Value Stories: Putting Human Values into Requirements Engineering

Christian Detweiler and Maaike Harbers

Interactive Intelligence Group, Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands
{c.a.detweiler,m.harbers}@tudelft.nl
http://ii.tudelft.nl

Abstract. Software systems can give rise to ethical issues, with human values such as privacy, autonomy and responsibility at their heart. Such issues often do not become apparent until software has been put to use, and are left to be dealt with after harm has been done. However, these ethical issues are influenced by decisions made during design. A range of approaches to technology design aims to consider ethical issues and their underlying values during design, when there is more room to shape a technology. These values-oriented approaches guide designers in identifying and analyzing value issues with technology, but do not focus on deriving requirements from identified issues. As a result, the knowledge that can be gained from values-oriented methods does not find its way into requirements engineering processes. To address these challenges, we present a method to extract values and related elements obtained through existing value elicitation techniques, and to document these elements in a format that is amenable to use in further requirements engineering activities. We present a case study as a proof of concept.

Keywords: requirements elicitation, values, ethical issues

1 Introduction

Software introduces new capabilities or modifies existing ones, and changes the way people do things. This can have desirable and undesirable consequences or ethical implications, which center on a system's impact on human values. Examples of such ethical implications include privacy and accountability issues with electronic patient record systems, bias in search engines, and user autonomy issues in decision support systems.

A system's impact on human values often is not considered until the system has been put to use and its desirable and/or undesirable consequences have come to light. To an important extent, this impact is the result of the system's functions and qualities. These are, in turn, the result of decisions made during the design process. A system's impact on human values need not be an afterthought; rather, relevant values should be considered early during design, when a system is still malleable. The design process should identify key values and ensure that the functions and qualities of the system support these values as much as possible.

In recognition of the need to address value issues during design, values are receiving increasing attention in some technology design disciplines. Methods that deal with values explicitly have emerged and matured perhaps most prominently within the Value Sensitive Design (VSD) framework [1]. These methods guide designers in identifying, conceptualizing, and analyzing values in light of a technology, evaluating technology, and pro-actively designing technology to support values.

The majority of work on VSD appears in the field of Human-Computer Interaction (HCI). This suggests that the concepts and methods introduced by VSD and related approaches receive limited attention outside HCI. As these methods tend to be geared towards HCI, it is also not clear how well they can be integrated with disciplines outside of HCI, such as Requirements Engineering (RE). Few methods offer little guidance on expressing identified values in design by specifying system functions and/or qualities to support the values in question, though the need for such specification is recognized (e.g., [2]). In RE, the few approaches that address values (e.g., [3, 4]) have not been tested and tried to the extent that more mature methods in VSD have, and lack widespread recognition. Thinking about values is not common practice in RE. As a result, values are not considered to the extent that they could be in design processes that involve RE.

We argue that this calls for ways to identify and analyze value issues in RE, and to specify functionality to support relevant values. To address these issues, we present an approach that helps 1. identify (potentially overlooked) groups of stakeholders; 2. identify and concretize their values; and 3. specify user stories that describe features to support these values. Our approach helps designers and stakeholders explore new ideas and issues beyond users, clients, and business goals, by considering both direct and indirect stakeholders and by identifying their values before focusing on system features.

In Section 2, we will discuss related work in VSD and RE. In section 3 we will present our approach. We will illustrate this approach in a case study in Section 4, and discuss the approach and draw some conclusions in Section 5.

2 Related work

In this section, we discuss key concepts in Value Sensitive Design, the most prominent values-oriented framework to have emerged within Human-Computer Interaction. Furthermore, we discuss approaches in Requirements Engineering that deal with values to some extent.

2.1 Value Sensitive Design

Value Sensitive Design (VSD) [1] draws on a number of key concepts. *Direct stakeholders* are those individuals or organizations that interact directly with a system or its output. *Indirect stakeholders* are all other parties affected by the use of the system. The latter are often ignored in the design process [1]. *Values* are another key concept, which can be defined as "what a person or group of people considers important in life" [1, p. 70]. VSD distinguishes between three types of values. *Explicitly supported values* are those that are required to be supported by the technology. *Designer values* refer to the designers or researchers personal values that implicitly guide design decisions. *Stakeholder values* are values that are important to some, though not necessarily all, stakeholders of a technology [5]. Differences among values can give rise to *value tensions* (such as privacy versus security) among stakeholder groups, when supporting one value in a technology challenges another.

These concepts are central to VSD's three-part methodology, consisting of conceptual investigations of key stakeholders and values; empirical investigations of actual or potential stakeholders and contexts-of-use; and technical investigations aimed at designing new technology to support selected values or examining how existing technologies support or hinder certain values [6]. These investigations can be applied iteratively and integratively throughout the design process. Suggestions for using VSD include starting with a value, technology or context of use; identifying direct and indirect stakeholders; identifying benefits and harms for each stakeholder group; mapping benefits and harms onto corresponding; conducting conceptual investigations of key values; and identifying potential value conflicts [1].

Several specific methods have been developed within VSD, some of which are suited to elicit knowledge about values *in situ*. In the Value Scenarios technique, designers or stakeholders write scenarios that consider stakeholders, pervasiveness, time, systemic effects and value implications of a technology to support long-term, systemic thinking in interactive design practice [7]. The Value Dams and Flows method conducts stakeholder surveys and, based on these, accounts for values in design by avoiding problematic features, by identifying and designing for values that stakeholders do wish to see the system embody, and systematically addressing design tradeoffs that concern values. [8]. Envisioning Cards [9] are a design toolkit that guides the design process through explanations of key themes and associated concepts, and provides focused design activities to address issues related to these themes in a systematic way. The toolkit consists of cards that fall into one of four categories of envisioning criteria: stakeholders, time, values, and pervasiveness. These methods help consider stakeholders, their values and a system's (long-term) implications on values, but do not guide the designer in specifying system functions and qualities to support the values in question.

2.2 Requirements Engineering

Before value issues can be dealt with in design, they have to be identified or elicited. A variety of elicitation techniques is available in Requirements Engineering (RE). These differ depending on the effort required and type of information they help elicit. Zowghi and Coulin [10] identify eight core techniques that cover the spectrum of available techniques: interviews, domain analysis, groupwork, ethnography, prototyping, goal-based approaches, scenarios, and viewpoints. As a requirements elicitation technique, scenarios or use cases describe interactions between users and the system, and help understand requirements [10] or even represent functional requirements [11]. The agile development method Extreme Programming (XP) uses user stories to describe features that provide business value to a customer [11]. User stories are comparable to use cases and the process of writing user stories can be seen as brainstorming in RE, which helps generate creative solutions to specific problems [11]. Though these techniques often involve stakeholders and their needs, few focus explicitly on both direct and indirect stakeholders and their values.

Notable exceptions with regard to values include [3, 4]. Thew and Sutcliffe's method uses a taxonomy of users' values, motivations and emotions, and provides process guidance to elicit and analyze these issues [3]. Though the method draws attention to values, it focuses on their implications for the Requirements Engineering process rather than specifying functions or qualities to support those values. Koch and colleagues present a method that focuses on approximating or identifying users' values based on their preferences for key (work) tasks [4]. The method helps identify user values, but uses these to adapt existing systems rather than generate new ideas for functions or qualities. Neither of the methods takes indirect stakeholders into account.

3 Our proposal: from values to requirements

In this section, we propose a technique to elicit requirements of a system to be developed that is based on a value analysis of the systems stakeholders. By that, we aim to connect the concepts used in VSD to those used in RE, and provide researchers in both fields the opportunity to profit from each others work. The technique involves the following five steps.

1. Analyzing the systems stakeholders
2. Analyzing the stakeholders values
3. Providing concrete situations with the values
4. Determining stakeholder needs
5. Creating user stories

In the remainder of this section, we will explain each step, and give suggestions for a practical use of the technique.The first step is to identify the direct and indirect stakeholders of the system-to-be, by asking who will interact directly with the system or its output, and who will be affected by the system or its output without interacting directly with it, respectively.

The second step, a value analysis, involves identifying values that are relevant to the stakeholders identified in the first step. This step can build on various VSD techniques used to elicit values. For instance, the Envisioning Cards and Value Dams and Flows techniques mentioned in Section 2.1 involve the analysis of stakeholders and their values. The envisioning cards deck contains cards with assignments and question like "generate as list of as many potentially implicated values as possible", and "what views and values do stakeholders bring to a system?". The Values Dams and Flows technique involves identifying potential harms and benefits of the system to stakeholders, and then analyzing values underlying the harms and benefits.

The third step of our technique involves coming up with one or more examples of concrete situation(s) that explain(s) why or how a particular value is important for the stakeholder who holds the value. This is important because one value can have different meanings in different situations for a stakeholder. For example, the value of privacy for a user of a system can mean that the users personal information stored in the system should not be shared with anyone, or that the user should be able to use the system on his own.

These examples of concrete situations are input for the fourth step, determining stakeholder needs. For each concrete situation, this step determines how to support the associated value for the stakeholder in question. For instance, to support or protect a users privacy with regard to undesired sharing of personal information, the user should be able to indicate which personal information can be shared with whom and under what conditions.

The fifth step is to write user stories in order to specify early requirements to meet the stakeholder needs expressed in the previous step. As discussed in Section 2.2, user stories can be compared to use cases in RE, which can help understand or represent functional requirements early in the development process. User stories commonly take the form: As a [role] I want [something] so that [benefit]. We propose to create user stories according to the following template: As a [stakeholder] I want [stakeholder need] so that my [value] is promoted/supported when [concrete situation]. Note that one stakeholder can have multiple values, one value can have multiple concrete situations, and one concrete situation can have multiple stakeholder needs, but that one user story is created for each stakeholder need (with the associated stakeholder, value and concrete situation). It clarifies and concretizes each elicited value in a specific situation, and describes a high-level requirement to support the value. This creates a result that draws on concepts from VSD. Moreover, the form of the result is familiar in agile development methods and arguably within RE, making it easier to integrate with existing practices in those fields.

One way to use this technique in practice is to organize a workshop with stakeholders of the envisioned system, e.g. domain experts, potential users, and developers. We suggest the following outline for such a workshop.

A. Short presentation to introduce the participants to values and VSD, e.g. by providing examples of values in design (assuming that the participants are not familiar with VSD yet)
B. Identify direct and indirect stakeholders
C. Per stakeholder (this may be a selection of stakeholders identified in step B), identify one or more values and concrete situations
D. Per concrete situation, identify one or more stakeholder needs

The stakeholders identified in part B need not correspond with the participants, as part of the aim of our approach is to spark new ideas by having participants consider other perspectives than their own, similar to viewpoint approaches [10]. Stakeholder analysis (step 1), value analysis (step 2) and concrete situations (step 3) are taken together in one part (part C) because in a workshop setting it is natural to immediately explain a value to the other participants by providing a concrete situation. In a large group (e.g., when $n > 8$), part C and D can be done in subgroups of 4 to 5 participants. In that case, we suggest to perform part C and D together with all participants for the first stakeholder so that everybody understands what is expected of them. The last step of the technique, creating user stories, does not require new input of the participants, and can thus be performed by the workshop organizers afterward, though the resulting user stories should be examined by stakeholders to validate them.

To conclude, we proposed a technique to elicit requirements while accounting for values. The technique connects important concepts in VSD (stakeholders and values) to user stories, which can be refined into more specific requirements.. The proposed technique has two benefits as compared to other requirement elicitation techniques. First, paying explicit attention to values of direct and indirect stakeholder may lead to requirements that would not have been discovered otherwise. Second, knowing the values behind requirements provides an extra perspective in case of design trade-offs. This perspective allows developers to also reason and decide about underlying values instead of mere system features.

4 Case study

In this section we describe a case study in which we tested the technique proposed in the previous section by organizing a value-requirements workshop. First, we will introduce the project in the context of which we performed the workshop, then we will describe the workshop results, and based on that, we provide an evaluation of the proposed technique.

4.1 Context

The case study was performed in the context of the IQmulus project [12]. This is a 4-year European project in the area of intelligent information management. The main objective of the project is to enhance decision making by developing a system (the IQmulus system) that extracts relevant information from large, heterogeneous geo-spatial data sets. Such a system is needed because new data acquisition techniques are providing fast and efficient means for multidimensional spatial data collection, e.g. stereophotogrammetry, airborne LIDAR surveys, and SAR satellites. These techniques provide extremely high volumes of raw data, but in order to be useful, the heterogeneous data sets require harmonization and integration. The IQmulus project aims to make these data more accessible, and by that, help decision makers to make better choices, e.g. in case of flooding, flash floods or industrial accidents.

The project consortium involves partners with technical expertise, e.g. in the development of algorithms for data integration and information filtering, the visualization of data, or the development of architectures, and partners with domain expertise, e.g. collecting and using geo-spatial data for marine spatial planning or for rapid response and territorial management on land. The latter group are potential future users of the IQmulus system. The case study was performed in year 1 of the project. At that point, several other user workshops and requirement elicitation activities had been organized.

4.2 Value Stories Workshop

The value-requirement workshop was performed with representatives of several of the IQmulus consortium partners with domain expertise, i.e., some of the potential users of the IQmulus system. More specifically, the workshop involved 9 participants with representatives of the following institutes: 2 of IGN (French National Institute of Geographical Information and Forestry), 2 of Fomi (Hungarian Institute of Geodesy, Cartography and Remote Sensing), 2 of Regione Liguria (Genova, Italy), 1 of Ifremer (French Institute for Exploitation of the Sea), 1 of UBO (European Institute for Marine Studies), 1 of HR Wallington (Independent Research and Consultancy in Civil Engineering and Environmental Hydraulics). The workshop was performed such as described in Section 3, and led by the authors of this paper. The total duration of the workshop was 4 hours (excluding breaks). None of the participants was familiar with VSD before participation in the workshop. In the remainder of Section 4.2 the results of the workshop are provided.

In the first step, stakeholder analysis, the following 13 direct and indirect stakeholders of the IQmulus system were identified: consortium partners, public, European Committee, and users, which were divided into managers, GIS experts, water authorities, scientists, disaster management people, risk assessment people, spatial planners, data providers, operators (hardware and infrastructure), and decision makers. The participants agreed that the public and European Committee are indirect stakeholders, but that the other stakeholders can be either direct or indirect stakeholders of the IQmulus system.

From this list, three stakeholders were selected for further analysis. It was tried to select three stakeholders that are very distinct, but yet influenced by each other. The following selection was proposed by the workshop leaders, and agreed upon by the participants.

Decision makers who do not directly interact with the system (indirect stakeholder)
GIS experts who directly interact with the system in order to provide information to decision makers (direct stakeholder)
Residents of an area with flood risk, representing public (indirect stakeholder)

For these three stakeholders 31 values were identified, 50 concrete situations and 94 stakeholder needs. This resulted in 94 user stories. Due to space limitations, we cannot provide all these results here. Instead, we selected two representative user stories for each stakeholder. Table 1 shows the user stories with the associated stakeholders, values, concrete situations, and stakeholder needs, respectively.

Several interesting observations can be made from the table. The decision makers value of personal job security can lead to two conflicting stakeholder needs. On the one hand, personal job security may yield a need for metrics about the uncertainty of information (user story 1), but on the other hand, it may yield the need to not receive these metrics (user story 2). In the former case, the decision maker takes responsibility for the interpretation of (processed) data, and in the latter case, GIS experts take this responsibility. This tension is related to the GIS experts value of accountability. The more responsibility is shifted from the decision maker to GIS experts, the more important their need for information about data and algorithms becomes to account for the information they delivered to the decision maker (user story 3).

The table also shows a relation between the values of residents and those of the decision maker. For residents of a flood area, having information about the flood risks can increase residents trust in the information provided by decision makers (user story 5). This is in tension with user story 2, according to which the decision maker wants information of the type yes or no and safe or unsafe, instead of information about risk sizes.

User stories 1, 2, 3 and 5 are all related to the question of how much information the IQmulus system should provide about risks, uncertainty, and accuracy of the data it produces. The user stories show the impact of design choices regarding that question on the values of all three stakeholders. User story 4 and 6 are not direct requirements of the IQmulus system. However, the design of the IQmulus system may be impacted by the stakeholder needs in these user stories. For example, the IQmulus system could have a decision support function that selects central, but safe locations from where voluntary actions could be coordinated.

4.3 Observations

The workshop participants generally indicated that they liked the workshop, that they found it interesting to learn about VSD, and that they thought that VSD provided a valuable perspective on the use and development of the IQmulus system, a perspective they had not encountered before. The participants particularly mentioned that it helped them to reflect on the main aims of the project. Some of the workshop participants had been involved in organizing user workshops to obtain requirements for the IQmulus system themselves. They remarked that the experience and knowledge obtained in these user workshops helped them a lot in the value-requirement workshop.

The workshop leaders thought that the value-requirement workshop was successful in several respects. First, the participants seemed to understand the main ideas of VSD and what was expected of them in the workshop. Second, the participants were able to accomplish all steps in the workshop. Third, the workshop evoked discussions and information exchanges among the participants, and that seemed to contribute in developing a common view on the goals of the IQmulus projects. A potential drawback of the technique is that it requires a considerable

Stake-holder	Value	Concrete situation	Stakeholder need	User story
Decision maker	Personal job security	Being fired because of a wrong decision	information should be accompanied by metrics about its (un)certainty and/or accuracy if possible (this may increase responsibility of DM)	1. As a decision maker I want information with metrics about its (un)certainty and/or accuracy if possible to support my personal job security
			avoid fuzziness in information, provide 'yes' or 'no' type of information (this may increase responsibility of expert)	2. As a decision maker I want information without fuzziness to support my personal job security
GIS expert	Accounta bility	expert has to rely on processing, and tools when accounting for data provided	information about the quality of input data, quality of algorithms, knowledge of expert, and rightness of choices made	3. As a GIS expert I want information about the quality of input data, quality of algorithms, knowledge of expert, and rightness of choices made to support my accountability for data provided
		Awareness of accuracy of data in order to interpret results correctly	Training so that GIS experts know what is possible and why, which algorithm does what	4. As a GIS expert I want training to know what is possible and why, and which algorithm does what to support my accountability
Resident of an area with flood risk	Trust	believing that decision makers, authorities, system will provide correct information	information about potential risks	5. As a resident of an area with flood risk I want information about potential risks to support my trust in information provided by decision makers
	Solidarity	Having a way to help in crisis, play a part	information, coordination, and distribution of tasks depending on each person's expertise	6. As a resident of an area with flood risk I want a mechanism that coordinates and distributes tasks depending on each person's expertise to support solidarity

Fig. 1. Identified stakeholders and values

amount of time. In this workshop, 13 stakeholders were identified, but in the total duration of the workshop, 4 hours, only 3 stakeholders were analyzed.

5 Discussion and conclusions

In this paper, we introduced our approach to integrate values-oriented techniques with a requirements engineering process. Our approach helps requirements engineers and stakeholders identify values and translate them into a form that is suitable for use in the further requirements engineering process. In a case study, we demonstrated that our approach yields value stories user stories, a concept familiar in requirements engineering, that include values.

Future work should evaluate this approach in a design process, focusing on its ease of use and usefulness. In particular, evaluations should aim to assess the extent to which the approach helps designers without experience in dealing with values identify and understand value issues, and successfully formulate requirements to address these issues. Furthermore, future work can further support the integration of values into further requirements engineering activities by developing ways to formalize values and their relationships with other entities that currently figure in requirements specification.

Another promising direction for future work is the development of tool support. Tools could provide functions to support documentation of identified values, value stories, and related concepts, as well as provide means to visualize the resulting structure. By using such tools across design projects, designers could build up a reusable knowledge base or design patterns of values, containing the range of values encountered, as well as the requirements formulated to support these values.

References

1. Friedman, B., Kahn, P., Borning, A.: Value sensitive design and information systems. Human-Computer Interaction and Management Information Systems: Foundations. ME Sharpe, New York (2006) 348–372
2. Flanagan, M., Howe, D., Nissenbaum, H. In: Embodying values in technology: Theory and practice. Cambridge University Press, Cambridge, UK (2008) 322–353
3. Thew, S., Sutcliffe, A.: Investigating the role of 'soft issues' in the re process. 16th IEEE International Requirements Engineering, 2008. RE'08 (2008) 63–66
4. Koch, S.H., Proynova, R., Paech, B., Wetter, T.: How to approximate users' values while preserving privacy: experiences with using attitudes towards work tasks as proxies for personal value elicitation. Ethics and information technology **15**(1) (2013) 45–61
5. Borning, A., Friedman, B., Davis, J., Lin, P.: Informing public deliberation: Value sensitive design of indicators for a large-scale urban simulation. In: ECSCW 2005, Springer (2005) 449–468
6. Czeskis, A., Dermendjieva, I., Yapit, H., Borning, A., Friedman, B., Gill, B., Kohno, T.: Parenting from the pocket: value tensions and technical directions for secure and private parent-teen mobile safety. In: Proceedings of the Sixth Symposium on Usable Privacy and Security. SOUPS '10, New York, NY, USA, ACM (2010) 15:1–15:15
7. Nathan, L., Klasnja, P., Friedman, B.: Value scenarios: a technique for envisioning systemic effects of new technologies. In: CHI'07 extended abstracts on Human factors in computing systems, ACM (2007) 2585–2590
8. Miller, J., Friedman, B., Jancke, G.: Value tensions in design: the value sensitive design, development, and appropriation of a corporation's groupware system. In: GROUP '07 Proceedings of the 2007 international ACM conference on Supporting group work, ACM (2007) 281–290
9. Friedman, B., Hendry, D.: The envisioning cards: a toolkit for catalyzing humanistic and technical imaginations. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '12, New York, NY, USA, ACM (2012) 1145–1148
10. Zowghi, D., Coulin, C.: Requirements elicitation: A survey of techniques, approaches, and tools. In: Engineering and managing software requirements. Springer (2005) 19–46
11. Paetsch, F., Eberlein, A., Maurer, F.: Requirements engineering and agile software development. In: 2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE Computer Society (2003) 308–308
12. `http://www.iqmulus.eu`