



Software Engineering Seminar

Information Retrieval Fault Localization

Description

Automated fault localization techniques assist developers with the task of pointing out program elements that are most probable to be responsible for a detected error. Over the years, many different techniques have been developed [6].

An interesting group of techniques can be classified as fault localization based on *information retrieval (IR)*. IR-based fault localization methods try to rank program elements based on, e.g., existing bug reports, textual similarity metrics, code complexity, etc.

The goal of this topic is to examine and discuss the current state of the art of IR-based fault localization techniques, to evaluate their relevancy and to analyze/estimate their capabilities compared to other fault localization techniques.

References

- [1] David W. Binkley and Dawn J. Lawrie. Information retrieval applications in software development. 2010.
- [2] T. Dao, L. Zhang, and N. Meng. How does execution information help with information-retrieval based bug localization? In *2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC)*, pages 241–250, May 2017.
- [3] R. K. Saha, J. Lawall, S. Khurshid, and D. E. Perry. On the effectiveness of information retrieval based bug localization for c programs. In *2014 IEEE International Conference on Software Maintenance and Evolution*, pages 161–170, Sept 2014.
- [4] R. K. Saha, M. Lease, S. Khurshid, and D. E. Perry. Improving bug localization using structured information retrieval. In *2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 345–355, Nov 2013.
- [5] Qianqian Wang, Chris Parnin, and Alessandro Orso. Evaluating the usefulness of ir-based fault localization techniques. In *Proceedings of the 2015 International Symposium on Software Testing and Analysis, ISSTA 2015*, pages 1–11, New York, NY, USA, 2015. ACM.
- [6] W. Eric Wong, Ruizhi Gao, Yihao Li, Rui Abreu, and Franz Wotawa. A survey on software fault localization. *IEEE Transactions on Software Engineering*, 42(8):707–740, 2016.

Contacts

Simon Heiden (heiden@informatik.hu-berlin.de)
Software Engineering Group
Institut für Informatik
Humboldt-Universität zu Berlin