BRINGING ETHICAL VALUES INTO AGILE SOFTWARE ENGINEERING

Olaf Zimmermann, Mirko Stocker, Stefan Kapferer
OST Eastern Switzerland University of Applied Sciences (Switzerland)
{firstname.lastname}@ost.ch

ABSTRACT

In principle, it is well understood how software engineers should behave; codes for ethics and professional conduct collect principles providing related guidance (ACM (2018)). However, these codes do not translate seamlessly into tangible advice for software engineering routines on development projects, for instance those applying agile principles. Value statements and principles in documents can easily be ignored, e.g., by busy engineers. Conflicts arise in practice, for instance, between public and commercial interests and between stakeholder groups. To improve the situation, we investigate three research questions: 1. How can ethical awareness be stimulated and integrated into agile software practices? 2. How can ethical concerns be actively identified and weighted against other requirements? 3. How can methods and tools trigger, assist, and validate ethical behaviour on agile projects? To answer these three questions, we propose Ethical Software Engineering (ESE) as an active, integrated approach to value-based software engineering advancing the existing passive, retrieval-based state of the art. In this paper, we report on first method engineering results and outline our plans for future work on ESE.

KEYWORDS: Agile Software Development, Design Decisions, IEEE 7000, Moral Values, Normative Ethics, Requirements Engineering, User Stories, Value-Based Systems Design

1 INTRODUCTION AND BACKGROUND INFORMATION

An ethical value is a "value in the context of human culture that supports a judgment on what is right or wrong" (IEEE 2021). Ethics should concern all project stakeholders, in particular software engineers as initial creators of possibly harmful software. Acting ethically is not a binary, absolute virtue but a multi-faceted, relative, and highly context-depending effort (Ozkaya (2019), Spiekermann (2019)). Stakeholder concerns differ across business sectors, application genres, and organizational units; tradeoffs between entrepreneurial goals and human values must be found (Whittle (2019)).

Professional societies describe the behavior they expect from their members in terms of ethics and professionalism in codes of conduct. The Association for Computing Machinery (ACM), the IEEE Computer Society, and other organizations have issued such codes. To give an example, general principle 1.6 in the ACM code is "respect privacy" and professional responsibility principle 2.9 is "design and implement systems that are robustly and usably secure" (ACM (2018)). It is worth noting that not only engineers but also the software they develop should behave ethically.

Agile practices became popular after the above-mentioned codes of conduct were published; for example, predecessors of the current ACM code (ACM (2018)) were released in 1966, 1972, and 1992. Agile practices bring novel challenges; some of them emphasize early and continuous delivery, which may contradict or hinder careful ethical thinking, planning, and execution (Spiekermann (2019), Gibson et al. (2022)). Certain agile practices, however, might be well-suited to identify potential issues; for instance, having business representatives and end users work with the development team on a daily base reduces the risk of misunderstanding and failing to meet their expectations. Ethics are not mentioned explicitly but touched upon in the "Manifesto for Agile Software Development" from 2001, which is based on four value statements itself; technical excellence is established as one of twelve principles in the Manifesto. Working software is the primary measure of progress and success, not its ethical properties.[1]

[1] https://agilemanifesto.org/

2 CURRENT STATE OF RESEARCH AND PRACTICE

2.1 State of the Art in Academia

Many researchers highlight the relevance of ethics in software engineering and the threats posed by recent developments in related fields such as artificial intelligence, big data, and Web development. An IEEE Software editorial positioned ethics as a "software design concern" (Ozkaya (2019)). Hole (2019) called for five principles: "ensure openness, avoid lock-in, pay for user information, provide multiple solutions with similar services, and combine minds and machines." Safety and privacy as well as robustness have received more attention than other values so far (IEC (2000), GDPR (2016)). Application domains differ in their adoption and maturity w.r.t. these values and qualities; e.g., medical device controllers can be expected to do better than situational apps for leisure and entertainment.

Few research projects address the problem domain from a method engineering or design science point of view; managing ethical values and risks on agile projects has received little attention so far. Issues have been reported (Gregory and Taylor (2013), Dindler (2022)) and the connection between technical debt and ethics has been identified (Gibson et al. (2022)). Economics researchers define digital value systems (Spiekermann (2019), Diethelm and Sennhauser (2019)).

2.2 State of the Practice in Industry

In many countries, ethics education receives increasing attention in computer science and software technology curricula (Dodig-Crnkovic and Feldt (2009)). The Software Engineering Body of Knowledge (SWEBOK)[2] references the ACM and IEEE codes (ACM (2018, IEEE Computer Society (2013)) in its Chapters 1 and 10. While it clearly emphasizes the importance of ethically responsible behavior, it does not provide related adoption and application advice in the form of practices.

The gray literature raises awareness. An online article points at general ethics decision making guidelines,[3] an industry thought leader points out that software engineers are "not just code monkeys" (according to M. Fowler in his OOP 2014 keynote)[4], and practitioners launch initiatives to collect and share more concrete guidance, for instance Code:Ethics in the United Kingdom.[5] An example of valid but rather generic and abstract advice to practitioners is to focus on service delivery quality (of people) and "act with integrity" and value "respect, trust, responsibility" (Hall (2009)).

The IEEE standard 7000-2021, "Standard Model Process for Addressing Ethical Concerns during System Design", defines five analysis and design processes to support this advice; it also suggests (but does not norm) an initial value catalog. IEEE 7000-2021, which we refer to as IEEE Std. 7000 from now on, "aims to support organizations in creating ethical value through system design. Creating ethical value is a vision for organizations that recognizes their central role in society as shapers of well-being and carriers of societal progress that benefits humanity. Implementing IEEE Std. 7000 can help them to strengthen their value proposition and avoid value harms. It is applicable to all kinds of products and services" (IEEE (2021)). Key concepts in IEEE Std. 7000 are (in alphabetical order):

- Concept of Operations (ConOps), from ISO/IEC/IEEE15288: 2015: "Verbal and/or graphic statement, in broad outline, of an organization's assumptions or intent in regard to an operation or series of operations."
- Ethical Value Requirement (EVR): "Organizational or technical requirement catering to values that stakeholders and conceptual value analysis identified as relevant for the SOI." (SOI: System of Interest)
- Value: "A conception that influences the selection from available modes, means and ends of action."
- Value-Based System Requirement (VBSR): "System requirement that is traceable from ethical value requirements, value clusters, and core values."
- Value Cluster: "Group containing one core value and several values instrumental to, or related to, the core value." IEEE Std. 7000 defines: "An information store created for transparency and

[2] https://www.computer.org/education/bodies-of-knowledge/software-engineering

[3] https://www.infoq.com/articles/ethical-software-engineer

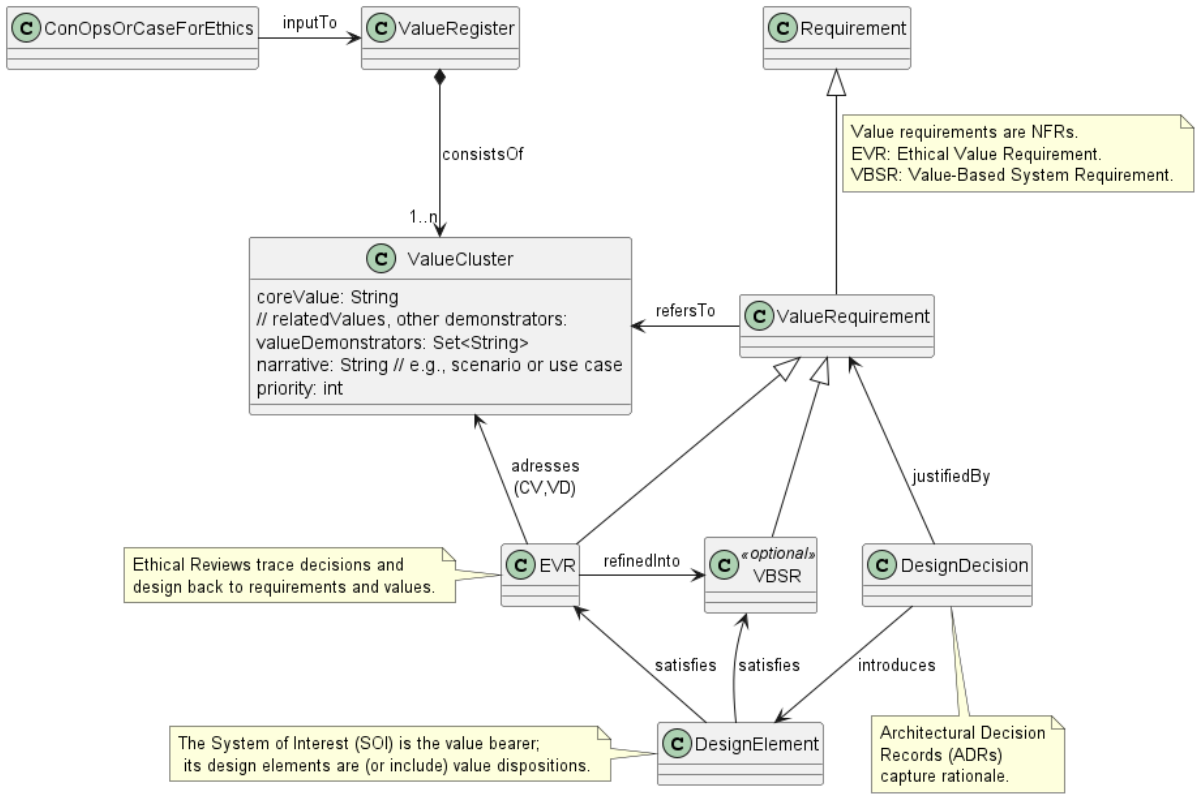
[4] https://martinfowler.com/tags/technical%20leadership.html

[5] https://www.codeethics.org/

traceability reasons, which contains data and decisions gained in ethical values elicitation and prioritization and traceability into ethical value requirements." (IEEE (2021))

IEEE Std. 7000 advises how to go from context to value to requirements to design; Figure 1 illustrates its core concepts and their relations (note that ADRs are not mentioned explicitly in it).

Figure 1. IEEE Std. 7000 artifacts addressing ethical concerns (UML class diagram)



Source: self-elaboration based on IEEE Std. 7000 (2021)

2.3 Open problems

The following deficits in the current state of the art must be overcome to advance from opportunistic, passive knowledge sharing to active guidance that respects context and conflicts:

1. Existing knowledge and advice are comprehensive but not always actionable in a given project context and not integrated into methods and tools applied in practice. It must be looked up (*pulled*) and consulted on demand (Berenbach and Broy (2009)).
2. Methods for dealing with the "dual use dilemma" (i.e., most projects and most software can do good, but also be harmful) are missing; it is hard to deal with conflicts such as "the right to privacy vs. the need to protect vulnerable user groups" (Rashid, Weckert, and Lucas (2009)).
3. Tools to promote and ensure ethically responsible, trustworthy behavior of software engineers and the software they produce are missing. It is not clear yet whether ethical behavior can be expected to be understood and demonstrated by software at all; hence, such tools might be desirable but neither theoretically nor practically feasible – and ethically acceptable (Spiekermann (2019)).

We propose to overcome these deficits by integrating ethical values into contemporary agile development routines. We do so in the form of novel and extended/enhanced agile practices.

3 RESULTS OVERVIEW

As explained in Section 2, existing work has focused on creating awareness. It followed a passive, document-oriented approach requiring project teams to pull knowledge and advice from the

literature; methods and tools to stimulate ethically responsible behavior are missing. In contrast, we propose to overcome these deficits by integrating ethical values into contemporary agile development routines. We do so in the form of an extended set of agile practices. We contribute an active push approach that makes the elicitation and prioritization of ethical values mandatory, effectively bringing value-based design into development workflows.

Our contributions fall in three categories: knowledge, methods, and tools. Our method, called *Ethical Software Engineering (ESE)*, balances both human values such as fairness and diversity with agile values such as customer collaboration and responding to change. We inject value-based ethical engineering in the agile software development mainstream by way of a novel approach to method engineering and tool design. ESE is released publicly via a git repository that renders Markdown pages to HTML; it is available at: https://github.com/ethical-se/ese-practices.

Knowledge. We compiled a set of essential questions to ask on agile development projects, derived and distilled from existing software engineering codes of ethics and professionalism as well as related sources on value-based software engineering and agile coaching (Agile Alliance (2022), IEEE (2021)). This compilation is disseminated in the form of two novel agile practices called *Story Valuation* and *Ethical Review*; the existing practice of user storytelling is amended with value information complementing the business benefits in the "so that [benefit]" part of the story template (that also has "As a [role]" and "I want to [capability]" parts).

Methods. In line with (Spiekermann (2019)), we propose a decision support and tradeoff method for value-based resolution of conflicts between ethical and other design concerns. Our method adopts the process defined in IEEE 7000 (IEEE (2021)) and complements them with agile practices, working with its ConOps, value register, Ethical Value Requirement (EVR) and Value-Based System Requirements (VBSRs) artifacts. Existing agile practices for requirement prioritization, project planning, and reflection (e.g., definition of ready, definition of done, retrospective) are updated; we also integrate the existing agile concepts of product backlog, sprint planning, and acceptance testing. Each ethically desired behavior on projects is derived from a) the existing body of knowledge (methods, guidelines, codes of conduct) and b) current project context and requirements. Values and resulting requirements are articulated in several different formats inspired by the agile user story template, including value narratives, value weightings and decision-oriented "context-criteria-options" triples. Such template-based value statements help to raise awareness for ethical concerns and make it harder to behave unethically because they force certain questions. To stimulate ethical thinking even further, we also envision concrete, actionable conflict resolution advice that leaves professional responsibility with the engineer (where it belongs) but moderates the decision-making process.

Tools. We experimented with a demonstrator of a continuous ethics linter as a first tool that actively places ethical awareness in the development mainstream. This tool looks for ethical smells (i.e., suspects that a value might be harmed), inspecting source code and supplemental artifacts in project repositories. A first, basic, text-based prototype of such a linter tool showed technical feasibility but also unveiled ethical concerns; further research is required to set an adequate direction here.

4 SELECTED METHOD AND KNOWLEDGE RESULTS

In our method design, we use the term *practices* as a generalization of artifacts and activities here; artifacts serve as input to and output from activities. Our roles originate from IEEE Std. 7000 (IEEE (2021)) and are presented in a rather short/dense/terse form in Release 1.0 of ESE.[6]

The remainder of this section is structured in the following way: Section 4.1 gives an overview of ESE. Section 4.2 maps its agile practices to the processes in IEEE Std. 7000 (and back); Section 4.3 covers Story Valuation, Section 4.4 Ethical Review. Section 4.5 lists additional repository content.

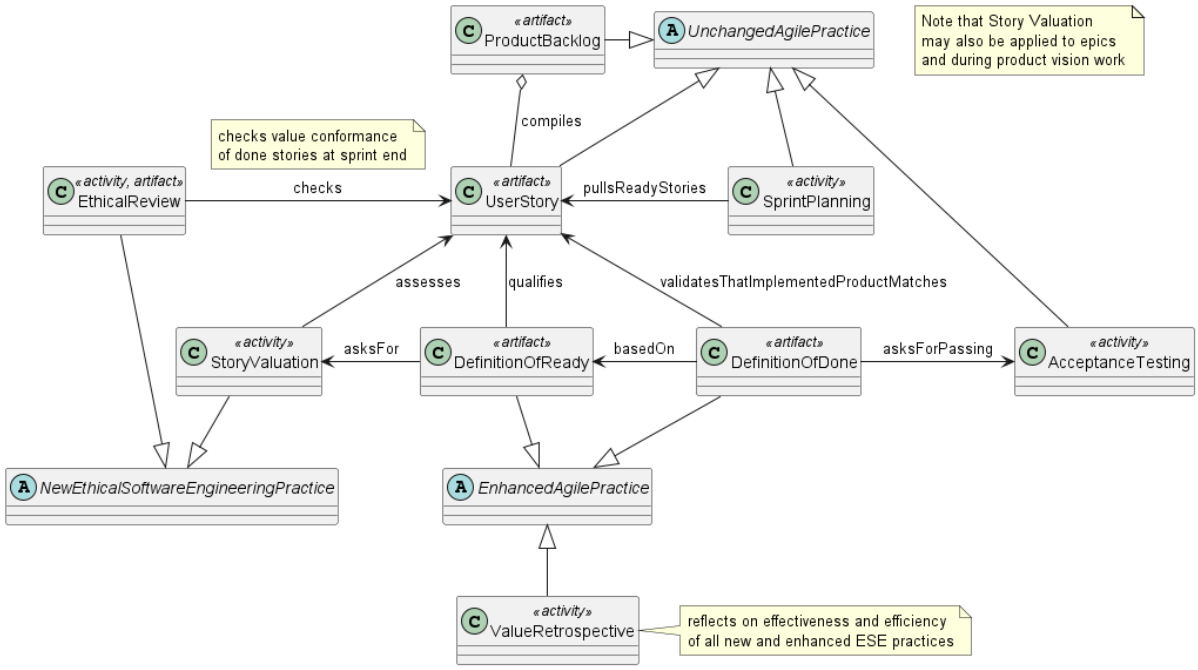
4.1 Practices Overview

ESE includes a total of nine agile activities and artifacts: (new) *Story Valuation* activity, the entry point for ESE usage and (new) *Ethical Review* activity and artifact; (extended artifact) *Definition of Done* and (extended artifact) *Definition of Ready*; (extended artifact) *Value Retrospective*; (unchanged

[6] https://github.com/ethical-se/ese-practices/tree/v101/roles

activity) Acceptance Testing, (unchanged activity) Product Backlog, (unchanged activity) Sprint Planning, (unchanged artifact) User Story. The Agile Alliance Glossary provides compact and concise reference information for the mentioned existing practices (Agile Alliance). Figure 2 gives an overview of our practices and where they come from.

Figure 2. Overview of new, enhanced, unchanged ESE practices (UML class diagram)



Source: self-elaboration (2023)

4.2 Mappings between Agile Practices and IEEE Std. 7000

Table 1 shows how agile teams can integrate concepts from IEEE Std. 7000 into their practices, including those featured in the previous Section 4.1. The mapping is the result of a literature research and reflection of own practices by one author of this paper; the other two authors then reviewed and commented on their own experiences, which led to two incremental refinements of the mapping.

Table 1 Mapping from Agile Practices to IEEE Std. 7000 concepts

Agile Practice	Related Concept in IEEE Std. 7000	Comments
User Story (as Product Backlog Item)	Functional system requirement	see ESE FAQs[7]
Sprint Planning	no direct mapping	parts of all processes executed in each iteration
Definition of Ready	EVRs elicited (Clause 9)	V in INVEST widened
Definition of Done	Design artifacts produced and reviewed (Clause 10)	from Scrum, many templates/criteria
Sprint Review	Verification and validation activities in all processes	Novel practice in ESE: Ethical Review
Retrospective	no direct mapping, contributes to Transparency Management Process	many variations

Source: own presentment (ESE repository, Background Information page)

Table 2 maps in the opposite way and suggests agile practices to adopters of IEEE Std. 7000. It came to be in the same way as Table 1.

[7] https://github.com/ethical-se/ese-practices/blob/v101/ESE-FAQ.md

Table 2 Mapping from IEEE Std. 7000 processes to Agile practices

IEEE Std. 7000	Related Agile Practice	Comments
Clause 7: Concept of Operations (ConOps)1 and Context Exploration Process 2	no direct pendant; epics as input	Concepts in HCI/UX community such as personas, Context Diagram in DPR
Clause 8: Ethical Values Elicitation and Prioritization Process	User Stories and related practices (e.g., backlog refinement/grooming)	Novel practice in ESE: Story Valuation
Clause 9: Ethical Requirements Definition Process	User Stories and related practices	Mapping, splitting; estimation/planning games such as planning poker
Clause 10: Ethical Risk-Based Design Process	n/a (implicit, evolutionary/emerging)	Related literature: "Domain-Driven Design" (E. Evans), "Just Enough Architecture" (G. Fairbanks)
Clause 11: Transparency Management Process	no direct mapping	Ethical Review Meeting/Report, in ESE design decision logs from DPR

Source: own presentment (adopted from ESE repository)

4.3 Story Valuation: Notations and Techniques

We introduce this practice in the form of a user story, the agile practice for requirements engineering and iteration scoping (rationale: user stories are also suited in method engineering):

As a responsible software engineer, I want to craft working software that delivers value to users and other stakeholders while not harming any individuals, society and/or the planet. I also want to identify goal conflicts so that adequate trade-offs can be found.

We now provide usage instructions, discuss notations briefly and list techniques to perform the activity. The online ESE method repository also specifies input and output, covers notation in depth, describes the techniques in detail and provides examples and pointers to the literature.

Instructions. The practice description in Release 1.0 of ESE advises to: "Add individual, societal, and environmental values to the business and user values in the "so that" part of epics, user stories or other types of product backlog items. Do so from the perspective of different stakeholder groups; compare and prioritize their value clusters and derive value requirements from them. Start this activity in Product Vision (or Sprint 0 or Minimum Viable Product development); return to it and resume valuation in each sprint/iteration as/if needed. Apply one of the techniques in ESE to do so and record your results in one of the proposed notations; alternatively, work with your own (or other recognized) techniques and notations; ESE is suggestive and not normative in this regard."

Notations. ESE does not mandate a certain format for the IEEE Std. 7000 Value Register; overview figures and comparison tables can be well suited. That said, ESE still suggests three novel formats: Value Epic, Value Weighting and Value Narrative; see ESE repository for notation templates as well as examples.[8] EVRs may take this form:

```
As a [role]
I want to [action/feature]
so that [benefit] is achieved
   and that [values a, b, c] are promoted,
   accepting that [values x, y, z] are reduced.
```

The first three clauses ("As-a", "I want to", "so that") are well-established convention/template in the Agile community specifying role, feature, and benefit of the story; the new clauses "and that" and "accepting that" add positive and negative values and other ethical consequences of an implementation of the feature described by the story.

Techniques. The valuation techniques proposed in ESE are a) *Goals and Vision First*: Question-Based Elicitation, b) *User Requirements First*: Story-Driven Value Jam and c) *Individual Values First*: Catalog-Guided Value Mapping.

[7] https://github.com/ethical-se/ese-practices/blob/v101/practices/ESE-StoryValuation.md#notations

When applying technique a), Question-Based Value Elicitation (Goals and Vision First), you may want to ask the following questions when developing and hardening the product vision to identify ethical values and elicit EVRs/VBSRs:

- "How does the system (product, service) under construction make the world a better place?
- Even if there were no explicit, external stakeholder goals and business drivers, why is it still a good idea to develop the system? Which positive ethical values does it promote?
- Which positive ethical values are degraded by any realizations of the envisioned functionality? Which negative values are promoted? What are the related elements of risk, those with high probability and huge impact in particular?
- How do positive and negative values, as well as benefits and harms, relate to each other? What is their relative and/or absolute weight?
- Which resources will the system consume, and can this consumption be justified by the business/user and ethical values that it delivers?"

The ESE technique b), User Requirements First (Story-Driven Value Jam), asks the following value-related questions:

- "What are the individual and collective values of the persona/role in the "As a" part of the story (aka stakeholder groups)?
- Which responsibilities do their user interfaces have that may promote or degrade ethical values (both positive and negative ones)?
- Which ethical values are affected when executing the program code realizes the "I want to" part of the story?
- What is the desired and, presumably, actual impact (good and bad) of the "so that" part of the story on individuals, society, and planet?"

It also asks the following more technical questions:

- "Which values are affected positively or negatively when end user input is received and validated, and when computation and query output is displayed?
- How does the data processing (application/business/domain logic) do w.r.t values?
- How do data access components and data storage (persistence mechanisms) behave w.r.t. values?
- What is the ethical risk of APIs and other transport channels, as well as shared services such as loggers?"

Technique c), Individual Values First (Catalog-Guided Value Mapping), works with Appendix G of IEEE Std. 7000. It suggests the following three steps:

1. "Pick 2-3 core values from the table. Use the information in the columns Related value and Opposing value in Table G.1 to make them more complete, concrete, and tangible.
2. Explain the relevance of each core value in the given project/product context by way of example and/or refinement. You might want to tell a story and/or point out the positive and negative consequences of this value in the form of a narrative or demonstrator (see column in table above). This information may come from the Value Register of the project/product development effort when IEEE Std. 7000 is followed (if no Value Register exists yet, it is a good time to create one now).
3. Prioritize the value(s), either absolutely or relatively. For instance, you may want to use a writing style akin to that used in the Agile Manifesto ('We value mm over nn')."

4.4 Ethical Review (Meeting and Report)

We follow a similar presentation structure here as in the previous section.

An Ethical Review Report captures the results from an ethical values-enhanced (or -centric) sprint/project Ethical Review Meeting. It may include mere meeting notes and/or an assessment with follow up actions (aka recommendations and findings).

Instructions. The practice description in Release 1.0 of ESE advises to: "Inspect the user story implementations with regards to the prioritized Value Cluster demonstrators/narratives and Value Requirements from Story Valuation in a meeting. Capture the meeting outcome in a report.

Record answers to the following questions during the review meeting:

- Have the Ethical Value Requirements (EVRs) been included in the acceptance testing and did these tests pass?
- Has any feature been introduced that stands in conflict with the EVRs as well as the team's values and personal beliefs of each team member? If yes, have tradeoffs and mitigation tactics been discussed?
- Would all team members use the new features themselves, and let their closest relatives use them (assuming that these individuals are in the target audience of the system under construction)?
- Has the behavior of the team and its members been in line with the eligible Codes of Conduct (CoCs), for instance the ACM Code of Ethics and Professional Conduct? Does it respect the values of the involved organizations (sponsor, clients, business partners)?
- Do any of the IEEE Std. and ESE artifacts (i.e., ConOps, Value Register, EVRs) require further updates, caused by the review results?

There are no easy answers to many of these questions typically; this is inherent, discussing them (as well as follow-on questions) is as important as the outcome of this activity to create awareness, for instance when planning the next sprint/iteration."

Notations. The review results can be recorded in free form in a new text document or added to the reviewed artifacts.

4.5 Other Repository Content

The ESE repository also contains Markdown templates for important artifacts specified in IEEE Std. 7000 (e.g., Case for Ethics), application hints per practice, and pointers to the literature including standards and books (e.g., on material ethics).

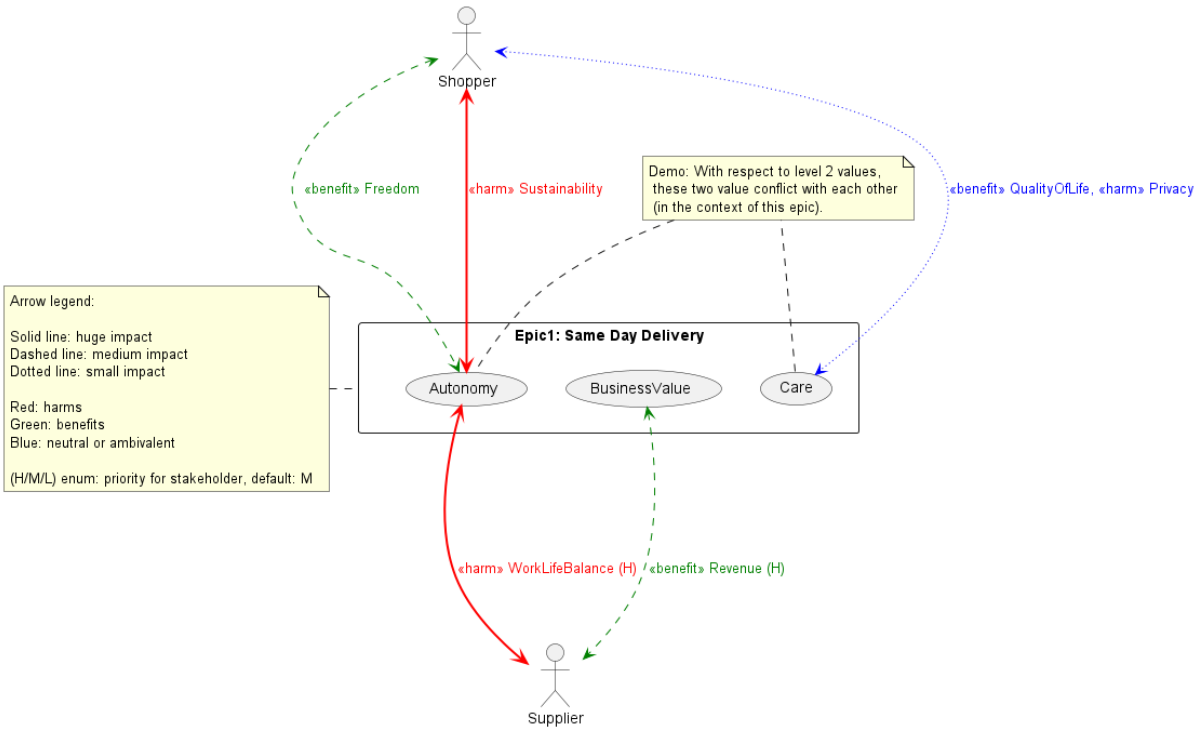
5 VALIDATION

We validated our method engineering results in action research and self-experiments so far. Two of the authors of the paper that did not author the first version of the method content, reviewed an intermediate version of it and applied it to two of their own real-world projects. This early feedback led to a substantial revision of the entire method content and the introduction of two additional Story Valuation techniques. The three authors then designed a fictitious but realistic "same day delivery" product vision for an online shop and applied the revised second version of the method content in a joint half-day workshop. General feedback on readability, applicability, and usefulness was positive; scalability, time management, dealing with conflicting stakeholder interests, and visualization challenges were identified and partially addressed (see Section 7 for future work about addressing these challenges). Reviewers commented that more examples would be welcome, on all levels of analysis and design refinement (i.e., values, value requirements, and architectural decisions addressing these requirements) to make the method even more accessible and useful.

An external reviewer who contributed to the standard appreciated the diagrams and the method organization but requested some terminology clarification so that utilitarian values (harms and benefits) are not confused with positive and negative values from any ethics school (and their consequences). The reviewer suggested to not only focus on stories and epics but also on business-level ideation activities such as product envisioning on Scrum; we also discussed whether the Value Lead role from IEEE Std. 7000 should be taken by a single person or be a shared responsibility of the development team (both positions have pros and cons attached).

Figure 3 shows an example from the validation workshop. Autonomy and care are core values from Annex G of IEEE Std. 7000 (IEEE (2021)); freedom and sustainability, quality of life and privacy are listed as related values in "Table G.1—Typical ethical values for systems design" of the standard.

Figure 3. Values and conflicts in the Same Day Delivery example (UML use case diagram)



Source: self-elaboration (2023)

Online experiment instructions and review questions are available online. They can be found at https://github.com/ethical-se/ese-practices/tree/main/experimentation. We invite readers to participate.

6 DISCUSSION OF DESIGN DECISIONS

We reviewed the OMG SPEM metamodel for methods and methods engineering (OMG (2008)) as well as other inputs before compiling the practice templates for our Design Practice Repository (Zimmermann and Stocker (2021)), now adopted and further refined for ESE.

Revisiting the research questions and contribution types from Sections 2 and 3, we decided to focus on method engineering primarily and less on tool development because of a) risk-benefit issues and b) importance and complexity of method engineering (early feedback, see below), and c) existence of IEEE Std. 7000 and supporting literature.

We decided not to feature the entire standard due to its size and general-purpose nature (i.e., its usage is not limited to software systems). Furthermore, we made several standards concepts optional (e.g., VBSRs) to achieve the goal of being attractive to our target audience, agile software development teams. The Frequently Asked Questions (FAQ) page in the ESE repository provides further information.

7 CONCLUSION AND OUTLOOK

Ethical Software Engineering (ESE) integrates ethical values and IEEE Std. 7000 with agile software development practices. ESE introduces two new practices, extends three existing ones, and reuses four existing ones without changing them. In this paper, we motivated the need for such methods in the state of the art and practice, provided a method overview, featured the two new/novel practices in some more detail, and then discusses early validation feedback as well as critical success factors for a broader adoption. The full ESE method is available at https://github.com/ethical-se/ese-practices, a public open-source repository under the Creative Commons Attribution 4.0 International License.

In our future work, we consider including pre-defined value catalogs and assessments of their relevance w.r.t. project phases and architectural layers (presentation, business logic, data access and storage) into our approach. We also consider developing additional templates and notations, emphasizing usability, scalability, and conflict management in our method engineering. Other

directions for future work include tool support for the method and its content, for instance in the form of a questions-answers moderation or conflict visualization and resolution support. Another area that we consider is starting even earlier – evaluating and assessing product visions with respect to their ethical ramifications. Such early start would support "go"-"no go" decisions for software development before it even starts.

ACKNOWLEDGEMENTS

This research was supported by the Hasler-Foundation. Bärbel Bohr answered our many questions about IEEE Std. 7000 and reviewed parts of the ESE repository.

REFERENCES

ACM (2018). "ACM Code of Ethics and Professional Conduct 2018."

ACM, and IEEE Computer Society (2013). Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. New York, NY, USA: ACM.

Agile Alliance (2022). "Code of Ethical Conduct for Agile Coaching, Version 2.0." https://www.agilealliance.org/agile-coaching-code-of-ethical-conduct/ (last access: Jan. 11, 2024)

Agile Alliance. "Agile Glossary" https://www.agilealliance.org/agile101/agile-glossary/ (last access: Jan. 11, 2024)

Berenbach, B., and M. Broy (2009). "Professional and Ethical Dilemmas in Software Engineering." Computer 42 (01): 74–80.

Diethelm, C., and P. Sennhauser. (2019). "Digitale Ethik, HWZ Whitepaper."

Dindler, C., P. G. Krogh, K. Tikær, and P. Nørregård (2022). "Engagements and Articulations of Ethics in Design Practice*." International Journal of Design* 16 (2): 47–54.

Dodig-Crnkovic, G., and R. Feldt. (2009). "Professional and Ethical Issues of Software Engineering Curriculum Applied in Swedish Academic Context." In HAoSE 2009 First Workshop on Human Aspects of Software Engineering. Online Proceedings.

GDPR. (2016). "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC." *OJ L 119/1*.

Gibson, J. P., M. Narouwa, D. Gordon, D. O'Sullivan, J. Turner, and M. Collins (2022). "Technical Debt is an Ethical Issue." Proc. of ETHICOMP 2022.

Gregory, P., and K. Taylor. (2013). "Social and Communication Challenges for Agile Software Teams.", Proceedings of ETHICOMP 2013, 186–91.

Hall, D. (2009). "The Ethical Software Engineer." *IEEE Software* 26 (4): 9–10.

Hole, K. J. (2019). "Dominating Software Systems: How to Overcome Online Information Asymmetry." *IEEE Software* 36 (4): 81–87.

IEEE. (2021). "IEEE 7000 Standard Model Process for Addressing Ethical Concerns During System Design."

IEC. (2000). "IEC 61508: Functional safety of electrical/electronic/programmable electronic safety related systems."

OMG. (2008). Software & Systems Process Engineering Metamodel Version 2.0

Ozkaya, I. (2019). "Ethics Is a Software Design Concern." *IEEE Software* 36 (3): 4–8. https://doi.org/10.1109/MS.2019.2902592

Rashid, A., J. Weckert, and R. Lucas. 2009. "Software Engineering Ethics in a Digital World." Computer 42 (6): 34–41.

Spiekermann, S. (2019). *Digitale Ethik: Ein Wertsystem für das 21. Jahrhundert*. Droemer Verlag.

Spiekermann, S. (2023). *Value-Based Engineering: A Guide to Building Ethical Technology for Humanity*. De Gruyter.

Whittle, J. (2019). "Is Your Software Valueless?" *IEEE Software* 36 (3): 112–15.

Zimmermann, O., and M. Stocker (2021). *Design Practice Repository*, LeanPub.

The extended abstract corresponding to this paper was peer-reviewed and accepted for ETHICOMP 2024. This is the authors copy of the full paper that appears in the ETHICOMP conference book "The leading role of smart ethics in the digital world".