BRINGING ETHICAL VALUES INTO AGILE SOFTWARE ENGINEERING

Olaf Zimmermann, Mirko Stocker, Stefan Kapferer

OST Eastern Switzerland University of Applied Sciences (Switzerland)

itolz@bluewin.ch, mirko.stocker@ost.ch, stefan.kapferer@ost.ch

EXTENDED ABSTRACT

Motivation. 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 behavior on agile projects?

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 results and outline our plans for future work.

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; e.g., 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.[5]

Current state of research. 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, for instance in IEC 61508 and the General Data Protection Regulation (GDPR).[6] Application domains differ in their adoption and maturity w.r.t. these values and qualities; e.g., software controlling medical devices can be expected to do better than situational apps for leisure and entertainment.

The Software Engineering Body of Knowledge (SWEBOK)[7] picks up the ACM and IEEE codes. In many countries, ethics education receives increasing attention in computer science and software technology curricula (ACM and IEEE Computer Society (2013), Dodig-Crnkovic and Feldt (2009)). The gray literature also raises awareness. 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 recently published standard IEEE 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 2021).

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)).

In summary, 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. 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.

Results. Our *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

[5] https://www.agilealliance.org/agile101/the-agile-manifesto/ (2001)

[6] https://en.wikipedia.org/wiki/IEC_61508 and https://eugdpr.org/

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

a novel approach to method engineering and tool design. Our contributions fall in three categories:

1. *Knowledge.* A compilation 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 *Ethics Review*; the existing practice of user storytelling is amended with value information complementing the business benefits in the "so that" part of the story template (that also has "As a [role]" and "I want to [capability]" parts).
2. *Methods.* We envision a decision support and tradeoff method for value-based resolution of conflicts between ethical and other design concerns. This method adopts and complements the process defined in IEEE 7000 (IEEE (2021)), 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 is distilled a) from the existing body of knowledge and b) current project context and requirements. Values and resulting requirements are articulated in several different formats that are 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. 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.
3. *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 unveiled technical feasibility but also ethical concerns; further research is required to set an adequate direction here.

   We validated our method engineering results in action research so far, with case studies and surveys planned; the project results are available in a public git repository at https://github.com/ethical-se. 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.

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

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/agilecoachingethics

Diethelm, C., and P. Sennhauser. 2019. “Digitale Ethik, HWZ Whitepaper.”

Dindler, Krogh, C. 2022. “Engagements and Articulations of Ethics in Design Practice.” *International Journal of Design* 1 (2): 47–54. https://doi.org/10.57698/v16i2.04

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. http://www.es.mdh.se/publications/1616

Gibson, John Paul, Massamaesso Narouwa, Damian Gordon, Dympna O’Sullivan, Jonathan Turner, and Michael Collins. 2022. “Technical Debt is an Ethical Issue.” Proc. of ETHICOMP 2022.

Gregory, Peggy, and Katie Taylor. 2013. “Social and Communication Challenges for Agile Software Teams.”, *Proceedings of ETHICOMP 2013*, 186–191.

Hall, D. 2009. “The Ethical Software Engineer.” *IEEE Software* 26 (4): 9–10. https://doi.org/10.1109/MS.2009.106

Hole, K. J. 2019. “Dominating Software Systems: How to Overcome Online Information Asymmetry.” *IEEE Software* 36 (4): 81–87. https://doi.org/10.1109/MS.2019.2903075

IEEE. 2021. “IEEE 7000 Standard Model Process for Addressing Ethical Concerns During System Design.”

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

Spiekermann, S. 2019. *Digitale Ethik: Ein Wertsystem Für Das 21. Jahrhundert*. Droemer Verlag.

Whittle, Jon. 2019. “Is Your Software Valueless?” *IEEE Software* 36 (3): 112–15. https://doi.org/10.1109/MS.2019.2897397