Editor: Jeffrey C. Carver
University of Alabama
carver@cs.ua.edu

Gender, Sentiment and Emotions, and Safety-Critical Systems

Jeffrey Carver, Rafael Capilla, Birgit Penzenstadler, Alexander Serebrenik, and Alejandro Valdezate

THIS ISSUE'S ARTICLE reports from the 40th International Conference on Software Engineering (ICSE 18) and the 17th International Conference on Software Reuse (ICSR 18). The ICSE papers focus on sociotechnical issues related to gender and sentiment or emotion. The ICSR paper focuses on safety-critical systems. Feedback or suggestions are welcome. In addition, if you try or adopt any of the practices described in the article, please send Jeffrey Carver and the paper authors a note about your experiences.

Gender Diversity in Software Engineering Education

"Barriers to Gender Diversity in Software Development Education: Actionable Insights from a Danish Case Study," by Valeria Bosotti, reports on an exploratory case study at the IT University of Copenhagen (ITU).[1] Bosotti interviewed 38 undergraduate or high-school students and surveyed 395 undergraduate or high-school students to investigate the main sociocultural barriers to female participation in ITU's Bachelor of Software Development program. The most influential factors were students' perceptions of stereotypes and role models, their attitudes and behavior regarding software development education, and gender differences in pre-university coding skills.

These results led to the design of local interventions to remove the stereotype biases. Some interventions focused on outreach:

- Teachers and advanced female students taught an IT camp for high-school girls, which led to 20 percent of the attendees enrolling in the Bachelor of Software Development program.
- An ad campaign included a promotional video shared on social media, in which male and female students interviewed each other regarding computer science and their motivation for studying it.

Other interventions focused on bridging the differences in pre-university coding skills:

- Free programming workshops took place just before a semester.
- Tutors ran a peer-to-peer study lab.
- Teaching assistants offered live coding sessions.

Because of the interventions, the percentage of women in the Bachelor of Software Development program doubled from 11 in 2016 to 22 in 2017.

Although not all of these interventions are directly transferable to industry, it's important to offer diverse role models. If software development companies represent themselves in all their diversity and offer insights into their employees' motivations for performing their particular kind of work, they might significantly increase gender diversity in computer science programs and ultimately in the workforce.

Access the paper at http://bit.ly/PD_2018_Nov_1.

GenderMag

"Open Source Barriers to Entry, Revisited: A Sociotechnical Perspective," by Christopher Mendez and his colleagues, discusses the barriers faced by people, especially women, interested in joining open source communities.[2] Adding to previous research in this area, this paper focuses on the impact of tools such as GitHub on newcomers' attempts to perform common tasks such as submitting a pull request or setting up their environment.

Mendez and his colleagues customized a research-based persona

0740-7459/18/$33.00 © 2018 IEEE

named Abby: an open-source-systems newcomer in her final year as a computer science major. Although Abby isn't representative of all women, she embodies problem-solving strategies that are statistically more common among women than among men. For example, Abby tends to gather fairly complete information before proceeding to her next step. (In contrast, men are statistically more likely to pursue the first promising idea and then backtrack if needed.)

Five teams of open source professionals put themselves in Abby's shoes to walk through various standard GitHub processes. At each step, they analyzed whether Abby would know what to do and whether she would know if she made the correct choice. In the past, this walkthrough process, called GenderMag (Gender Inclusiveness Magnifier), has helped software teams find issues, particularly regarding gender biases, in software. For example, collaborators from Microsoft made these observations about GenderMag:[3]

> *[Software revised using GenderMag ideas produced] very positive responses from both novice and experienced customers.*
>
> *[GenderMag] uncovered many general usability- and gender-related issues. … [Team name] along with two other key teams are addressing [issues] in the next version of [product name].*

Using GenderMag, the five teams identified issues "baked into" tools that were much broader than tool and UI bugs. For example, some tool-based processes were unclear to newcomer Abby (for instance, one participant suggested that Abby would "be like 'I know my stuff works,' but 'I don't really know what a pull request looks like'"). Also, some terminology was unclear (for instance, "Yeah this terminology … 'Push upstream' …").

In addition, issues in the infrastructure (for example, GitHub) showed gender biases. For instance, one respondent conjectured that Abby "might take a while [navigating this information] because she has comprehensive information processing."

Overall, 92 percent of the issues identified by GenderMag related to previously identified barriers to newcomer entry, including lack of patience; shyness; difficulties in finding a task to start with; getting contributions accepted; and outdated, incomplete, or spread-out documentation. Of these barriers, 73 percent showed gender biases.

So, newcomers who employ the problem-solving strategies common among women are at a disadvantage when those strategies aren't well supported by tools and infrastructure.

This finding might suggest at least a partial explanation for why women are underrepresented in open source communities. If open source (or closed-source) communities build tools that inclusively support the diversity of problem-solving strategies enumerated in GenderMag, those tools can contribute to efforts to recruit and retain more diverse participation in the software industry.

Access this paper at http://bit.ly/PD_2018_Nov_2.

Sentiment and Emotion

Software engineering is an intellectual activity requiring creativity and problem-solving skills, which are influenced by emotion. So, sentiment and emotion in software engineering received significant attention at ICSE 18 and its collocated events. The covered topics included sentiment analysis tools and techniques, applying sentiment and emotion to mobile-app acceptance testing, developers' well-being, assessing automation's impact, and measuring Scrum team meetings' success.

In prior research, applying general-purpose sentiment analysis tools to software engineering produced inconsistent results (both with other tools and with humans). To address this issue, researchers have developed sentiment analysis tools targeting software engineering. One such tool is Senti4SD, which Fabio Calefato and his colleagues describe in "Sentiment Polarity Detection for Software Development."[4] Senti4SD outperformed general-purpose sentiment analysis tools on Stack Overflow data. This result suggests that Senti4SD could detect negative sentiment, such as anger and hostility, toward peers. Because StackOverflow itself recently recognized the need to deal with this issue,[5] such a tool could ultimately make it and similar platforms more welcoming. Access this paper at http://bit.ly/PD_2018_Nov_3.

Regarding the use of emotion in software engineering, two papers at collocated events present interesting results. "Acceptance Testing of Mobile Applications: Automated Emotion Tracking for Large User Groups," by Simon Scherr, Frank Elberzhager, and Konstatin Holl, describes an approach for using emotion to support mobile-app acceptance testing.[6] Their tool translates a user's facial expression into an emoji and provides the emoji to the app developer. This approach lets developers consider real users and real scenarios rather than more-contrived laboratory settings. Access this paper at http://bit.ly/PD_2018_Nov_4.

IEEE SOFTWARE ISSUE ON EMOTIONS IN SOFTWARE ENGINEERING

An upcoming issue of *IEEE Software* will showcase recent advances in research and practice on, and the latest tools for supporting and enhancing, emotion awareness in software development. Besides regular-length articles, we seek short experience reports from practitioners or tool developers sharing practical insights, challenges faced, solutions attempted, and results obtained. Submit your paper by 1 February 2019. For more details, see http://bit.ly/PD_2018_Nov_6.

"Effects of Automated Competency Evaluation on Software Engineers' Emotions and Motivation: A Case Study," by Gul Calikli and her colleagues, discusses using self-reported emotions to assess the automation of developers' competency evaluations at a Swedish software consultancy.[7] The results show that even partial automation has positive effects on developers' and managers' emotions and motivations. Access this paper at http://bit.ly/PD_2018_Nov_5.

Given the increasing understanding of emotions' role in software engineering, *IEEE Software* is planning a theme issue on this topic (see the sidebar).

Safety-Critical Software Product Line Engineering

"Variability Management in Safety-Critical Software Product Line Engineering," by André Luiz de Oliveira and his colleagues, adapts software product line engineering (SPLE) to the development of safety-critical systems.[8] The authors extend traditional SPLE to address dependability and safety issues. During hazard analysis and risk assessment, developers model safety-critical requirements and dependability properties using software variabilities. This approach contributes to

- reducing the cost of developing safety assets for a large number of product variants;
- analyzing the impact of design and usage context variations on properties of the system architecture, behavior, and dependability, by integrating variant management into compositional dependability analysis; and
- modeling component failure by analyzing how variability affects failure propagation across other components in the face of system hazards.

Oliveira and his colleagues illustrate their DEPendable-SPLE approach with a case study of the Tiriba Flight Control (TFC) avionics product line. They identified candidate usage contexts and product variants for consideration during SPL dependability analysis during domain engineering. The results show that context features might influence the selection of product features that change system behavior and affect the dependability properties.

DEPendable-SPLE enabled systematic reuse and automatic generation of design and dependability artifacts. Almost 100 percent of the dependability information produced during domain engineering was reusable. The approach also helped reduce the effort required for dependability analysis (that is, hazard analysis, risk assessment, generation of fault trees, and generation of FMEA [failure mode and effects analysis] artifacts) by 60 to 90 percent for each TFC system variant. Access this paper at http://bit.ly/PD_2018_Nov_7.

References

1. V. Bosotti, "Barriers to Gender Diversity in Software Development Education: Actionable Insights from a Danish Case Study," *Proc. 40th Int'l Conf. Software Eng.: Software Eng. Education and Training* (ICSE-SEET 18), 2018, pp. 146–152.
2. C. Mendez et al., "Open Source Barriers to Entry, Revisited: A Sociotechnical Perspective," *Proc. 40th Int'l Conf. Software Eng.* (ICSE 18), 2018, pp. 1004–1015.
3. M. Burnett et al., "Gender HCI and Microsoft: Highlights from a Longitudinal Study," *Proc. 2017 IEEE Symp. Visual Languages and Human-Centric Computing* (VL/HCC 17), 2017, pp. 139–143.
4. F. Calefato et al., "Sentiment Polarity Detection for Software Development," *Empirical Software Eng.*, vol. 23, no. 3, 2018, pp. 1352–1382.
5. J. Hanlon, "Stack Overflow Isn't Very Welcoming. It's Time for That to Change," blog, 26 Apr. 2018; https://stackoverflow.blog/2018/04/26/stack-overflow-isnt-very-welcoming-its-time-for-that-to-change.
6. S. Scherr, F. Elberzhager, and K. Holl, "Acceptance Testing of Mobile Applications: Automated Emotion Tracking for Large User Groups,"

ABOUT THE AUTHORS

JEFFREY CARVER is a professor in the University of Alabama's Department of Computer Science. Contact him at carver@cs.ua.edu.

ALEXANDER SEREBRENIK is an associate professor in Eindhoven University of Technology's Department of Mathematics and Computer Science. Contact him at a.serebrenik@tue.nl.

RAFAEL CAPILLA is an associate professor of computer science at Rey Juan Carlos University of Madrid. Contact him at rafael .capilla@urjc.es.

ALEJANDRO VALDEZATE is a software engineer at Ibermática and a PhD candidate at Rey Juan Carlos University of Madrid. Contact him at alejandro@valdezate.net.

BIRGIT PENZENSTADLER is an assistant professor of software engineering at California State University, Long Beach. Contact her at birgit.penzenstadler@csulb.edu.

Proc. 5th Int'l Conf. Mobile Software Eng. and Systems (MOBILESoft 18), 2018, pp. 247–251.

7. G. Calikli et al., "Effects of Automated Competency Evaluation on Software Engineers' Emotions and Motivation: A Case Study," *Proc. 3rd Int'l Workshop Emotion Awareness in Software Eng.* (SEmotion 18), 2018, pp. 44–50.
8. A.L. de Oliveira et al., "Variability Management in Safety-Critical Software Product Line Engineering," *New Opportunities for Software Reuse*, LNCS 10826, Springer, 2018, pp. 3–22.