CMPUT 402/501
List of related research papers
Dr. Sarah Nadi -- University of Alberta

This is a list of research papers on topics related to software quality. You can use these papers as inspiration for your course project or as examples of how a research paper looks like.

1. The Seven Sins: Security Smells in Infrastructure as Code Scripts, Rahman et al., ICSE '19.

2. BugSwarm: Mining and Continuously Growing a Dataset of Reproducible Failures and Fixes, Dmeiri et al., ICSE '19. -- paper should be available on the website under publications soon. You can also directly contact the authors if it doesn't get posted soon.

3. Mining Historical Test Logs to Predict Bugs and Localize Faults in the Test Logs, Amar and Rigby, ICSE '19. -- preprint should be available soon. You can also directly contact the authors for a version.

4. A Large-Scale Study of Test Coverage Evolution, Hilton et al., ASE '18.

5. An Automated Approach to Estimating Code Coverage Measures Using Execution Logs, Chen et al., ASE '18.

6. One Size Does Not Fit All: An Empirical Study of Containerized Continuous Deployment Workflows, Zhang et al., FSE '18.

7. Modern Release Engineering in a Nutshel -- Why Researchers Should Care, Adams and McIntosh, SANER '16.

8. The Art of Testing Less without Sacrificing Quality, Herzig et al., ICSE '15.

9. Predicting Bugs from History, Zimmermann et al.

10. An Empirical Study on the Effect of Modern Code Review Practices on Software Quality, Adams and McIntosh, ESE '16.

11. Are Refactorings to Blame? An Empirical Study of Refactorings in Merge Conflicts,

Mahmoudi and Nadi, SANER '19.

12. Automatic Identification of Bug-Introducing Changes, Kim et al., ASE '06.

13. Mining Metrics to Predict Component Failures, Nagappan et al., ICSE '06.

14. Forked and integrated variants in an open-source firmware project, Stănciulescu et al., ICSME '15.

15. Software Practitioners View on Merge, McKee et al., ICSME '17.

16. Why We Refactor? Confessions of GitHub Contributors, Silva et al., FSE '16.

17. Mining API patterns as partial orders from source code: from usage scenarios to specifications, Acharya et al., FSE '07.

18. API Change and Fault Proneness: A Threat to the Success of Android Apps, Linares-Vásquez et al., FSE '13.

19. Security Versus Performance Bugs: A Case Study on Firefox, Zaman et al., MSR '11.

20. Mining Analogical Libraries in Q&A Discussions -- Incorporating Relational and Categorical Knowledge into Word Embedding, Chen et al., SANER '16.

21. Learning from Examples to Improve Code Completion Systems, Bruch et al., FSE '09.

22. Categorizing the Content of GitHub README Files. G. A. A. Prana, C. Treude, F. Thung, T. Atapattu, and D. Lo. Empirical Software Engineering, 2018.

23. Adding sparkle to social coding: an empirical study of repository badges in the npm ecosystem. Trockman et al., ICSE ’18

24. Detecting Missing Information in Bug Descriptions, Chaparro et al., FSE ’17.

25. Coverage is not strongly correlated with test suite effectiveness, ICSE '14

26. Are Mutants a Valid Substitute for Real Faults in Software Testing?, FSE '14

27. Configuration Smells in Continuous Delivery Pipelines: A Linter and A Six-Month Study on GitLab, FSE '20

28. Detecting Numerical Bugs in Neural Network Architectures, FSE '20

29. FrUITeR: A Framework for Evaluating UI Test Reuse, FSE '20