



Laws, Ethics, and Fairness in Software Engineering

Miroslaw Staron¹, Silvia Abrahão², Alexander Serebrenik³, Birgit Penzenstadler⁴, Jennifer Horkoff⁵, and Chetan Honnenahalli

SOFTWARE ENGINEERING IN the era of generative AI, large datasets, and the superfast pace of software development often tends to focus on technology, tools, and methods, putting aside us, the software engineers. In this edition of the “Practitioners’ Digest,” we focus on softer aspects of software engineering and report from two conferences: the 28th International Conference on Evaluation and Assessment in Software Engineering (EASE 2024) and the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2024). The selection of papers provides a glimpse on handling privacy, documenting ethical considerations in AI models, and trustworthy AI. Feedback or suggestions are welcome. In addition, if you try or adopt any of the practices included in the column, please send us and the authors of the paper(s) a note about your experiences.

Is Agile a Culture?

Cultural factors have long been observed to influence workplace behavior, and software developers are no exception. This has led both researchers and practitioners to adopt existing theories of national culture, such as those proposed by Geert Hofstede. However, software developers often

differ from the general population in terms of demographics and work dynamics, such as the emphasis on collaboration, which may limit the applicability of these theories in understanding cultural aspects specific to software development.

To investigate this issue, Michael Neumann, Klaus Schmid, and Lars Baumann, in their paper “What You Use Is What You Get: Unforced Errors in Studying Cultural Aspects in Agile Software Development,” compared Hofstede’s cultural dimension scores for Japan and Germany with individual scores from two agile software development teams in those countries. The study found that developers’ cultural values did not always align with the general population’s. For example, developers showed similar views to the broader population in terms of societal power disparities (as measured by Hofstede’s power distance index), but in line with agile work practices, they placed greater value on collaboration (masculinity/femininity) and were more open to new ideas and divergent opinions within teams (uncertainty avoidance). Overall, the scores obtained by Neumann et al. significantly diverged from Hofstede’s national scores, particularly in areas like uncertainty avoidance and long-term orientation in Germany and masculinity/femininity, uncertainty avoidance, and long-term orientation in Japan.

The implications of this study are relevant for both researchers and managers interested in cultural aspects, be it in international and multinational contexts or simply trying to understand cultural elements in a single team. While it is important to remain mindful of cultural differences, it is equally critical to recognize additional factors that shape the values and behaviors of software developers. The authors recommend considering at least two levels when discussing culture, such as national versus organizational culture, but ideally looking at the cultural specificities of the team of interest. This paper was published as part of the Vision and Emerging Results track of EASE 2024 and is available at <https://tinyurl.com/y8sddfrw>.

Communities and Fairness

How do software developers experience (un)fairness in the workplace? To explore this question, Emeraldal Sesari, Federica Sarro, and Ayushi Rastogi, in their paper “Understanding Fairness in Software Engineering: Insights From Stack Exchange,” curate and analyze 136 posts from various websites on the Stack Exchange network. Their study reveals that the majority of fairness discussions revolve around income, indicating that many software practitioners are deeply concerned with issues related to compensation and its fair distribution. However,

these discussions on income fairness are diverse and often controversial. Another prominent topic is fairness in hiring, such as the frustration of managers seeking to recruit more developers from underrepresented groups but encountering challenges in doing so. The authors observed that most posts focus on fairness in decision-making processes (e.g., work allocation or job interviews) and the distribution of rewards (e.g., pay). Additionally, the study shows that discussions on fairness often extend beyond protected attributes like gender, age, or ethnicity.

Based on these insights, the authors recommend that companies reflect on specific aspects of fairness or particular situations where employees might experience (un)fairness. For instance, fairness in decision-making procedures is strongly correlated with job satisfaction, underscoring the importance for companies to invest in ensuring greater consistency, accuracy, and minimization of bias. The paper was published at ESEM 2024 and is available at <https://tinyurl.com/2tn54adr>.

Responsible, Fair, and Trustworthy AI Models

The academic and practice communities are paying increasing attention to the ability of AI systems to operate responsibly and ethically. A plethora of frameworks and guidelines have emerged to support practitioners in implementing trustworthy AI applications (TAI), but so far there has been little research into whether and how these frameworks are being used. Maria Teresa Baldassare, Domenico Gigante, Marcos Kalinowski, Azzurra Ragone, and Sara Tibido, in their paper “Trustworthy AI in Practice: An Analysis of Practitioners’ Needs and Challenges,” study the vision of 34 AI practitioners on the principles of TAI. The study addresses how these 34 AI

practitioners approach TAI principles and what they would like to have when trying to incorporate these principles into the systems they develop.

Through a survey and semistructured think-aloud interviews, they investigated practitioners’ visions, challenges, and needs in developing TAI systems. Practitioners mainly focus on privacy, often overlooking fairness. Noteworthy is the fact that half of the participants stated that they did not fix TAI issues after discovering them in their projects. The strategies most employed to build TAI applications focus on data quality enhancement or choosing the most self-explanatory algorithm. Practitioners felt the need for tools to monitor the model after production deployment and knowledge bases and actionable guidelines to help them implement trustworthiness throughout the entire SDLC. This paper was published as part of the Research track of EASE 2024 and is available at <https://tinyurl.com/5emye76a>.

Ethics and AI Models

As AI becomes increasingly more popular, software developers become more aware about the ethical considerations related to using AI models. In their paper “Documenting Ethical Considerations in Open Source AI Models,” Haoyu Gao, Mansoor Zahedi, Christoph Treude, Sarita Rosenstock, and Marc Cheong focus on the ethical content of existing open source AI project documentation. The authors mine model cards and readme files from both GitHub and HuggingFace AI projects, using KeyBERT to expand keyword searchers and filtering documents for relevance. Finally, the authors focus on 265 documents, applying thematic coding to look for ethical themes. Six major themes are found, including 1. data quality issues,

where findings indicate documented issues with private or objectionable (offensive) data and data labeling bias; 2. model behavioral risks, focusing on biased, objectionable, or unethical answers; 3. model risk mitigation, mostly focusing on high-level recommendations for developers downstream of the model development; 4. model use cases, describing out-of-scope use cases, malicious model uses, and licenses and regulations; 5. reference to other materials, covering external and internal materials that explore ethics issues in more detail; and 6. others, covering codes that do not fit in the previous categories, including future ethical research directions, limited exploration of ethics in the documentation, disclaimers from model developers, and documentation of environmental impact.

The results show that the most focus is placed on model behavior and expected use cases, with less focus on data quality concerns. Descriptions of actionable mitigations are scarce; the most emphasis is on describing the potential issues only. The paper’s results are used to formulate recommendations per role. AI model developers should take more care to document ethical concerns, consider data quality concerns, and detail possible downstream mitigations. To compensate for these gaps, software developers should work through and document mitigations for ethical issues. Finally, recommendations are given for researchers, educators, and policy members. Overall, this work gives an empirically grounded view of ethical documentation for those projects that document this topic, giving a grounded view of the state-of-practice, including current foci and limitations. The paper was published at ESEM 2024 and is available at <https://tinyurl.com/488pay49>.



MIROSLAW STARON is a full professor in the Interaction Design and Software Engineering division in the Computer Science and Engineering Department shared by Chalmers University of Technology and the University of Gothenburg, Sweden, 412 96 Sweden. Contact him at <https://www.staron.nu> or mirosław.staron@cse.gu.se.



SILVIA ABRAHÃO is a full professor at the Department of Computer Science, Universitat Politècnica de València, Valencia, 46022, Spain. Contact her at <https://sabrahao.wixsite.com/dsic-upv> or sabrahao@dsic.upv.es.



ALEXANDER SEREBRENİK is a professor at Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands. Contact him at a.serebrenik@tue.nl.



BIRGIT PENZENSTADLER is an associate professor at Chalmers University of Technology and the University of Gothenburg, Gothenburg, 412 96 Sweden. Contact her at <https://www.chalmers.se/en/staff/pages/birgit.aspx> or birgitp@chalmers.se.



JENNIFER HORKOFF is an associate professor at the Interaction Design and Software Engineering division in the Computer Science and Engineering Department shared by Chalmers University of Technology and the University of Gothenburg, Sweden, 412 96 Sweden. Contact her at <https://www.chalmers.se/en/persons/jenho/> or jenho@chalmers.se.



CHETAN HONNENAHALLI is an engineering manager at Meta, Menlo Park, CA 94025 USA. Contact him at <https://www.linkedin.com/in/hschetan/> or chetanhs@meta.com.

open source software (OSS) into commercial and noncommercial projects. In their paper “An Exploratory Mixed-Methods Study on General Data Protection Regulation (GDPR) Compliance in Open-Source Software,” Lucas Franke, Huayu Liang, Sahar Farzanehpour, Aaron Brantly, James C. Davis, and Chris Brown explore an important area within software engineering: how data privacy regulations, specifically GDPR, influence the development and compliance processes of OSS.

The authors conduct a mixed-methods study with the aim to investigate how GDPR impacts OSS development, addressing four key research questions: the effect of GDPR on development activity, the challenges developers face in implementing GDPR, developers’ perceptions of these requirements, and how developers assess compliance. The results highlight that GDPR introduces significant challenges for OSS developers, regarding user data management, compliance assessments, and the associated costs in time and effort. The authors shine a light on the sentiment among OSS developers toward GDPR as many view it as an onerous and complicated regulation. These challenges are further exacerbated by limited access to legal expertise and automated tools, which are more accessible to developers in commercial environments.

By underscoring the complexities OSS developers face, the authors advocate for better policy resources and automated tools to support GDPR implementation. This research lays the groundwork for continued exploration into regulatory impacts on the broader OSS ecosystem and how legal frameworks can better align with the practical realities of software development. The paper was published at ESEM 2024 and is available at <https://arxiv.org/pdf/2406.14724>.

General Data Protection Regulations and Open Source

With the growing adoption of privacy laws worldwide, including General


Data Protection Regulation (GDPR), ensuring software compliance with these regulations is paramount, especially given the extensive integration of

Avoiding Conflicts or Handling Them Automatically: That Is the Question

Merge conflict resolution is essential in practical software development as roughly 15% of branch merges result in one or more merge conflicts, hampering cooperation and continuous integration. Based on previous work, the hypothesis is that most merges are derivable, i.e., they are constructed from the base version and/or one of the parent versions without modification. In other words, if you and I have a merge conflict in a chunk of code, we are likely to resolve the conflict by choosing either my version, your version, the original version, or a sequence of them. Alexander Boll, Yael van Dok, Manuel Ohrndorf, Alexander Schultheiß,

and Timo Kehrler in their paper “Towards Semi-Automated Merge Conflict Resolution: Is It Easier Than We Expected?” collect 10,000 GitHub projects over 10 programming languages capturing more than 4.5 million merges. The authors further create MeGA, a tool for the analysis of merges in Git repositories. Overall, 87.9% of merge chunks are resolved using a derivable pattern, and 62.4% of merge chunk resolution use the version of the current branch (i.e., our version). Some variation is seen among languages. Looking at entire merges, rather than mergeable chunks, the more chunks per merge, the lower the likelihood of the whole merge being derivable. Still, whole merge commits are 30%–40% derivable, and this is

more consistent among languages. The results also show that types of merges are relatively consistent at the file level.

The implications of these results are that automated merge tooling at the merge chunk level is likely possible, and the chances of successful automation of merges with few conflicting chunks is relatively high. Automation at the file level is also possible by propagating chunk merging decisions to the whole file. Thus, overall, this paper works toward recommendations and design ideas for automated merging tools, with a focus on empirically validated, likely suggestions at the chunk and file levels. This paper won the Distinguished Paper Award at EASE 2024 and is accessible at <https://tinyurl.com/bvk9vpcc>. 



IEEE Security & Privacy magazine provides articles with both a practical and research bent by the top thinkers in the field.

- stay current on the latest security tools and theories and gain invaluable practical and research knowledge,
- learn more about the latest techniques and cutting-edge technology, and
- discover case studies, tutorials, columns, and in-depth interviews and podcasts for the information security industry.



computer.org/security



IEEE
COMPUTER
SOCIETY

