Electrical & Computer ENGINEERING

18-642:

Testing Overview

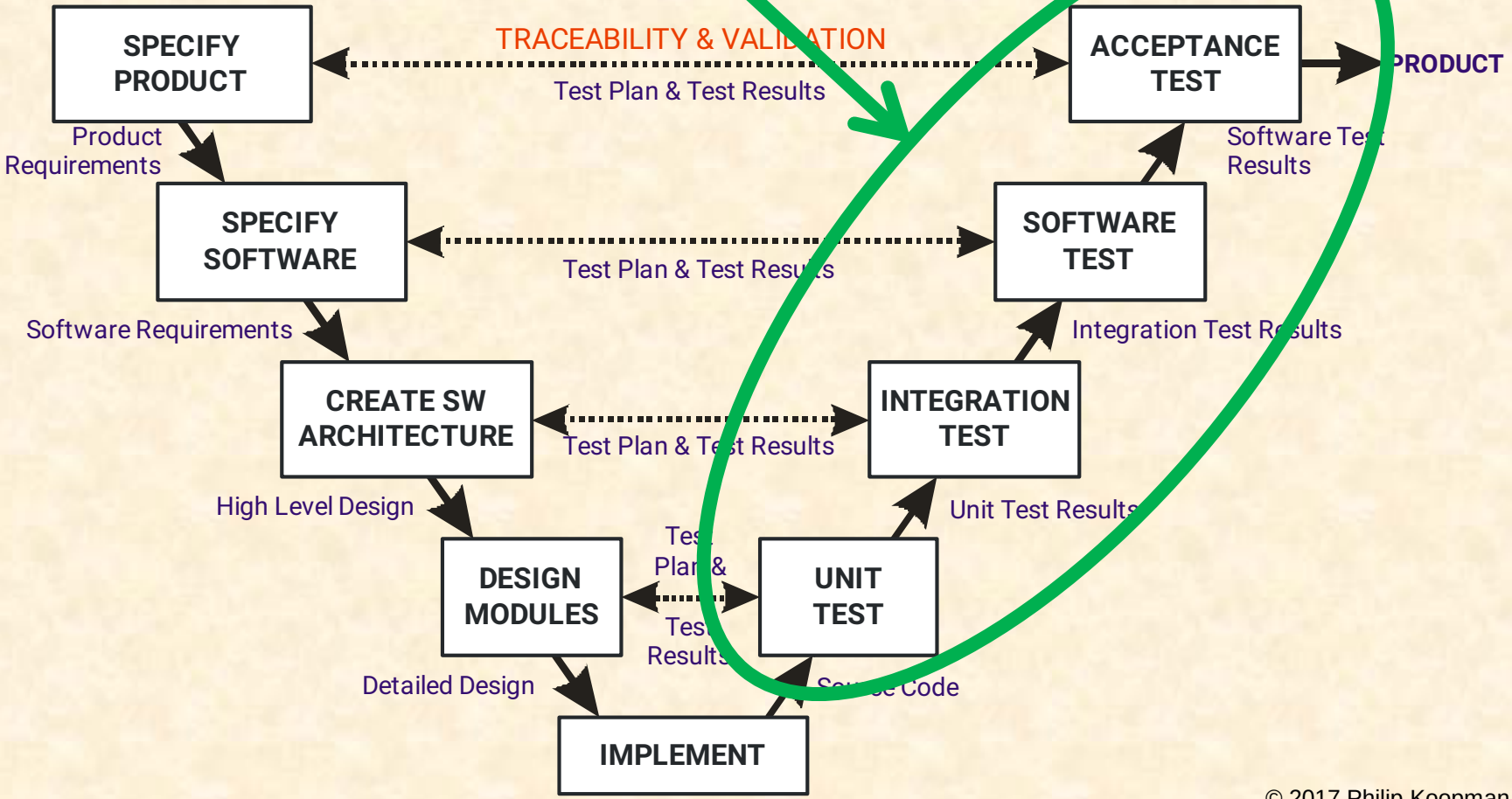
9/25/2017

"In September of 1962, a news item was released stating that an $18 million rocket had been destroyed in early flight because "**a single hyphen was left out of an instruction tape**."... The nature of programming being what it is, there is no relationship between the "size" of the error and the problem it causes. Thus, it is difficult to formulate any objective for program testing, short of "the elimination of all errors" - an impossible job. "

- Gerald M. Weinberg, goo.gl/w42tqZ

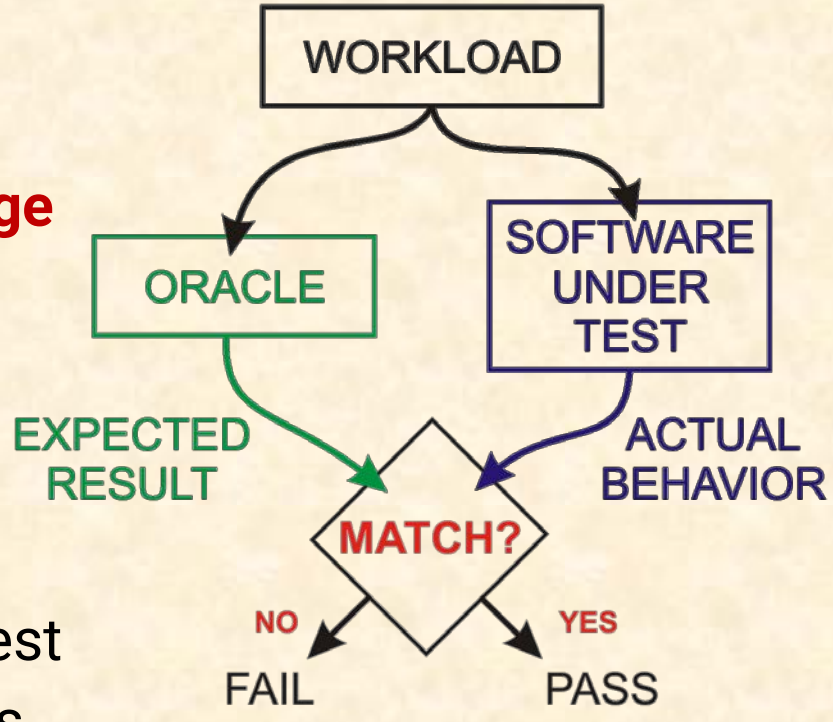
Carnegie Mellon University

YOU ARE HERE



Testing

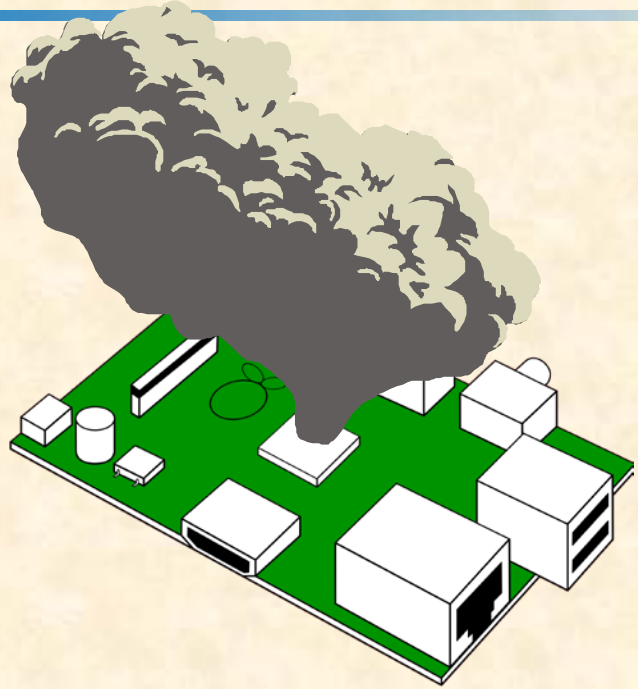
- **Anti-Patterns:**
  - **Primarily ad hoc, informal testing**
  - **No defined notion of testing coverage**
  - **No defined pass/fail test results**



- **Testing executes software**
  - A workload is applied to SW under test
  - An oracle generates expected results
  - The test passes if SW matches oracle

Unstructured Testing Approaches

- **"Smoke Test"**
  - Quick check to see if the software runs at all
  - Valuable pre-test if it exercises major functions
- **Exploratory Testing**
  - Expert tester beats on the system
  - Can find low hanging fruit to fix if tester is good
- **Problems with unstructured approaches**
  - Hard to know how much you've tested
  - Test coverage is often subjective
  - Effectiveness heavily dependent upon tester skill

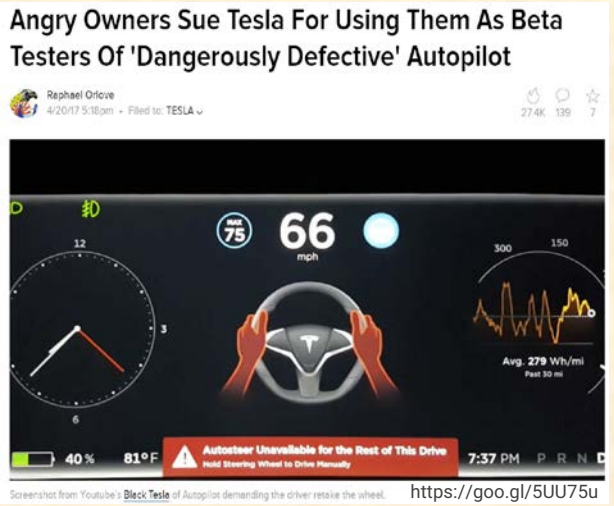


Methodical Testing Approaches

- **Black Box Testing**
  - Use requirements to guide test design
  - Exercise all code functions
- **White Box Testing**
  - Use structure of code to guide test design
  - Exercise all paths through code
- **Need both black box and white box testing**
  - Black box testing can miss implementation corner cases
  - White box testing doesn't exercise missing functionality
- **Coverage: how completely have you tested?**
  - White box coverage: fraction of code exercised in test
  - Black box coverage: fraction of requirements exercised
  - Other coverage: e.g., fraction of state chart arcs traversed

Testing Across Development Cycle

- **Testing within V process**
  - Unit test: single executable procedure
  - Integration test: do modules play nice?
  - Software test: exercise system or subsystem
  - Acceptance test: whole-product test with customer use scenarios
- **Other types of testing**
  - Beta test: see if representative users discover defects
  - Regression test: test to see if a previously fixed bug comes back
  - Performance testing: measure capacity and identify bottlenecks
  - Robustness testing: what happens with invalid inputs?
  - Security testing: can an attacker penetrate the system?
  - Fault injection: what happens when a component fails?



https://goo.gl/5UU75u

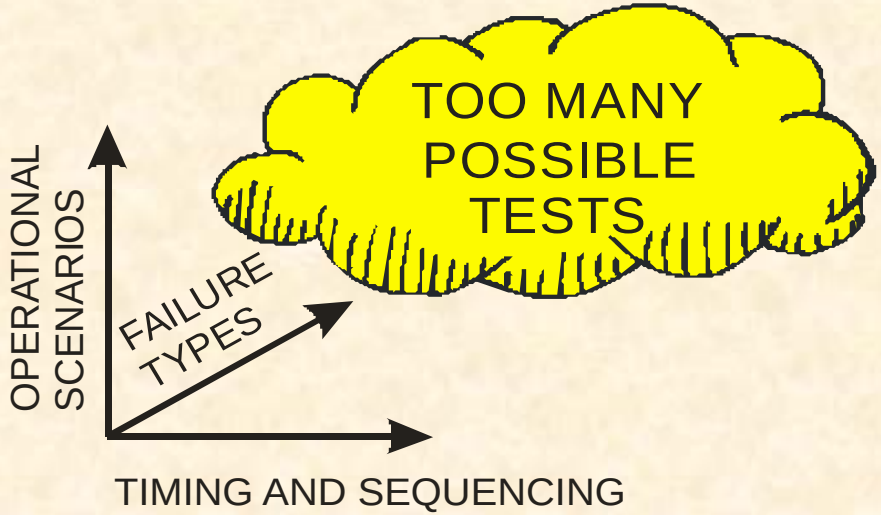
Testing Philosophy

- **Testing finds bugs**
  ➔ but why are you finding bugs?
- **Testing Low Quality Software:**
  - Find bug, remove bug, iterate until out of time
  - How many bugs are left?

- **Testing Safety Critical Software:**
  - All bugs removed via peer review, unit test, integration test
    - Testing is to prove hypothesis that software is essentially perfect
  - Product-level tests should ***never*** find a SW bug
    - If your find a bug, **your process is broken**

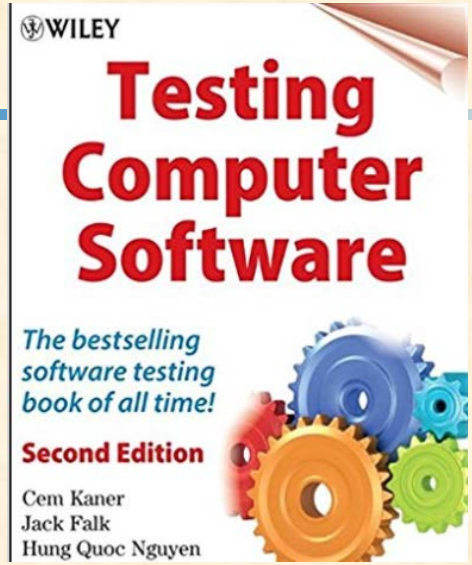
Testing Best Practices

- **Get high testing coverage**
  - White box – most or all code paths
  - Black box – all requirements
  - Exploratory testing to reduce risk
    - Unexpected functionality
    - Low hanging fruit for buggy software
  - Other types of testing as appropriate



- **Testing pitfalls**
  - Testing all data values is impracticable
    - How many tests to cover all inputs for: `proc(int w, int x, int y, int z)`
  - Creating an oracle is the hard part of scaling up test generation
  - 100% coverage is not 100% tested
  - 100% passing tests is not 100% bug-free

Testing Resources



- **Good starting point for more info:**
  - http://www.softwareqatest.com/index.html

- **Useful references:**
  - Kaner, Cem, Jack Falk, and Hung Quoc Nguyen. Testing Computer Software, Second Edition. International Thomson Computer Press, 1993. ISBN 1-85032-847-1. *OR* John Wiley & Sons, Inc., 1999. ISBN 0-471-35846-0.
  - Beizer, Boris. Black Box Testing. John Wiley & Sons, 1995, ISBN 0-471-120904-4.
  - Free Training: http://www.testingeducation.org/BBST/