In-tree Unit Testing

Vassil Vassilev, PhD

Unit testing

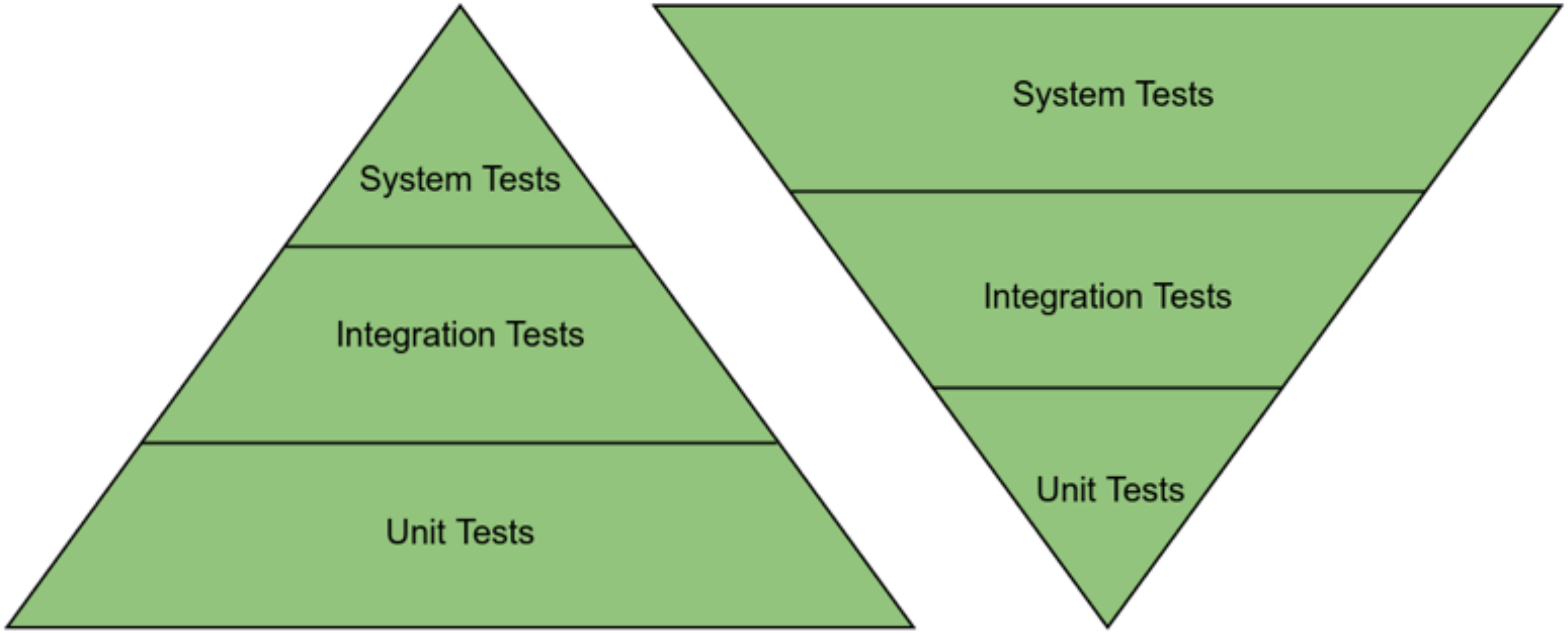
In computer programming, unit testing is a **software testing method** by which **individual units of source code**, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

What Unit Testing Solves

Testing the whole app as a unit by pretending to be a user (see references):

- Slow & Flacky
  - **Green**:
    - high confidence paths are OK
    - high initial coverage; impossible to get corner cases
  - **Red** = Could be OK, not sure
    - Hard to reproduce failures
    - Need debugger to figure out what went wrong

Test Classification



System Tests

Integration Tests

Unit Tests

Amount of tests

System Tests

Integration Tests

Unit Tests

Test execution time

Testing Guidance

No guidance how to write a test, but a lot of guidance how to write testable code.

Writing Scoped Tests

- Unit testing leafs is easy (unit test for std::sort is trivial to write)
- Testing entities which construct other entities outside of the scope of testing is hard

Friendly Objects

- Objects which substitute the real objects and breaking the dependency graph to simplify and speedup execution
- An implementation of this concept is gtest’s mock library

Pros and Cons of In-tree Tests in ROOT

- + Allows tools to reason about never instantiated code. Eg. TDataFrame

- + Introduces the pros of unit testing and writing faster tests

- + We have already such folders in math. Test will provide just better testing facilities

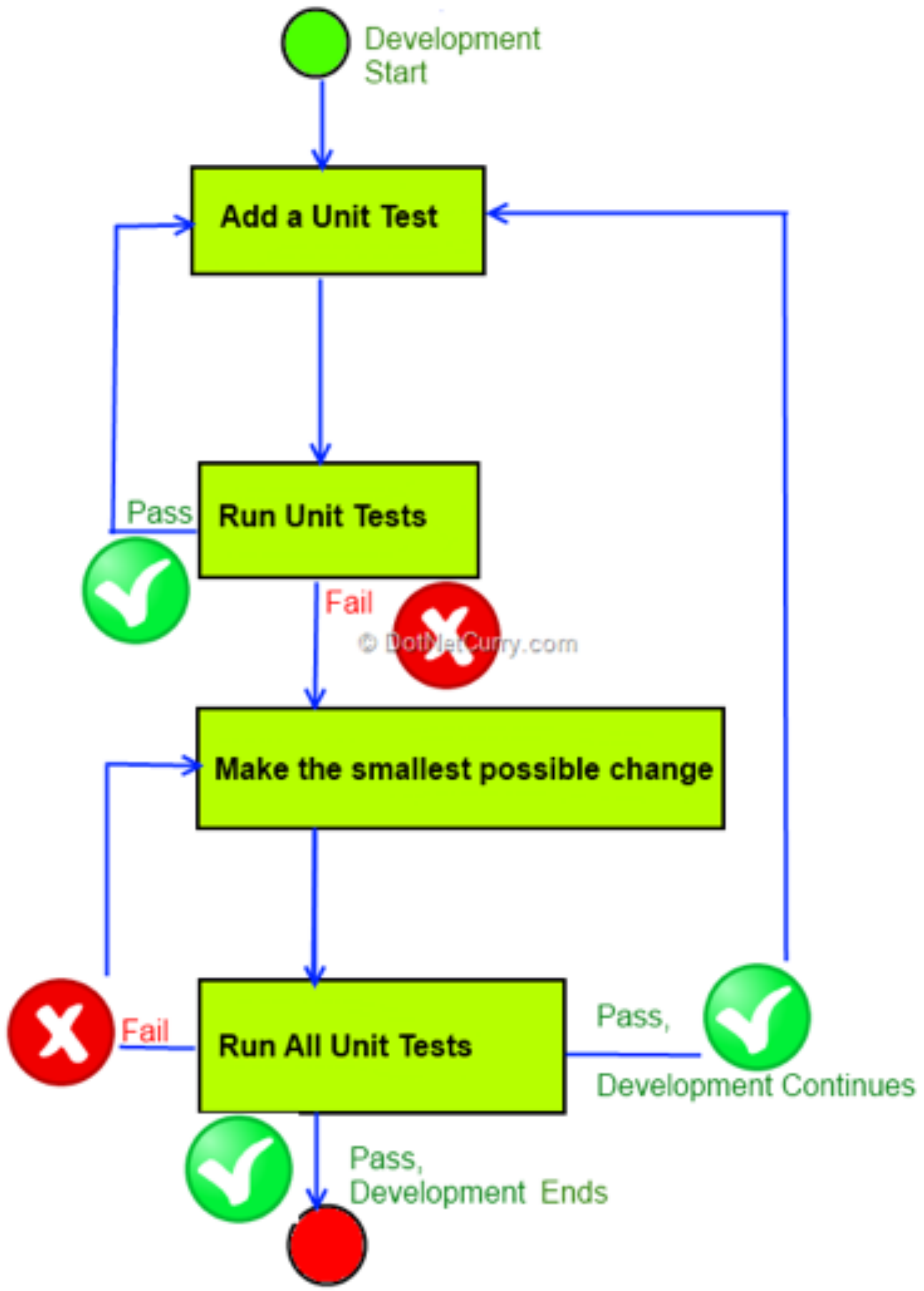
- + Improves GitHub pull request review process because all code goes in one place

- - Bigger codebase (only if we write *many* tests to make it significant to the 3.5MSLOC)

Adding GTest to ROOT

- A first step towards writing in-tree unit tests is PR451 which is ready to land.
- It allows to write not-so-complicated unit tests
- It is meant to trigger incremental progress in this direction

A Foreseen Pull Request Testing Flow



References

Google Test Primer: https://github.com/google/googletest/blob/master/googletest/docs/Primer.md

Google Mock: https://github.com/google/googletest/blob/master/googlemock/docs/ForDummies.md

The Clean Code Talk — Unit testing: https://www.youtube.com/watch?v=wEhu57pih5w&feature=youtu.be