A Comparison of Program Comprehension Strategies by Blind and Sighted Programmers

Ameer Armaly, Paige Rodeghero, and Collin McMillan
University of Notre Dame
Notre Dame, IN
[aarmaly,prodeghe,cmc]@nd.edu

CCS CONCEPTS

• **Software and its engineering → Software maintenance tools**;

KEYWORDS

program comprehension, accessibility, visual impairments

ACM Reference Format:
Ameer Armaly, Paige Rodeghero, and Collin McMillan. 2018. A Comparison of Program Comprehension Strategies by Blind and Sighted Programmers. In *Proceedings of ICSE '18: 40th International Conference on Software Engineering (ICSE '18).* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3180155.3182544

Programmers who are blind use a screen reader to speak source code one word at a time, as though the code were text. For example, "float f = 5.23;" can be read as "float f equals five point two three semicolon". This process of reading is in stark contrast to sighted programmers, who skim source code rapidly with their eyes. At present, it is not known whether the difference in these processes has effects on the program comprehension gained from reading code. These effects are important because they could reduce both the usefulness of accessibility tools and the generalizability of software engineering studies to persons with low vision. Furthermore, a lack of knowledge about blind programmers contributes to a bias against employing blind programmers. Employers are unfamiliar with the idea of a blind programmer and as a result may feel unsure about hiring one.

In this paper [1], we present an empirical study comparing the program comprehension strategies of blind and sighted programmers. We recruited twelve blind programmers to read and summarize a series of Java methods. We tracked their cursor using a screen reader extension and compared the results to a previous eye-tracking study [2]. The eye-tracking study used the same java methods which were taken from a variety of different application domains. Both studies examined three code areas: method signatures, method invocations and control flow statements. These areas have previously been suggested as being important to code comprehension. The eye-tracking study measured three metrics: gaze time, fixations and regressions. Our study measured reads and regressions. All metrics were adjusted to take into account the size of the code area relative to the size of the method. We found that both blind and sighted programmers prioritize reading method signatures over other areas of code. Blind programmers also returned to method signatures more often than the rest of the method. This finding emphasizes the importance of readable method signatures to the process of code comprehension and suggests that blind and sighted programmers use very similar strategies when reading code.

We then recruited independent experts to compare the summaries produced by the blind and sighted groups. The goal was to measure whether there was any difference between the degree of comprehension on the part of blind and sighted programmers. The experts took an online survey that presented them with a method and a summary for that method. The experts were not told whether the summary had been produced by a blind or sighted programmer. The experts answered five questions meant to rate the quality of the summary. Possible scores ranged from "strongly disagree" to "strongly agree." There were no differences between the scores given to the blind group versus the sighted group. Both groups of programmers obtained an equally high degree of comprehension.

Our conclusions are that 1) blind programmers employ similar code comprehension strategies as sighted programmers, and 2) blind programmers comprehend code to the same degree as sighted programmers. We hope that these findings will contribute to an increased familiarity with and understanding of blind programmers on the part of potential employers. We also hope that these findings will lay the foundation for further investigation into potential differences between blind and sighted programmers.

ICSE '18, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5638-1/18/05…$15.00
https://doi.org/10.1145/3180155.3182544

ACKNOWLEDGMENTS

The authors would like to thank the several participants in our user studies for their careful attention to detail. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1313583. This work is supported in part by the NSF CCF-1452959 and CNS-1510329 grants. Any opinions, findings, and conclusions expressed herein are the authors' and do not necessarily reflect those of the sponsors.

REFERENCES

[1] Ameer Armaly, Paige Rodeghero, and Collin McMillan. 2017. A Comparison of Program Comprehension Strategies by Blind and Sighted Programmers. *IEEE Transactions on Software Engineering* (2017). https://doi.org/10.1109/TSE.2017.2729548

[2] Paige Rodeghero, Collin McMillan, Paul W. McBurney, Nigel Bosch, and Sidney D'Mello. 2014. Improving Automated Source Code Summarization via an Eye-Tracking Study of Programmers. In *Proceedings of the 36th international conference on Software engineering (ICSE '14).* 12.