2021

Four Opportunities for SE Ethics Education

Alicia M. Grubb
Smith College, amgrubb@smith.edu

Four Opportunities for SE Ethics Education

Alicia M. Grubb
Department of Computer Science
Smith College, Northampton, MA, USA
amgrubb@smith.edu

Abstract— **Many software engineers direct their talents towards software systems which do not fall into traditional definitions of safety critical systems, but are integral to society (e.g., social media, expert advisor systems). While codes of ethics can be a useful starting point for ethical discussions, codes are often limited in scope to professional ethics and may not offer answers to individuals weighing competing ethical priorities. In this paper, we present our vision for improving ethics education in software engineering. To do this, we consider current and past curricular recommendations, as well as recent efforts within the broader computer science community. We layout challenges with vignettes and assessments in teaching, and give recommendations for incorporating updated examples and broadening the scope of ethics education in software engineering.**

I. Introduction

Software engineering (SE) has historically focused on the production of safety and mission critical systems. At a high level of abstraction, the IEEE/ACM SE Code of Ethics asks practitioners to make SE a respected profession by committing to the health, safety, and welfare of the public [1]. This code builds on a history and culture of ethics from other engineering disciplines. Early work focused on ensuring the ethical behavior of software engineers (*engineers*) within projects (e.g., negotiating contracts, conflicts of interest, honest benchmarking). More recent SE curricular recommendations concretize this code by asking engineers to consider ethics in the context of making tradeoff decisions, evaluating both costs and benefits [2]. They also emphasize that ethics concepts need to be presented as recurring themes throughout the SE learning experience, with early courses introducing the material and later courses reinforcing, expanding, and deepening engagement with concepts. In this paper, we focus specifically on educating SE students to become ethical practitioners.

Concrete Example. Standard practice in engineering ethics is to ask students to review a series of vignettes and answer questions about how they would respond in the situation described. For example, Fig. 1 shows a vignette based on the Waymo-Uber lawsuit, which was reproduced from McNamara et al. [3]. Using this example, students are expected to learn a set of ethical skills, such as recognizing, raising, and responding to an ethical concern raised by a colleague or subordinate, by proposing an ethical solution. However, whether this training results in individuals who behave more ethical has yet to be demonstrated in the literature. McNamara et al. found that telling survey participants to indicate how they would resolve ethical vignettes, such as the one in Fig. 1, using the ACM Code of Ethics *did not impact* ethical decision making [3]. Lau et al. confirmed that students perceive different actions as unethical depending on their personal values [4]. Bloodgood et al. found that teaching ethics to individuals who score high on measures of Machiavellianism may lead them to become more ethically skilled, but use these skills to be more unethical (e.g., cheating) [5]. Finally, Aderonmu et al. described how fourth-year computer science (CS) students (or those who have completed internships) are less convinced than first-year students that the field of CS is an ethical one [6].

In an industry where a generation of entrepreneurs were told to "move fast and break things" [7], when is disruption ethical and what are the consequences for being unethical? Our own training and, now, teaching has demonstrated that these are not simple questions and current curricular recommendations and codes may not provide sufficient guidance for our students. Thus, we hope to engage the community in a discussion about opportunities in SE ethics education.

Contributions. In this position paper, we present a vision for improving SE ethics education by exploring four challenge areas. We look at (1) the robustness of ethical examples and (2) how these examples are used in assessment toward curricular learning outcomes. We then (3) consider expanding the learning outcomes to recognize that ethical decisions are based on values and (4) explore students' disconnect with professionalism.

In the remainder of this paper, we review codes and curricula in Sect. II, and present challenges and recommendations for the teaching and learning of SE ethics in Sect. III. Finally, we summarize our call to action in Sect. IV.

II. Codes, Guidelines, and Related Efforts

In this section, we give background on the codes, curricular documents, and notable initiatives in SE Ethics.

A. Professional Codes of Ethics

The IEEE Code of Ethics asks members to ensure that they and their colleagues uphold the highest standards of professional activities (including conduct and behavior) and treat all persons fairly and with respect [8]. The IEEE code is part of the IEEE Policies, and as such, we do not believe it was intended for educational purposes, but rather to describe acceptable behavior for members of a professional society.

The ACM's recent revision to their Code of Ethics fills in some of these gaps [9]. In addition to calling members to the highest standards of professionalism and emphasizing public good as paramount, the ACM code addresses some

A deadline is quickly approaching for a project that you are working on. You realize that you will not be able to meet the deadline if you only work during normal hours. You are not allowed to take your computer out of the office. What do you do?
(a) Download the data on a personal hard drive so you can continue development at home
(b) Unsure
(c) Stay at work longer in order to continue development.

Fig. 1: Sample vignette reproduced from [3].

of the challenges of interpreting these high standards into actions by acknowledging conflicting ethical priorities. The code discusses some key challenges in developing software, such as how systems have emergent properties that may cause ethical issues, how merging data sets may compromise privacy, and how a system might become more integral over time (i.e., requiring adaptation) [9]. With this updated code, the ACM formed the Committee on Professional Ethics to expand resources for interpreting the code, and has added case studies as examples on how to apply the code to complex situations.

The IEEE-CS/ACM Joint Task Force on Software Engineering Ethics published the SE Code of Ethics [1] (first published in [10]). Similar to the previous two, this code contains eight guiding principles at three levels of abstraction: general human ethics, professional specific ethics, and SE specific ethics. The SE Code is complementary to the ACM and IEEE codes because of its more detailed recommendations that are specific to software (e.g., testing and installation); however, without the examples and updated materials in the general ACM code, the SE specific one may be difficult for students to implement.

B. Curricular Recommendations for Teaching Ethics

Next, we look at curricular recommendations for teaching ethics in SE. Since SE straddles both engineering and computer science, we consider recommendations from both the ACM/IEEE and North American accreditation bodies.

The Canadian Engineering Accreditation Board (CEAB) specifies twelve attributes that students must demonstrate competency in by the completion of their engineering program [11]. It is clear that the CEAB views professional ethics as a fundamental component of education because four of the twelve attributes directly refer to elements of ethical behavior. For example, Ethics and equity (i.e., Attribute 9) directly calls upon learners to be able to apply professional ethics but does not define how this should be accomplished. Similarly, the US accreditor ABET lists student outcomes for both their engineering [12] and computing [13] accreditation. The requirements are similar to those of the CEAB describing ethical principles in multiple abilities, such as public safety, social impacts, and professional responsibilities [12]. Yet, the computing requirements are less strong, requiring students to recognize professional responsibilities and make informed ethical judgments [13]. Neither of these bodies give additional details about how to teach ethics.

Both computer scientists and engineers require training in software ethics. The ACM/IEEE Joint Task Force on Computing Curricula makes recommendations on teaching ethics through their SE and CS Curricula [2], [14]. CS students are recommended to have one to two hours of lecture time as part of software engineering skills, as well as information assurance and security skills [14]. These recommendations are much stronger for engineers. The SE curricula document mirrors the accreditation guidelines described above with expected student outcomes, where ethics is prominent in three of the seven outcomes. The core ethics introduction should occur during six hours of lecture that focuses on professionalism, where students are expected to gain *knowledge* (i.e., recall information) and/or *comprehension* (i.e., compare/contrast elements) on topics such as licensing, ethics, contracts, and social/legal issues of concern. Ethics is also considered a cross-cutting theme that should be recurrent throughout the curriculum and raised frequently, citing that “professional practice is concerned with the knowledge, skills, and attitudes that software engineers must possess to practice software engineering professionally, responsibly, and ethically” [2]. In summary, based on these codes and curricula, it is imperative that we, as educators, teach students about SE ethics.

C. Research on Teaching Ethics

In Sect. I, we introduced studies that prompt for improvements to ethics education, such as, recommending more guidance on how to apply the ACM code of ethics [3] to understanding changes in students perceptions about whether CS, as a field is ethical [6]. We are not the first to investigate ways of teaching ethics both in SE and more generally, and here we consider recent efforts.

Based on the curricular guidelines described above ethics was primarily discussed in individual modules in software engineering courses or as part of optional ethics courses (usually at a senior-level). Recent efforts have focused at pushing ethics education throughout the CS curriculum. Since 2018, these efforts have concentrated within the Responsible Computer Science Challenge [15]. At time of writing, the challenge supports seventeen teams actively researching how to broaden the impact of ethics education in CS. The challenge will also create repositories of classroom resources and white papers. For example, Grosz et al. are integrating class sessions on ethical reasoning into courses throughout Harvard’s computer science curriculum [16]. In another example, the Markkula Center for Applied Ethics at Santa Clara University is developing course modules on cryptography, data science, and security, as well as SE ethics, which is of particular relevance to this paper [17]. In the SE module, Vallor and Narayanan provide case studies with open-ended questions for students to discuss both inside and outside of the classroom [17]. We revisit this work in Sect. III.

III. Challenges and Opportunities

In this section, we reflect on four challenges and propose opportunities for change.

Challenge 1: Disconnected Examples

SE ethics education primarily uses examples from the safety critical domain and professionalism. Current examples do not encapsulate the breadth of situations engineers experience [18]. Over the past two decades, engineers are increasingly focusing their efforts on internet applications for commerce and social media (e.g., Facebook, YouTube, Netflix). While these applications have been documented to have broad societal impacts [19], they are not represented in ethical examples. Engineers may feel accomplished when they improve the software architecture to enable users to receive content faster or improve recommendation algorithms, but they may not be aware of the downstream effects of these changes (e.g., increase screen time for users). Furthermore, given the narrow scope of many capstone courses, it is unlikely that college students will naturally encounter these broad ethical issues with sufficient nuance in their course work.

While some examples are clear violations of codes of ethics, other examples are not clear-cut. For example, we can ask if it is *unethical* to not collect requirements from key stakeholders or users of a system (e.g., [20])? Are software designers and architects responsible for their designs for the lifetime of the system, or is that the responsibility of the software maintainers? Who should be responsible for adding higher levels of scrutiny when a previously non-safety critical system evolves into a safety critical system (as in [21])? Who is responsible for the unknown ethical impacts of algorithms [22]? If an application is found to be addictive and detrimental to their users, should the ethical practitioner develop techniques to reduce dependence (as in [23]) or choose to *sunset* the application and avoid involvement in similar applications? Thus, there is a mismatch between the examples provided in SE courses and the situations practitioners face.

Opportunity 1: Create a Robust Collection of Examples

Our first recommendation for improving SE ethics education is the creation and expansion of realistic case studies with full descriptions from a variety of domains. Students should be exposed to the messiness of actual ethical situations faced by real practitioners. When we describe case studies here, we are not referring to research case studies as in the work in [24], but instead similar to those in legal studies. A case study is one or more paragraphs giving contextual information in addition to the actions of the actors involved. In the next challenge, we will explain why we recommend creating case studies rather than vignettes.

The ACM has already begun this effort by creating case studies coupled with the code, as well as crowd sourcing videos and podcasts to promote the importance of ethics [9]. The ACM examples contain an expert analysis of the ethical scenario. They can be used in teaching as exemplars to help students initially encountering ethics, as well as demonstrating how to write about ethical violations; however, they are insufficient because they present a correct answer hindering discussion. The SE ethics module by Vallor and Narayanan (introduced in Sect. II) provides examples of appropriate case studies for teaching ethics in SE.

There are multiple ways for our community to create these collections. Case studies may be created by instructors, practitioners, or students. For example, instructors could create an assignment in public writing, for upper-level undergraduate or graduate students as part of an ethics specific course, where students generate a case study of a SE ethical dilemma by interviewing alumni. In a broader engineering context, Rottmann et al. conducted interviews with alumni and students to elicit first hand accounts of ethical scenarios experienced by participants and how they handled them [18]. The same may be accomplished with alumni networks from SE programs and young professionals to generate SE specific case studies. Additionally, new case studies can be generated from recent events, such as notable whistleblowers. We have begun to develop new case studies for our own courses and hope to partner with others to develop and share more.

Challenge 2: Vignettes Limit Exploration

Educators primarily use lectures and reading to introduce ethical concepts and vignettes, such as the one introduced in Sect. I, to engage students with ethical scenarios. The advantage of vignettes with multiple choice answers is that they are easy for a student to digest and quick to assess. Vignettes only describe the viewpoint of a single individual and ask students to place themselves in the situation of that individual deciding on a narrowly defined set of potential actions (i.e., binary thinking). In these scenarios, students usually imagine themselves as an intern with a benevolent boss (unless otherwise indicated by the scenario), yet real situations are rarely this simplistic.

We argue that in reality, these vignettes fall short of enabling students to explore situations with conflicting imperatives and do not satisfy the learning objectives set forth in the curricula. For example, the vignette in Sect. I does not allow students to consider the many other options in this situation. Perhaps it is not the fault of the individual that the task is not completable—it may be the result of poor *sprint planning*. The individual could work with their supervisor, mentors or other team members to triage what must be done by the deadline vs. what can wait. They could also request allocation of more resources to meet the deadline or request the deadline be moved. Further, this does not take into account power dynamics. If they are a senior developer seen as reliable and experienced, a statement that the deadline cannot be met might be acknowledged; conversely, newly employed junior developers may risk harming their reputation in the company by raising the same concerns.

Opportunity 2: Learning in Workshops with Case Studies

Creating robust case studies (as described in Opportunity 1) is a necessary but insufficient step to address Challenge 2. We argue for replacing lectures with exploratory workshops to increase student engagement, as well as embedding ethics throughout SE, as recommended in the ACM curriculum [2].

Being able to identify alternatives is an ethical reasoning skill that can be taught [17]. We envision creating workshops where, after learning basic skills, students can explore ethical situations and engage in the process of reasoning using the case studies introduced in Opportunity 1. Given a case study, students can engage in groups to explore the problem and the possible causes and solutions for the core issue (consistent with the recommendations in [25]). As with some of the projects in the Responsible CS Challenge [15], educators may want to partner with other disciplines (e.g., philosophy) to experiment with using role playing, debate, and writing formal arguments. Furthermore, using alternative ethical frameworks, we can help student see opposing viewpoints and develop a more nuanced position. An acceptable resolution for a utilitarian framework might be unacceptable for an egoist framework [26]. Even the most ethical resolution might have negative outcomes for some scenario agents.

With regards to assessment of student learning, we echo the argument of Bairaktarova and Woodcock, who argue the importance of requiring students to answer ethical questions in their own words, rather than multiple choice answers, to bring forth the motivational factors that influence the students' ethical awareness and decision making [27]. We recommend moving from assessments with a correct answer to evaluating students ability to create a variety of alternative scenarios and provide insight into each alternative by extrapolating likely consequences. Thus, we recommend engaging learners in multiple modes of instruction and assessment (e.g., lectures, open-ended group discussions, brainstorming possible scenario resolutions, and argumentation using ethical frameworks).

Challenge 3: Ethical Decisions Beyond the Cognitive Domain

The curricular documents described in Sect. II treat ethics as a cognitive skill to be acquired and provide learning outcomes in terms of the cognitive domain of Bloom-Anderson's taxonomy (e.g., knowledge and comprehension) [28]. Students can perform test questions to demonstrate knowledge and how to behave in situations and avoid conflicts of interest. Being ethical requires vigilance to identify situations and find ethical resolutions. If the learner does not believe it is worth it (i.e., affective domain) then they may not apply the ethical skills they have developed, or might use them to be skillfully unethical and not get caught [5]. This raises the importance of affect domain in addition to the cognitive domain for learning.

Bloom-Anderson-Krathwohl's taxonomy contains three simultaneously active domains: cognitive (knowledge-based), affective (emotion-based), and psychomotor (action-based) [28]. The affective domain looks at one's attitudes, values, and appreciation for elements in learning, and is measured on five levels: receiving, responding, valuing, organizing, and characterizing. For example, receiving is associated with an awareness or willingness to listen, whereas responding involves students reacting to a stimuli. How students respond to elements of a case study on the affective domain can inform educators to what extend they value ethical reasoning.

Opportunity 3: Introduce Affective Learning Objectives

There is an opportunity to assess learners' progress towards becoming ethical professionals in the affective domain. Curricular recommendations can be improved by including affective learning outcomes. For example, in addition to cognitive learning outcomes such as "identify ethical issues that arise in software development and determine how to address them technically and ethically" [14], the ACM could also recommend that students "commit to treating colleagues fairly" thereby *valuing* principle seven of the code [1]. Similarly, another learning outcome could be to "defend intended actions as ethical" thus *characterizing* affect. Once we, as a community, create affective learning outcomes, we can measure students' appreciation for the ethical responsibilities of the profession and mitigate against mechanistic answers to ethical questions.

Challenge 4: Disconnect with Professionalism

Our final challenge of teaching SE ethics is that many students do not see SE as a profession, and do not value the shared community in a profession. Recent work confirms this assertion for software developers [3], and we have confirmed this anecdotally in our own teaching experience. When students are given the opportunity to explore why professions exist and what interests drive some occupations to define themselves as a profession, many students rule out the need for a professional body and common code, while others become heavily vested in these ideals. Opponents argue that computer instructions, on a rudimentary level, are neither ethical nor unethical, and can be evaluated as complete and correct given a specification. While proponents argue that it is the emergent properties of a system that determine whether the developers of that system behaved ethically. The opponents respond that each engineer may behave ethically following the codes discussed in Sect. II; however, the system may still have unintended consequences. The justification given by students in opposition is that not one person has the ability to fully comprehend an entire software system, and thus engineers should not be held responsible for their designs.

Opportunity 4: Encouraging Professionalism

It is important to advocate for students to see SE as a profession, both as part of their educational program and beyond. When ethics is taught as a stand-alone module separate from technical concepts in SE, students partition this learning; instead, ethical concepts should be integrated with technical ones [16]. We believe this will reinforce professionalism as integral to engineering excellence. Thus, we recommend creating a variety of ethics activities throughout SE courses to help students value ethical thinking and learn to further examine and resolve ethical situations. We envision embedding ethics activities into project-based learning at each stage of the software lifecycle. For example, ethical reasoning has been considered in requirements elicitation [29], design [30], and testing [31], as well as communication between developers and users [32]. To engage students with professionalism beyond the classroom, we can connect them with professional societies. One proposed approach for this is through mentorship programs, where students are paired with near-peers who are actively engaged in professional societies.

As noted in Sect. II, ethical values play a stronger role in the ABET accreditation for engineers than for computing professionals. Since software engineers straddle this separation, computing codes and accreditation can be strengthened to be more comparable. We call on professional societies to play a stronger role in fostering moral courage and enforcing codes of ethics (building on the call in [21]).

IV. Summary and Future Work

In this paper, we presented our position for opportunities in teaching SE ethics. After reviewing curricular documents, codes, and current best practices, we recommend that educators should add descriptive case studies as examples of ethical scenarios and embed these examples in workshops at each stage of the SE curriculum. Educators may also connect with practitioners and computing societies to foster a sense of professionalism in computing. As with all educational interventions, we must implement these opportunities with intent and measure the outcomes.

To support this work, we, as scholars, can assist educators in developing adaptable case studies that can be integrated into multiple SE contexts, as well as partner with our industry collaborators to expose more nuanced ethical situations. In future work, a systematic review of SE ethics interventions could explore the prevalence of the anecdotal observations made in this paper and understand to what extent they hold both within North America and globally. We call on scholars who have created case studies and successful ethical interventions to share them with the broader community.

Acknowledgments. We thank Jeffrey S. Castrucci for brainstorming ideas and discussing early drafts of this paper.

References

[1] D. Gotterbarn et al., "Code of ethics," IEEE-CS/ACM joint task force on Software Engineering Ethics and Professional Practices (SEEPP). Available: https://www.computer.org/education/code-of-ethics, 1999.

[2] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society, *Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. New York, NY, USA: ACM, 2015.

[3] A. McNamara, J. Smith, and E. Murphy-Hill, "Does acm's code of ethics change ethical decision making in software development?" in *Proc. of ESEC/FSE'18*, 2018, pp. 729–733.

[4] L. K. Lau, B. Caracciolo, S. Roddenberry, and A. Scroggins, "College students' perception of ethics," *Journal of Academic and Business Ethics*, vol. 5, p. 1, 2012.

[5] J. M. Bloodgood, W. H. Turnley, and P. E. Mudrack, "Ethics Instruction and the Perceived Acceptability of Cheating," *Journal of Business Ethics*, vol. 95, no. 1, pp. 23–37, 2010.

[6] O. Aderonmu, C. L. Aasheim, and P. Rutner, "Development and Testing of a Survey Instrument to Assess Ethical Perceptions of IT and IS Students," in *Proc. of SEINFORMS'13*, 2013, pp. 217–241.

[7] J. Taplin, *Move Fast and Break Things: How Facebook, Google, and Amazon Cornered Culture and Undermined Democracy*. Little, Brown, 2017. [Online]. Available: https://books.google.com/books?id=NX7UDAAAQBAJ

[8] IEEE Board of Directors, "IEEE Policies, Section 7 - Professional Activities (Part A - IEEE Policies)," Available: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/ieee-policies.pdf, accessed 01/16/21, 2020.

[9] ACM Code 2018 Task Force: Exec. Committee, "ACM Code of Ethics and Professional Conduct," Association for Computing Machinery. Available: https://www.acm.org/code-of-ethics, accessed 01/16/21, 2018.

[10] D. Gotterbarn, K. Miller, and S. Rogerson, "Software engineering code of ethics," *Commun. ACM*, vol. 40, no. 11, pp. 110–118, 1997.

[11] Canadian Engineering Accreditation Board, "2019 Accreditation Criteria and Procedures," Available: https://engineerscanada.ca/accreditation/accreditation-resources/2020-2021-cycle, accessed 01/16/21, November 2019.

[12] ABET Engineering Accreditation Commission, "Criteria for Accrediting Engineering Programs, 2019–2020," Available: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2019-2020/, access 08/08/19, 2019.

[13] ABET Computing Accreditation Commission, "Criteria for Accrediting Computing Programs, 2019–2020," Available: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2019-2020/, accessed 08/08/19, 2019.

[14] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society, *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. New York, NY, USA: ACM, 2013.

[15] The Mozilla Foundation, "Responsible Computer Science Challenge," Available: https://foundation.mozilla.org/en/what-we-fund/awards/responsible-computer-science-challenge/, access 01/01/21.

[16] B. J. Grosz, D. G. Grant, K. Vredenburgh, J. Behrends, L. Hu, A. Simmons, and J. Waldo, "Embedded ethics: Integrating ethics across cs education," *Commun. ACM*, vol. 62, no. 8, pp. 54–61, Jul. 2019.

[17] S. Vallor, "Moral deskilling and upskilling in a new machine age: Reflections on the ambiguous future of character," *Philosophy & Technology*, vol. 28, no. 1, pp. 107–124, 2015.

[18] C. Rottmann, D. Reeve, R. Sacks, and M. Klassen, "Where's my code? engineers navigating ethical issues on an uneven terrain," in *Paper presented at the American Society of Engineering Education Annual Conference and Exposition*, Salt Lake City, UT, USA, 2018, Available: http://hdl.handle.net/1807/89246.

[19] E. Qualman, *Socialnomics: How Social Media Transforms the Way We Live and Do Business*. Wiley, 2012.

[20] A. Finkelstein and J. Dowell, "A comedy of errors: the london ambulance service case study," in *Proc. of IWSSD'96*, 1996, pp. 2–4.

[21] J. Herkert, J. Borenstein, and K. Miller, "The boeing 737 max: Lessons for engineering ethics," *Science and Engineering Ethics*, vol. 26, no. 6, pp. 2957–2974, 2020.

[22] C. O'Neil, *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown, 2016.

[23] D. Cemiloglu, E. Arden-Close, S. Hodge, T. Kostoulas, R. Ali, and M. Catania, "Towards ethical requirements for addictive technology: The case of online gambling," in *Proc. of REthics'20*, 2020, pp. 1–10.

[24] L. Yu, "Ethical considerations in case studies," in *Proc. of REthics'20*, 2020, pp. 15–21.

[25] C. Rottmann and D. Reeve, "Equity as Rebar: Bridging the Micro/macro Divide in Engineering Ethics Education," *Canadian Journal of Science, Mathematics and Technology Education*, pp. 1–20, 2020.

[26] J. M. Kizza, "Ethics, technology, and values," in *Ethical and Social Issues in the Information Age. Undergraduate Texts in Computer Science*. Springer, New York, NY, 1998.

[27] D. Bairaktarova and A. Woodcock, "Engineering ethics education: aligning practice and outcomes," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 18–22, 2015.

[28] D. R. Krathwohl, "A Revision of Bloom's Taxonomy: An Overview," *J. Theory Into Practice*, vol. 41, no. 4, pp. 212–218, 2002.

[29] F. B. Aydemir and F. Dalpiaz, "A roadmap for ethics-aware software engineering," in *Proc. of FairWare'18*, 2018, pp. 15–21.

[30] I. Ozkaya, "Ethics is a software design concern," *IEEE Software*, vol. 36, no. 3, pp. 4–8, 2019.

[31] H. Jeret-Mäe, "The 10 Commandments for Ethical Software Testers," Available: https://nortal.com/us/blog/10-commandments-ethical-software-testers/, accessed 01/18/21, June 2017.

[32] B. Paech and K. Schneider, "How do users talk about software? searching for common ground," in *Proc. of REthics'20*, 2020, pp. 11–14.