Appendix C

Paper C: User Stories

Top Prio User Stories

1. In my daily work; I want to get a rapid overview of tonight's testing; so I can understand if I need to do a deeper investigation or not.

2. As a developer; I need to see results from a branch; so that I can work rationally with my branch

3. In my daily work; I want to get lists of failing and invalid test cases; so I can investigate what went wrong.

4. As someone involved in branch merge; I need to see that there are not many failing tests in a branch; so that we can make decisions on merge.

5. At merge time; I want to be able to make use of the tool; so we can make better decisions about merge.

6. As a developer deciding on merge; the tool should give me all I need to decide on merge; so we can make decisions in the team without special competence such as test lead.

7. As someone involved in merge; I want to use web-based tools to get all the information I need; so that I don't have to use complicated command line tools.

Very High Prio User Stories

1. In my daily work; if a suite did not start I want to see details from the job (test framework branch, etc.); so I can better investigate why the job

did not start.

2. As a developer in an experimental feature branch; I want to be able to see results; so results should not only be available for "official features".

3. In my daily work; I want to be able to look at more than data source (such as branch) at a time; so I can compare a data source with another one (diff branches).

4. In my daily work; I want sensible default filters in all views; so I won't have to start by refiltering.

5. In my daily work; I want to be able to filter away some results; so I can look at results without known issues.

6. In my daily work; I want to be able to filter based on many things (outcome, hardware type, test system, . . .); so I can see what I care about.

7. My colleagues with different roles; must be able use find use of the tool; so it can be useful for many different stakeholders.

High Prio User Stories

1. As a developer; the tool should help me take responsibility of the product quality I am responsible of; so I can easily take responsibility.

2. As a branch responsible; I want it to be easy to set up a new branch; so that it's easier for me to start a new branch.

3. Whenever I work with the tool; I want to be able to filter in ways I had not thought about in the past; in order to explore the results in new ways.

4. In my daily work; I want heatmap visualizations when investigating test cases, branches, test systems; so I can rapidly get an overview.

5. In my daily work; I want it to be easy to get an overview of tests with known issues; so we don't overlook or forget things with known issues.

6. In my daily work; I need good access to log files, in particular when test cases fail (kernel logs, etc. from the DUT's); so that my work can be simplified

7. At merge time; I want to understand what is risky with the branch; so we can put resources on testing the risks.

8. When investigating a failed test; I want to understand how the test case was designed and why; so I don't waste time being frustrated about why a test case exists.

9. In my daily work; I want flaky tests to be pointed out; so I can stabilize it as soon as I realize it's flaky.

10. In my daily work; I want to rapidly get an overview of my branch; so I can be prepared before the daily standup.

11. In my daily work; I want a dashboard of nightly testing; so I can get an overview of the nightly testing.

12. In my daily work; I want to see long term trends of non-functional characteristics; so I can understand any long-term changes.

13. When I look at a trend plot; I want to be able to select time window (start and stop date); so I can understand results better.

14. In the future, when we have many virtual test systems that "come and go" rapidly, the tool should function as well as for physical systems; so I am not "punished" for using virtual systems.

15. In the future, when we have much more testing; the tool must still be able to work without scalability problems; so the tool can live for a long time.