Paper ID: Shi2017



Technological rule: To reduce wasteful text executions when tests are poorly placed in modules, use TestOptimizer to guide which tests should be moved reducing test execution time.

Problem
Understanding
The authors
mathematically
formalize the problem
of wasteful test
executions that occur

Problem Instance

Many large software projects have imprecise dependency graphs that lead to wasteful test executions which impacts on developer productivity



Evaluation approach:

TestOptimizer was applied to 5 large proprietary projects, calculating time saved by potential movements. Validation was carried out with developers who accepted 84% of the suggestions.



TestOptimizer, a technique which uses a greedy algorithm to suggest test movements between test nodes.providing a ranked list of suggestions to devs allowing them to choose and reduce the expected number of test executions.





Solution design approach:

A greedy algorithm is proposed to reduce # of test executions by suggesting test movements that consider historical build information & test dependencies



when tests are poorly

placed in project

modules.

Relevance: Relevant for developers during regression testing of large systems when tests may be poorly positioned, leading to wasteful test executions at build time.



Rigor: Case study with 5 large systems was carried out.



Novelty: Proposal of TestOptimizer, a tool for integration in a build environment for making suggestions on how to reduce wasteful test executions. The tool can also be used to reduce impact of flaky tests.