

# Fail Fast

Weifan Jiang (wfjiang@uw.edu)

Yuqi Huang (yh73@uw.edu)

# Motivation:

Situation: Running test suite multiple times

Problem:

- Many common unit test frameworks (JUnit) run tests with fixed ordering
- Possible to wait until whole test suite finishes to see part of result that you are curious about

Any improvements?

- First run test cases that are more likely to fail, then run others

# Test Prioritization

Calculate a priority score of each test case, based on:

- Percentage of failure in past runs of test suite.
- Has this test case failed consecutively in the last couple runs of test suite?
- Are there correlations between test cases?
  - For example, test case A and B always pass or fail together
  - Test A first, if A passes, we can move B to the end of the queue
  - Think about conditional probabilities!

Reorder the test cases based on priority scores: make test suite to fail as fast as possible!

# Limitation

- Appropriate way to convert three aspects of prioritization to a numerical score
- Extra computation & test result storage may get expensive as test suite size increases.
- Overkill for small test suites

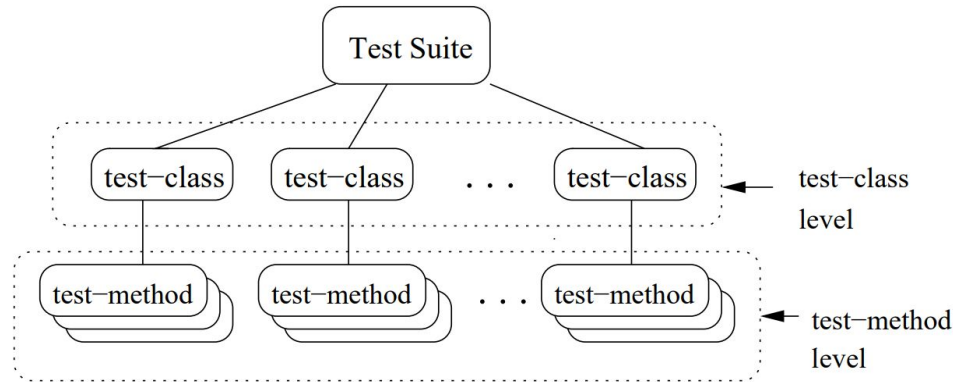


Figure 1: JUnit test suite structure