Agile as a Vehicle for Values
A Value Sensitive Design Toolkit

Umbrello, S.; Gambelin, Olivia

DOI
10.1007/978-3-031-25233-4_13

Publication date
2023

Document Version
Final published version

Published in
Rethinking Technology and Engineering

Citation (APA)
Umbrello, S., & Gambelin, O. (2023). Agile as a Vehicle for Values: A Value Sensitive Design Toolkit. In F. Albrecht, & A. Santa-María (Eds.), *Rethinking Technology and Engineering: Dialogues Across Disciplines and Geographies* (pp. 169–181). (Philosophy of Engineering and Technology; Vol. 45). Springer. https://doi.org/10.1007/978-3-031-25233-4_13

Chapter 13
Agile as a Vehicle for Values: A Value Sensitive Design Toolkit

Steven Umbrello and Olivia Gambelin

Abstract The ethics of technology has primarily focused on what values are and how they can be embedded in technologies through design. In this context, some work has been done to show the efficacy of several design approaches. However, existing studies have not clearly pointed out the ways which design team managers can use design-for-values approaches to organise and use technologies in practice properly. This chapter attempts to fill this gap by discussing the value sensitive design (VSD) approach as a valuable means of co-designing technologies as a toolkit for existing workflow management, in this case, Agile. It will be demonstrated that VSD shows promise as a way of democratically designing technologies as well as fostering democratic technology policy innovation.

Keywords Value sensitive design · VSD · Agile · Systems thinking

13.1 Introduction

Since at least the 1950s, the philosophy and ethics of technology have been contending that technology and society influence each other (Winner, 1980). Although this view has changed over time, there nonetheless persists views of analysing the technical and non-technical aspects of society (i.e., that artefacts are inseparable from society) (Ropohl, 1999). In recent years, nation-states have been paying attention to the effects of this phenomenon. In addition, there is regional attention to responsible

S. Umbrello (✉)
Delft University of Technology, Delft, Netherlands

Institute for Ethics and Emerging Technologies (IEET), Wellington, CT, USA
e-mail: s.umbrello@tudelft.nl; Steven.Umbrello@eurac.edu; steven.umbrello@unito.it

O. Gambelin
Ethical Intelligence Associates, Limited, Brussels, Belgium
e-mail: olivia@ethicalintelligence.co

A. Fritzsche, A. Santa-María (eds.), *Rethinking Technology and Engineering*, Philosophy of Engineering and Technology 45, https://doi.org/10.1007/978-3-031-25233-4_13

innovation, like the EU Horizon 2020 projects (Veugelers et al., 2015), and international ones, such as with the United Nations Sustainable Development Goals (SDGs) (United Nations, 2018). What underlies this global orientation towards responsible innovation of new technologies is the notion that they are fundamentally systemic: they interact with social forces. They hence cannot be reduced to mere neutral instruments, instead, they are best understood as important carriers of values.

The value sensitive design (VSD) approach was developed in the early 1990s as essentially a means to reflect human values in technology design, rather than arbitrarily referring to them *ex post facto* or ignoring them altogether; both of which can do more harm than good (c.f., Winner, 1980). For over two decades, VSD has been demonstrated to be uniquely capable of being adopted across a wide range of design domains such as online browsers (Millett et al., 2001), urban simulation (Davis et al., 2006), care robotics (Umbrello et al., 2021; van Wynsberghe, 2013), artificial intelligence (Umbrello, 2019), energy systems (Mouter et al., 2018), and manufacturing technologies (Longo et al., 2020). Despite the extensive scholarship on the topic, as well as the systemic nature of VSD, existing studies have not addressed the issue of how and why VSD should be implemented/adopted at managerial levels within design domains, and not only by dispersed groups of interested engineers. Of course, engineers and engineering teams are often hierarchical and part of larger organisational structures, which introduce policies that usually govern the working principles of the design team. If VSD seamlessly integrates into the design domain – a fundamental conceptual precept of the approach – it also needs to be applied to the field of management.[1]

This chapter aims to address this critical gap by exploring how VSD is a design approach capable of being applied not only to the design of technologies themselves (i.e., the artefacts) but also to the seamless integration of the context of use in which it is adopted (i.e., the design domain/workflow). Our discussion is divided into the following sections. Section 13.1 looks at systems thinking and systems engineering, which globally forms the ontological foundation of much of the current trend in technology. Section 13.2 briefly outlines the VSD approach as an implementation of systems engineering and shows how it can satisfy many of the conceptual requirements of systems engineering. Section 13.3 shows how VSD is attractive not only to designers and engineers themselves but also to engineering management, a direct stakeholder and a subject of change. The change in engineering management affects the whole team. Finally, we will illustrate the applicability of VSD to the entirety of sociotechnical systems design (i.e., the design of the artefact(s) and its adoptability by the design domain) by showing how it fits within the Agile methodology, an example of a widely adopted methodology for project management. Section 13.4 discusses how engineering management can begin to think about implementing a VSD toolkit in their Agile process as well as how to increase its symbioses through an explicit orientation towards systems thinking. Finally, Sect. 13.5 presents our conclusions.

[1] It is worth noting that innovation management has already its own approach to values (see, for example, Breuer and Lüdeke-Freund (2016)). Although, this approach to values in management has not been explored as intensively as that of the VSD approach to technology design.

13.2 Systems Thinking and Systems Engineering

There are multiple reasons why systems thinking and systems engineering needs to be introduced in this discussion. Firstly, systems thinking in and of itself characterises the various levels of systems. Thus, it not only looks at the system *per se* (the artefact) but also the system in which it forms a part (i.e., the organisation). Secondly, VSD is fundamentally a systems engineering approach. Because VSD is fundamentally predicated on the *interactional stance* of technology that we mentioned in the introduction, it examines systems rather than artefacts in isolation or the organisation with only ancillary relation to design. As such, the focal points of VSD, like systems thinking, are the plurality of actors, institutions, technologies, and their design histories. Hence, for responsible innovation to occur, these various connected elements need to be given explicit and primary attention in thinking about design.

Broadly, systems thinking is the interdisciplinary study of organised and complex systems (Whitchurch & Constantine, 2009). The nodes, or parts that form a system, are both covariant and co-constitutive of each other and are thus dynamic in their relationship and complexity (Gorokhov, 1998). A system's environment and context of use both support and constrain its function, the latter of which is teleological in the sense that its intended use is determined by its operation (Adams et al., 2014). This complexity allows systems to be described as greater than the sum of their parts. Furthermore, the covariance of elements creates behaviours that any of the individual components may not produce. This *synergy* of constituent parts and the emergent behaviour that it generates specify a goal of systems thinking: the behaviour mapping of patterns to help predict such behaviours in different environments of use (Haken, 2013).

Systems engineering then takes this more ontological understanding of systems as the theoretical basis for its application in engineering and design (Albers et al., 2010, 2016). It takes a similarly interdisciplinary approach to the understanding, design, management, and deployment of engineered systems to ensure optimised equifinality over their lifecycles (Thomé, 1993). To achieve such equifinality, a system is mapped to determine how the parts that co-constitute it work synergistically and thus provide predictable emergent behaviours in different environments of use (Aslaksen, 2012). One can already begin to see how such a framing of engineering can be helpful when we talk about technologies like artificial intelligence based on machine learning, which learns and adapts based on many (different) environmental inputs.

This interdisciplinary approach to engineering employs many related human-centred areas of research, such as risk analysis, organisational studies, and project management, alongside technical studies such as mechanical, electrical, software, and industrial engineering, among others (Strijbos, 1998). Doing this takes the discipline of engineering itself as a system to be managed, leading to the conception of the design of the technical artefacts as *systems-within-systems*. This approach aims to map and design the way managerial and organisational structures in which these systems are developed both support and constrain certain design decisions.

13.3 Value Sensitive Design (VSD)

For almost three decades, value sensitive design has been a popular approach to designing technologies *for* human values. Rather than relegating them to *ex post facto* or *ad hoc* additions to systems, VSD is espoused as a principled approach to technology design that makes human values the goals of the design. Pioneered by Batya Friedman and her colleagues at the University of Washington in the early 1990s, VSD has since become one of the leading design approaches in the field of engineering ethics and responsible research and innovation (Winkler & Spiekermann, 2021). Although its various applications are outside the scope of this chapter, it does merit discussing some of the theoretical underpinnings of the approach. In addition, it will help us later in describing how it is effectively *de facto* an approach that functions at all levels of a design domain, as is required by systems thinking and engineering.

As mentioned, one of the fundamental theoretical precepts on which VSD is founded is that technology is not purely deterministic, instrumental, and/or socially constructed, but rather is *interactional*: it both supports and constrains social structures and vice versa. Consequently, technologies, and thus the social structures they form and how they are formed, can be described as *sociotechnical systems* (Ropohl, 1999). In the design of sociotechnical systems then, VSD is often described as a tripartite methodology constituted of (1) conceptual investigations, (2) empirical investigations, and (3) technical investigations (see Fig. 13.1). They are often undertaken consecutively, in parallel, or iteratively. Respectively, they involve (1) conceptual investigations into values and possible value tensions, (2) empirical investigations of the relevant stakeholders that enables one to define the way one understands their values and priorities and determine their values, and (3) determination and evaluation of what the technical limitations of the technology are and of the way they support and/or constrain identified values and design requirements.

Although many VSD applications begin with conceptual investigations, it does not inflexibly specify which one to start with. Instead, the approach is highly flexible and dependent on the contextual particulars of any given domain. VSD projects can then being in one of three considerations: (1) the context of use, (2) a technology itself, and/or (3) a value(s) (Fig. 13.2).

Friedman and Hendry (2019) propose 17 more specific methods that can be used in VSD: (1) stakeholder analysis; (2) stakeholder tokens; (3) value source analysis; (4) coevolution of technology and social structure; (5) value scenarios; (6) value sketches; (7) value-oriented semi-structured interview; (8) scalable assessments of information dimensions; (9) value-oriented coding manual; (10) value-oriented mock-ups, prototypes, and field deployments; (11) ethnography focused on values and technology; (12) model for informed consent online; (13) value dams and flows;

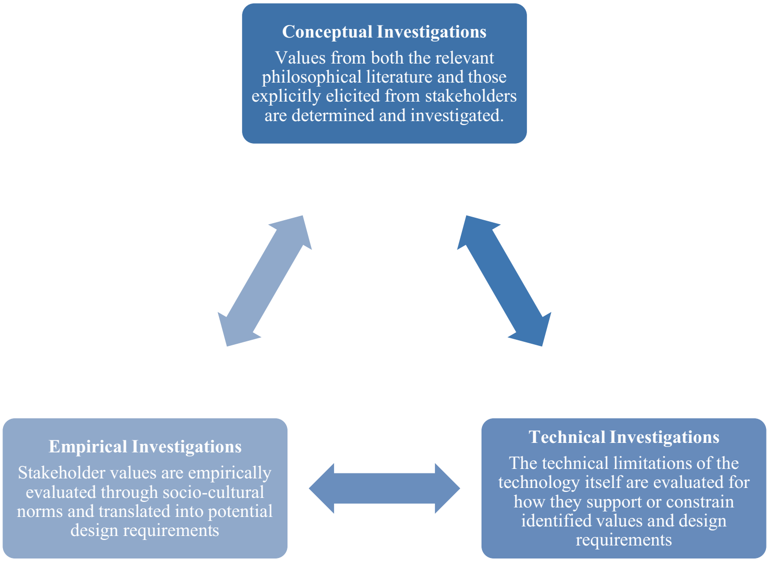

Fig. 13.1 The recursive VSD tripartite framework. (Source: Umbrello, 2020)

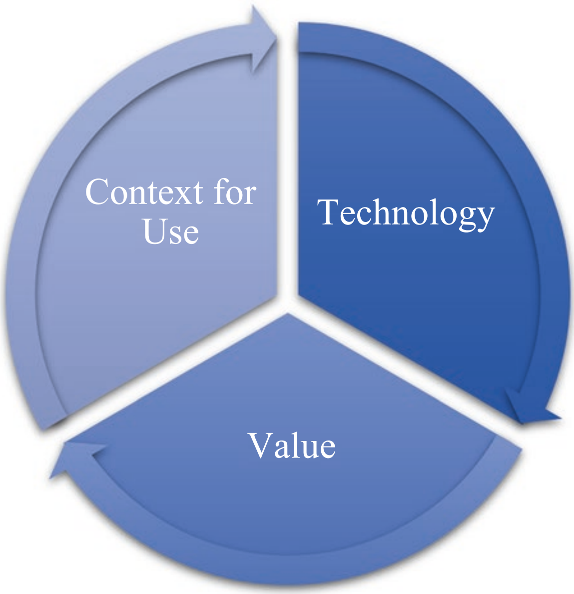

Fig. 13.2 Starting considerations for VSD. Typically, one of the three is most pertinent to any given design. (Source: Umbrello, 2021)

(14) value sensitive action-reflection model; (15) multi-lifespan timeline; (16) multi-lifespan co-design; and (17) envisioning cards. Each of the different methods used over the three decades to address different points within the VSD investigations is not mutually exclusive. Nor can they be exhaustive given that novel approaches from the social sciences and other disciplines emerge as useful means of carrying out the three investigations.

In its goal towards designing technologies *for* human values, the methods aim to identify the direct and indirect stakeholders and stakeholder legitimation, expand design spaces and identify value sources, elicit and represent values, and analyse values concerning social structure as multi-generational envisioning and design thinking. Different VSD methods are better suited depending on the chosen starting point and the system in question. However, we have to keep in mind that stakeholders are not only those outside the design domain, like public users, but also the designers themselves, management, and other members of design organisations. For this reason, VSD should be envisioned not only as a bottom-up approach adopted by designers but also as an approach supporting and constraining top-down structures in a design space, forming those structures and being informed by them. For this reason, VSD is not monolithic or confined to the design level proper but are useful across the various levels of technology design.

Consequently, describing VSD as having multiple levels of abstraction is essential for the ethical design of systems and ensuring that equifinality is obtained. Furthermore, the entire sociotechnical structure from which innovations are created is similarly constrained by the approach. The following section describes how VSD can be conceptualised across these levels of abstraction towards the goal of equifinality in both design and design spaces.

13.4 VSD to Design *for* Equifinality Via Agile

As already mentioned, VSD takes the sociotechnicity of technologies as a fundamental precept in its approach. However, if VSD aims to obtain true equifinality, a symbiosis in which a particular system or technology forms a part, then divorcing the approach from different levels of the broader structure like managerial domains and confining it primarily to the level of designers does a disservice. Up until this point, VSD has been applied and explored by designers themselves who have sought to explore its applicability and strengths. Its successes and methodological aptitude in designing *for* human values entitle it for larger-scale adoption. However, to promote it, we need to look at an instance that couples the various levels of abstraction, specifically the designer levels and the managerial levels, and illustrate how VSD works in both directions (bottom-up and top-down). We can look at Agile as a vehicle for this integration of VSD (Sacolick, 2022).[2] Agile methods call for collaborative, cross-functional, and self-organising teams, relying on managerial levels to choose which work to be prioritised while designer levels organise around the tasks.[3] This enables co-construction of the overall project, bringing in both bottom-up and top-down application of a central project vision. The result is a technology developed by all levels (i.e., *systems-within-systems*). Furthermore, agile creates an internal organisational environment that is otherwise lacking in VSD and provides a landscape on which VSD can be modulated.

The tripartite methodology of VSD, which comprises conceptual, empirical, and technical investigations, naturally fits into the iterative cycle of planning, executing, and evaluating that is characteristic of an agile team (see Fig. 13.3). Conceptual investigations begin during planning, as values are explored and provisionally designed into a system, and ends during evaluation. In contrast, teams reflect on potential value tensions and consider how to account for them in the next iteration. In between planning and evaluation, the execution phase creates space for technical investigations, as a team envisages the building and implementation of the design. Finally, due to Agile's focus on stakeholder feedback at the end of each iteration, empirical investigations into stakeholder values and priorities obtain the necessary insight in the evaluation phase. This maps on nicely to existing methods in Agile like Scrum where sprints are preceded by inception, product roadmap, release planning, sprint planning and other rituals, and are succeeded by sprint review and sprint retrospective (Mahalakshmi & Sundararajan, 2013).

Communication and adaptation is the core of Agile. Teams work within tight feedback cycles and strive for continuous improvement towards the central project vision. Agile has been criticised as a potential cause of poor decision-making, resulting in an unethical technology due to its rapid pace and occasional lack of high-level impact assessment. However, that does not mean it is not a helpful vehicle for value consideration, as Agile does create the structure by which VSD can be integrated into a project cross-functionally and multi-laterally throughout its

Fig. 13.3 The mapping of the VSD conceptual, empirical, and technical investigations onto the Agile cycle

[2] Although an argument can be made that the level of abstraction themselves can be designed by VSD, in fact, I have elsewhere shown this (Umbrello, 2021), the aim of this paper is to show that the VSD methodology can be easily adopted by Agile domains given that Agile's methodological structure allows for VSD's seamless integration.

[3] Agile can be considered an approach, a philosophy, or a set of frameworks that are used by design teams (such as Scrum, XP, SAFe, Lean Startup and the others).

lifecycle. In fact, the longer-term view and planning from VSD, if adequately embraced in a team, has the potential to solve these concerns of the short-termism caused by Agile practices.

13.5 Implementing VSD in Engineering Teams: An Agile Approach

In order to further explore the prospect of utilising Agile as a means for implementing VSD, let us examine the application of one of the 17 different VSD methods – the *Value Dams and Flows* – to the Agile cycle.

The *Value Dams and Flows* method focuses on avoiding system features that even a small number of stakeholders view as problematic (Friedman & Hendry, 2019). This involves identifying values as stakeholders want and seeking a design that best reflects them while continually addressing values-oriented trade-offs (Miller et al., 2007). A *Value Dam* refers to a technical feature or organisational policy a minority subgroup of stakeholders strongly opposes, which requires serious consideration of its negative impact. On the other hand, *Value Flows* are the values a large majority of the stakeholders would like to see embodied in the system, regardless of whether the resulting technical feature or organisational policy is necessary to the successful operational function of the system. The key is that once these dams and flows have been identified, designers must find a way to balance the two to address any values-oriented design trade-offs and/or moral overload (van den Hoven et al., 2012).

At the end of every iteration of the Agile cycle, a new system or feature is deployed to stakeholders for feedback. This feedback is then used throughout the next planning phase to determine what elements will be built in the following iteration. This stakeholder feedback mechanism within the Agile cycle is an ideal vehicle for the *Value Dams and Flows* toolkit. It first creates the engagement with stakeholders necessary for identifying the various dams and flows and then prioritises the insight gained from this feedback in designing the new features of a system.

Like most Agile workflows, they are broken down into 'sprints' or iterations, similar to how VSD iteratively designs innovations towards ever-greater equifinality. These pre-determined sprints allow for critical redesigning to take place as new information emerges across the workflow. Likewise, as Fig. 13.5 shows, redesigning as a function of mock-ups and prototypes is a crucial VSD method, just as it is inextricable from Agile. Doing so permits Agile teams to continue their current workflow patterns while simultaneously integrating a method of accomplishing it in a way that is explicitly oriented to designing *for* human values (Fig. 13.4).

The above figure is by no means an exhaustive representation of the potential of VSD tools, which can be employed at any given stage of the Agile cycle. Multiple tools should be used across the Agile process, given that the various stages of any given cycle parallels a tool in VSD that focuses on designing *for* human values and

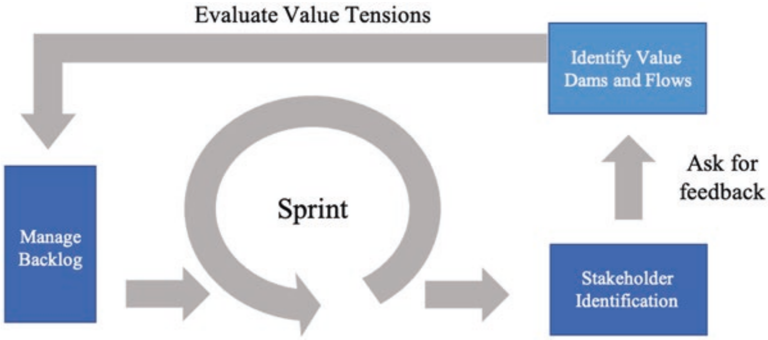

Fig. 13.4 *Value Dams and Flows* Toolkit in the Agile cycle

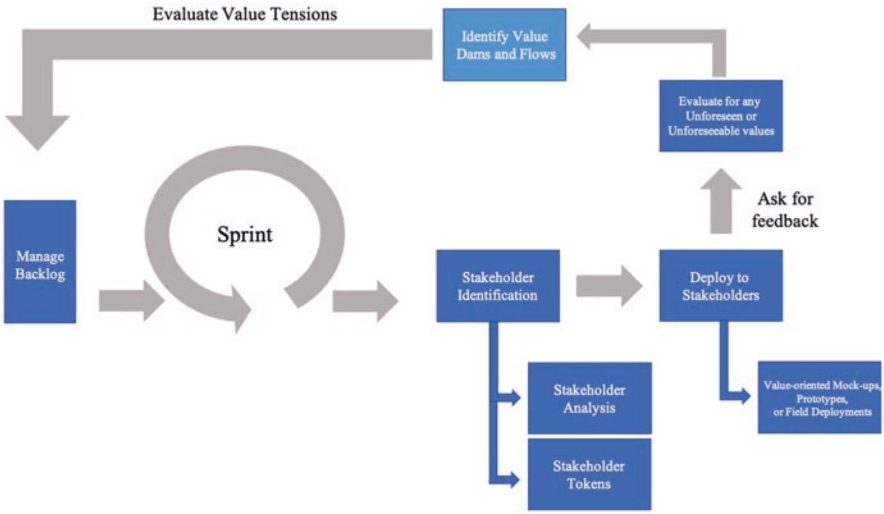

Fig. 13.5 Agile workflow with the addition of VSD stakeholder identification tools

can be mapped onto the cycle.[4] For example, ‘deploy to stakeholders’ requires knowledge of *who* the relevant direct and indirect stakeholders are as well as *what* constitutes safe yet salient deployment to those stakeholders. The former can be addressed by using two VSD tools: (1) *stakeholder analysis* for the identification and legitimation of the different relevant stakeholders (e.g., Czeskis et al., 2010) and (2) *stakeholder tokens* to facilitate the identification and articulate the interaction between stakeholders (Yoo, 2021). The latter regarding safe deployment can be

[4] This mapping is just that, and should not be construed as a ‘redesigned’ version of Agile *per se*. Inherent to VSD is its ability to be modulated to any design domain, thus increasing the potential for its uptake. Here we can see that Agile is *already* designed in such a way that allows for the seamless mapping of VSD methods.

resolved through *value-oriented mock-ups, prototypes, or field deployments* to determine if the technology safely embodies the designed values and to evaluate the system for potential recalcitrance, albeit in an incremental and controlled environment. By observing for any previously unforeseen (or unforeseeable) unwanted values, this can trigger another design iteration (i.e., redesign) (see Fig. 13.5).

In 'evaluating value trade-offs' or 'moral overload' as referred to in VSD scholarship, there are existing tools to help designers manage this stage of development (van den Hoven et al., 2012). The language of 'trade-offs', which is often framed in cost-benefit analysis language, is exchanged in VSD, and this paper, for a more inclusive approach to value tensions by co-creating value representations with stakeholders to determine points of overload and how to design systems to support and constrain such tensions via compromise. *Value scenarios, value sketches,* and the *value sensitive action-reflection model* are all potential tools that Agile team members can employ to establish a connection with the value understandings, and consequently, the technical design requirements for the values at play, all without replacing their already familiar day-to-day Agile practices. Figure 13.6 illustrates in greater detail the process for how Agile can use VSD. It's important to note the symbiotic relationship that VSD and Agile have when paired together; VSD is not overlaid on top of Agile, and neither does Agile become overly complicated with the addition of VSD. Agile already possesses many underlying ontological tools that allow an approach like VSD to integrate seamlessly into its workflow.

Umbrello and van de Poel (2021) argue that the VSD tool of 'value hierarchies' can be employed for artificial intelligence systems to help designers to translate different values (i.e., higher-order values, AI-specific values, and stakeholder/contextual values) through norms and then into design requirements (and vice versa). Similar heuristics can be used to streamline the evaluation of these apparent 'trade

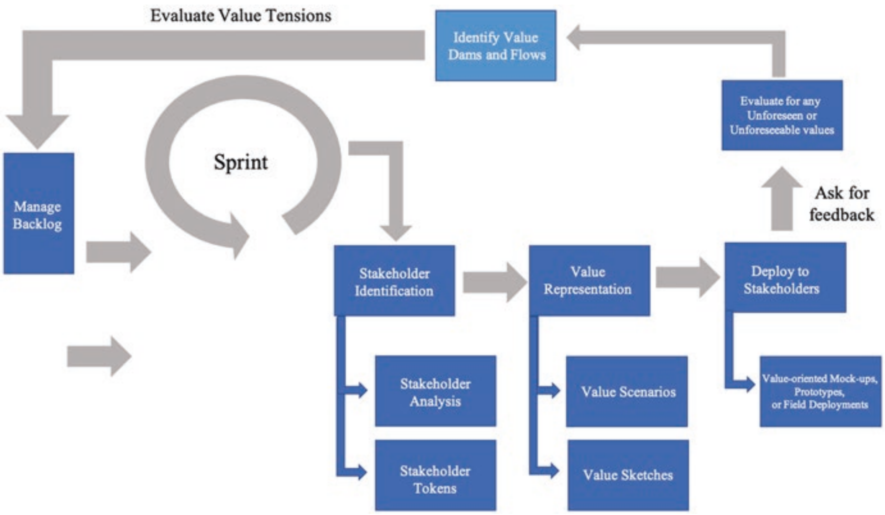

Fig. 13.6 Agile workflow with the addition of VSD value representation tools

off' scenarios by Agile design teams (e.g., Umbrello, 2018; van de Kaa et al., 2020; van de Poel, 2013, 2014, 2017; van den Hoven et al., 2012). Nonetheless, we have illustrated here that implementing a design-for-values approach, like VSD, is achievable not via wholesale overhauls of design domains and programmes but by invoking one of the fundamental tenets of VSD, i.e., that VSD can and should be integrated into existing practices and workflows. We chose to use Agile because it is a popular, globally-adopted workflow used by design teams. However, given that Agile is not fundamentally devised towards designing *for* human values, and given the growing importance of this in the modern world, we offer VSD as a toolkit that can be seamlessly integrated into current Agile workflows. A key benefit of the adoptability of VSD is the degree to which its use can be augmented as needed to any given situation while still promoting the benefits of designing *for* human values. Finally, Umbrello and Gambelin (2021) provide the actual field manual for putting VSD into action for Agile, bringing this more theoretical framework into actual practice.

13.6 Conclusions

Value sensitive design (VSD) is a principled approach to design that is increasingly being adopted for a wide range of technologies, and across many domains of use, given its potential for designing technologies *for* human values. To date, however, the literature on VSD focuses almost exclusively on the technologies themselves or the nature and state of human values in the methodological framework. Because VSD is predicated on the concept that technologies are sociotechnical, social structures influence and are being influenced by technology. This merits examining the social structures that support and constrain the domains themselves, e.g., managerial levels of design domains. This chapter argues that for VSD to achieve a genuinely salient design *for* human values, its tenets of support and constraint must operate seamlessly across all workflow levels, from the design domain to the domain of management. We have shown how team members at the managerial levels who employ project management methodologies like Agile can adopt VSD as a toolkit and how VSD can be used to iteratively redesign the internal roles and policies of the design domain towards a greater equifinality of innovation.

Acknowledgements We would like to thank Batya Friedman, David Hendry, and other participants at the VSD Quarterly Seminar on 8 April 2021 for providing helpful comments and suggestions on the initial draft of this work. All remaining errors are the authors' alone. The views expressed in this chapter do not necessarily reflect those of the Institute for Ethics and Emerging Technologies or Ethical Intelligence Associates, Limited.

References

Adams, K. M. G., Hester, P. T., Bradley, J. M., Meyers, T. J., & Keating, C. B. (2014). Systems theory as the foundation for understanding systems. *Systems Engineering, 17*(1), 112–123.

Albers, A., Braun, A., & Muschik, S. (2010). Uniqueness and the multiple fractal character of product engineering processes. In *Modelling and management of engineering processes* (pp. 15–26). Springer.

Albers, A., Reiss, N., Bursac, N., & Richter, T. (2016). iPeM–integrated product engineering model in context of product generation engineering. *Procedia CIRP, 50*, 100–105.

Aslaksen, E. W. (2012). *The system concept and its application to engineering*. Springer Science & Business Media.

Breuer, H., & Lüdeke-Freund, F. (2016). *Values-based innovation management: Innovating by what we care about*. Macmillan International Higher Education.

Czeskis, A., Dermendjieva, I., Yapit, H., Borning, A., Friedman, B., Gill, B., & Kohno, T. (2010). 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* (pp. 1–15).

Davis, J., Lin, P., Borning, A., Friedman, B., Kahn, P. H., Jr., & Waddell, P. A. (2006). Simulations for urban planning: Designing for human values. *Computer, 39*(09), 66–72.

Friedman, B., & Hendry, D. G. (2019). *Value sensitive design: Shaping technology with moral imagination*. MIT Press.

Gorokhov, V. (1998). A new interpretation of technological progress. *Society for Philosophy and Technology Quarterly Electronic Journal, 4*(1), 16–20.

Haken, H. (2013). *Principles of brain functioning: A synergetic approach to brain activity, behavior and cognition* (Vol. 67). Springer Science & Business Media.

Longo, F., Padovano, A., & Umbrello, S. (2020). Value-oriented and ethical technology engineering in industry 5.0: A human-centric perspective for the design of the factory of the future. *Applied Sciences, 10*(12), 4182.

Mahalakshmi, M., & Sundararajan, M. (2013). Traditional SDLC vs scrum methodology–a comparative study. *International Journal of Emerging Technology and Advanced Engineering, 3*(6), 192–196.

Miller, J. K., Friedman, B., Jancke, G., & Gill, B. (2007). Value tensions in design: the value sensitive design, development, and appropriation of a corporation's groupware system. In *Proceedings of the 2007 international ACM conference on supporting group work* (pp. 281–290).

Millett, L. I., Friedman, B., & Felten, E. (2001). Cookies and web browser design: Toward realizing informed consent online. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 46–52).

Mouter, N., de Geest, A., & Doorn, N. (2018). A values-based approach to energy controversies: Value-sensitive design applied to the Groningen gas controversy in the Netherlands. *Energy Policy, 122*, 639–648.

Ropohl, G. (1999). Philosophy of socio-technical systems. *Society for Philosophy and Technology Quarterly Electronic Journal, 4*(3), 186–194.

Sacolick, I. (2022). What is agile methodology? Modern software development explained. *InfoWorld*. https://www.infoworld.com/article/3237508/what-is-agile-methodology-modern-software-development-explained.html

Strijbos, S. (1998). Ethics and the systematic character of modern technology. *Society for Philosophy and Technology Quarterly Electronic Journal, 3*(4), 160–169.

Thomé, B. (Ed.). (1993). *Systems engineering: Principles and practice of computer-based systems engineering*. Wiley.

Umbrello, S. (2018). The moral psychology of value sensitive design: The methodological issues of moral intuitions for responsible innovation. *Journal of Responsible Innovation, 5*(2), 186–200.

Umbrello, S. (2019). Beneficial artificial intelligence coordination by means of a value sensitive design approach. *Big Data and Cognitive Computing, 3*(1), 5.

Umbrello, S. (2021). Conceptualizing policy in value sensitive design: A machine ethics approach. In *Machine law, ethics, and morality in the age of artificial intelligence* (pp. 108–125). IGI Global.

Umbrello, S., & Gambelin, O. (2021). *Value sensitive design: An agile toolkit*. ResearchGate.

Umbrello, S., & Van de Poel, I. (2021). Mapping value sensitive design onto AI for social good principles. *AI and Ethics, 1*(3), 283–296.

Umbrello, S., Capasso, M., Balistreri, M., Pirni, A., & Merenda, F. (2021). Value sensitive design to achieve the UN SDGs with AI: A case of elderly care robots. *Minds and Machines, 31*(3), 395–419.

Umbrello, S. (2020). Meaningful Human Control Over Smart Home Systems. *HUMANA.MENTE Journal of Philosophical Studies, 13*(37), 40–65.

United Nations. (2018). *Sustainable development goals*. UN Department of Economic and Social Affairs Sustainable Development. https://sdgs.un.org/goals

van de Kaa, G., Rezaei, J., Taebi, B., van de Poel, I., & Kizhakenath, A. (2020). How to weigh values in value sensitive design: A best worst method approach for the case of smart metering. *Science and Engineering Ethics, 26*(1), 475–494.

van de Poel, I. (2013). Translating values into design requirements. In *Philosophy and engineering: Reflections on practice, principles and process* (pp. 253–266). Springer.

van de Poel, I. R. (2014). Conflicting values in design for values. In *Handbook of ethics, values, and technological design: Sources, theory, values and application domains* (pp. 1–23). Springer.

van de Poel, I. (2017). Dealing with moral dilemmas through design. In *Designing in ethics* (pp. 57–77). Cambridge University Press.

van den Hoven, J., Lokhorst, G.-J., & Van de Poel, I. (2012). Engineering and the problem of moral overload. *Science and Engineering Ethics, 18*(1), 143–155.

van Wynsberghe, A. (2013). Designing robots for care: Care centered value-sensitive design. *Science and Engineering Ethics, 19*(2), 407–433.

Veugelers, R., Cincera, M., Frietsch, R., Rammer, C., Schubert, T., Pelle, A., Renda, A., Montalvo, C., & Leijten, J. (2015). The impact of horizon 2020 on innovation in Europe. *Intereconomics, 50*(1), 4–30.

Whitchurch, G. G., & Constantine, L. L. (2009). Systems theory. In *Sourcebook of family theories and methods* (pp. 325–355). Springer.

Winkler, T., & Spiekermann, S. (2021). Twenty years of value sensitive design: A review of methodological practices in VSD projects. *Ethics and Information Technology, 23*(1), 17–21.

Winner, L. (1980). Do artifacts have politics? *Daedalus, 109*(1), 121–136.

Yoo, D. (2021). Stakeholder tokens: A constructive method for value sensitive design stakeholder analysis. *Ethics and Information Technology, 23*(1), 63–67.