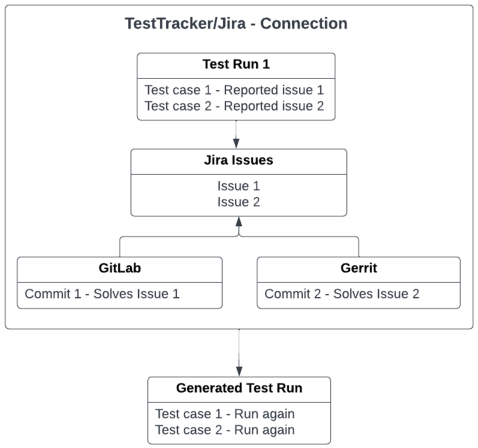
Test Case Selection based on code changes and risk of regression

Problem

When changes in code are made there is always a risk of error and regression. Manual system testing is done before a new launch but since there is a limited amount of time the test cases must be prioritized. The QA team at Axis uses an algorithm that analyzes code changes and evaluates the risk of errors being introduced by the change. But the test lead has to manually analyze the results and pick relevant test cases. To save time and resources, the QA team at Axis Communications want to develop a new algorithm that automatically selects the test cases that match the high risk commits.

Method

The work process was split into four different phases.The initial phase was gathering information. This was done by conducting a literature study on previous work related to the topic. The next method used to gather information was by having interviews with employees at Axis, including an interview with a test leader. The final method of information gathering was by manually testing one of the products. The second phase was spent on developing, testing and evaluating a prototype to the first solution, the TestTracker-Jira connection. The third phase was spent on developing, testing and evaluating a prototype to the second solution, the text analysis and keyword extraction. Finally the fourth phase was spent on merging the two prototypes developed into a finished product, also testing and evaluating it.



Solution

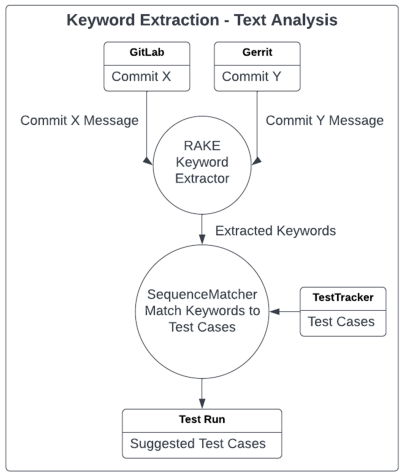
We developed an algorithm that uses two methods for selecting test cases. The first draws a connection between reported issues, the commits that fixed them, and the test cases used to validate that the affected feature works.
If a commit fixes an issue, connected test cases should be included in the test run.

The second method makes use of text analysis and keyword extraction. It extracts keywords from the commit messages and matches the relevance of the keywords to the test cases.

The algorithm combines the two methods and produces a list of suggested test cases for the upcoming test run.

Result

The algorithm developed in this project collects commits from multiple version control software, such as Gitlab and Gerrit, then matches them with Trace test cases from any chosen project. The algorithm then generates a list of suggested and prioritized test cases. The suggested test cases are shown in a browser where test leaders can look at data for every commit and see why each test case was chosen, helping them create their own test run.

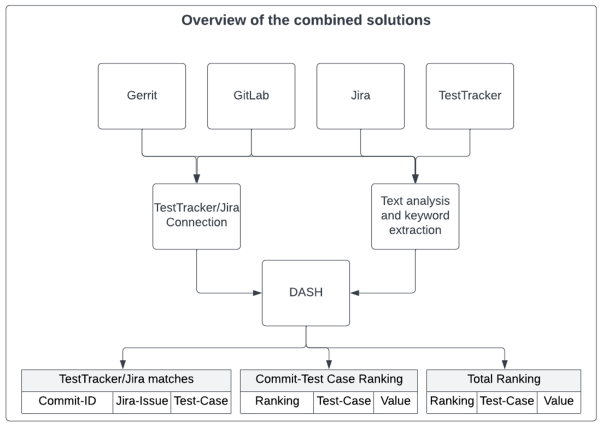


Conclusion

The result of this thesis is an algorithm that automatically suggests test cases based on recent code changes and the risk of introduced errors or regression. The algorithm contains two methods for suggesting test cases. The first method, the TestTracker-Jira connection, draws a direct connection between the test-case that reported an issue or bug and the commit that solved it. This method yields the most accurate suggestions but demands that developers and testers always references Jira tickets.

The second method makes use of text analysis and keyword extraction and suggests test cases based on the keywords extracted from commit messages. This method yields less accurate suggestions compared to the first method, but it does not require developers to always include references to Jira tickets. In order for this method to work correctly and accurately, the commit mesages have to be descriptive enough to extract accurate keywords.

The algorithm opens a graphical user interface in a web browser where the QA test leader can view rankings of the test cases as well as suggestions for individual commits.



Alexander Olofsson
Christoffer Larsson