Software Engineering Education for the Next Generation: SEENG 2023 Workshop Report

Stephan Krusche
Technical University of Munich
krusche@tum.de

Jonathan Bell
Northeastern University
j.bell@northeastern.edu

Bastian Tenbergen
SUNY Oswego
bastian.tanbergen@oswego.edu

DOI: 10.1145/3617946.3617959
https://doi.org/10.1145/3617946.3617959

ABSTRACT

The 5th International Workshop on Software Engineering Education for the Next Generation was held on May 16, 2023 in Melbourne, Australia. The workshop was part of the 45th International Conference on Software Engineering. It specifically supported the general theme of "Educating the Next Generation of Software Engineers". Building on its predecessors, the workshop used a highly interactive format, structured around eight short paper presentations to generate discussion topics, an activity to select the most interesting topics, and structured breakout sessions. This enabled the participants to discuss the most interesting topics in detail. Participants presented the results of the breakout sessions using mind maps.

1. INTRODUCTION

Following the previous editions of the workshop, our goal in this 5th edition was to continue to bring together key stakeholders that shape the future education for aspiring software engineers. We wanted to discuss the unique needs and challenges of software engineering education for the next generation, including educators and representatives of STEM[1] education.

We solicited papers addressing a variety of related topics:[2] software engineering education for new and emerging technologies; novel approaches to designing software engineering curricula; skills and continuing education for software engineering educators; classroom formats that cater to diverse learning styles; teaching approaches that leverage technology-enhanced education in software engineering courses; balancing teaching of soft and hard skills; rigor and practicality in software engineering education; experience in educating students in software engineering programs.

The eclectic international committee, comprising 22 members from both the academic and industrial realms of software engineering, hailed from eight different countries. They conducted a tough selection process, in which at least three reviewers reviewed each submission. They accepted eight out of the 14 submitted papers, demonstrating a selective acceptance rate of approximately 57 %. These accepted papers encompass a broad range of topics, including project-based courses, teaching design by contract, software testing methodologies, student feedback mechanisms, assessment practices, and more.

The driving force behind this 5th edition was the aspiration to nurture and expand a community passionate about and committed to educating future generations. A group of 23 individuals participated in the workshop, congregating onsite to exchange ideas, wisdom, and experiences, provide and receive guidance on teaching techniques, and seek potential partnerships for fresh initiatives in software engineering education.

2. AGENDA

We based the workshop agenda on the structure and practices proposed in designing interactive workshops for software engineering educators [PED19]. With most participants attending locally and only a few attending remotely, we offered a hybrid framework, including the use of an online collaborative whiteboard platform called Miro.[3]

The workshop began with a brief introduction of all participants. This was followed by a series of short presentations of accepted papers to remind everyone of the problems identified and how others had solved them. Short clarification questions initiated first discussions directly after the presentation. To identify common interests, participants recorded interesting findings from the presentations on virtual sticky notes in Miro.

Based on the insights gathered during the presentations, we performed an affinity mapping activity to identify and select discussion topics. The topics that emerged as of most interest to the participants were:

1. Grading and Assessment
2. ChatGPT and Large Language Models
3. Teaching Practices for Engagement and Critical Thinking

We closed the workshop with a retrospective and identification of future action items. In the evening, participants discussed their opinions in the workshop dinner.

3. PAPERS

Eight papers were accepted for publication and invited for presentation at the workshop. These included two full-length 8-page papers and six shorthand position papers, each four pages in length. The workshop was held in person in Melbourne, Australia and online on May 16, 2023. The accepted papers were presented in brief sessions. The majority of session time was dedicated to discussion, questions, answers, and developing new ideas together.

Topics were pleasantly diverse, varying in their level of technicality as well as applicability in terms of SE courses. A key aspect of SE education is teaching students quality. This takes many forms, e.g., code quality [BW23], aesthetics [MF23], product quality and user-centeredness [Pé23], but also process quality. For the latter, Ma and Lopes [HM23] propose a code repository tool that automatically reads students' commit messages and suggests methods of improving them, specifically about "what" was changed in the commit and "why" that change took place.

[1] STEM stands for science, technology, engineering and mathematics and refers to any subjects that fall under these four disciplines.

[2] The workshop website with the call for paper can be found online at the following link: https://conf.researchr.org/home/icse-2023/seeng-2023

[3] Miro is a digital whiteboard that makes it easy to collaborate with others: https://miro.com

Their results show a clear positive impact on perceived informativeness, clarity, and level of detail.

Since the advent of user stories in software engineering, usability and user-centered design are concerns no longer relegated for specialists, but are at the forefront of every technical stakeholder in a SE team. To enable students to write high-quality user stories, Cécile Péraire [Pé23] proposes a template containing four Cs: Context, Card, Conversation, and Confirmation. This template helps students specify the rationale as well as success criteria for a user story just as much as additional information background that goes with the feature expressed therein. Experimentation revealed a strongly positive impact on students' ability to write effective user stories that meet stakeholder needs while at the same time fostering innovative solutions.

A generically relevant issue impacting software quality is investigated by Fedorova et al. in [MF23]. The authors maintain that aesthetics must play a major role in SE education to the point of making it a specific curricular learning outcome. In their position paper, the authors make a strong argument for aesthetics being a key aspect of the perceived quality of a software product and supply striking evidence from a student survey in support of this argument. However, not only the external appeal of software must be beautiful, a considerable number of respondents attribute aesthetic appeal as a vital property of internal quality as well. In other words: beautiful code may beget beautiful software.

Quality, of course, also depends to a large degree on correctness. While correctness is easy to ascertain, learning how to achieve it can be very challenging. To this end, Wanjiru et al. propose the notion of a generic model to classify code into correctness levels [BW23] and apply it to teaching students SQL. The key benefit of such an approach lies in automatically differentiating student solutions to assignments that are not quite perfect with regard to the kind of errors. By doing so, generically applicable feedback patterns can be applied, thereby systematically leading students to improve their queries and achieving higher levels of correctness while feasibly reducing instructor effort.

Yet, quality is not merely limited to the outcomes of SE, but also expands to how we teach SE. One successful way to do this is to employ gamification. In [SHBB23], Speth et al. suggest a novel web-based platform that implements gamification in SE education called Gamify-IT. The particular benefit of such a platform is to remove the pedagogical triviality of gamification which occurs when the instructor merely uses "levels" and "scores" instead of assignments and grades. Rather than planting a game-theme on a SE course, Gamify-IT allows students to immerse themselves into roleplay, playing out software engineering tasks in a virtual world using mini-games.

Other than gamification and aesthetics [MF23], another relatively novel aspect instructors and students struggle with is the emergence of human-like artificially intelligent agents like ChatGPT. Products like this may offer new challenges in teaching, and learning software engineering and cannot be ignored. Therefore, Neumann et al. [MN23] review grey literature to leverage possible advantages and avenues to integrate AI-based instruction into SE education to the benefit of both the students (e.g., get feedback quickly and amply), and the instructor (e.g., be supported with partial scoring of results).

The way we teach SE also largely impacts the quality of the products we can expect future graduates to produce. Because of this, Ardic and Zaidman make a strong case for integrating testing skills not just into dedicated courses or letting aspiring professionals self-teach. In their research paper [BA23], the authors investigate curricula of 100 highly ranked universities and conducted a survey among practitioners into where their knowledge of testing and quality assurance comes from. Their results indicate that the earlier testing skills are included in a students' SE education, the stronger their skills to ensure high-quality software products.

Another set of real-world considerations is proposed by Morrison and Slankas in [PM23]. In their paper, the authors report on experiences from project-based programming and software engineering instruction as a core aspects of a novel, non-consecutive Master's program aimed at students with non-IT backgrounds (i.e., finance or economics). The authors offer lessons learned based on results from offering such instruction to small and large classes to over 100 students.

The workshop concluded with a collaborative session, where the participants jointly created a mind map of topics that represent the needs and requirements future software engineers need to address in the coming decade.

4. DISCUSSIONS

Based on the presentations of these papers, we held a short mind-mapping session as part of the session summaries to identify the main themes of the papers presented. Participants divided themselves into working groups to further discuss the selected topics, one group per topic.

For each theme, participants were asked to define a "big hairy audacious goal," that is, a very ambitious goal toward which future generations of educators should work. Group participants created a mind map to consider what success looks like and how to achieve the goal. Finally, each group presented their mind map to the other workshop participants.

4.1 Grading and Assessment

The goal of the first breakout group was to consider how to structure grading and assessment in software engineering classes so as to help each student achieve their maximum potential. Assessment is critical to understand learners' progress, and to help identify gaps in knowledge transfer. Particularly in the context of project-based software engineering courses, designing effective assessments can be challenging. Figure 1 shows the results of the discussion in a mind map.

Much of the discussion focused on how to design assessments that help students receive fast, reliable feedback. Instructors might rely on different feedback mechanisms, considering self-assessment and peer-assessment as effective techniques that are complementary to more traditional means like TA/instructor grading and automated grading. However the assessment is performed, participants agreed that there should be a far greater prioritization on feedback over grading. One approach for grading that was discussed for improving learning outcomes focuses on putting feedback as the most important aspect, rather than the precise numeric grade. Rubrics that sort student submissions into coarse buckets (e.g. check, check plus, check minus) can reduce the time spent assigning points and research has shown that this approach increases student motivation and produces higher-quality work.

The discussion also considered what systemic barriers to improving assessments in software engineering courses. Two key themes were identified: 1) instructors do not have sufficient resources to provide learners with fast, frequent and detailed feedback, and 2) institutions can be slow to change and impose external barriers. As an example of an institutional barrier: some participants noted that their institutions require that students' grades be ranked in a total order with no ties. At such an institution, it would be impossible to implement a coarse-bucket grading scheme, as no ties are permitted.

4.2 ChatGPT and Large Language Models

The discussions revolved around the application of large language models, such as ChatGPT, in software engineering education.

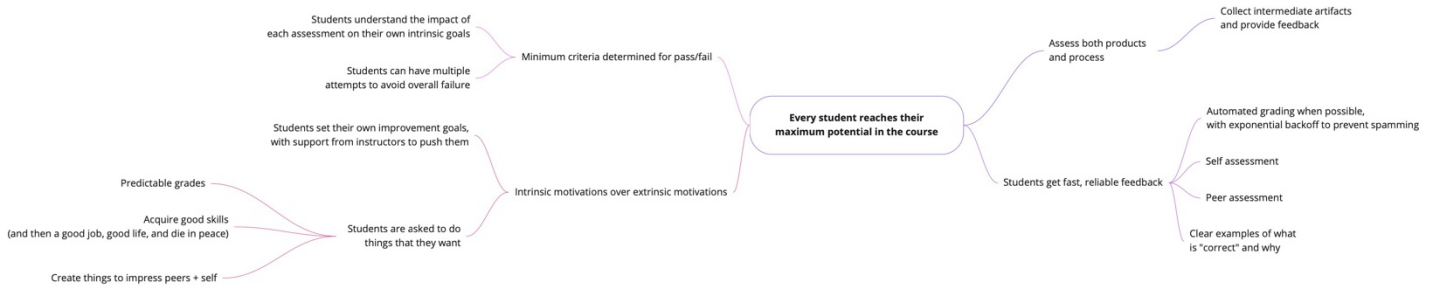

Figure 1. Mind map for discussion on Assessment and Grading.

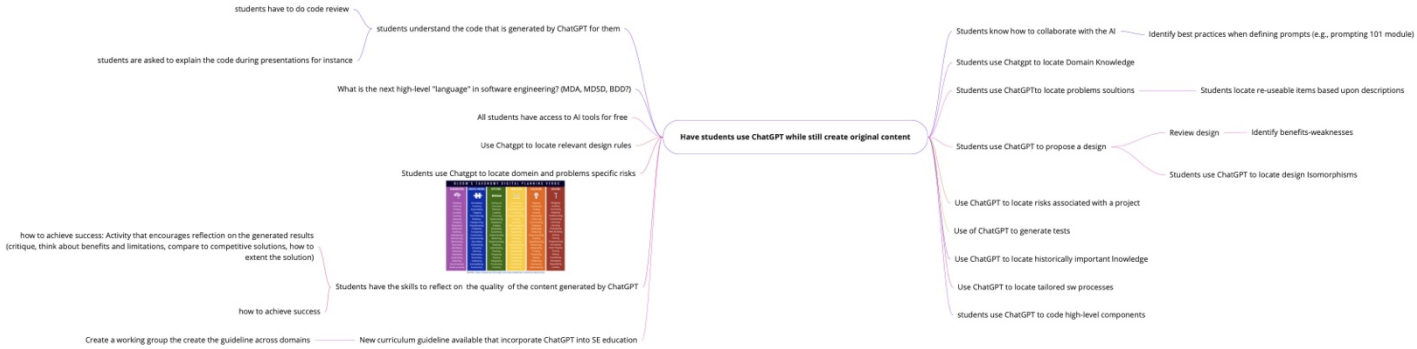

Figure 2. Mind map for discussion on ChatGPT and Large Language Models.

Observations showed a considerable number of students intensively utilizing, and at times over-relying on, artificial intelligence. However, it became apparent that there was a disparity in students' access to and proficiency with these technologies. Figure 2 shows the results of the discussion in a mind map.

The participants actively suggested that educators should incorporate generative AI into their curriculums as an educational tool and advocate for its responsible usage. They unanimously agreed that effectively utilizing AI tools represents a significant part of the future of programming and software engineering education. This necessitates a comprehensive debate about the skills and competencies students will need in the future.

Key discussions emphasized the importance of educating students on AI tool utilization and fostering their ability to critically analyze the quality of content generated by AI platforms such as ChatGPT. The group proposed that students must use AI as a productivity-enhancing tool and not merely rely on it.

Concurrently, the group recognized that students should retain their ability to critically assess the quality of AI outputs. They further recommended encouraging students to incorporate AI-provided solutions into larger frameworks, rather than treating them as independent solutions. The workshop concluded with a mutual understanding of AI's evolving role in education and a dedication to developing strategies that responsibly and effectively empower students to leverage its potential.

4.3 Engagement and Critical Thinking

The discussion centered on the theme of "Learningverse", which advocates for an education system that is primarily powered by the inherent drive of learners. It emphasized the creation of an environment that fosters self-directed learning, thereby encouraging personal curiosity and genuine interest in students. Another significant part of the discussion focused on reshaping the traditional dynamic between students and teachers. The participants shared the consensus that educators should act as "coaches" rather than authority figures, fostering an atmosphere of collaboration and shared intellectual exploration. Figure 3 shows the results of the discussion in a mind map.

A crucial facet of the discourse was the role of community in the learning process. Techniques were explored to encourage student engagement within a communal setting, highlighting the benefits of peer-to-peer learning and collaborative problem-solving. This dialogue transitioned smoothly into the concept of "lean learning", which champions broad, interdisciplinary learning. The importance of nurturing intellectual curiosity beyond academic confines was stressed, bringing to light the potential of learning in a wide array of fields.

Appreciation for open-ended questions was another topic that garnered significant attention. Participants underscored the need to teach students to value ambiguity and complexity in intellectual discourse. This would, in turn, encourage students to see the journey of exploration as a learning opportunity in itself. Additionally, the group also tackled the paradigm shift from numeric grading to curated portfolios in student assessment. They were unanimous in their belief that portfolios provided a more holistic representation of a student's abilities, interests, and growth, thereby empowering them to better chart their future paths.

The final parts of the workshop revolved around rethinking physical learning environments and content delivery methods. Suggestions were made to modify learning spaces to encourage collaboration, creativity, and independent thinking. Likewise, participants shared various approaches to delivering educational content to cater to diverse learning styles. Techniques such as integrating technology, employing experiential learning methods, and adopting multi-modal instruction were among the strategies discussed. The workshop concluded with a rich exchange of ideas and experiences, shedding light on the evolving dynamics of modern education.

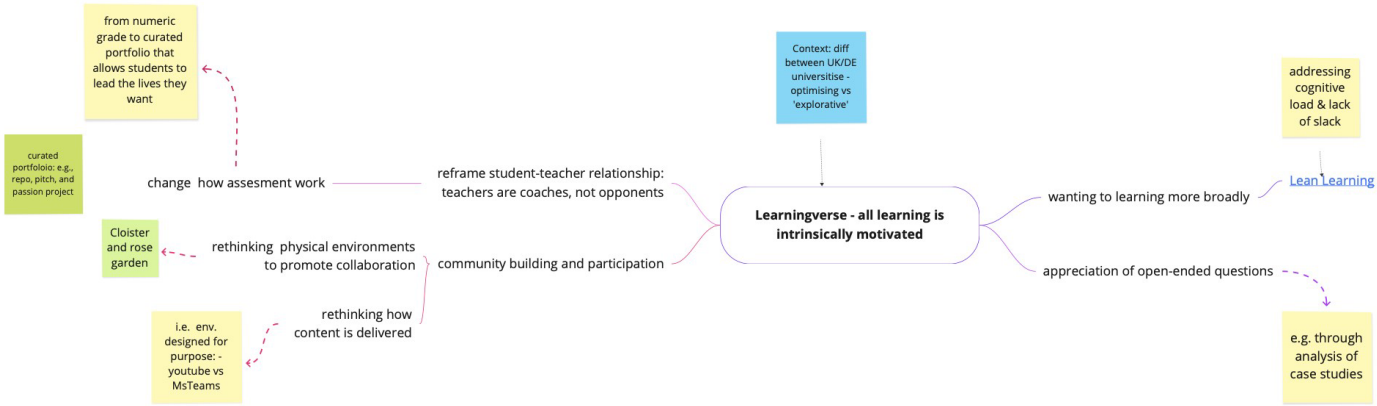

Figure 3. Mind map for discussion on engagement and critical thinking.

4.4 Workshop Retrospective

The last activity of the workshop was a retrospective in which the participants inserted their opinion on sticky notes in Miro. They provided suggestions for the questions "What went well during the workshop?", "What went wrong and could be improved?" and "What should be the next steps?".

Positive feedback included the interactive format, interesting discussions, and praise for the engaged participants. Diverse talks, great idea generation, topic selection and informative presentations were helpful. The use of mind maps in Miro was viewed positively. It was good to meet again in person after many online meetings during the pandemic.

Suggestions for improvement included a desire for more time for discussion, a discussion that the hybrid setup was tricky and a proposal for using flip charts for note taking.

As for next steps, participants suggested starting collaborative projects and writing blog posts to advertise their work. Authors should begin to evaluate their ideas related to ChatGPT and have a special workshop on the design of the future software engineering syllabus.

5. CONCLUSIONS

The feedback was consistently positive. Participants appreciated the interactive workshop format, the use of the Miro collaboration tool, the short presentations, and the opportunity to discuss with other participants in an online workshop. The main suggestion for improvement involved the use of onsite microphones for all participants, as remote participants sometimes had difficulty hearing what was being said on-site. Action items identified included using the workshop blog to present the results of the breakout sessions and to summarize and promote the contributions.

The 6th International Workshop on Software Engineering Education for the Next Generation will be part of the ICSE 2024[4] conference in Lisbon, Portugal. Krusche, Tenbergen, and Bell will organize the successor workshop keeping an interactive format with a strong focus on discussions.

Acknowledgments. We want to thank all program committee members for their work and selection of high-quality papers. We would also like to thank all participants for attending the 5th edition of ICSE's software engineering education workshop. The workshop was a success thanks to your enthusiasm, active participation, insights, and experiences.

[4] https://conf.researchr.org/home/icse-2024/

REFERENCES

[BA23] A. Zaidman B. Ardic. *Hey teachers, teach those kids some software testing*. In Proceedings of the 5th Workshop on Software Engineering Education for the Next Generation, 2023.

[BW23] D. Hiemstra B. Wanjiru, P. van Bommel. *Towards a generic model for classifying software into correctness levels and its application to SQL*. In Proceedings of the 5th Workshop on Software Engineering Education for the Next Generation, 2023.

[HM23] C. Lopes H. Ma. *Improving the quality of commit messages in students' projects*. In Proceedings of the 5th Workshop on Software Engineering Education for the Next Generation, 2023.

[MF23] M. Mazmanian M. Fedorova, P. Dourish. *Not just a matter of style: Does aesthetics have a place in software engineering curriculum?* In Proceedings of the 5th Workshop on Software Engineering Education for the Next Generation, 2023.

[MN23] E. Schön M. Neumann, M. Rauschenberger. *"We need to talk about ChatGPT": The future of AI and higher education*. In Proceedings of the 5th Workshop on Software Engineering Education for the Next Generation, 2023.

[PED19] Cécile Péraire, Hakan Erdogmus, and Dora Dzvonyar. *Designing interactive workshops for software engineering educators*. In International Workshop on Frontiers in Software Engineering Education, pages 217–231. Springer, 2019.

[PM23] J. Slankas P. Morrison. *"Work in the morning instead of midnight" and other lessons learned in fintech 512*. In Proceedings of the 5th Workshop on Software Engineering Education for the Next Generation, 2023.

[Pé23] Cécile Péraire. *Learning to write user stories with the 4c model: Context, card, conversation, and confirmation*. In Proceedings of the 5th Workshop on Software Engineering Education for the Next Generation, 2023.

[SHBBB23] S. Speth, L. Hofmeister, U. Breitenbücher, and S. Becker. *Gamify-it a web-based gaming platform for software engineering education*. In Proceedings of the 5th Workshop on Software Engineering Education for the Next Generation, 202