicit, articulate, and measure values in SE practice? In this paper we describe the V-QS, a values elicitation and measuring tool that adapts values-study instruments (e.g., Schwartz’s values survey [14]) to the SE context (Section 5). We also introduce the ‘Values-Retro’ as an example of how emotional response to values (‘affect’) can be included in agile retrospectives (Section 8).

RQ3- What can be learned from applying the identified research framework (RQ1) and tools (RQ2) to SE practice? Specifically, what can we learn about software practitioners’ understanding and articulation of values, and their relation to SE practice? Our findings report on both quantitative measurements of SE practitioners’ values orientations (Section 6), and rich narratives explaining practitioners’ interpretations and instantiations of values in SE practice (Section 7). We

then draw ten lessons learned and their implications for SE research and practice (Section 9).

11.2 Future Directions

Emerging research directions outlined in previous sections include the need to further investigate how values orientations may change with job roles and specialisms (Section 6), and the possible effect of different interpretations (L2) and enactments (L3) of the same value on teams and organisations (Section 7). In Section 8, ‘Work in Progress,’ we adapt the agile retrospective to explore the relationship between values and affect observed during the V-QS qualitative data elicitation (Section 7); emerging findings suggest that further research should be conducted investigating the extent to which affect can signal critical values alignments and misalignments in teams and organisations.

Crucially, a key objective of this paper was to study how values are interpreted and instantiated by SE practitioners. Though a variety of values’ interpretations and instantiations was expected, we argue that much of the observed wide-ranging diversity is linked to a lack of precision of values constructs in our participants’ SE practice. Future work should investigate if interventions such as those described in Section 9 (for instance, the adoption and adaptations of values tools and SE techniques described in Section 9.2) can affect - e.g., reduce - the range of values’ interpretations and instantiations within teams and organisations, and, most importantly, can do so without silencing dissonant voices, but by fostering constructive and transparent debates.

Less ambiguity in values constructs and a greater transparency in their documentation can also help a more reflective [60] and SE-relatable application of Ethics in SE (Section 3.5). Our argument is that we need to understand and articulate what values mean, how they relate to each other, and how they can be enacted, before consciously choosing between the values that should be prioritised in SE practice (i.e., those that are ethically ‘right’) from those that are potentially harmful to self and society (i.e., ethically ‘wrong’) [61], [62].

ACKNOWLEDGMENTS

We thank Mika Lassander for introducing us to Q-Sort methodology, Greg Maio for talking through his values theory, Lucy Hunt for helping facilitate the Values-Retros, and Digital Share for their time, welcome and insight.

REFERENCES

- [1] J. Grundy, H. Khalajzadeh, and J. McIntosh, “Towards human-centric model-driven software engineering,” *Proc. 15th Int. Conf. Eval. Novel Approaches Softw. Eng.*, 2020, pp. 229–238.
- [2] J. Whittle, M. A. Ferrario, W. Simm, and W. Hussain, “A case for human values in software engineering,” *IEEE Softw.*, vol. 38, no. 30, pp. 106–113, Jan.-Feb. 2021.
- [3] J. Grundy, “Human-centric software engineering for next generation cloud-and edge-based smart living applications,” in *Proc. 20th IEEE/ACM Int. Symp. Cluster, Cloud Internet Comput.*, 2020, pp. 1–10.
- [4] J. Whittle, “Human values in software: A new paradigm for requirements engineering?,” in *Proc. IEEE 27th Int. Requirements Eng. Conf.*, 2019, pp. 4–4.
- [5] K. Czarniecki, “Requirements engineering in the age of societal-scale cyber-physical systems: The case of automated driving,” in *Proc. IEEE 26th Int. Requirements Eng. Conf.*, 2018, pp. 3–4.

- [6] M. A. Ferrario, W. Simm, S. Forshaw, A. Gradinar, M. T. Smith, and I. Smith, "Values-first se: Research principles in practice," in *Proc. IEEE/ACM 38th Int. Conf. Softw. Eng. Companion*, 2016, pp. 553–562.
- [7] K. Shilton, J. A. Koepfler, and K. R. Fleischmann, "How to see values in social computing: Methods for studying values dimensions," in *Proc. 17th ACM Conf. Comput. Supported Cooperative Work Soc. Comput.*, 2014, pp. 426–435.
- [8] K. W. Miller and D. K. Larson, "Agile software development: Human values and culture," *IEEE Technol. Soc. Mag.*, vol. 24, no. 4, pp. 36–42, 2005.
- [9] W. Hussain *et al.*, "Human values in software engineering: Contrasting case studies of practice," *IEEE Trans. Softw. Eng.*, to be published, doi: [10.1109/TSE.2020.3038802](https://doi.org/10.1109/TSE.2020.3038802).
- [10] E. Winter, S. Forshaw, and M. A. Ferrario, "Measuring human values in software engineering," in *Proc. 12th ACM/IEEE Int. Symp. Empir. Softw. Eng. Meas.*, 2018, pp. 1–4.
- [11] E. Winter, S. Forshaw, L. Hunt, and M. A. Ferrario, "Towards a systematic study of values in SE: Tools for industry and education," in *Proc. 2019 IEEE/ACM 41st Int. Conf. Softw. Eng.: New Ideas Emerg. Results*, 2019, pp. 61–64.
- [12] E. Winter, S. Forshaw, L. Hunt, and M. A. Ferrario, "Advancing the study of human values in software engineering," in *Proc. IEEE/ACM 12th Int. Workshop Cooperative Hum. Aspects Softw. Eng.*, 2019, pp. 19–26.
- [13] G. R. Maio, "Mental representations of social values," *Adv. Exp. Soc. Psychol.*, vol. 42, pp. 1–43, 2010.
- [14] S. H. Schwartz, "Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries," *Adv. Exp. Soc. Psychol.*, vol. 25, pp. 1–65, 1992.
- [15] P. H. Hanel, L. F. Litzellachner, and G. R. Maio, "An empirical comparison of human value models," *Front. Psychol.*, vol. 9, 2018, Art. no. 1643.
- [16] D. Gotterbarn, "Software engineering ethics," *Encyclopedia of Software Engineering*. Hoboken, NJ, USA: John Wiley & Sons, Ltd, 2002, doi: [10.1002/0471028959.sof314](https://doi.org/10.1002/0471028959.sof314).
- [17] J. Van den Hoven, G.-J. Lokhorst, and I. Van de Poel, "Engineering and the problem of moral overload," *Sci. Eng. Ethics*, vol. 18, no. 1, pp. 143–155, 2012.
- [18] J. Van den Hoven, "Human capabilities and technology," in *The Capability Approach Technology and Design*, Berlin, Germany: Springer, 2012, pp. 27–36.
- [19] B. Friedman, "Value-sensitive design," *Interactions*, vol. 3, no. 6, pp. 16–23, 1996.
- [20] B. Friedman, D. G. Hendry, and A. Borning, "A survey of value sensitive design methods," *Found. Trends Hum.-Comput. Interact.*, vol. 11, no. 2, pp. 63–125, 2017.
- [21] B. Friedman and D. G. Hendry, *Value Sensitive Design: Shaping Technology With Moral Imagination*, Cambridge, MA, USA: MIT Press, 2019.
- [22] C. A. Le Dantec, E. S. Poole, and S. P. Wyche, "Values as lived experience: Evolving value sensitive design in support of value discovery," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2009, pp. 1141–1150.
- [23] B. Knowles, L. Blair, S. Walker, P. Coulton, L. Thomas, and L. Mullagh, "Patterns of persuasion for sustainability," in *Proc. Conf. Designing Interactive Syst.*, 2014, pp. 1035–1044.
- [24] J. Whittle, "Is your software valueless?," *IEEE Softw.*, vol. 36, no. 3, pp. 112–115, May./Jun. 2019.
- [25] M. Harbers, C. Detweiler, and M. A. Neerinx, "Embedding stakeholder values in the requirements engineering process," in *Proc. Int. Work. Conf. Requirements Eng.: Found. Softw. Qual.*, 2015, pp. 318–332.
- [26] D. Mougouei, H. Perera, W. Hussain, R. Shams, and J. Whittle, "Operationalizing human values in software: A research roadmap," in *Proc. 26th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2018, pp. 780–784.
- [27] S. Thew and A. Sutcliffe, "Investigating the role of 'soft issues' in the re process," in *Proc. 16th IEEE Int. Requirements Eng. Conf.*, 2008, pp. 63–66.
- [28] S. Thew and A. Sutcliffe, "Value-based requirements engineering: Method and experience," *Requirements Eng.*, vol. 23, no. 4, pp. 443–464, 2018.
- [29] H. Perera *et al.*, "A study on the prevalence of human values in software engineering publications," in *Proc. ACM/IEEE 42nd Int. Conf. Softw. Eng.*, 2020, pp. 409–420.
- [30] A. Sutcliffe, P. Sawyer, W. Liu, and N. Bencomo, "Investigating the potential impact of values on requirements and software engineering," 2021, *arXiv:2103.01309*.
- [31] S. Galhotra, Y. Brun, and A. Meliou, "Fairness testing: Testing software for discrimination," in *Proc. 11th Joint Meeting Found. Softw. Eng.*, 2017, pp. 498–510.
- [32] S. Russell, D. Dewey, and M. Tegmark, "Research priorities for robust and beneficial artificial intelligence," *Ai Mag.*, vol. 36, no. 4, pp. 105–114, 2015.
- [33] S. H. Schwartz *et al.*, "Refining the theory of basic individual values," *J. Pers. Soc. Psychol.*, vol. 103, no. 4, 2012, Art. no. 663.
- [34] A.-S. Cheng and K. R. Fleischmann, "Developing a meta-inventory of human values," *Proc. Amer. Soc. Inf. Sci. Technol.*, vol. 47, no. 1, pp. 1–10, 2010.
- [35] S. H. Schwartz, "An overview of the schwartz theory of basic values," *Online Readings Psychol. Culture*, vol. 2, no. 1, pp. 2307–0919, 2012.
- [36] S. H. Koch, R. Proynova, B. Paech, and T. Wetter, "How to approximate users' values while preserving privacy: Experiences with using attitudes towards work tasks as proxies for personal value elicitation," *Ethics Inf. Technol.*, vol. 15, no. 1, pp. 45–61, 2013.
- [37] T. Partala and S. Kujala, "Exploring the role of ten universal values in using products and services," *Interacting Comput.*, vol. 28, no. 3, pp. 311–331, 2016.
- [38] P. G. Kirchschlaeger, *Digital Transformation and Ethics*, Baden-Baden, Germany: Nomos Verlagsgesellschaft mbH & Co KG, 2021.
- [39] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for SE research" in *Guide to Advanced Empirical Software Engineering*, Berlin, Germany: Springer, 2008, pp. 285–311.
- [40] A. S. Lee, "A scientific methodology for MIS case studies," *MIS Quart.*, vol. 13, no. 1, pp. 33–50, 1989.
- [41] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empir. Softw. Eng.*, vol. 14, no. 2, 2009, Art. no. 131.
- [42] R. K. Yin, *Case Study Research and Applications: Design and Methods*, Newbury Park, CA, USA: Sage Pub., 2017.
- [43] C. Pierce, *Philosophical Writings of Pierce*, J. Buchler ed., New York, NY, USA: Dover, 1955, pp. 5–20.
- [44] S. Watts and P. Stenner, "Doing q methodology: Theory, method and interpretation," *Qualitative Res. Psychol.*, vol. 2, no. 1, pp. 67–91, 2005.
- [45] G. Terry, N. Hayfield, V. Clarke, and V. Braun, "Thematic analysis," in *The Sage Handbook of Qualitative Research in Psychology*, Newbury Park, CA, USA: SAGE, 2017, pp. 17–37.
- [46] ACM, ACM code of ethics and professional conduct, 2018. [Online]. Available: <https://www.acm.org/code-of-ethics>
- [47] R. E. Anderson, D. G. Johnson, D. Gotterbarn, and J. Perrolle, "Using the new acm code of ethics in decision making," *Commun. ACM*, vol. 36, no. 2, pp. 98–107, 1993.
- [48] G. R. Maio and J. M. Olson, "Values as truisms: Evidence and implications," *J. Pers. Soc. Psychol.*, vol. 74, no. 2, 1998, Art. no. 294.
- [49] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, Reading, MA, USA: Addison-Wesley, 2012.
- [50] M. A. Ferrario, W. Simm, P. Newman, S. Forshaw, and J. Whittle, "Software engineering for 'social good': Integrating action research, participatory design, and agile development," in *Proc. Companion Proc. 36th Int. Conf. Softw. Eng.*, 2014, pp. 520–523.
- [51] E. M. Bender, T. Geburu, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?," in *Proc. ACM Conf. Fairness, Accountability, Transparency*, 2021, pp. 610–623.
- [52] G. R. Maio, V. M. Esses, K. H. Arnold, and J. M. Olson, *The Function-Structure Model of Attitudes: Incorporating the Need for Affect*, Hove, U.K.: Psychology Press Ltd, 2004.
- [53] D. Hassabis *et al.*, "Asilomar AI principles," Aug. 2017. [Online]. Available: <https://futureoflife.org/2017/08/11/ai-principles/>
- [54] P. H. Hanel, K. C. Vione, U. Hahn, and G. R. Maio, "Value instantiations: The missing link between values and behavior?," in *Values and Behavior*, Berlin, Germany: Springer, 2017, pp. 175–190.
- [55] P. H. Hanel *et al.*, "Cross-cultural differences and similarities in human value instantiation," *Front. Psychol.*, vol. 9, 2018, Art. no. 849.
- [56] C. Miller and R. Coldicott, Ken-q analysis, 2019. [Online]. Available: <https://www.doteveryone.org.uk/report/workersview/>
- [57] J. W. Creswell and J. D. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Newbury Park, CA, USA: Sage, 2017.
- [58] L. Stark and A. L. Hoffman, "Data is the new what? Popular metaphors & professional ethics in emerging data culture," *J. Cultural Anal.*, vol. 4, no. 1, pp. 1–22, 2019.

- [59] S. Vakil, "Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education," *Harvard Educ. Rev.*, vol. 88, no. 1, pp. 26–52, 2018.
- [60] M. A. Ferrario *et al.*, The Denver Manifesto, chi2017, May, 2017, [Online]. Available: <http://www.valuesincomputing.org/background/chi2017-values-in-computing-workshop/the-denver-manifesto/>.
- [61] M. A. Ferrario, W. Simm, J. Whittle, C. Frauenberger, G. Fitzpatrick, and P. Purgathofer, "Values in computing," in *Proc. CHI Conf. Extended Abstr. Hum. Factors Comput. Syst.*, 2017, pp. 660–667.
- [62] C. Becker, G. Engels, A. Feenberg, M. A. Ferrario, and G. Fitzpatrick, "Values in computing (dagstuhl seminar 19291)," in *Dagstuhl Rep.*, vol. 9, no. 7, pp. 40–77, 2019.



Maria Angela Ferrario (Member, IEEE) received the BA and MA degrees in philosophy and social psychology from Università Cattolica di Milano, the MSc degree in multimedia systems from Trinity College Dublin, and the PhD degree in on-line intelligent systems from University College Dublin, in 2002. She is a senior lecturer with the School of Electronics, Electrical Engineering, and Computer Science (EECS), Queen's University Belfast. Her expertise lies in both Human Computer Interaction (HCI) and Software Engineering (SE), with focus

on human-centric issues of SE. She adopts participatory methods to software systems design and investigates the role of human values in software practice and of digital technology in environmental and societal change.



Emily Winter received the PhD degree from Lancaster University, in 2017. He is a lecturer in computer science with the School of Computing and Communications, Lancaster University. Her main area of expertise lies in the socio-technical aspects of software engineering. Her recent research has investigated two key themes: The role of human values in software development; and developers' perspectives about automatic program repair and how automatic program repair tools can best be designed for the human user.

▷ **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.**