

# Behavioral Science and Diversity in Software Engineering

**Citation for published version (APA):**

Carver, J. C., Muccini, H., Penzenstadler, B., Prikladnicki, R., Serebrenik, A., & Zimmermann, T. (2021). Behavioral Science and Diversity in Software Engineering. *IEEE Software*, 38(2), 107-112. Article 9354400. <https://doi.org/10.1109/MS.2020.3042683>

**Document license:**

TAVERNE

**DOI:**

[10.1109/MS.2020.3042683](https://doi.org/10.1109/MS.2020.3042683)

**Document status and date:**

Published: 01/03/2021

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.



# Behavioral Science and Diversity in Software Engineering

Jeffrey C. Carver, Henry Muccini, Birgit Penzenstadler, Rafael Prikladnicki, Alexander Serebrenik, and Thomas Zimmermann

**THE “PRACTITIONERS’ DIGEST”** department in this issue of *IEEE Software* covers two topics: the behavioral science of software engineering and diversity in software engineering (this issue’s theme) and includes papers from the 42nd International Conference on Software Engineering (ICSE20), 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME19), 13th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE20), Empirical Software Engineering and Measurement 2020 (ESEM20), and Association for Computing Machinery Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE20). Feedback or suggestions are welcome. In addition, if you try or adopt any of the practices included in this article, please send me and the authors of the paper(s) a note about your experiences.

## Developer Emotions and Programming

“Recognizing Developers’ Emotions While Programming” by Girardi et al.<sup>1</sup>

describes the results of a study on the range of developers’ emotions during a programming task, correlation of those emotions with perceived progress, triggers for developers’ emotions, and strategies developers implement to deal with negative emotions. To study this question, the authors had 27 students write a small Java program using the Stack Exchange application programming interface while wearing the NeuroSky BrainLink headset to record electroencephalographic (EEG) waves and the Empatica E4 wristband for electrical activity of the skin, blood volume pulse, and heart-related metrics.

During the 30-min programming task, the researchers interrupted the students every 5 min, asking them to report their emotions, the reasons for their emotions, and their perceived progress. The authors observed a wide range of emotions during the activity; most of them were negative. In addition, the majority of the emotions were intense (“high arousal”). The researchers also observed that the valence of the emotion (i.e., whether it was positive or negative) was positively correlated with perceived progress on the task.

Students were more likely to experience positive emotions when they thought they were progressing well. Indeed, when asked about the triggers of their emotions, students referred to the effectiveness of the implemented solution. Students reported positive emotions when they perceived the solution as being simple or upon finding an incremental approach toward the solution. Conversely, they reported negative emotions when encountering unexpected output, unforeseen usage of libraries, and unavailable documentation. Also, time pressure and self-perceived low productivity triggered negative emotions.

These results have potential implications for practice. Because the researchers were able to gather reliable measurements of developers’ emotions using only a noninvasive sensor like the Empatica E4 wristband, there are multiple opportunities to integrate biometric measurements into the daily routine of developers. The authors envision applications including a just-in-time suggestion of corrective actions when negative emotions are detected, integration of daily emotional records into retrospective meetings at the

Digital Object Identifier 10.1109/MS.2020.3042683  
Date of current version: 11 February 2021

end of an iteration or sprint, prevention of burnout, and improvement of techniques aiming at preventing interruptions of developers when they are “in flow.” This paper appears in the main research track of ICSE20. Access it at <http://bit.ly/PD-2021-March-01>.

### Cognitive Biases

“A Tale From the Trenches: Cognitive Biases and Software Development” by Chattopadhyay et al.<sup>2</sup> studies the effects of cognitive biases on professional software developers. Cognitive biases are behaviors that influence developer actions, which may set them on an incorrect course and require backtracking. An example of a cognitive bias negatively impacting software development is ownership bias, where a developer might choose to reuse his or her own work rather than using a solution designed elsewhere. To understand the effects of cognitive biases, the authors observed 10 developers during their normal daily routine by recording the screen, physical workspace, and audio from participants’ verbalizations of their thoughts. The authors then conducted follow-up interviews to confirm their analysis of the collected data.

Approximately half of the developer actions could be associated with biases. The developers reverted almost 80% of those actions by undoing, re-doing, or discarding them. These reversions required a significant amount of time. The most frequently occurring biases related to fixation—i.e., participants “fixated” on a specific solution, rejecting warnings and errors that contradicted their belief. The second-most frequent bias—convenience bias—was associated with developers choosing the most convenient solution rather than the optimal one.

The biases affected multiple aspects of problem solving, including how adequately developers explore the solution space, how thoroughly they engage in sensemaking, how effectively they retain context, and how efficiently they invest their attention. The biases led the authors to propose a series of developer tool suggestions that can reduce the negative impact of cognitive biases.

For example, an IDE could track developer actions and detect situations where a developer is “fixated.” As one of the interview participants suggested, “If you change [code] and you always get the same error, it can say, hey, you have been doing the same thing five times. But you always get the same error—maybe try something different?” Similarly, a tool that can identify suboptimal code changes and recommend “clean” or “nonsmelly” code might warn a developer opting for the most convenient solution over an optimal one.

The paper also describes a series of practices developers can employ to reduce the negative effects of cognitive biases. It appears in the main research track of ICSE’20. Access it at <http://bit.ly/PD-2021-March-02>.

### Neurology and Code Writing

“Neurological Divide: An fMRI Study of Prose and Code Writing” by Krueger et al.<sup>3</sup> used functional magnetic resonance imaging (fMRI) to understand the similarities and differences in brain activity between writing code and writing prose. This work was motivated, at least in part, by Edgar Dijkstra’s 1975 statement that “besides a mathematical inclination, an exceptionally good mastery of one’s native tongue is the most vital asset of a competent programmer.”<sup>4</sup> This statement suggests a relationship between code writing

and prose writing. However, is that relationship really present? Krueger et al. focused on two questions:

1. Are code writing and prose writing similar neural activities?
2. Is being good at writing actually associated with being a good software developer?

The authors found that, overall, code writing and prose writing operate via distinct neural mechanisms (i.e., the brain treats these activities differently). More specifically, code writing engages the right brain hemisphere, which involves spatial ability and planning, while prose writing engages the left brain hemisphere, which involves language production.

While it is unlikely that an fMRI device will be used any time during actual software development (due to size and cost), the findings do have important implications for practice. First, the paper demonstrates the potential of using fMRI and similar technologies to test assumptions and hypotheses made about software development, which will ultimately help to design better software tools and improve software practices. Software companies already use EEG devices to monitor changes in brainwaves to build better software features. The advantage of fMRIs compared to EEGs is that they offer high spatial resolution, that is, where exactly the brain activity is happening (in contrast, EEGs have higher temporal resolution).

Second, a deeper understanding of how the brain works when writing code at the neurological level will help to more effectively educate and train the next generation of software engineers. The paper also provides empirical evidence that code and prose writing are not as intertwined

as previously thought and may encourage more diverse participation in computer science. This paper appears in the main research track of ICSE20. Access it at <http://bit.ly/PD-2021-March-03>.

### Measuring Emotion

“EmoD: An End-to-End Approach for Investigating Emotion Dynamics in Software Development” by Neupane et al.<sup>5</sup> describes EmoD, a systematic approach to identify, quantify, model, store, and present emotion dynamics. EmoD automatically collects project communication records, identifies the emotions, calculates their intensities, and presents the results in a time-series visualization. EmoD uses five steps to collect issues and pull requests from GitHub (data collection), preprocess the data (to deal with stop words, special characters, and tokenization), classify emotions into four categories (anger, fear, joy, and sadness), calculate the intensity of the emotions, and store them in a time-series database.

The EmoD approach is automated in a tool that implements these five steps by collecting data from GitHub, using the Natural Language Toolkit preprocessing library for Python, employing the tool-EmoTXT emotion detection tool to classify comments into emotions, utilizing a lexicon-based novel method based on NRC to identify the emotion intensities, and storing the computed emotion intensities into InfluxDB, which stores time-stamped data sets. The paper provides a running example along with a sample of the emotion visualization.

Overall, EmoD provides an automated approach, using data in a GitHub repository, that practitioners can use to visualize the emotional dynamics of each team member.

This paper appears in the short paper track of ICSME19. Access it at <http://bit.ly/PD-2021-March-04>.

### Gender Inclusivity

“Engineering Gender Inclusivity Into Software: Ten Teams’ Tales From the Trenches” by Hilderbrand et al.<sup>6</sup> describes how to use GenderMag, an evaluation method to find and fix gender inclusivity bugs and systematize the practice within software teams. GenderMag uses personas that represent a range of users built upon five cognitive facets [motivations, computer self-efficacy, risk averseness, information processing style, and tech learning style (tinkering versus process-oriented learning)].

The authors used action research in a longitudinal study in collaboration with 10 teams, five from academic IT and five from industry. They collected data from the teams (over 9 months to 3.5 years, depending on the team) using GenderMag sessions, interviews, design meeting notes, and emails. The authors present the teams’ experiences “in the trenches,” in the form of 12 practices and three potential pitfalls, to provide their insights to other real-world software teams trying to engineer gender inclusivity into their software products.

Overall, the teams found that GenderMag improves inclusivity because teams were trying to go beyond just making their software work to having it work equally well for different genders. This paper appears in the main research track of ICSE20. Access it at <http://bit.ly/PD-2021-March-05>.

### Female Participation in Hackathons

“A Preliminary Study About the Low Engagement of Female Participation in Hackathons” by Paga-

nini and Gama<sup>7</sup> studies why so few women participate in hackathons, the motivations for participating in hackathons, and the main problems related to gender in hackathons. The paper describes the results from interviews of four women and one man who participated in a female-focused hackathon organized by female undergraduate students in June 2019 in Recife, Brazil.

The interviewees indicated that while women are motivated to join hackathons by a desire to learn, network, and gain experience, they are deterred by low self-esteem, being in the minority, and toxic masculinity. Women who do join hackathons experience man interrupting, mansplaining, underestimation, verbal harassment, and inappropriate comments. Despite these negative experiences, the interviewees also reported many positive aspects of participation in women-friendly hackathons, such as mentoring by other women, feelings of inspiration and solidarity, or sisterhood.

Companies that organize hackathons might consider some of these findings to increase participation in their hackathons (e.g., mentoring by women) and reduce instances of negative interactions (e.g., harassment). Based on the insights of the interview, the authors intend to design and promote better hackathon environments for women to join. This paper appears in CHASE20. Access it at <http://bit.ly/PD-2021-March-06>.

### Women in Open Source Software

“Work Practices and Perceptions From Women Core Developers in OSS Communities” by Canedo et al.<sup>8</sup> investigates how common female core developers are in open source software (OSS), whether their

work practices differ from male core developers, and how they perceive gender bias. The authors mined OSS repositories to identify core developers and their gender and then surveyed the identified female core developers. They found that only 2.3% of open source core developers are women. (Of all contributors, 5.35% are women.)

To conduct their investigation, the authors surveyed the female core developers. First, they found no significant difference between the work practices of female and male core developers. However, the results showed that gender bias does occur among core developers, with 34% of female core developers surveyed reporting they had observed gender bias at least once while contributing to OSS projects. On a positive note, the majority (66%) of female core developers reported they had not ever experienced gender discrimination when contributing to OSS projects.

To help remedy gender bias problems, the authors asked survey respondents about ideas that would help increase women's participation in open source communities. Respondents' suggestions include the following:

- Promote women-specific mentorship programs.
- Promote women to senior roles: some communities, e.g., Open Stack, already implement this.
- Organize women-specific events, such as local meetups or even tech groups, such as, e.g., the R ladies group or the Pyladies group.
- Avoid gendered language (e.g., using "guys" when "folks" would work).

These suggestions build women's confidence to contribute to OSS communities and might make OSS more welcoming to individuals of other minority groups as well. According to survey participants, gender diversity might contribute to improving communication among team members and help generate different ideas while designing a software product. This paper appears in ESEM2020. Access it at <http://bit.ly/PD-2021-March-07>.

### Reducing Gender Biases

"Reducing Implicit Gender Biases in Software Development: Does Intergroup Contact Theory Work?" by Wang and Zhang<sup>9</sup> investigates whether intergroup contact theory, a social psychology approach, is helpful in reducing gender bias within a set of students working on software development teams in a university course. The basic idea of intergroup contact theory is that increasing interactions among different groups of people can help reduce bias if certain conditions are met, including equal status, intergroup cooperation, common goals, and institutional support.

To test this theory, the researchers compared the change in gender bias among students who participated in teams of four that either 1) were all of one gender, 2) had an equal number of male and female participants, or 3) had a majority of one gender. The study found that intergroup contact theory reduced both implicit and explicit bias. This effect was maximized in the teams where the number of females was greater than or equal to the number of males.

These findings suggest some practical recommendations for software organizations that wish to reduce gender bias. First, it is important to

satisfy all of the conditions listed. Second, using mixed-gender teams can help reduce bias as long as the females are not in the minority on the teams. This paper appears in ESEC/FSE20. Access it at <http://bit.ly/PD-2021-March-08>.

### Software Developer Diversity and Inclusion Workshop

The second edition of the Software Developer Diversity and Inclusion workshop (SDDI) sought to raise awareness about SDDI challenges faced by industry, brainstorm concrete goals to address those challenges, and gather recommendations and best practices to share with developers. One of the key takeaways from the workshop is that many open source projects and software companies recognize the value of diversity and inclusion but are unsure where they should make investments. At the same time, researchers are beginning to gather a body of knowledge about software developer diversity and inclusion, but they need deeper collaborations with industry and open source professionals to increase the impact of that research.

For this reason, a group of workshop participants worked together, in collaboration with the Linux Foundation, to create the SDDI project to discover, evaluate, and promote best practices from research and industry to increase diversity and inclusion in software engineering (<https://sddiproject.org/>). A press release about the project can be accessed at <http://bit.ly/PD-2021-March-09>. The workshop report can be accessed at <http://bit.ly/PD-2021-March-10>. All presentations are available online at <http://bit.ly/PD-2021-March-11>. 🌐



**JEFFREY C. CARVER** is a professor in the University of Alabama's Department of Computer Science, Tuscaloosa, Alabama, 35487, USA. Further information about him can be found at <http://carver.cs.ua.edu>. Contact him at [carver@cs.ua.edu](mailto:carver@cs.ua.edu).



**RAFAEL PRIKLADNICKI** is an associate professor at the School of Technology and one of the managers of the Science and Technology Park (Tecnopuc) at the Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, RS 90619-900, Brazil, where he also leads the MuNDDoS research group. He was the chair of *IEEE Software's* advisory board from 2017 to 2020. Further information about him can be found at <https://www.pucrs.br/researchers/rafael-prikladnicki/>. Contact him at [rafaelp@pucrs.br](mailto:rafaelp@pucrs.br)



**HENRY MUCCINI** is a professor with the Department of Information Engineering, Computer Science, and Mathematics, University of L'Aquila, L'Aquila, 67100, Italy. Further information about him can be found at <http://www.henrymuccini.com/>. Contact him at [henry.muccini@univaq.it](mailto:henry.muccini@univaq.it).



**ALEXANDER SEREBRENIK** is a professor at Eindhoven University of Technology, Eindhoven, 5600MB, The Netherlands. Further information about him can be found at <https://www.win.tue.nl/~aserebre/>. Contact him at [a.serebrenik@tue.nl](mailto:a.serebrenik@tue.nl).



**BIRGIT PENZENSTADLER** is an assistant professor at Chalmers University of Technology and the University of Gothenburg, Gothenburg, 412 96, Sweden. Further information about her can be found at <https://www.chalmers.se/en/Staff/Pages/birgit.aspx>. Contact her at [birgitp@chalmers.se](mailto:birgitp@chalmers.se).



**THOMAS ZIMMERMANN** is a senior principal researcher at Microsoft, Redmond, Washington 98052, USA. Further information about him can be found at <http://thomas-zimmermann.com>. Contact him at [tzimmer@microsoft.com](mailto:tzimmer@microsoft.com).

## References

1. D. Girardi, N. Novielli, D. Fucci, and F. Lanubile, "Recognizing developers' emotions while programming," in *Proc. ACM/IEEE 42nd Int. Conf. Softw. Eng. (ICSE '20)*, 2020, pp. 666–677. [Online]. Available: <http://bit.ly/PD-2021-March-01>. doi: 10.1145/3377811.3380374.
2. S. Chattopadhyay et al., "A tale from the trenches: cognitive biases and software development," in *Proc. ACM/IEEE 42nd Int. Conf. Softw. Eng.*, 2020, pp. 654–665. [Online]. Available: <http://bit.ly/PD-2021-March-02>. doi: 10.1145/3377811.3380330.
3. R. Krueger, Y. Huang, X. Liu, T. Santander, W. Weimer, and K. Leach, "Neurological divide: an fMRI study of prose and code writing," in *Proc. ACM/IEEE 42nd Int. Conf. Softw. Eng.*, 2020, pp. 678–690. [Online]. Available: <http://bit.ly/PD-2021-March-03>. doi: 10.1145/3377811.3380348.
4. E. W. Dijkstra, *Selected Writings on Computing: A Personal Perspective*. New York: Springer-Verlag, 1982.
5. K. P. Neupane, K. Cheung, and Y. Wang, "EmoD: An end-to-end approach for investigating emotion dynamics in software development," in *Proc. 2019 IEEE Int. Conf. Softw. Maintenance Evol. (IC-SME)*, Cleveland, OH, pp. 252–256. [Online]. Available: <http://bit.ly/PD-2021-March-04>. doi: 10.1109/ICSME.2019.00038.
6. C. Hilderbrand et al., "Engineering gender-inclusivity into software: Ten teams' tales from the trenches," in *Proc. ACM/IEEE 42nd Int. Conf. Softw. Eng.*, 2020,

pp. 433–444. [Online]. Available: <http://bit.ly/PD-2021-March-05>. doi: 10.1145/3377811.3380371.

7. L. Paganini and K. Gama, “A preliminary study about the low engagement of female participation in hackathons,” in *Proc. IEEE/ACM 42nd Int. Conf. Softw. Eng. Workshops*, 2020, pp. 193–194. [Online]. Available: <http://bit.ly/PD-2021-March-06>. doi: 10.1145/3387940.3392163.

8. E. D. Canedo, R. Bonifácio, M. V. Okimoto, A. Serebrenik, G. Pinto,

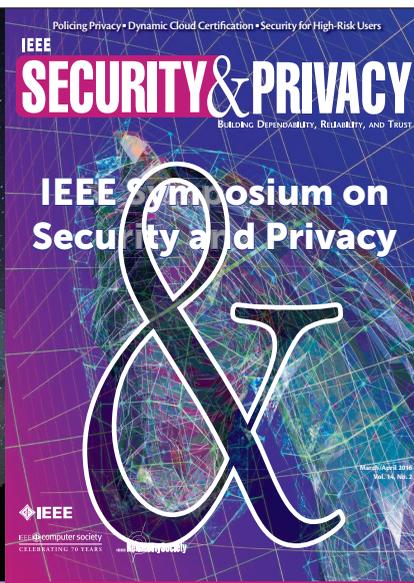
and E. Monteiro, “Work practices and perceptions from women core developers in OSS communities,” in *Proc. 14th ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, 2020, pp. 1–10. [Online]. Available: <http://bit.ly/PD-2021-March-07>. doi: 10.1145/3382494.3410682.

9. Y. Wang and M. Zhang, “Reducing implicit gender biases in software development: does intergroup contact theory work?” in *Proc. 28th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2020,

pp. 580–592. [Online]. Available: <http://bit.ly/PD-2021-March-08>. doi: 10.1145/3368089.3409762.



Access all your IEEE Computer Society subscriptions at [computer.org/mysubscriptions](http://computer.org/mysubscriptions)



**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](http://computer.org/security)



Digital Object Identifier 10.1109/MS.2021.3051540