

Lead engineer

Result visualization

- great addition
- summary at the top of the report with score? Other metrics?
no need to scroll if scores are perfect

Metrics

- agree with the justification of categories
- split $0\% < m\% < 100\%$ into 2 groups
they felt like e.g. 20% lines would have different underlying reasons than e.g. 80%

Test generation

- valuable addition on its own
- (pseudo-)equivalent detection could spare lots of time

Engineer using the mutation testing tool

Learning process

- mutation testing:
 - concept – when first explained, 0.5-1 hour presentation
 - application in this context – when tool was demonstrated, 1 hour
- tool:
 - understood how it works when first demonstrated, 1 hour
 - usage: could've looked at the documentation, but after the session where I helped setting up the project, it was not needed (~1 hour)

Initial version of tool

- added value:
 - live mutants show possible mistakes they did not think of → they can correct by adding more tests and also learn from their mistakes
 - killed mutants reassure them that a tested requirement is truly covered
- missing features:
 - graphical interface for results
 - for starting test sessions, a UI could be more comfortable, they're used to it from the test definition tool
- worth the time required to use: yes
 - time:
 - onboarding: 1-2h
 - environment setup: 1h+waiting for IT department
 - executing: could be done overnight
 - result evaluation: 10 min/day
 - meanwhile added value is undoubted
- would use: yes
 - they would use it evaluate the quality of "finished" test sets
- how would use:
 - after thinking they finished testing a module/set of modules
 - check for missing test cases
 - implement missing test cases
 - run again to check if covered
 - repeat until all killable mutants are killed

Improved version of tool (test report, metrics)

- added value:
 - metrics:
 - per-LOC mutation score: good estimate of type of deficiency
 - line-based mutation score ratio:
 - does not help them directly, but good for showing progress towards management
 - [later]: evaluate state of module when handed over, e.g. change request
 - report: much better, no need to look at report in Excel and code in CODESYS
- missing features: UI to start tests
- worth the time required to use: [same as before]
 - new reports make result evaluation more effective
- would use: [same as before]
- how would use: [same as before]
 - if project isn't new (change request), would start by running to check status of existing TCs, the report helps visualize the state of a module

Test generation

- added value:
 - (pseudo-)equivalent mutant detection
 - suggestion of test inputs for live mutants
- would use:
 - for analyzing live mutants: yes
 - decreases time spent with analyzing live mutants
 - imagined it as an extra status column next to the result of the mutant
 - for test generation:
 - initially only for simpler modules
 - if there are no mutants, no tests are generated – a problem for simpler modules
 - tracing generated tests to requirements can be complicated, especially for more complicated modules