

Measuring Human Values in Software Engineering

Emily Winter, Steve Forshaw, and Maria Angela Ferrario
School of Computing and Communications
Lancaster University, UK
{e.winter, s.w.forshaw, m.ferrario}@lancaster.ac.uk

ABSTRACT

Background: Human values, such as prestige, personal security, social justice, and financial success, influence software production decision-making processes. Whether held by developers, clients or institutions, values are both highly subjective and deeply impactful on software outcomes. While their subjectivity makes some values difficult to measure, their impact on software motivates our research. **Aim:** To contribute to the scientific understanding and the empirical investigation of human values in Software Engineering (SE). **Approach:** Drawing from experimental psychology, we consider values as mental representations to be investigated on three levels: at a system (universal, L1), personal (abstract, L2), and instantiation level (concrete, L3). **Method:** We design and develop a selection of tools for the investigation of values at each level. As an example, we focus on the design, development, and use of a *Values Q-Sort* built by mapping Schwartz’s universal values model onto the ACM Code of Ethics. **Results:** Q-statistic sorts work with smaller samples than R-statistic surveys; from our study with 12 software practitioners, it is possible to extract 3 values ‘prototypes’ indicative of an emergent typology of values considerations in SE. **Conclusions:** The Values Q-Sort combines the extraction of quantitative values prototypes that indicate the connections between values (L1) with rich personal narratives (L2) reflective of specific software practices (L3), and as such, it supports a systematic, empirically-based approach to capturing values in SE.

CCS CONCEPTS

• General and references → Empirical Studies

KEYWORDS

Software engineering, mixed methods, Q-Sort, human values

INTRODUCTION

Deeply held values influence software practitioners’ decisions in software production [2] and values tensions can lead to high-impact actions. The 4000-employee Google walk-out objecting to Google’s involvement with the DoD in the project Maven “*business of war*”¹ was fundamentally driven by values: “*Google’s stated values make this clear: Every one of our users is trusting us. Never jeopardize that.*”¹. The employees’ mobilization resulted in Google’s withdrawal from the \$15M project. Values considerations

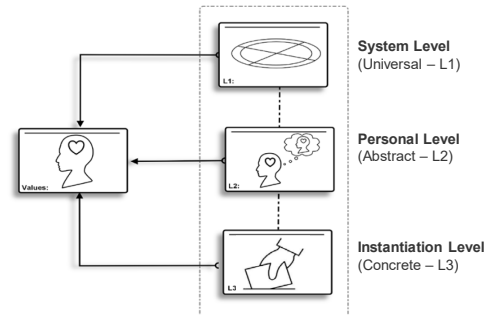


Figure 1: Values as mental representations to be studied on three levels: system (L1), personal (L2), and instantiation (L3).

also impact software systems design. For example, Facebook (FB) “*social by default*” original Graph API, which allowed access to users’ friends’ data and, through a ‘loophole’, put FB at the center of the Cambridge Analytica scandal, was driven by FB business-model values left unquestioned by an alleged “*ethical tunnel vision of software engineers*”². Although media views of SE ethical standards may be harsher than deserved, it is true that human values within SE seem to be considered either too difficult to define [12] or ‘soft’ aspects of SE [18]. There is indeed a growing body of SE that studies individual values (e.g. ‘fairness’ [7]). This research is important, yet we argue that studying values in isolation misses system level values interdependence (L1) [15]. Similarly, to solely carry out values systems research fails to capture the variety of personal meanings associated with each value (L2) and how they translate into concrete actions (L3).

In this paper we argue that the measurement of values in SE is complex and multilayered, but it is both a *possible* and *necessary* step for the systematic reasoning of values in SE. Our research goal is the development of a systematic approach for the elicitation, articulation, deliberation and representation of values across SE decision-making processes. The consideration of values as mental representations to be investigated on these three levels (Fig. 1) constitutes our theoretical underpinning and it is based on an established body of work from experimental psychology, drawing on Schwartz’s universal values model [15] and Maio’s work [10].

Our key research question is: How can human values be systematically studied and measured in SE? This paper explores this question through three sub-questions:

¹ <https://static01.nyt.com/files/2018/technology/googleletter.pdf>

² <https://gu.com/p/8cphg/stw>

1. What empirically-tested values theory can provide a robust theoretical framework to our study?
2. What existing subjectivity measuring methods can be adopted and adapted to values investigation in SE?
3. What values considerations influence software practitioners' decision making processes?

Our research is addressing such questions by:

- a. Anchoring SE values research to an empirically-tested values theory drawn from experimental psychology [10][15].
- b. Iteratively designing, piloting, and developing a selection of values elicitation and measuring tools specifically designed for the SE domain.
- c. Utilizing the tools developed in user studies with software practitioners

This paper outlines (a), and introduces the design, development and use of the Values Q-Sort (VQ-S, Fig. 2) as an example of (b) and (c). We then present 3 initial values 'prototypes' extracted from the Values Q-Sort study with 12 software practitioners. Such prototypes are indicative of a typology of values considerations in software production. We conclude with a reflection on the implications for future values study in SE.

2 RELATED WORK

2.1 Values, Morals and Ethics

Values can be defined as the guiding principles influencing our decision-making processes as individuals, groups and organizations. Ethics provide moral guidance through codified principles; morals describe the 'goodness' or 'badness' of actions [3]. There is a constant interplay between values, morals and ethics, making it challenging to study values in isolation. For example, Friedman's Value Sensitive Design (VSD) "emphasizes values with an ethical import" [5]; similarly, van den Hoven focuses on ethics and engineers' "moral overload" [20]. This work is important, but, in contrast to what we observe in [6], we argue for the study of the role of *all* human values in SE - not just an ethically oriented subset - as this is necessary for the systematic discovery of interdependences between values.

2.2 Measuring Values and Subjectivity in SE

It seems that little research on the measurement of human values in SE exists. Broadly related ESEM research focusing, for example, on human factors [1, 4, 9, 13, 14] appears to differ from ours in terms of approach and thematic focus.

Qualitative approaches - some of the qualitative research [13], for example, positions itself within the framework of grounded theory, an approach that is "well suited for situations where the researcher does not have pre-conceived ideas, and instead is driven by the desire to capture all facets of the collected data and to allow the theory to emerge from the data" [13]. Whilst grounded theory may present a useful approach (the opportunities and challenges for SE research use are exposed in [19]), social psychology already offers several empirically-tested theoretical frameworks for studying values. We thus position our research within an established body of theory [10][15] that offers a robust departure

point, but also enables a process of iterative reflection [3] on this theory's usefulness and limitations for SE research.

Quantitative approaches - survey-based quantitative research by Meyer et al. [11] identifies six 'types' of software developers in terms of their attitude towards productivity. These types are identified through the statistical analysis of an online survey. The Q-Sort we adopt has a similar output to [11] in that it offers a statistical method for prototype extraction. However, it also captures qualitative data about participants' interpretations of statements.

Thematic focus - research that investigates software practitioners' subjective views tends to focus on productivity [11] skills [9], and work motivation and satisfaction [4][14]. Such research is often motivated by a desire to improve software engineers' working practices and foster more productive workplaces [14]. Whilst this is an important aim, our approach is motivated by a set of wider societal concerns [2][3].

Articulating values - Miller and Larson [15] argue that software engineers have difficulty in "expressing ideas about human values with language that is not as precise or articulate as the language routinely used to express technical ideas". By mapping values statements to the highly structured ACM Code of Ethics [8] and using a systematic exercise like the Q-Sort we attempt to address such difficulty.

3 EMPIRICAL APPROACH

A series of tools were developed to examine values at the three study levels. We here focus on the VQ-S (Fig. 2). The Q-Sort is an established mixed method that was developed in the 1930s by the psychologist and physicist William Stephenson [17]. It is specifically designed for the systematic study of subjectivity by providing structure to subjective opinions [21]. The method involves asking participants to sort a series of statements onto a grid according to their level of agreement with each statement. One significant benefit of the Q-Sort is that it yields both qualitative and quantitative data. The sort is accompanied by a semi-structured interview, while the results of multiple sorts can be statistically analyzed. The interview helps to uncover the personal interpretations for each value (L2), while the patterns emerging from statistical analysis help to understand values at a system level (L1). Asking the participants to focus the sorting on a specific software project helps to anchor the reasoning at (L3).

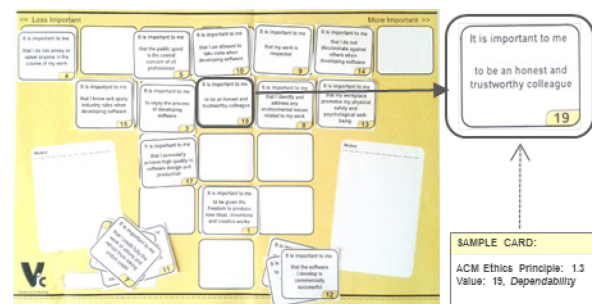


Figure 2. Sample VQ-S; participants order the statements on the grid according to their importance. ACM Code of Ethics principles are mapped to Schwarz Values [16].

4 METHOD

Values Q-Sort design - The VQ-S has been designed so that the chosen statements are both related to an appropriate model of human values and relevant to the community being studied (the SE community) [21]. Schwarz’s universal values model was used. After a first pilot iteration involving a senior software engineer and two team members using the original 57 values, it became apparent that the 57 statements were too many to keep the sorting focused, and that the original values descriptors were not specific enough for the SE community. As a result, the most recent [16] version of Schwartz’s value system, which identifies 19 distinct value types, was used. For SE relevance, two researchers examined the latest draft of the ACM Code of Ethics [8] and dual-coded its 24 principles according to the 19 Schwartz value types [16]. These codes were then compared; considerable similarity was found, areas of discord were discussed, and value duplicates removed.

This resulted in the most appropriate Code of Ethics principle for each value type being used as a Q-Sort statement. All value types were represented in the Code except four (‘Pleasure’, ‘Stimulation’, ‘Conformity’, and ‘Face’ [16]). In such cases, additional statements were developed. For example, for the value ‘Pleasure’, the statement “*It is important to me...to enjoy the process of developing software*” was added to the sort. The resulting statement set went through a second piloting cycle involving four computing researchers external to the authors’ institution before being finalized in its current form.

Values Q-Sort study - Q-Sort exercises and accompanying interviews have been carried out with 12 software practitioners, from a variety of sectors and with varying levels of expertise (Table 1). Interviews ranged in length from 45 minutes to 1.5 hours. All interviews followed a semi-structured interview guide. At the beginning of the interview, participants were asked about their role, career trajectory, and current projects. Participants were then invited to fill in the Q-Sort for a specific project they were working on. Upon completion of the exercise, questions were asked about the sort, focusing particularly on the participant’s top- and bottom-ranked statements. At the end, feedback about the exercise was collected. The project receive full ethical clearance from the Faculty of Science and Technology, Lancaster University. Informed consent was obtained prior each session.

Table 1: Values Q-Sort Participants

Participant	Gender	Experience years	Sector
P1	M	10-20	Public –research/envirom
P2	M	10-20	Public – tech support
P3	M	10-20	Public - media broadcast
P4	F	<10	Public – media broadcast
P5	M	10-20	Private – freelance
P6	M	<10	Private – industry
P7	M	>20	Private – industry
P8	M	10-20	Public – media broadcast
P9	M	10-20	Public - media broadcast
P10	M	10-20	Public –research/defense
P11	F	10-20	Public –research/health
P12	M	10-20	Private – freelance

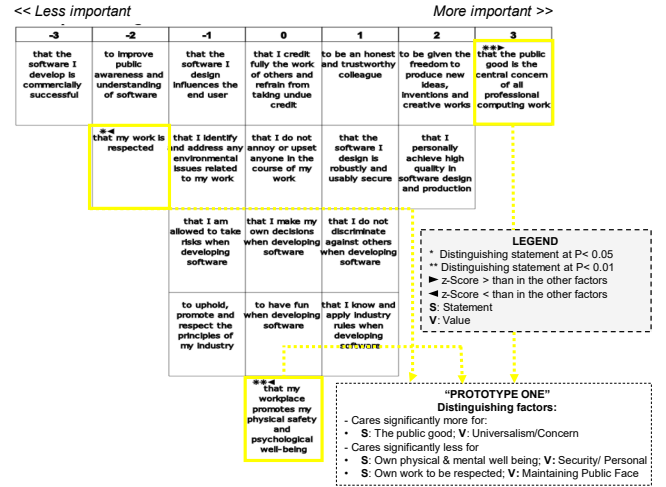


Figure 3: Composite Q-Sort for Prototype 1, *The Intrinsically-driven, Socially-concerned Software Engineer*.

5 EMERGING RESULTS

The Q-Sort data from the 12 participants were inputted into an online Q-analysis program - <http://shawnbanasick.github.io/ken-q-data>. The central element of this program is enabling factor analysis that attempts to identify “*patterns of similarity in the Q-Sort configurations produced*” [21]. The factors extraction allows for the emergence of statistically significant patterns. Factors are produced through centroid factor analysis, and each factor is given an Eigenvalue – the sum of the square of each of the individual Q-Sort loadings onto the factor – and a factor variance. High scoring Eigenvalues and factor variances indicate that a factor has “*strength and potential explanatory power*” [21]. According to the Kaiser-Guttman criterion, the Eigenvalue should be greater than 1; if it is less than 1, it accounts for less variance than an individual Q-Sort.

Of the eight factors extracted from our data, three had Eigenvalues greater than 1 and were thus 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). Our data was analysed using 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 [21]. Each of the three factor ‘viewpoints’ can be considered as a ‘prototype’ - an abstract type of software engineer. Fig. 3, introduces Prototype 1 as Factor 1 ‘viewpoint’. By examining its distinguishing factors, we can describe Prototype 1 as “*intrinsically-driven and socially-concerned*”. Early analysis of these three prototypes is included below, though space prohibits a full analysis of the associated qualitative data.

Prototype 1 - The Intrinsically-driven, Socially-concerned Software Engineer - This factor explains 19% of the total study variance, and three Q-Sorts (P2, P10, P12) have significant factor loadings (<0.59) to this factor. The highest rated statement by this

software engineer 'type' is 'that the public good is the central concern of all professional computing work'. This statement is more highly ranked, according to the z-score, than in the other factors, and is significant for $p < 0.01$. In addition, this software engineer values highly autonomy of thought and creativity and is driven by a desire to achieve high quality in software production. However, this achievement is interpreted in their own terms, and seems to be intrinsically motivated. Ranked lower than in other factors is 'that my work is respected' (significant at $p < 0.05$), so gaining external credit or recognition is of less importance. Commercial success is ranked lowly, as is improving public awareness of software. The other statistically significant item is 'that my workplace promotes my physical safety and psychological well-being'. This is ranked as less important than seven other items, and is lower than in other factors (at $p < 0.01$)

Prototype 2- The Autonomous, Nonconforming Risk-taker - This factor explains 17% of the variance, and two Q-Sorts (P1 and P5) have significant factor loadings to this factor. This software engineer likes to take risks (at $p < 0.01$), have freedom of thought, and for their psychological wellbeing and physical safety to be looked after. Of least importance for this software engineer is improving public awareness of software, the security of the software that they produce (at $p < 0.01$), and upholding the principles of the software industry. The statements relating to being an honest and trustworthy colleague (at $p < 0.01$) and the achievement of high quality (significant at $p < 0.01$) are lower ranked than in other factors. Of higher ranking than in other factors are the statements related to the software being commercially successful (at $p < 0.05$), the software influencing the end user (at $p < 0.01$) and addressing work environment issues (at $p < 0.01$).

Prototype 3- The Fun-loving, Extrinsically-driven Software Engineer - This factor explains 17% of the variance, and three Q-Sorts (P4, P8, P11) have significant factor loadings. This software engineer values most having fun at work (at $p < 0.01$) and achieving high quality. They place low value on the commercial success of the software they build, addressing environmental issues, and improving public awareness and understanding of software. Of the other statements, they are less likely than the other factors to place the public good at the centre of what they do (at $p < 0.01$) and to value autonomy of thought and creativity (at $p < 0.01$). They seem to be more conformist than the other factors being more likely to rank higher the upholding of the software industry's principles (at $p < 0.05$).

6 CONCLUSIONS AND NEXT

In this paper we argue that measuring values in SE is complex, but that it is also possible and necessary for the systematic study of values in SE. We outline our approach by introducing its theoretical underpinning drawn from experimental psychology [10][16] and consider values as mental representations to be investigated on three levels: at a system (L1), personal, (L2), and instantiation level (L3). We then outline the design, development, and use of a mixed-method tool, the VQ-S, specifically designed to study values on these three levels. Reporting on the first results of the Q-Sort data statistical analysis, we introduce three 'software engineer'

prototypes. With their defined structure, we consider such prototypes as a useful guide for the systematic reasoning of values in SE. Future work will include complementing the statistical analysis with the narratives collected. With the potential for this approach to be used more widely (we are looking at the design implications for VQ-S digital delivery and customization), limitations and concerns need to be flagged. Prototypes should be used as guidance only; without the analysis of the qualitative data, individual interpretations of each value are likely missed and the study may suffer from over-generalizations.

REFERENCES

- [1] Chagas, A.B., Melo, F.A., dos Santos, W.F., de Oliveira, A.A.N., Bora, S.M. and da Silva, F.Q.B., 2017. Analysis of the Understanding of the Concepts of Task and Skill Variety by Software Engineering Professionals. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (pp. 217-222). IEEE.
- [2] Ferrario, M.A., Simm, W., Forshaw, S., Gradinar, A., Smith, M.T. and Smith, I., 2016. Values-first SE: research principles in practice. In *Comp. Proc. of the 38th International Conference on Software Engineering* (pp. 553-562). ACM.
- [3] Ferrario, M.A., Simm, W., Newman, P., Forshaw, S. and Whittle, J., 2014. May. Software engineering for 'social good': integrating action research, participatory design, and agile development. In *Comp. Proc. of the 36th International Conference on Software Engineering* (pp. 520-523). ACM.
- [4] França, C., Da Silva, F.F. and Sharp, H., 2018. Motivation and Satisfaction of Software Engineers. *IEEE Transactions on Software Engineering*.
- [5] Friedman, B., 1996. Value-sensitive design. *interactions*, 3(6), pp.16-23.
- [6] Friedman, B., Hendry, D.G. and Borning, A., 2017. A survey of value sensitive design methods. *Foundations and Trends® in Human-Computer Interaction*, 11(2), pp.63-125.
- [7] Galhotra, S., Brun, Y. and Meliou, A., 2017. Fairness testing: testing software for discrimination. In *Proc. of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 498-510). ACM.
- [8] Gotterbarn, D.W., Bruckman, A., Flick, C., Miller, K. and Wolf, M.J., 2018. *ACM code of ethics: a guide for positive action*.
- [9] Iivonen, J., Mäntylä, M. & Itkonen, J., 2010. Characteristics of high performing testers: a case study. In *Proc. of the 2010 International Symposium on Empirical Software Engineering and Measurement* (p. 60). ACM.
- [10] Maio, G.R., 2010. Mental representations of social values. In *Advances in experimental social psychology* (Vol. 42, pp. 1-43). Academic Press.
- [11] Meyer, A.N., Zimmermann, T. and Fritz, T., 2017. Characterizing Software Developers by Perceptions of Productivity. In *Empirical Software Engineering and Measurement (ESEM), 2017* (pp. 105-110). IEEE.
- [12] Miller, K.W. and Larson, D.K., 2005. Agile software development: human values and culture. *IEEE Technology and Society Magazine*, 24(4), pp.36-42.
- [13] Racheva, Z., Daneva, M. and Herrmann, A., 2010. A conceptual model of client-driven agile requirements prioritization: Results of a case study. In *Proc. of the 2010 International Symposium on Empirical Software Engineering and Measurement* (p. 39). ACM.
- [14] Sach, R., Sharp, H. and Petre, M., 2010. Continued involvement in software development: motivational factors. In *Proc. of the 2010 Int. Symposium on Empirical Software Engineering and Measurement* (p. 44). ACM.
- [15] Schwartz, S.H., 1992. Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries. In *Advances in experimental social psychology* (Vol. 25, pp. 1-65). Academic Press.
- [16] Schwartz, S.H., et al. 2012. Refining the theory of basic individual values. *Journal of personality and social psychology*, 103(4), p.663.
- [17] Stephenson, W., 1993. Introduction to Q-methodology. *Operant Subjectivity*, 17(1), pp.1-13.
- [18] Stol, K., Ralph, P. and Fitzgerald, B., 2016, May. Grounded theory in software engineering research: a critical review and guidelines in *ICSE '16 Proceedings of the 38th International Conference on Software Engineering* (pp. 120-131), ICSE
- [19] Thew, S. and Sutcliffe, A., 2008, September. Investigating the Role of 'Soft issues' in the RE Process. In *International Requirements Engineering, 2008. RE'08. 16th IEEE* (pp. 63-66). IEEE.
- [20] Van den Hoven, J., Lokhorst, G.J. & Van de Poel, I., 2012. Engineering & the problem of moral overload. *Science and engineering ethics*, 18(1), pp.143-155.
- [21] Watts, S. and Stenner, P., 2012. *Doing Q methodological research: Theory, method & interpretation*. Sage.
- [22] Webler, T., Danielson, S. and Tuler, S., 2009. Using Q method to reveal social perspectives in environmental research. *Greenfield MA: Social and Environmental Research Institute*, 54, pp.1-45.