



| SUPERVISOR INFORMATION | |
|--|--|
| First and Last name | José Campos |
| URL of supervisor webpage | https://jose.github.io |
| Department | Department of Informatics Engineering |
| Field(s) of research | Software Engineering, Software Testing & Debugging, Fault Localization |
| PROJECT PROPOSAL | |
| Title (optional) | GZoltar 2.0: A Framework for Automatic Debugging |
| Brief project description | |
| <p>The continued use of traditional debugging techniques (e.g., prints and breakpoints) has contributed to the high cost of software failures [1]. To mitigate this, researchers have proposed automatic techniques to help software developers identify the location of faults in software programs and ultimately repair them [2]. The most successful technique to date is Spectrum-based Fault Localization (SBFL) [3]. SBFL relies on the execution behavior (“spectrum”) of the program’s test cases and their outcome (i.e., pass or fail). Given a spectrum, SBFL uses statistical methods [4] to compute the likelihood of each code component (e.g., statement) being the truly faulty one. Components are then ranked by their likelihood, i.e., suspiciousness score, and the ones with this highest value are more likely to be the ones responsible for the software failure.</p> <p>Several tools and plug-ins that use SBFL have been proposed for Java, e.g., Vida, Falcon, GZoltar [5], Jaguar, iFL4Eclipse, and recently FLACOCO [10]. These tools have been successfully used in several research studies (e.g., [11, 12, 13]), and some have been integrated into other tools, for example, both GZoltar and FLACOCO have been integrated into the automatic program repair tool ASTOR [14]. Despite their success, neither GZoltar nor FLACOCO supports the latest advances in automatic fault localization (e.g., dynamic slicing [15] or the usage of stack traces from bug reports in the absence of failing tests [16]).</p> <p>Thus, in this project, we aim to (1) gather the most recent and successful approaches or simple improvements made to SBFL, (2) integrate those into [GZoltar](https://github.com/GZoltar/gzoltar), the most popular and used tool in empirical studies, and (3) investigate which combination of approaches works best. In a nutshell, we aim to develop a framework that allows others to integrate and evaluate new fault localization approaches easily. The framework will be available online as an open-source project.</p> | |



[1] Glenford J. Myers, Corey Sandler, and Tom Badgett. The Art of Software Testing. 3rd. Wiley Publishing, 2011. ISBN: 1118031962, 9781118031964.

[2] Martin Monperrus. The Living Review on Automated Program Repair. Tech. rep. hal-01956501. HAL/archives-ouvertes.fr, 2018.

[3] W. Eric Wong, Ruizhi Gao, Yihao Li, Rui Abreu, and Franz Wotawa. "A Survey on Software Fault Localization". In: IEEE Transactions on Software Engineering 42.8 (2016), pp. 707–740. DOI: 10.1109/TSE. 2016.2521368.

[4] Rui Abreu, Peter Zoetewij, and Arjan J.c. Van Gemund. "An Evaluation of Similarity Coefficients for Software Fault Localization". In: 2006 12th Pacific Rim International Symposium on Dependable Computing (PRDC'06). 2006, pp. 39–46. DOI: 10.1109/PRDC.2006.18.

[5] José Campos, André Ribeiro, Alexandre Perez, and Rui Abreu. "GZoltar: an eclipse plug-in for testing and debugging". In: Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. ASE '12. Essen, Germany: Association for Computing Machinery, 2012, 378–381. ISBN: 9781450312042. DOI: 10.1145/2351676.2351752.

[10] André Silva, Matias Martinez, Benjamin Danglot, Davide Ginelli, and Martin Monperrus. FLACOCO: Fault Localization for Java based on Industry-grade Coverage. 2023. arXiv: 2111.12513 [cs.SE].

[11] Spencer Pearson, José Campos, René Just, Gordon Fraser, Rui Abreu, Michael D. Ernst, Deric Pang, and Benjamin Keller. "Evaluating and Improving Fault Localization". In: 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE). 2017, pp. 609–620. DOI: 10.1109/ICSE.2017.62.

[12] Prantik Chatterjee, Abhijit Chatterjee, José Campos, Rui Abreu, and Subhajit Roy. "Diagnosing software faults using multiverse analysis". In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. IJCAI'20. Yokohama, Yokohama, Japan, 2021. ISBN: 9780999241165.

[13] Prantik Chatterjee, José Campos, Rui Abreu, and Subhajit Roy. "Augmenting automated spectrum based fault localization for multiple faults". In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. IJCAI '23. Macao, P.R.China, 2023. ISBN: 978-1-956792-03-4. DOI: 10.24963/ijcai.2023/350.

[14] Matias Martinez and Martin Monperrus. "ASTOR: a program repair library for Java (demo)". In: Proceedings of the 25th International Symposium on Software Testing and



Analysis. ISSTA 2016. Saarbrucken, Germany: Association for Computing Machinery, 2016, 441–444. ISBN: 9781450343909. DOI: 10.1145/2931037.2948705.

[15] Ezekiel Soremekun, Lukas Kirschner, Marcel Böhme & Andreas Zeller. “Locating faults with program slicing: an empirical analysis”. In Empirical Software Engineering, 2021. <https://link.springer.com/article/10.1007/s10664-020-09931-7>

[16] Lorena Barreto Simedo Pacheco, An Ran Chen, Jinqiu Yang, Tse-Hsun (Peter) Chen. “Leveraging Stack Traces for Spectrum-based Fault Localization in the Absence of Failing Tests”. 2024. <https://arxiv.org/pdf/2405.00565>