Informatics 43

LECTURE 7-2

EMILY NAVARRO

Last Time

- We use HCI/UCD methods…
  - Interviews/observations, personas, scenarios, storyboards, swimming lane diagrams, site maps, mockups, design guidelines, heuristic evaluation, user testing
- …for good reasons
  - Sales double
  - Performance doubles
  - Traffic counts increase
- It's all about the user!

Today's lecture

- Failures: a second look
- Quality assurance
- Testing
- Quiz 4 study guide

Today's lecture

- Failures: a second look
- Quality assurance
- Testing
- Quiz 4 study guide

What do these have in common?

- Airbus 320
- Toyota
- Mariner 1 launch
- AT&T telephone network
- Ariane 5
- Mars Polar Landing
- Radiation therapy machine
- NSA
- Y2K

They all failed!

- Airbus 320
  - http://catless.ncl.ac.uk/Risks/10.02.html#subj1.1
- Toyota
  - "unintended" acceleration problem
- Mariner 1 launch
  - http://catless.ncl.ac.uk/Risks/5.73.html#subj2.1
- AT&T telephone network
  - Ripple effect, from switch to switch, network down/dark for 2-3 days
- Ariane 5
  - http://catless.ncl.ac.uk/Risks/18.24.html#subj2.1
- Mars Polar Landing
  - http://spaceflight.nasa.gov/spacenews/releases/2000/mpl/mpl_report_1.pdf
- Radiation therapy machine
  - http://courses.cs.vt.edu/~cs3604/lib/Therac_25/Therac_5.html
- Y2K

Toyota Failure

- Overly complex spaghetti code
- Violated standards set by the industry
  - Single-point failures
- No peer code reviews
- System threw away error codes

Toyota did not follow good quality assurance practices

Y2K Facts

- Bug description
  - Date formats were MM/DD/YY, e.g., 01/01/98, 02/02/99, 03/03/00
  - Does 00 mean 2000 or 1900?
  - Does 1999 turn to 19100?
- Effects
  - Relatively minor
- Cost: $300 billion!

Impact of Failures

- Not just "out there"
  - Space shuttle
  - Rocket
  - Mars Polar Landing
- But also "at home"
  - Your car
  - Your call to your mom
  - Your wireless network, social network, mobile app
  - Your homework
  - Your hospital visit

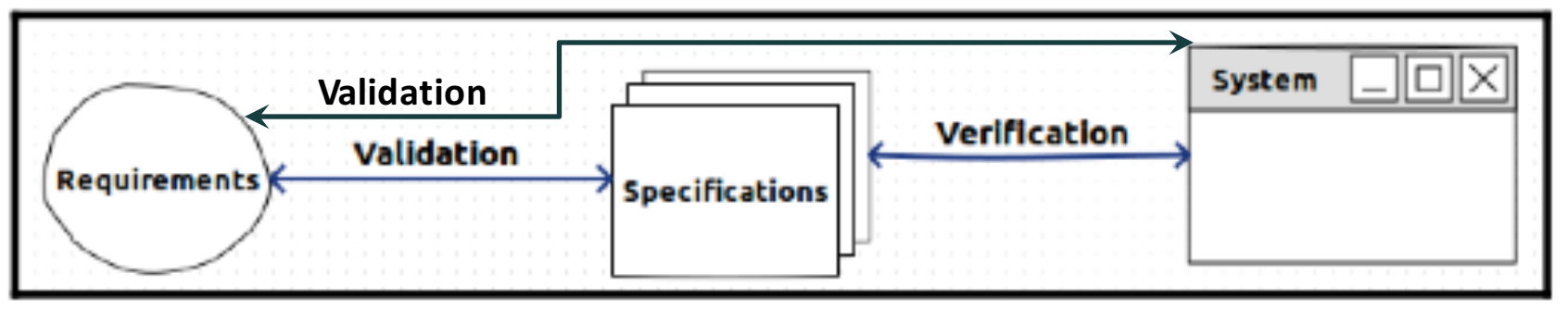
Peter Neumann's Risks Digest: http://catless.ncl.ac.uk/Risks

Today's lecture

- Failures: a second look
- Quality assurance
- Testing
- Quiz 4 study guide

QA Goals: Verification and Validation

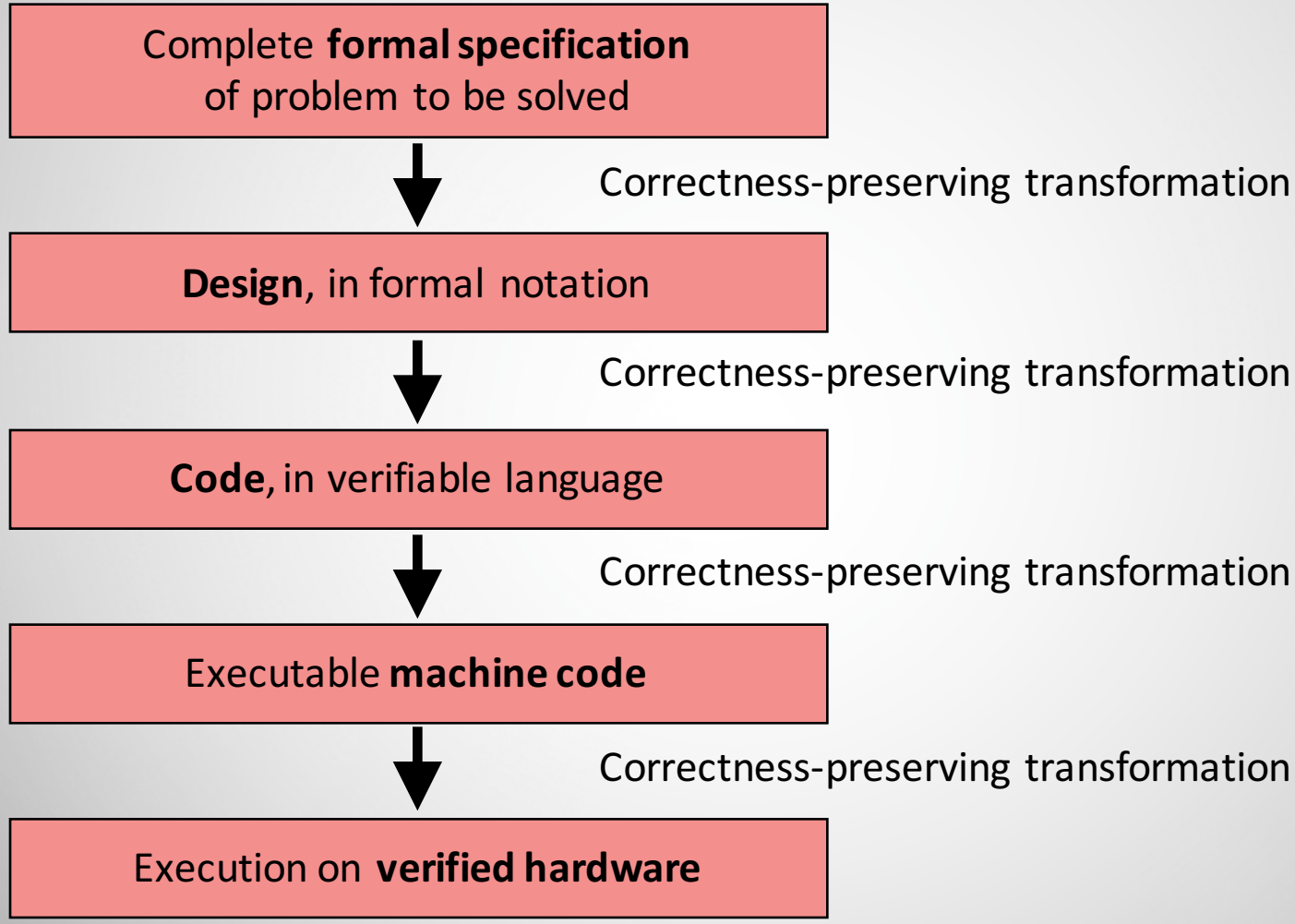
- Quality Assurance = All activities designed to measure and improve quality in a product



- Verification: "Implement the idea properly"
- Validation: "Implement the proper idea"

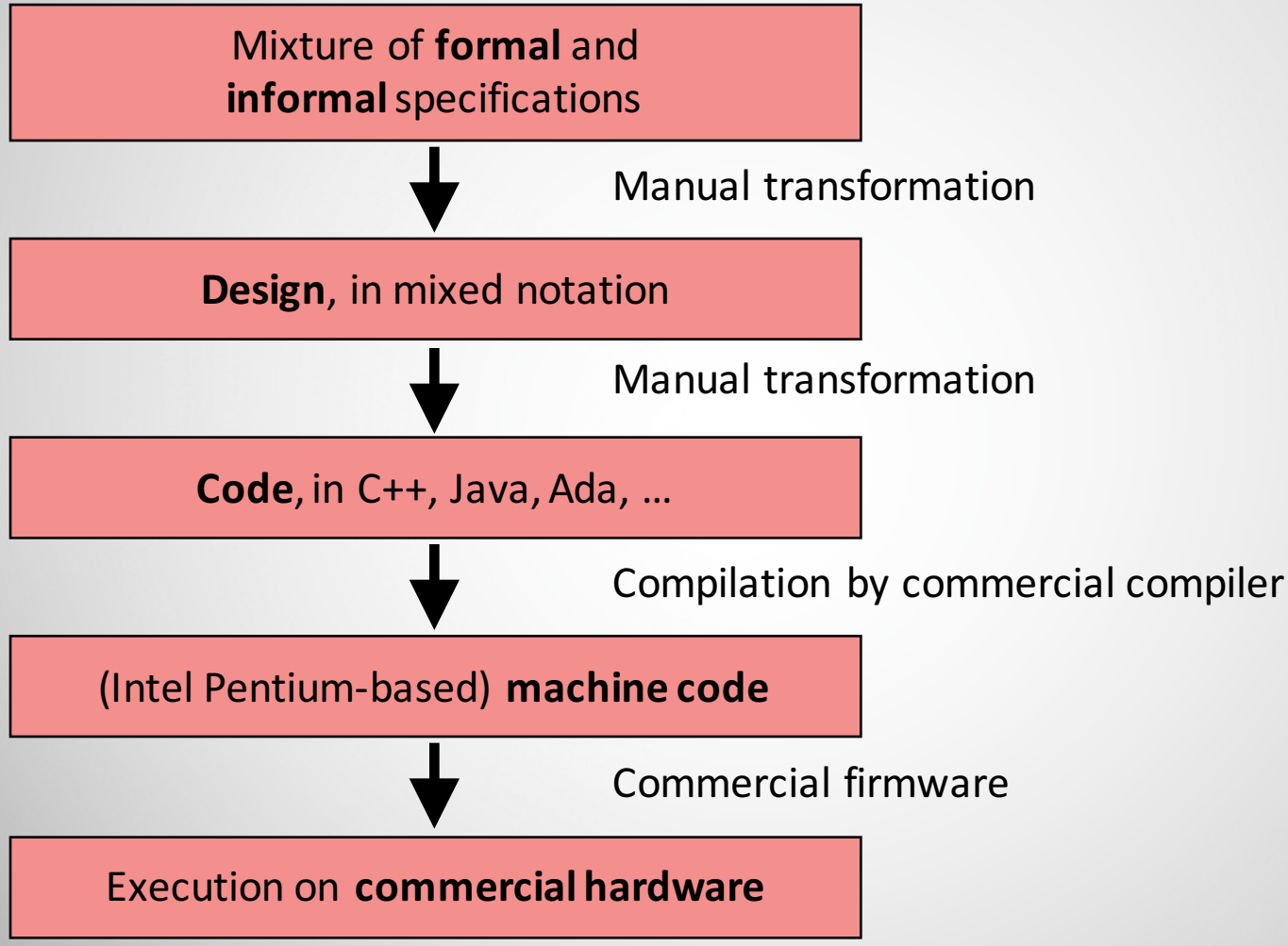
Why do we need QA?

Ideal world scenario:

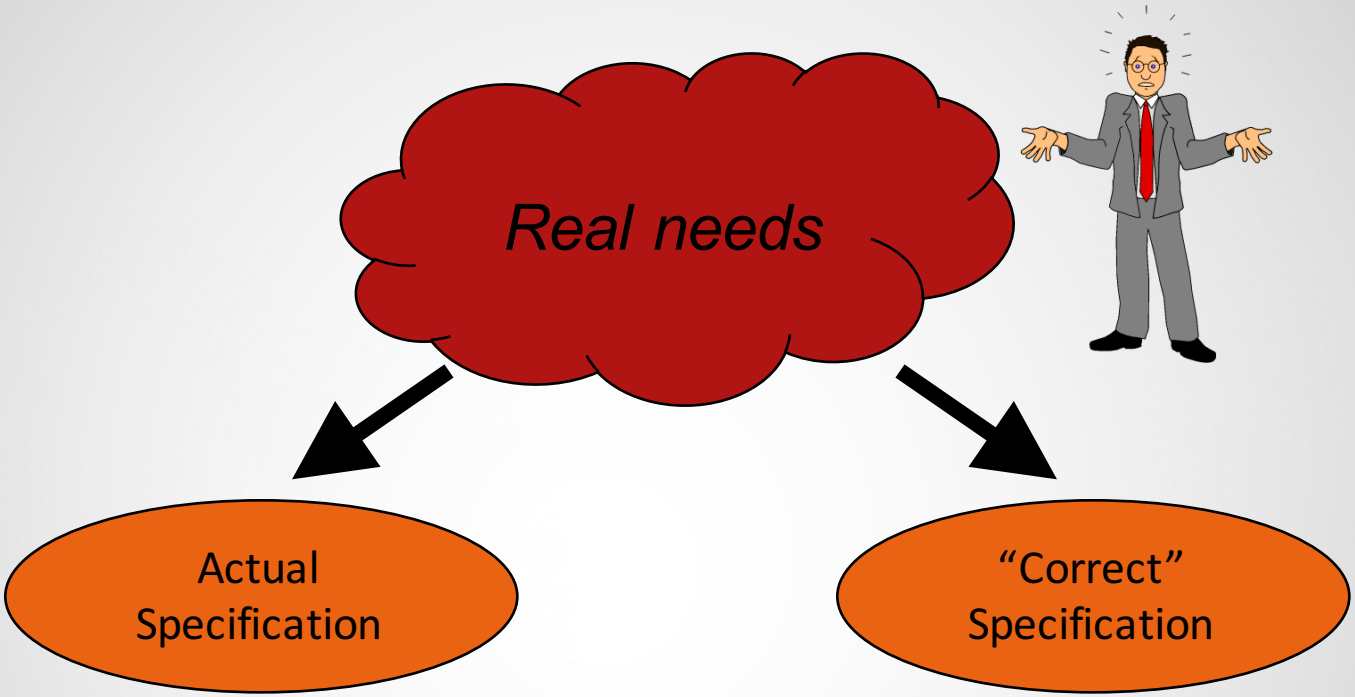


Why do we need QA?

Real world scenario:



Why is QA so difficult? (I)

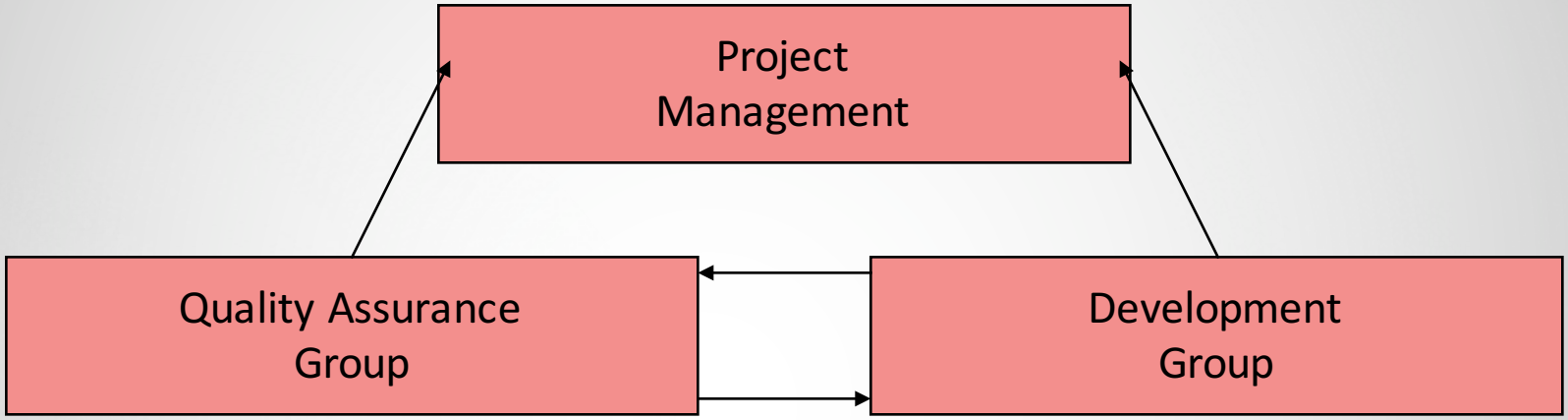


Why is QA so difficult? (II)

Why is QA so difficult? (III)

- Complex data communications
  - Electronic fund transfer
- Distributed processing
  - Web search engine
- Stringent performance objectives
  - Air traffic control system
- Complex processing
  - Medical diagnosis system

Why is QA so difficult? (IV)



Why is QA so difficult? (V)

- Quality assurance lays out the rules
  - You will check in your code every day
  - You will comment your code
  - You will…
- Quality assurance also uncovers the faults
  - Taps developers on their fingers
  - Creates image of "competition"
- Quality assurance is viewed as cumbersome, "heavy"
  - "Just let me code"

QA Techniques

- Formal methods
- Static analysis of program properties
- Reviews and inspections
- Testing

Use a mixture of techniques!

Today's lecture

- Failures: a second look
- Quality assurance
- Testing
- Quiz 4 study guide

Testing – Basic Process

- Detect and correct errors in a software product
- Exercise a module, collection of modules, or system
  - Devise test case (input)
    - Create expected output
  - Run test case
  - Capture actual output
  - Compare actual output to expected output

Actual output = expected output	**Actual output ≠ expected output**
Test case **SUCCEEDS**	Test case **FAILS** (report failure)

- (Lather, rinse, and repeat)

Testing Terminology

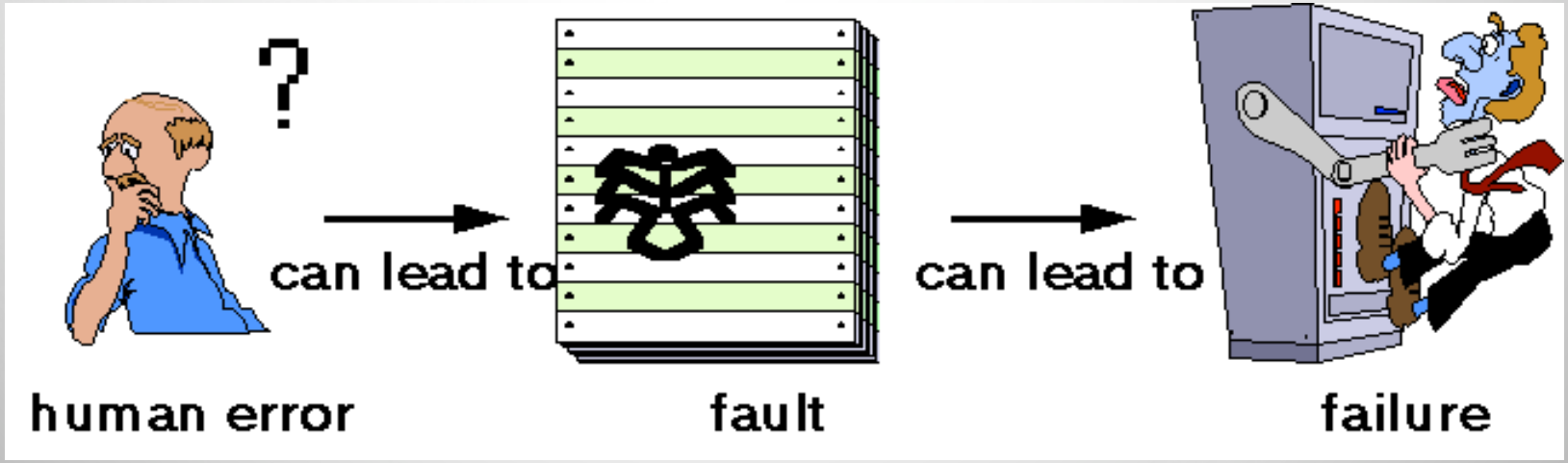
Error (human mistake in programmer's mind)

↓

Fault or defect (discrepancy in code)

↓

Failure (external behavior/execution/output is incorrect)



Poor Mrs. Null…

"I feel like I still have to do things the old-fashioned way."
– Jennifer Null

Source: http://www.bbc.com/future/story/20160325-the-names-that-break-computer-systems

Testing Goals

- Find and fix failures/faults/errors
- Improve confidence that the system performs as specified (verification) and as desired (validation)
- All in a manner that is
  - Accurate
  - ~~Complete~~ Thorough
  - Repeatable
  - Systematic

Program testing can be used to show the presence of bugs, but never to show their absence [Dijkstra]

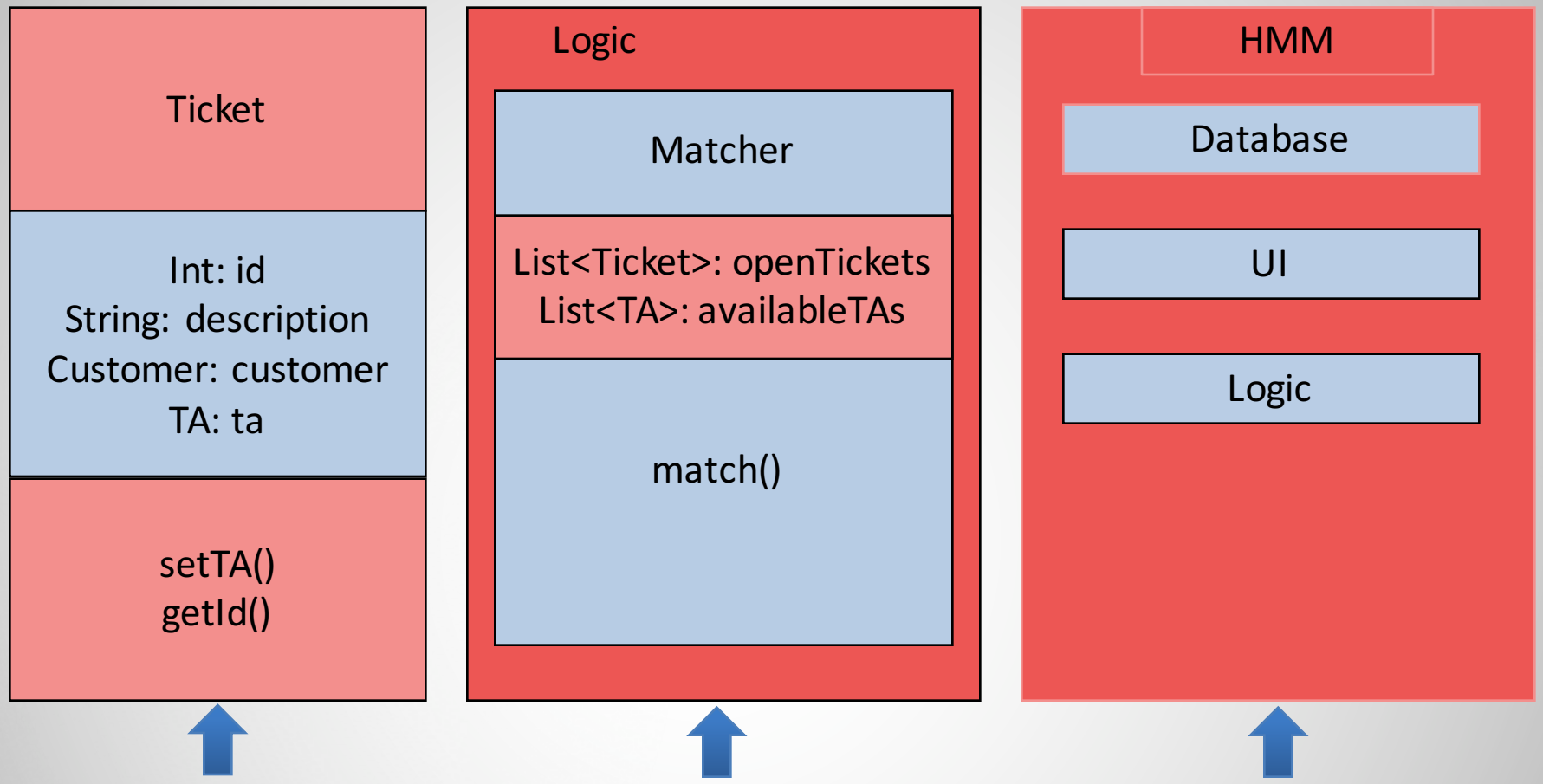
Who does the testing?

- Programmers
- Testers
- Users

Levels of Testing

- Unit testing
  - Testing of a single code unit
- Functional/integration testing
  - Testing of interfaces among integrated units
- System/acceptance testing
  - Testing of complete system for satisfaction of requirements

Levels of Testing



Unit testing

Ticket.setTA()

Functional/integration testing

Matcher.match()

System/acceptance testing

Create ticket

Pay ticket

Login

Logout

How to choose test cases?

- There are usually an infinite number of possible test cases, so we must take a small sample

```
int multiplier(int a, int b) {
      return a * b;
}
```

Ways to choose test cases

- Intuition
- Specification (black-box testing)
  - Equivalence class partitioning
  - Boundary-value analysis
- Code (white-box testing)
  - Path analysis
- Existing test cases (regression testing)
- Faults

Test Oracles

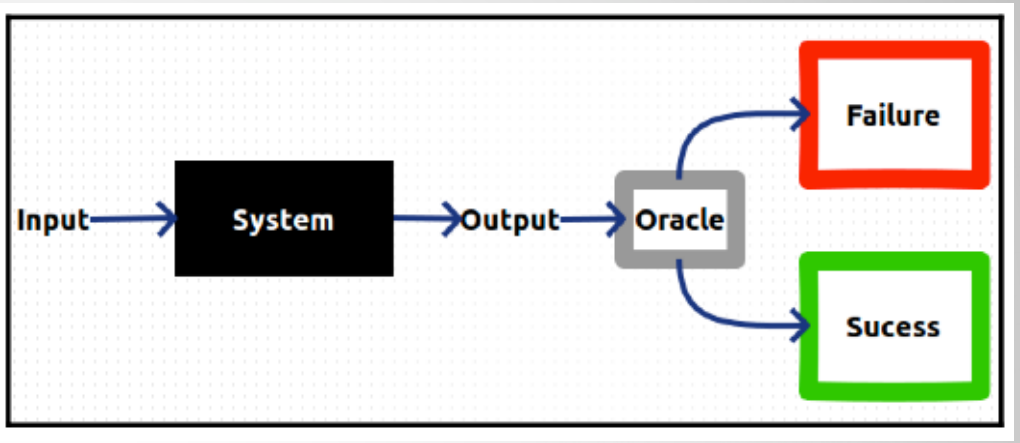
- A mechanism for deciding whether a test case execution succeeds or fails
- Difficult to automate

Oracle Example: Cosine

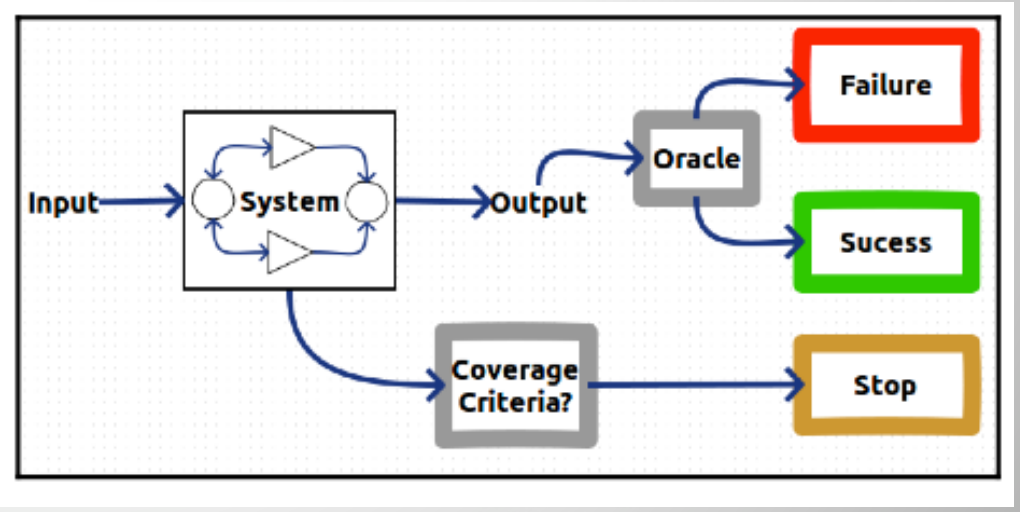
- Your test execution shows cos(0.5) = 0.87758256189
- Oracles?
  - Drawing a triangle and measuring the sides
  - A cosine table in a book
  - A website
  - Computing the value using Taylor series expansion
  - Calculator

Two Overall Testing Approaches

- Black box testing
  - Specification-based testing



- White box testing
  - Structural testing



All software has bugs!

"***All nontrivial code has defects***, *and the probability of nontrivial defects increases with code size. The more code you use to solve a problem, the harder it gets for someone else to understand what you did and to maintain your code when you have moved on to write still larger programs.*"

- Code Inflation, Holzmann (IEEE SW 2015)

All software has bugs!

Q: What is the required period of failure-free operation for conventional takeoffs and landings of the F35 Joint Strike Fighter?

A) 6,000 years
B) 600 years
C) 60 years
D) 6 hours

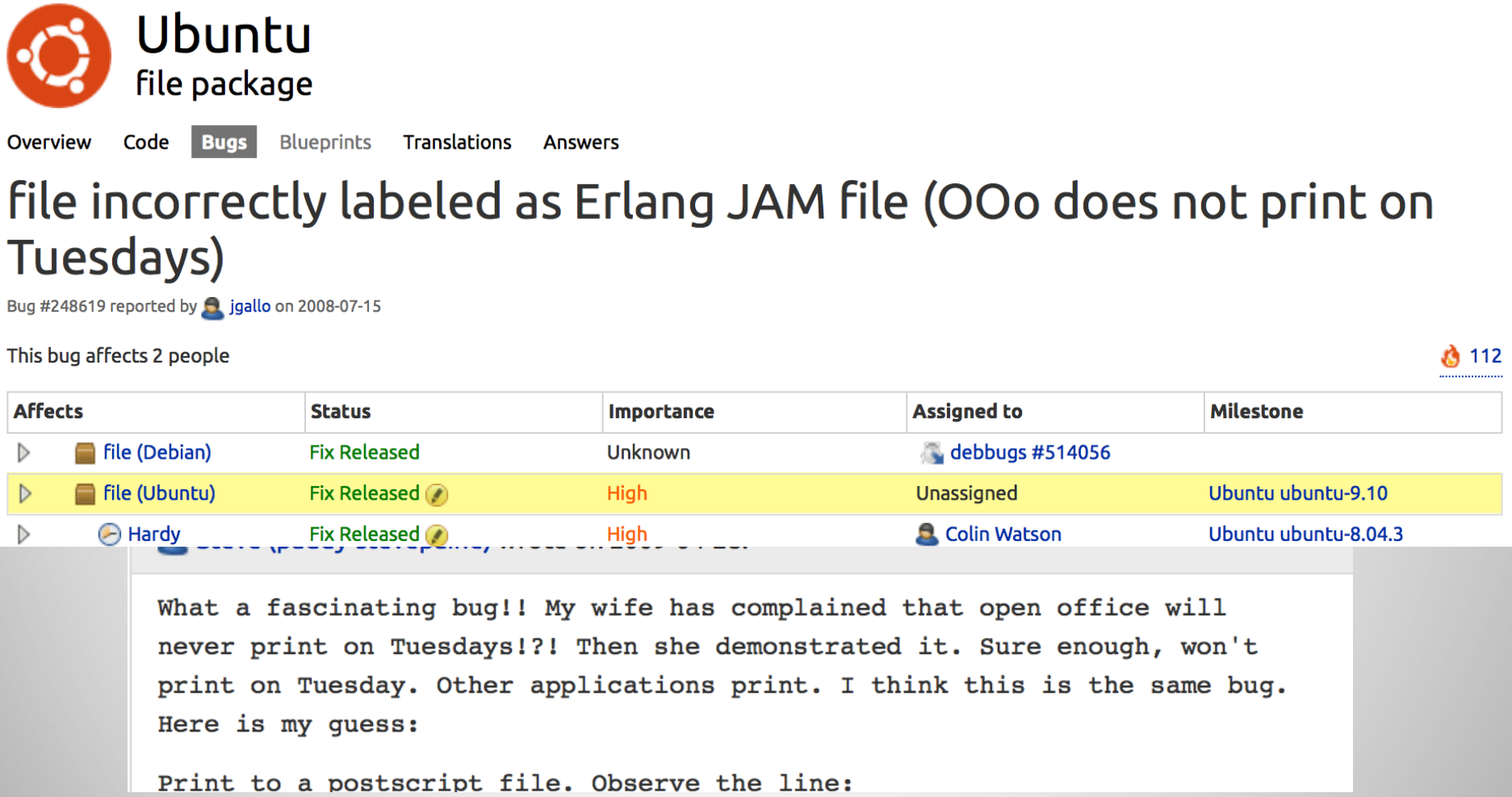
All software has bugs!

Q: What is the required period of failure-free operation for conventional takeoffs and landings of the F35 Joint Strike Fighter?

A) 6,000 years
B) 600 years
C) 60 years
D) 6 hours

AND a recent government report stated that this target had not yet been realized!

Some bugs are bizarre…



Ubuntu
file package

Overview Code **Bugs** Blueprints Translations Answers

file incorrectly labeled as Erlang JAM file (OOo does not print on Tuesdays)

Bug #248619 reported by jgallo on 2008-07-15

This bug affects 2 people 112

Affects	Status	Importance	Assigned to	Milestone
file (Debian)	Fix Released	Unknown	debbugs #514056	
file (Ubuntu)	Fix Released	High	Unassigned	Ubuntu ubuntu-9.10
Hardy	Fix Released	High	Colin Watson	Ubuntu ubuntu-8.04.3

What a fascinating bug!! My wife has complained that open office will never print on Tuesdays!?! Then she demonstrated it. Sure enough, won't print on Tuesday. Other applications print. I think this is the same bug. Here is my guess:

Print to a postscript file. Observe the line:

Source: https://bugs.launchpad.net/ubuntu/+source/cupsys/+bug/255161

Some bugs are long-lived…

Friday, September 25, 2015

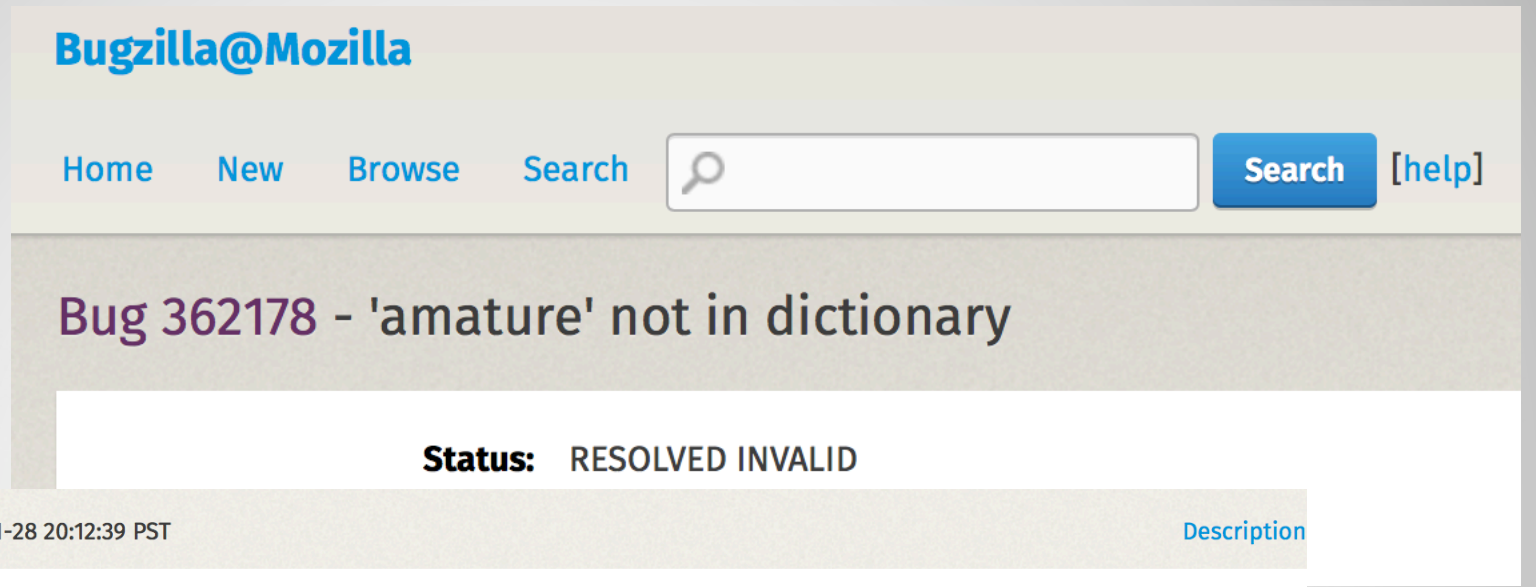
Thanks Google for Open Source TCP Fix!

The Google transport networking crew (QUIC, TCP, etc..) deserve a shout out for identifying and fixing a nearly decade old Linux kernel TCP bug that I think will have an outsized impact on performance and efficiency for the Internet.

Their patch addresses a problem with cubic congestion control

Source: http://bitsup.blogspot.com/2015/09/thanks-google-tcp-team-for-open-source.html

Some bugs are not bugs…

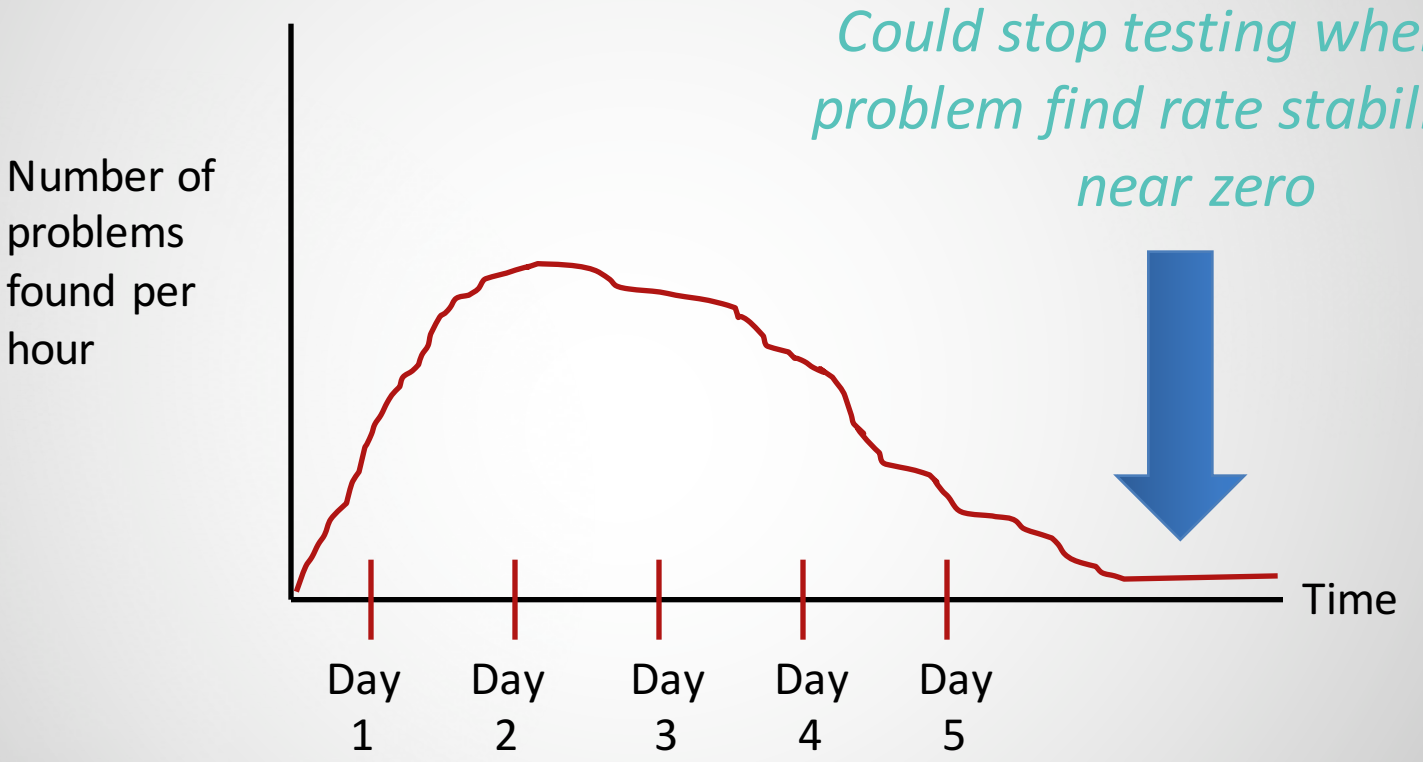


Source: https://bugzilla.mozilla.org/show_bug.cgi?id=362178

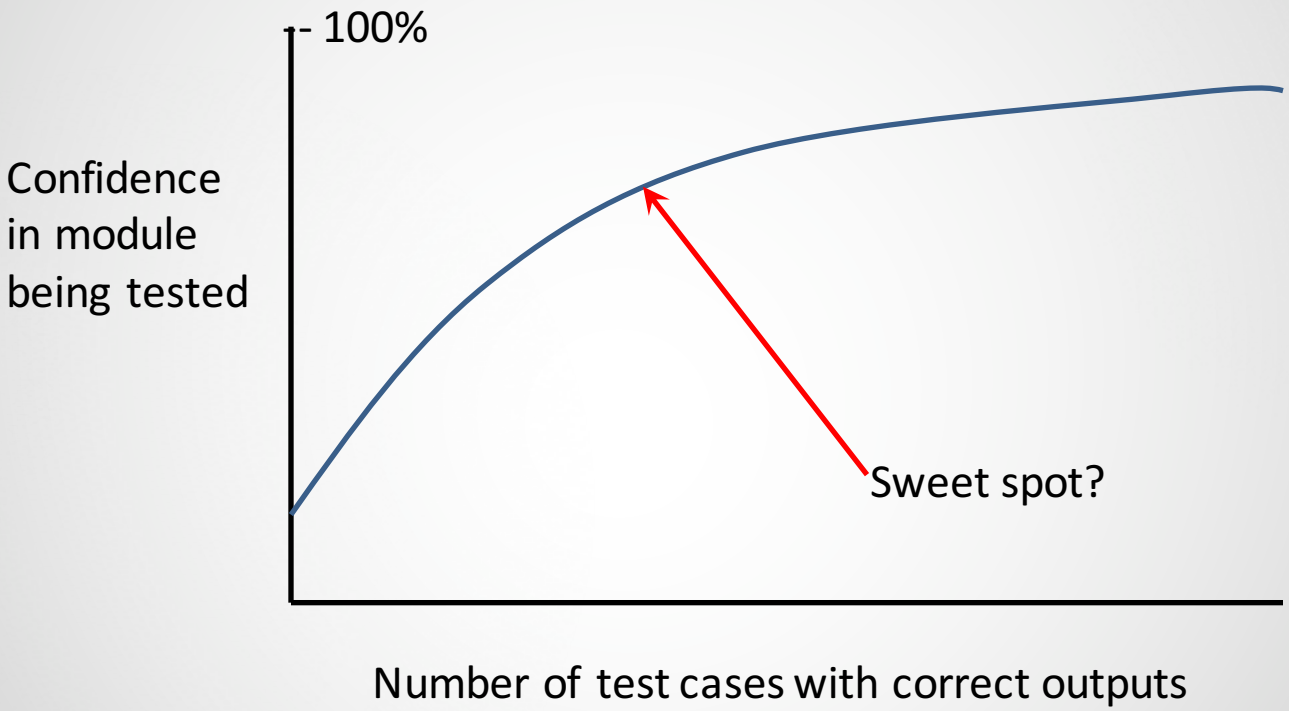
How do we know when we are done testing?

- Aim to reveal as many faults as possible in a fixed period of time with a fixed budget
  - Target specific areas of the system
- Aim to meet the quality requirements established for the project

How do we know when we are done testing?



How do we know when we are done testing?



How do we know when we are done testing?

- Pepper the code with defects and observe how many of the seeded defects are discovered
  - From this, infer how many bugs are left to find
- This technique assumes that nonseeded defects are similar to the seeded ones

Summary

- Many failures are caused by a lack of good quality assurance (QA) practices
- QA = All activities designed to measure and improve quality in a product
  - Validation & verification
- Testing is the most common QA activity
  - Different levels: unit/functional/system
  - Goal: find and fix failures/faults/errors
  - Can never be exhaustive
  - Can never prove a system's correctness
- All software has bugs!

Today's lecture

- Failures: a second look
- Quality assurance
- Testing
- Quiz 4 study guide

Quiz 4 – Topics (1)

- Designs/models/notations
  - Approaches to software design
  - Purposes of the different types of diagrams presented
    - For UML, only class diagrams
- User orientation
  - Understand the 10 user-centered design methods
    - Understand all the Nielsen heuristics (Discussion exercise)

Quiz 4 – Topics (II)

- Testing
  - Validation/verification
  - Error, fault, failure
  - Levels of testing
  - Oracles
- Readings - know main idea of:
  - LinkedIn story
  - "GoodUX" article
  - Toyota article

Next Time

- Black box testing
- Attend discussion tomorrow (user orientation)