Prioritizing Regression Tests for Manual Testing of Mobile Applications

Master's Thesis

Supervisor: Prof. Dr. Alexander Pretschner
Advisor(s): Roland Würsching, Oliver Wasiak (SAP)
Email: `roland.wuersching@tum.de`
Starting date: 15.10.2023

Fakultät für Informatik
Lehrstuhl 4
Software & Systems Engineering
Prof. Dr. Alexander Pretschner

Boltzmannstraße 3
85748 Garching bei München

Tel: +49 (89) 289 - 17362
https://www4.in.tum.de

Context

Regression testing is a fundamental procedure in software development to ensure that recent changes or modifications introduced into a software system do not unintentionally alter or disrupt the pre-existing system behavior or functionalities [1, 2]. This process plays a pivotal role since change is a constant in software engineering. Nevertheless, due to recent trends in software development, regression testing is getting more complex. For instance, component-based development often involves third-party black-box components, making it challenging to test changes that may impact the software system. Additionally, the shorter lifecycles in agile software development constrain regression testing when resources are limited [1].

While the most straightforward approach is retest-all, it is cost-intensive as it involves running every test case after each change [3]. Furthermore, with test suites tending to grow in size as software evolves, the retest-all approach becomes even more arduous [1]. Therefore, a more suitable technique can be regression test prioritization, which enables engineers to execute regression tests efficiently by ordering them to achieve testing objectives earlier based on coverage and priority criteria [4, 5, 6].

Research has extensively focused on prioritization approaches for various software types [7, 8, 9]. However, mobile app development remains overlooked despite the complexities of different device setups, operating systems, backend versions, geocaching, network conditions, multiple languages, screen sizes, touch interactions, and gestures. These factors make it challenging to automate the test setup, so testing is mainly done manually. Therefore, to ensure cost-efficient manual testing of mobile apps, it is crucial to conduct research on regression test prioritization in this area.

The SAP Mobile Engineering Munich Team is developing the native iOS app SAP Warehouse Operator and the cross-platform app SAP Direct Distribution. Since each app has a test suite of around 400-500 regression tests and quality engineers having limited time to test due to frequent releases, this is an ideal subject for studying prioritization approaches in an industrial setting.

Goal

The aim of this thesis is **(1)** to implement various regression test prioritization approaches for mobile apps, and **(2)** evaluate these approaches in the context of a case study based on the apps SAP Warehouse Operator and SAP Direct Distribution.

For **part 1**, various regression test prioritization approaches have to be studied and applied sequentially on the manual test suites of the aforementioned mobile apps.

After the manual test suites have been run prioritized by the different approaches, they have to be evaluated for **part 2**. To determine the most efficient prioritization approach, different metrics like Average Percentage of Faults Detected (APFD) could be applied to the various prioritized test runs [7]. After the different metrics have been applied, the approaches can be compared and a best approach can be determined. This can eventually be compared to similar studies for other types of software and developers and test engineers can be interviewed to make a statement about the efficiency.

Prioritizing Regression Tests for Manual Testing of Mobile Applications – Master's Thesis

Working Plan

1. Research: Review the literature about regression test prioritization.
2. Research: Learn about different regression test prioritization approaches.
3. Implementation: Setup the different approaches one by one.
4. Implementation: Run the tests prioritized by the different approaches.
5. Implementation: Apply different metrics to the various approaches.
6. Evaluation: Evaluate the metrics of the different approaches.
7. Evaluation: Compare the results and decide which approach works best.
8. Evaluation: Compare the results to state of the art literature and developer voices to be able to make a statement about its efficiency.
9. Evaluation: Identify limitations of the research and document them.

Fakultät für Informatik
Lehrstuhl 4
Software & Systems Engineering
Prof. Dr. Alexander Pretschner

Boltzmannstraße 3
85748 Garching bei München

Tel: +49 (89) 289 - 17362
https://www4.in.tum.de

Deliverables

- Technical report with comprehensive documentation of the different prioritization approaches, how they were applied to the test suites, and the metrics used for comparison.
- Source code of the evaluation of the different approaches.
- Final thesis report written in conformance with TUM guidelines.

References

[1] Shin Yoo, Mark Harman. 2012. Regression testing minimization, selection and prioritization: a survey. In Software Testing, Verification and Reliability. 67-120. https://doi.org/10.1002/stvr.430

[2] Hareton K. N. Leung, Lee White. 1989. Insights into regression testing. In Proceedings of the International Conference on Software Maintenance. 60-69.

[3] Daniel Elsner et al. 2022. Build System Aware Multi-language Regression Test Selection in Continuous Integration. In Proceedings of the International Conference on Software Engineering: Software Engineering in Practice. 87-96.

[4] W. E. Wong, J. R. Horgan, S. London, and H. Agrawal. 1997. A study of effective regression testing in practice. In Proceedings of the 8th International Symposium on Software Reliability Engineering. 264-274.

[5] S. G. Elbaum, A. G. Malishevsky, and G. Rothermel. 2000. Prioritizing test cases for regression testing. In Proceedings of the 2000 ACM SIGSOFT International Symposium on Software Testing and Analysis. 102–112.

[6] S. G. Elbaum, G. Rothermel, S. Kanduri, and A. G. Malishevsky. 2004. Selecting a cost-effective test case prioritization technique. In Software Quality Control. 185–210.

[7] R. Wuersching et al. 2023. Severity-Aware Prioritization of System-Level Regression Tests in Automotive Software. In IEEE Conference on Software Testing, Verification and Validation. 398-409.

[8] D. Di Nardo, N. Alshahwan, L. Briand, and Y. Labiche.2015. Coverage-based regression test case selection, minimization and prioritization: A case study on an industrial system. In Software Testing, Verification and Reliability. 371–396.

[9] Daniel Elsner, Florian Hauer, Alexander Pretschner, and Silke Reimer. 2021. Empirically evaluating readily available information for regression test optimization in continuous integration. In Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2021). Association for Computing Machinery, New York, NY, USA, 491–504. https://doi.org/10.1145/3460319.3464834