Advancing Cognitive Inclusivity in Software Engineering Tools and Practices

Faith Culas
University of Auckland
Auckland, New Zealand
fcul804@aucklanduni.ac.nz

***Abstract*—Software engineering is a discipline that relies on various tools and best practices to support the creation and maintenance of complex software systems. It is also a discipline that demands a significant amount of cognitive processing while using these tools and practices, and it is often challenging to grasp what's happening inside the mind of a software engineer. Recent research has found evidence of biases within software development tools and practices, which favour only certain cognitive styles and create additional overhead or challenges for people with different cognitive styles. Yet, very few software engineering tools and practices have been studied through the lens of cognitive style. This study will investigate how individuals with different cognitive styles interact with a variety of software tools and practices through empirical research. It will contribute to analysing different cognitive styles and processes employed in software engineering, identifying potential biases in software tools and practices and paving the way to mitigate them.**

***Index Terms*—cognition, tools, practices, diversity, inclusion**

I. Introduction

Software engineering (SE) emerged as a term in the 1960's due to a realisation that systematic processes were needed to cope with the complexity of software systems. Since then, the field of SE has advanced significantly with many tools and practices developed and accepted to help software engineers design, develop, test, maintain, and deliver software applications. However, recent research has identified biases in some popular SE tools, for example, GitHub, Code review tools and Stack Overflow [1]–[5]. The studies found that these tools were not well suited to all ways of thinking. Research from the field of psychology have shown that people have diverse ways of thinking, known as cognitive styles. Cognition is the "collection of mental processes and activities used in perceiving, remembering, thinking, and understanding, and the act of using those processes" [6]. The findings that some SE tools are biased towards certain cognitive styles highlight the fact that many SE tools and best practices, while widely accepted and utilised, are not always supported by sufficient empirical evidence. Ordoñez-Pacheco et al. [7] argued that many practices are accepted because they have been successful in specific contexts, but that this success does not necessarily imply they are universal.

Research in psychology also indicates that cognition often clusters by gender [8]. If cognitive biases exist in SE tools and practices, this can also imply gender biases exist. This highlights the need to identify and address the underlying factors that cause such disparities, to make these essential tools and practices more diverse and inclusive. This also raises the question of whether the additional cognitive overhead required for people with diverse cognitive styles can contribute to the low diversity in the industry[1].

GenderMag [4], a well-researched method to identify biases in software, presents five different cognitive styles: *Self-efficacy, Information processing style, Learning style, Attitude towards Risk and Motivation*, that help to analyse individual cognitive differences, which also cluster by gender [9], [10]. For instance, when it comes to learning style, men reportedly tinker more frequently than women. GenderMag has been proven effective in understanding cognitive styles and identifying biases in SE tools and practices [11]. Although GenderMag has been applied to some software tools to understand different user interactions with the software, it is yet to be applied to many other software tools and practices.

This research will focus on popular SE tools and practices to examine biases across cognitive styles. Based on the findings, new designs of common SE tools will be conceptualised and validated through further studies to examine if the suggested new designs improve inclusivity.

The remainder of this paper is divided into five sections: Section II highlights related work. In Section III, the expected contributions are presented. In Section IV, the approach is discussed. Section V concludes the paper.

II. Related work

Although there has been significant research conducted on some software tools and practices that considered factors like experience, only recently has gender and cognitive diversity been considered in such experiments. This section covers some of the research done on some popular software tools and practices.

Research on cognition in software tools has been explored to see how users interact with tools such as GitHub [5], Stack Overflow [3], Visual Studio [4], and various custom tools like Google's internal code review tool [2] and a debugging tool [12]. These studies revealed insights into cognitive processes involved in software development tasks. Some key findings were 73% of the barriers in GitHub had some form of gender

[1]https://www.accenture.com/us-en/about/corporate-citizenship/tech-culture-reset

bias and only 5.8% of the contributors are women in Stack Overflow. Additionally, women reported lower confidence than men, and some tinkering preferences did seem to be tied to confidence levels. Burnett et al. [4] emphasized on the importance of considering gender differences and thus cognitive differences when designing tools, but argued that such changes do not have to favour one gender at the expense of another. An effective example of achieving this is demonstrated by the work of Murphy-Hill et al. [2]. They found that the redesigned edit feature in the Google internal code review tool improved its discoverability by 2.4 times for women, removing a key barrier and enhancing discoverability for both men and women overall.

Most studies examining cognitive style in SE have focused on tools, though some have looked at software practices more generally. For example, a study at Google on the code reviewing process resulted in some recommendations to improve developer satisfaction by using techniques like customization [13].

There have also been research on other tools which are not directly involved with SE like Excel [14] and Microsoft search engine [9], which found barriers for certain cognitive styles. Research has also found that novices using tools like GitHub [5] and Debugger [12] face greater difficulty than experienced software engineers, revealing barriers to newcomers. So, while some biases have been found in some of the existing tools, there are many tools yet to be explored from a cognitive perspective to achieve cognitive parity. This prior work calls for more attention on the inclusivity of software tools and practices.

III. Expected Contributions

This PhD thesis aims to examine cognitive biases in software tools and practices in more detail and aims to make several contributions to the field of SE:

A. Improvements to existing tools and practices

The study aims to discover biases in some of the popular SE tools and practices. The identified biases will be fixed to evaluate the effectiveness of the proposed fixes. Based on the study's findings, recommendations will be suggested for SE teams to improve practices [e.g. code reviews, debugging, collaboration] and software tools [e.g. Integrated development environment tools, debugging tools]. These recommendations aim to support both productivity and inclusivity in teams.

B. Insight into different cognitive styles of software engineers

By analysing how software engineers interact with different tools and practices, this research will provide a deeper understanding of the cognitive processes employed by individuals with different cognitive styles. This insight can help design more tools and practices that align with diverse cognitive styles.

C. Call for action on existing metrics

Different metrics such LoC (lines of code), time taken, number of correct answers, explanation of code in plain English, eye-tracking data have been used to evaluate performance of software engineers. While literature has shown there are many different ways to conduct research and measure performance, we are interested in using a key aspect which is constantly overlooked which is the individual differences in cognitive styles along with existing metrics. For example, while its widely accepted that smaller functions improve code comprehensibility and is usually used as a metric to measure code quality, the fact that cognitive styles could influence code comprehension strategies hasn't been explored yet.

IV. Approach

There will be a series of quantitative and qualitative research over the course of two years, in the form of surveys and experiments with SE students and professionals to understand their cognitive styles and their interaction with different tools and practices. The first two studies currently being conducted are on a debugging tool and code comprehension. The studies have been designed to understand the cognitive style of the participants using the GenderMag questionnaire[2] and also ensure participant diversity in terms of the five facets and gender. Think-aloud experiments are in progress to analyse the interaction between the participant and the tool or practice. The data gathered will be analysed across the five facets to find biases and recommend improvements. The recommended improvements will be validated through further lab experiments. Furthermore, more insight on different cognitive processes and differences in metrics that can be introduced to include all cognitive styles will result through these experiments.

V. Conclusion

In conclusion, SE is a field that has seen substantial advancement over the years and continues to evolve, bringing further developments in the tools and practices that support it. This study will explore the biases in some selected existing software tools and practices to recommend suggestions to mitigate the cognitive biases and also lay the foundation to create more inclusive tools and practices. Furthermore, our research highlights the need for continuous evaluation and adaptation of SE tools and practices to ensure they meet the needs of all users with diverse cognitive styles.

References

[1] C. Sadowski, E. Söderberg, L. Church, M. Sipko, and A. Bacchelli, "Modern code review: A case study at google," *2018 Ieee/Acm 40th International Conference on Software Engineering - Software Engineering in Practice Track (Icse-Seip 2018)*, pp. 181–190, 2018.

[2] E. Murphy-Hill, A. Elizondo, A. Murillo, M. Harbach, B. Vasilescu, D. Carlson, and F. Dessloch, "Gendermag improves discoverability in the field, especially for women," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ACM, 2024.

[2] https://gendermag.org

[3] D. Ford, J. Smith, P. J. Guo, and C. Parnin, “Paradise unplugged: Identifying barriers for female participation on stack overflow,” *Fse’16: Proceedings of the 2016 24th Acm Sigsoft International Symposium on Foundations of Software Engineering*, pp. 846–857, 2016.
[4] M. Burnett, S. D. Fleming, S. Iqbal, G. Venolia, V. Rajaram, U. Farooq, V. Grigoreanu, and M. Czerwinski, “Gender differences and programming environments,” in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ACM, 2010.
[5] H. S. . S.-H. Z. . H. C. . H. A. . H. C. . S. L. . P. N. . S. A. . B. M. Mendez, Christopher ; Padala, “Open source barriers to entry, revisited: A tools perspective,” in *Proceedings of ACM ICSE conference, Sweden, May 2018 (ICSE’2018)*, 2018.
[6] M. H. Ashcraft, *Cognition*. Prentice Hall, 2002.
[7] R. Ordoñez-Pacheco, K. Cortes-Verdin, and J. O. Ocharán-Hernández, “Best practices for software development: A systematic literature review,” in *Advances in Intelligent Systems and Computing*, Advances in intelligent systems and computing, pp. 38–55, Cham: Springer International Publishing, 2021.
[8] J. S. Hyde and N. M. McKinley, “Gender differences in cognition,” *Gender differences in human cognition*, pp. 30–51, 1997.
[9] M. Vorvoreanu, L. Y. Zhang, Y. H. Huang, C. Hilderbrand, Z. Steine-Hanson, and M. Burnett, “From gender biases to gender-inclusive design: An empirical investigation,” *Chi 2019: Proceedings of the 2019 Chi Conference on Human Factors in Computing Systems*, 2019. Bn1he Times Cited:51 Cited References Count:63.
[10] A. Anderson, J. Noa Guevara, F. Moussaoui, T. Li, M. Vorvoreanu, and M. Burnett, “Measuring user experience inclusivity in Human-AI interaction via five user problem-solving styles,” *ACM Trans. Interact. Intell. Syst.*, May 2024.
[11] M. Burnett, S. Stumpf, J. Macbeth, S. Makri, L. Beckwith, I. Kwan, A. Peters, and W. Jernigan, “Gendermag: A method for evaluating software’s gender inclusiveness,” *Interacting with Computers*, vol. 28, p. 760–787, Nov. 2016.
[12] M. Hassan, G. Zeng, and C. Zilles, “Evaluating how novices utilize debuggers and code execution to understand code,” in *Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 1*, vol. 5, pp. 65–83, ACM, 2024.
[13] C. Sadowski, E. Söderberg, L. Church, M. Sipko, and A. Bacchelli, “Modern code review: A case study at google,” *2018 Ieee/Acm 40th International Conference on Software Engineering - Software Engineering in Practice Track (Icse-Seip 2018)*, pp. 181–190, 2018.
[14] M. M. Burnett, L. Beckwith, S. Wiedenbeck, S. D. Fleming, J. Cao, T. H. Park, V. Grigoreanu, and K. Rector, “Gender pluralism in problem-solving software,” *Interacting with Computers*, vol. 23, no. 5, pp. 450–460, 2011.