CSC 7003 : Basics of Software Engineering

J Paul Gibson, D311

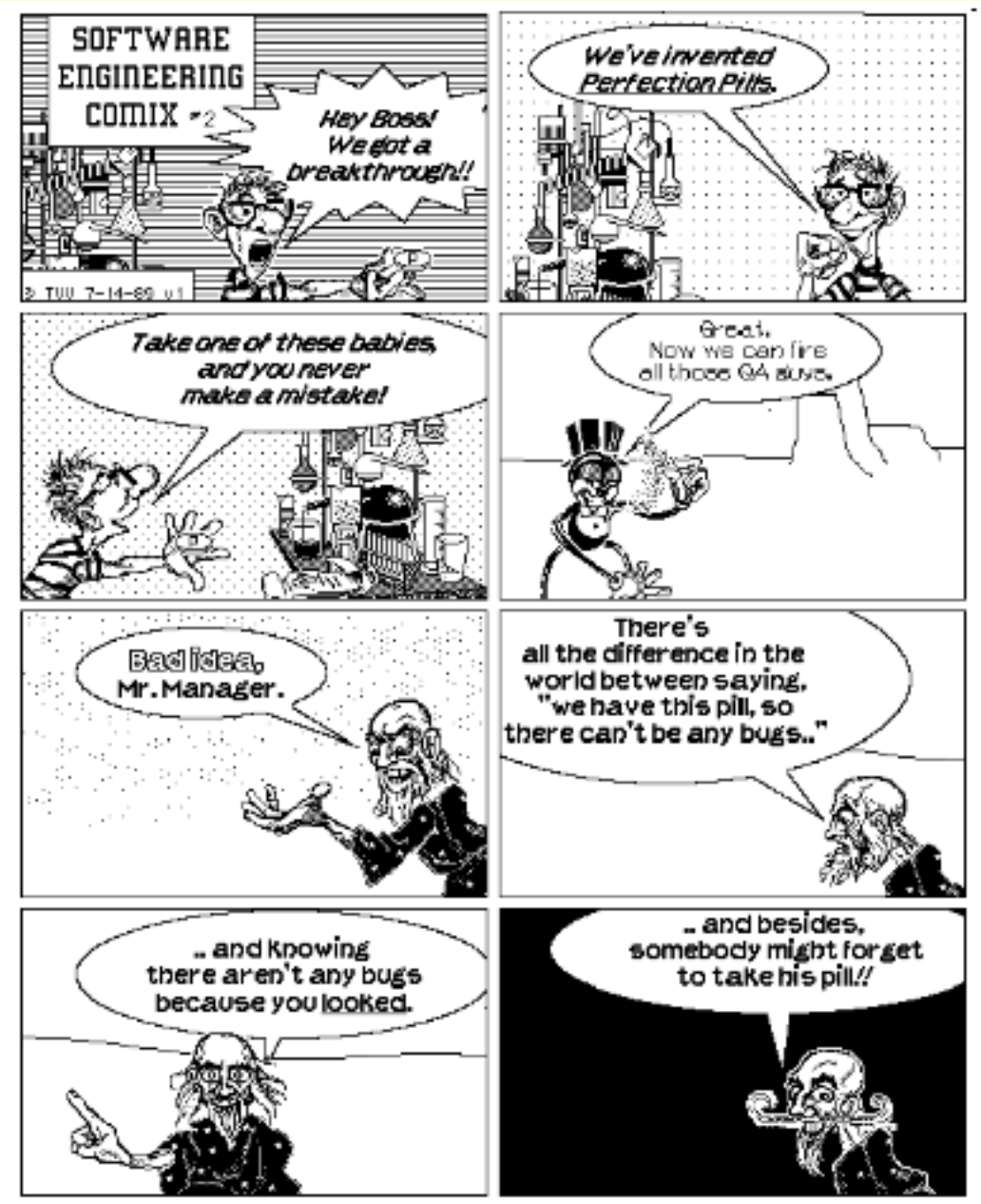
paul.gibson@telecom-sudparis.eu

http://www-public.telecom-sudparis.eu/~gibson/Teaching/CSC7003/

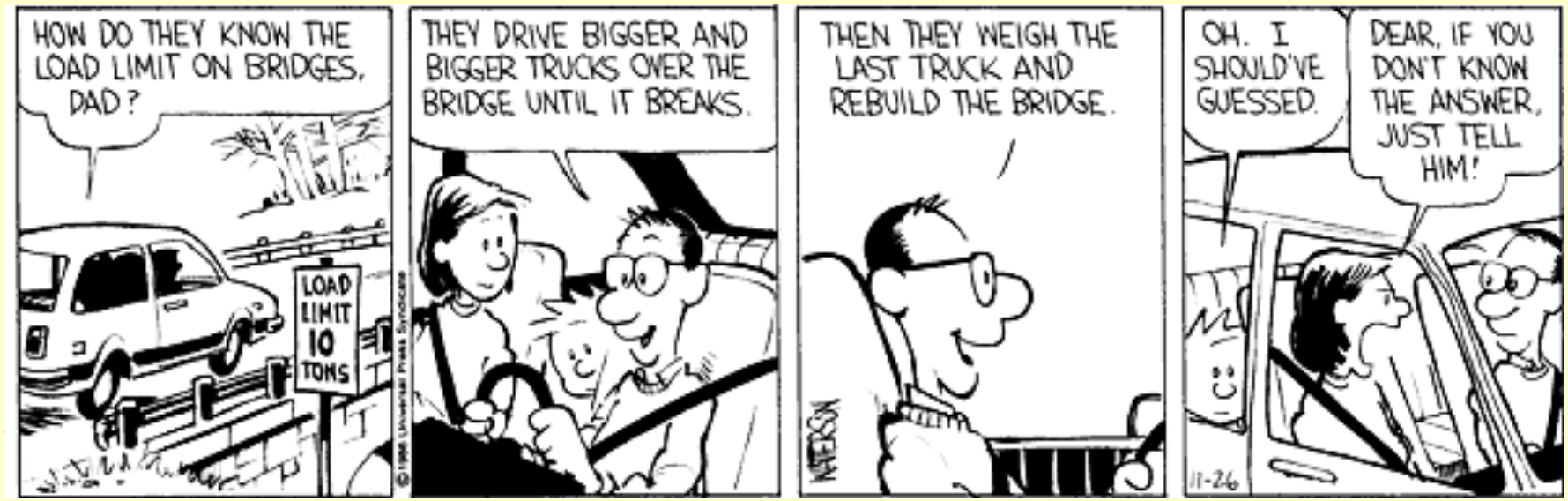
Testing

…/~gibson/Teaching/CSC7003/L7-Testing.pdf

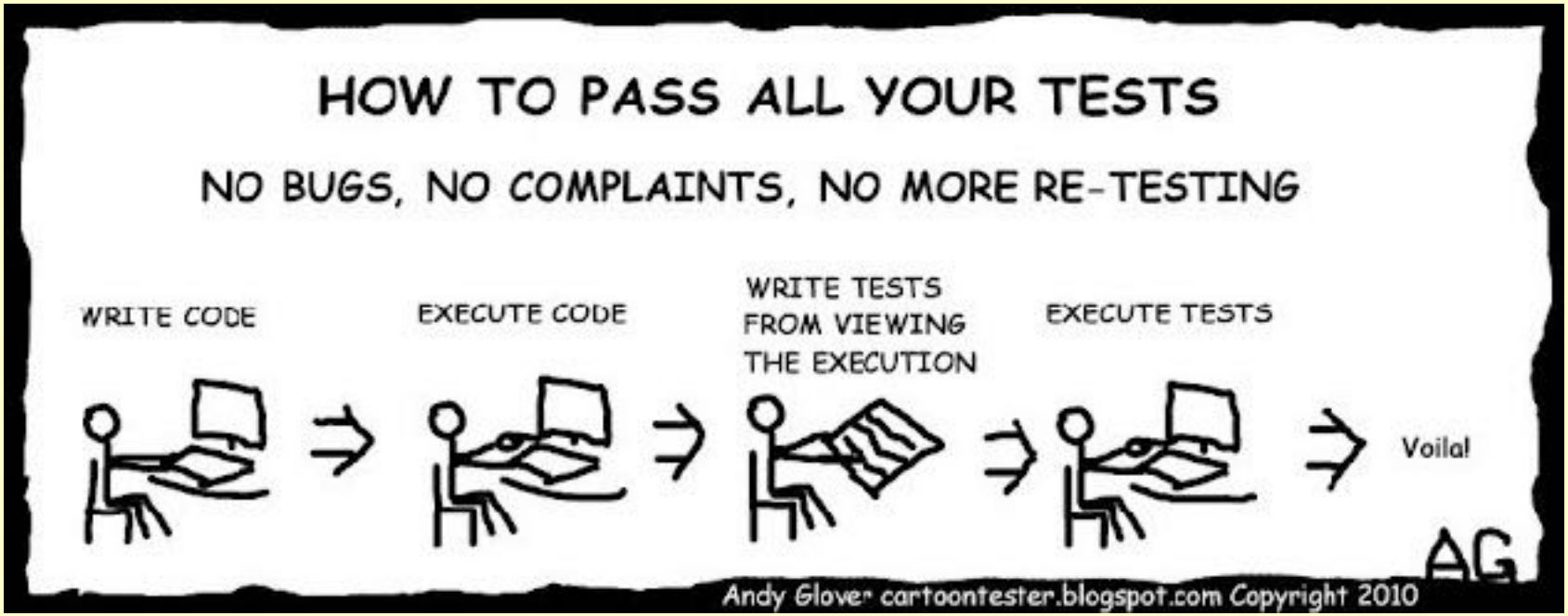
Why do we need tests?



Do we test our software like this?

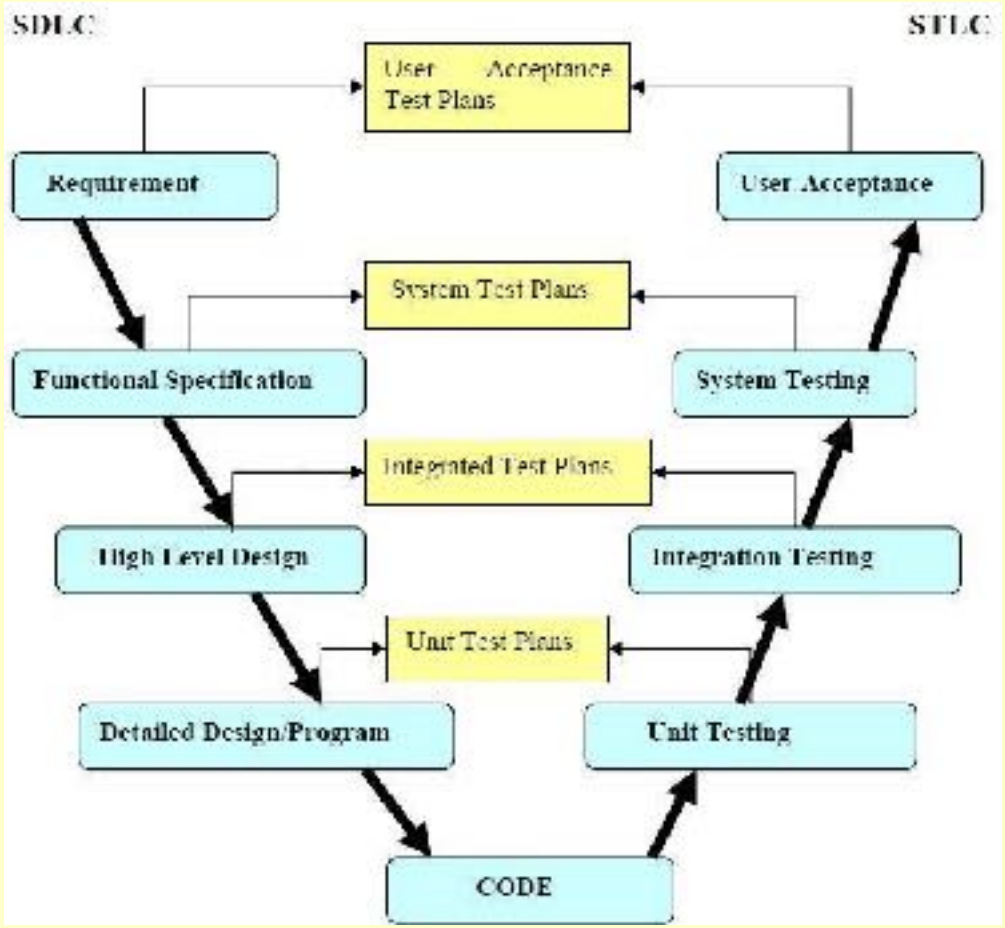


Do we test our software like this?

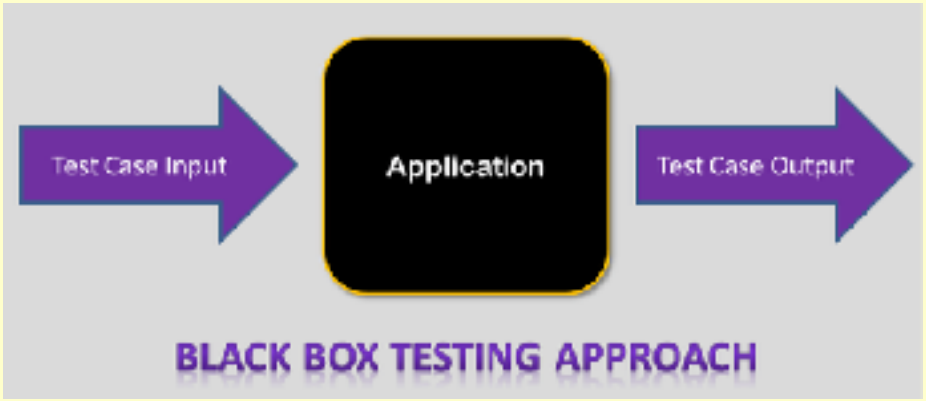


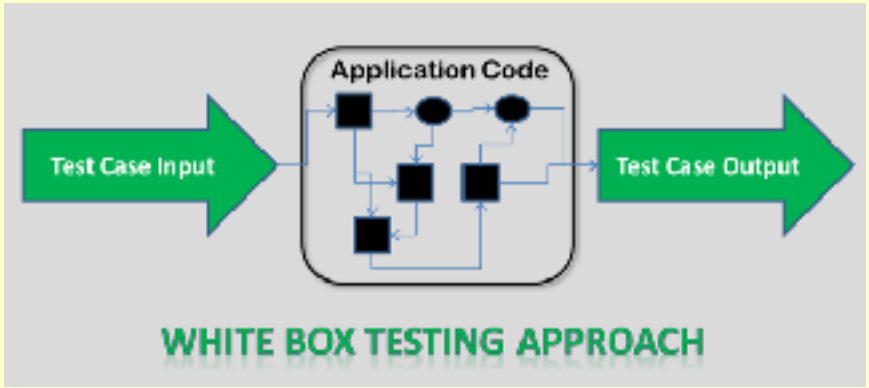
Testing : in the V life cycle

Software Development Life Cycle **Software Test Life Cycle**



Black box or White box Testing





Question: advantages and disadvantages of each?

Unit Testing

Most (nearly all) programming languages have automated tool support for unit testing (as well as other types of testing)

JUnit CUnit xUnit etc ….

Whenever you learn a new programming language, learn the testing tool(s) that come with it

Automated Unit testing is very valuable and beginners to programming need to learn it ASAP

http://blog.smartbear.com/automated-testing/a-short-lecture-on-the-value-and-practice-of-unit-testing/

Integration Testing

Why do we not just do unit tests?

"2 unit tests, zero integration tests"

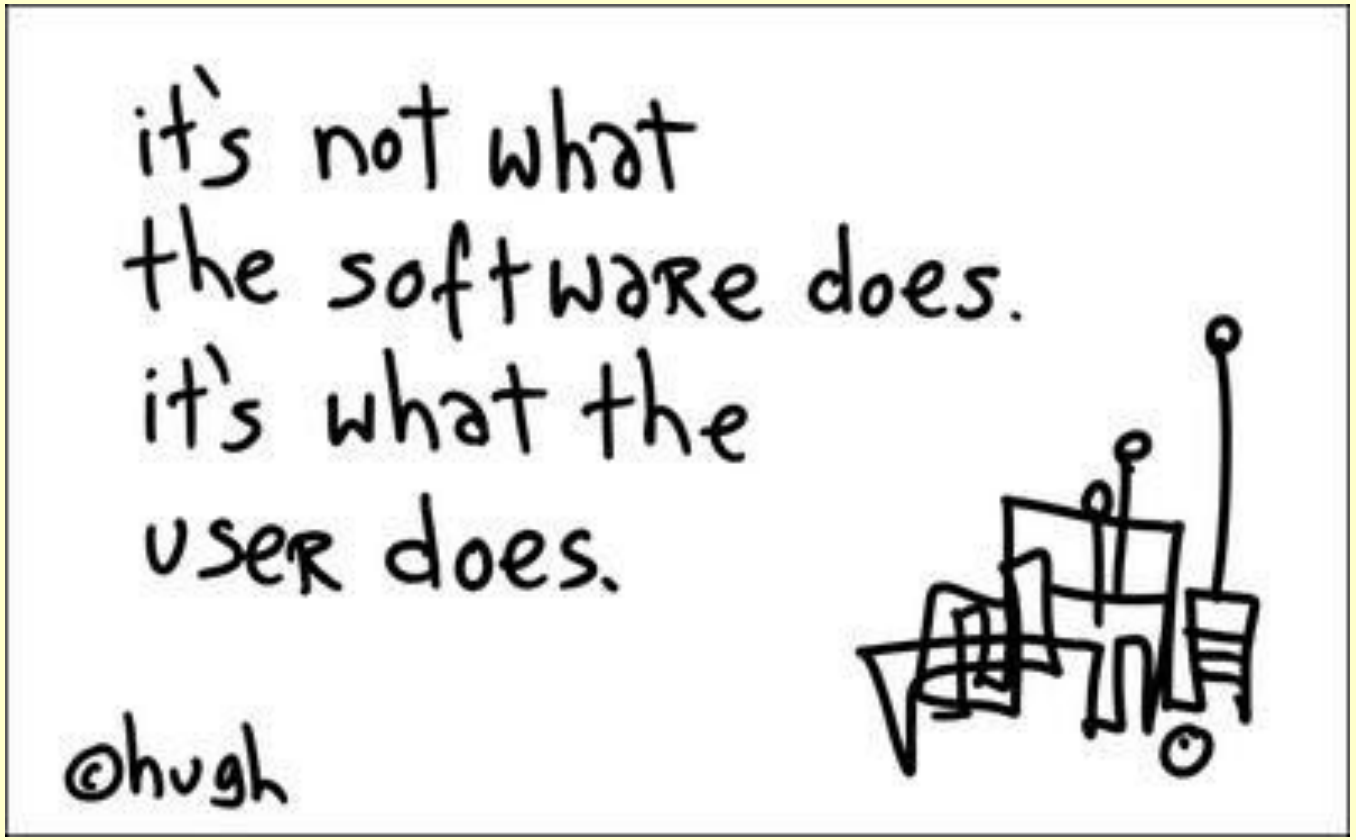
http://i.imgur.com/qSN5SFR.gifv

Why do we not just do validation tests?

Integration Testing

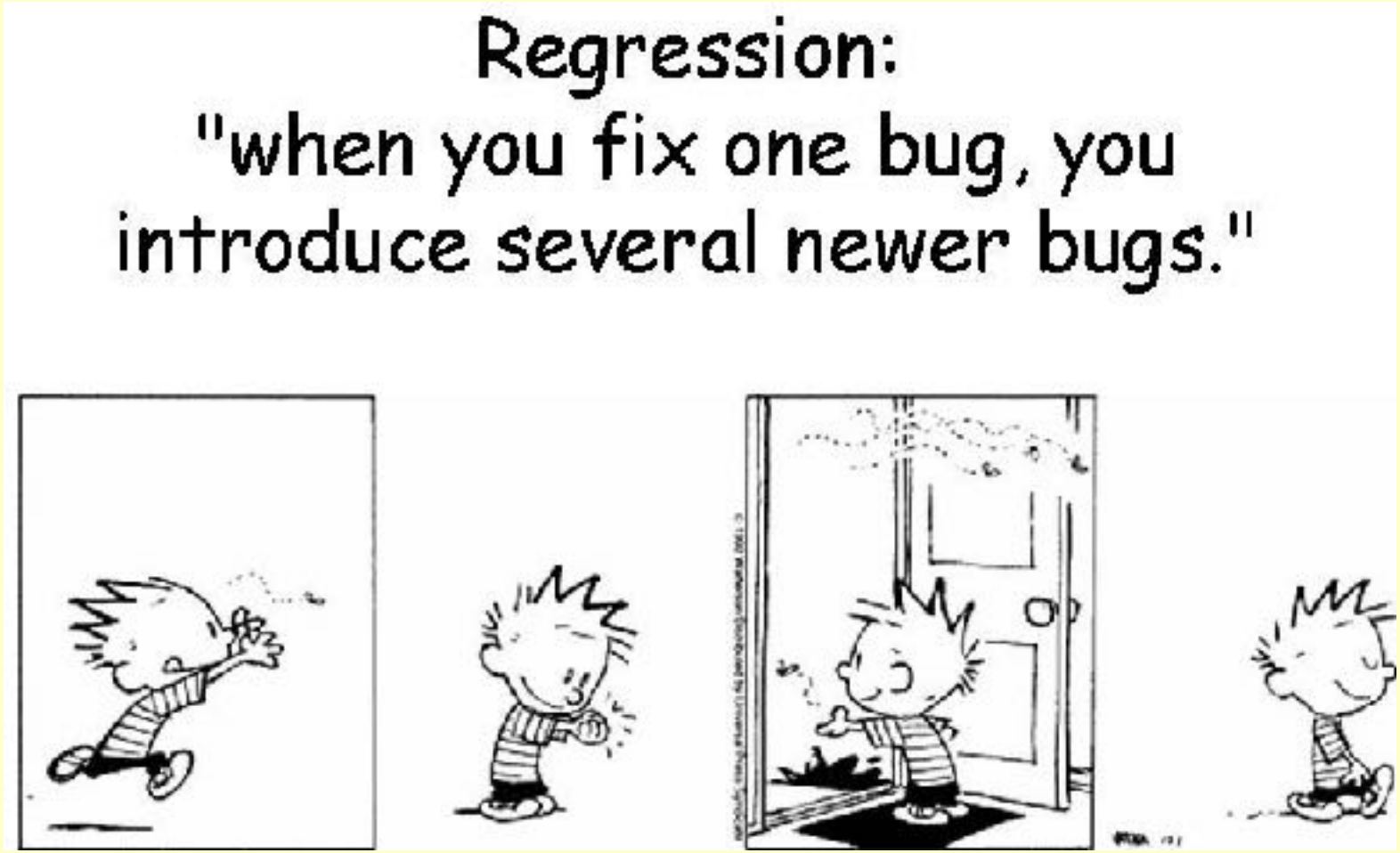
Fixing problems later in development can cost much more than fixing them earlier - but you have to detect them first

System Testing

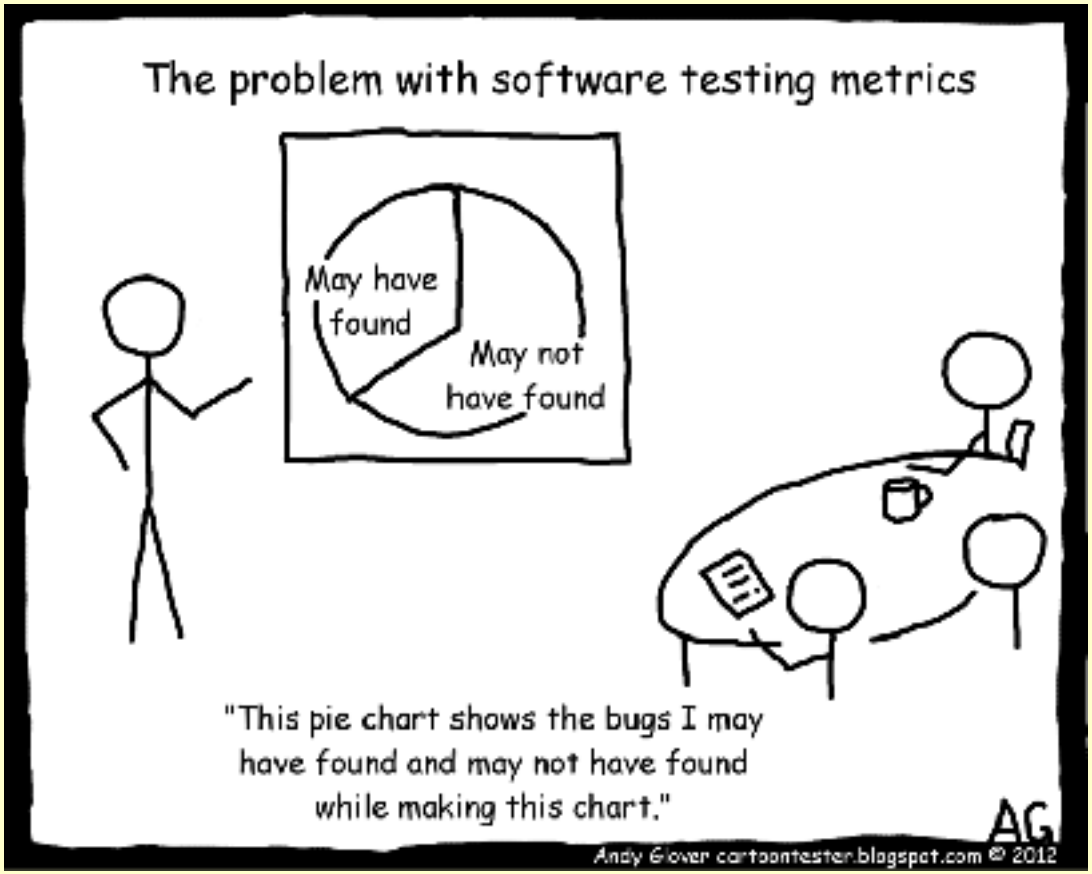


Never underestimate the users' ability to surprise

Regression Testing



Testing Metrics



Testing Code Coverage

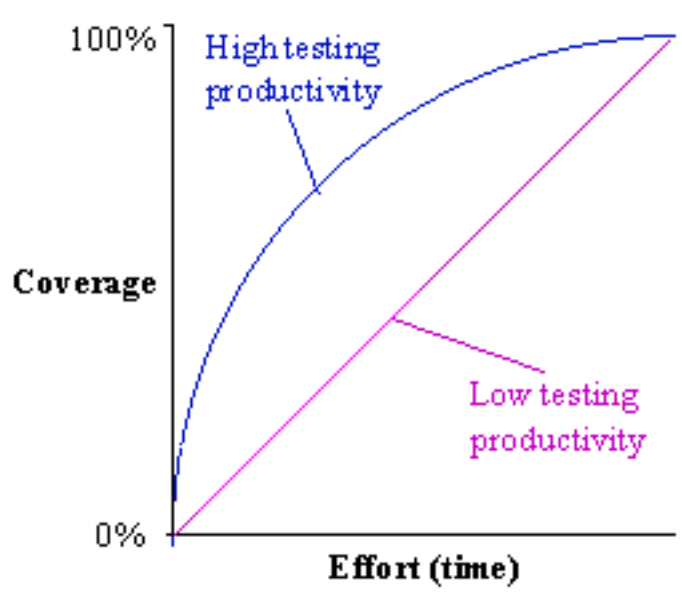


Figure 1: Coverage rate

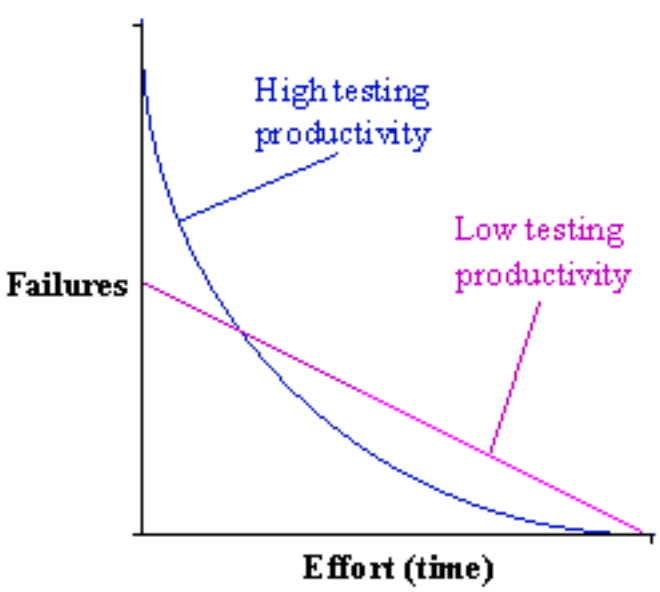
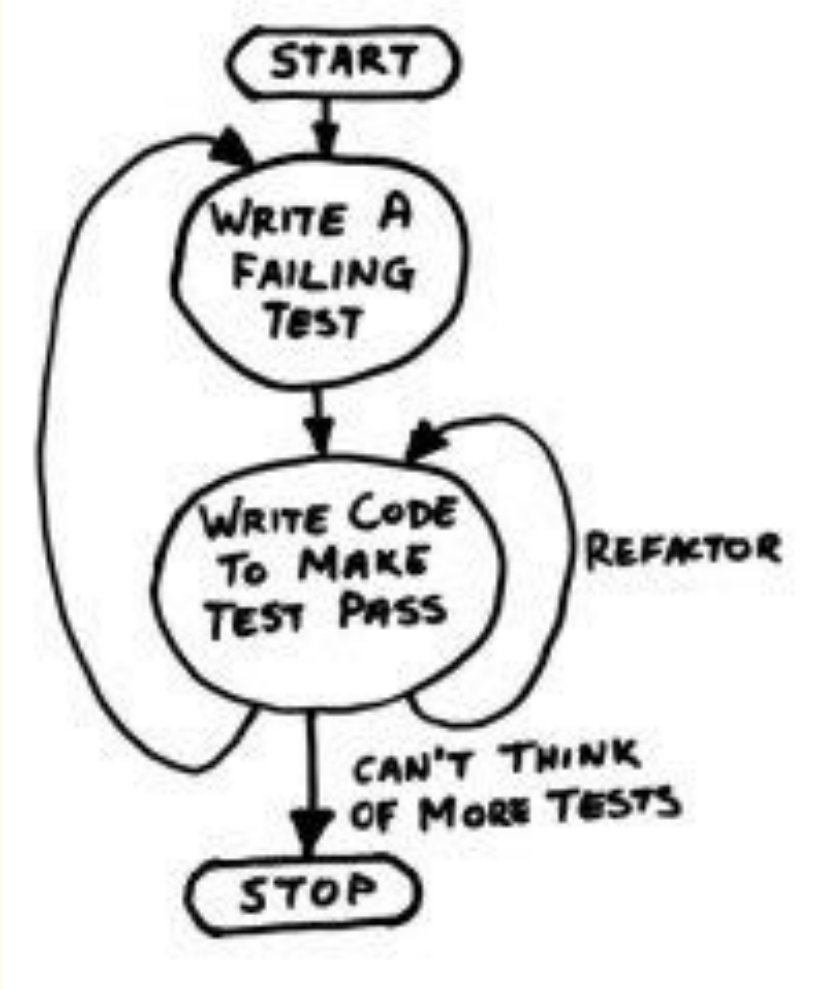


Figure 2: Failure discovery rate

Most testing tools come/work with coverage tools

Test first development



Some other testing types

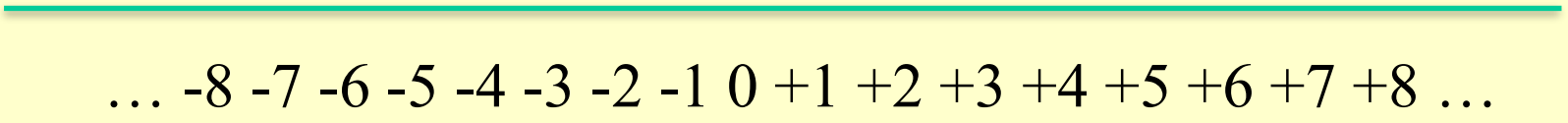
Functional - Nonfunctional
Performance
Usability
Security
Accessibility
Internationalisation/ localisation etc …

http://testautomation.applitools.com/post/98802238427/41-awesome-quotes-about-software-testing

Your test code needs testing?

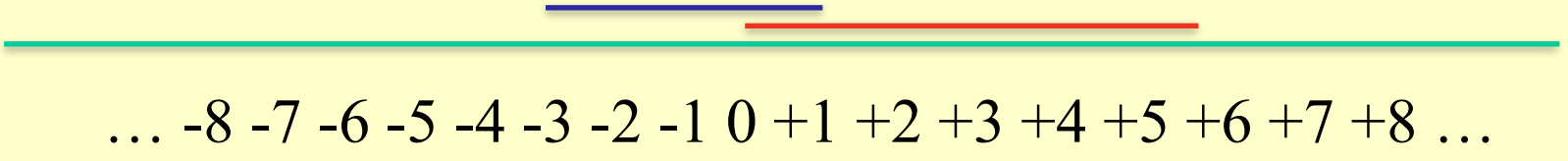
The Line Overlap Problem

Consider the integer number line:



We can define a segment on this line by a range (minimum … maximum)

Below, we illustrate 2 segments: **(-3, 1)** and **(0,6)**



In this example, the 2 segments are said to overlap on the line because they share at least 1 point in common.
The overlap in this case is the segment (0, 1).

The Line Overlap Problem

Requirements and Tests

The problem is to write a program that can calculate whether any 2 segments overlap on the integer line. It is to return the segment overlapped as the result of the program (an "empty" segment if there is no overlap)

You are to specify and implement a test set for this problem. You are not to code a working solution until after your tests are coded.

Your test code should be written in the same programming language as the solution(s) which you will be expected to test.

The code (including tests) must be well documented.

Illustrate that your tests can find errors in an incorrect solution

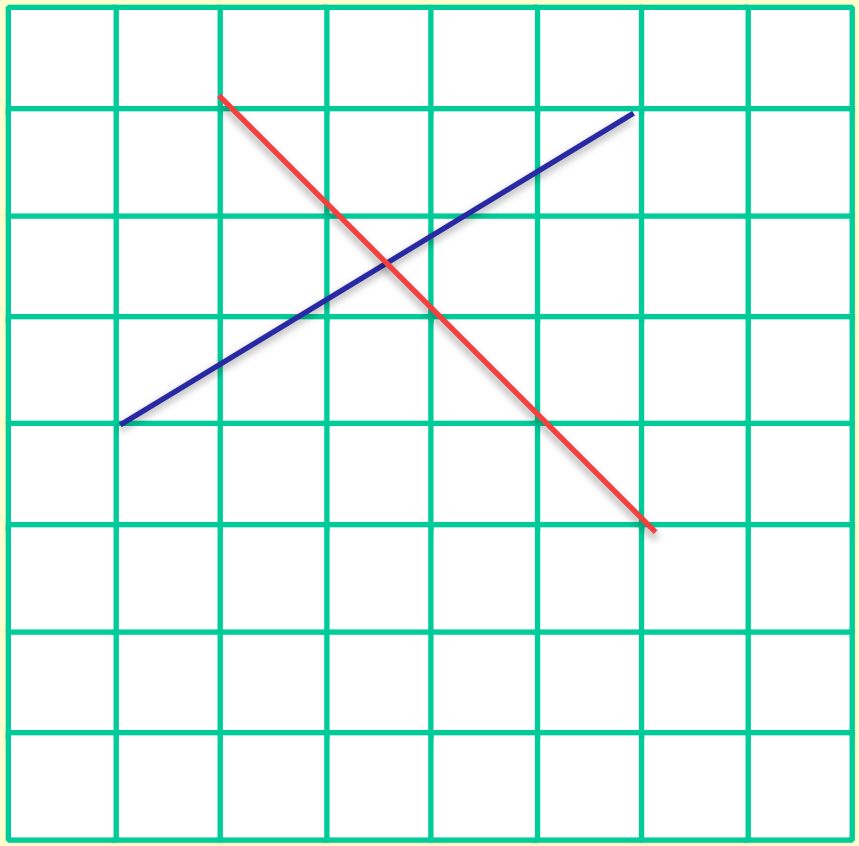
The Line Overlap Problem

Complete Tests

Given the minimum and maximum values, how many tests must be executed if we wish to test (exhaustively) every possible case?

The Line Overlap Problem: Part 2 - An extra dimension

Consider a 2-dimensional grid



Segment ((2,7), (6,3))
Segment ((1,4), (6,7))

The problem is to tell whether the 2 segments intersect on the specified integer *grid*.

In the example, the answer is clearly yes.

Note that we do not need to calculate the point of intersection

The Line Overlap Problem: Part 2 - An extra dimension

Requirements and Tests

The problem is to write a program that can calculate whether any 2 segments overlap on the integer grid.

You are to specify and implement a test set for this problem. You are not to code a working solution until after your tests are coded.

Your test code should be written in the same programming language as the solution(s) which you will be expected to test.

The code (including tests) must be well documented.

Illustrate that your tests can find errors in an incorrect solution

The Line Overlap Problem: Part 2 - An extra dimension

Testing a design hypothesis for re-use of code.

I would like to be able to re-use my code to the first 1-dimensional overlap problem in order to solve the problem in 2 dimensions.

It is suggested that 2 lines overlap in a grid if and only if they 'overlap vertically' and 'overlap horizontally'.

Intuitively, this seems right. However this needs to be tested. For example:

Segment ((2,7), (6,3))
Segment ((1,4), (6,7))

overlap iff

(2,6) and (1,6) overlap
and
(7,3) and (4,7) overlap

Is this true for all such segments in the grid space?

Problem Analysis

What did you learn about testing from this problem?

Did you try to use a testing tool, like JUnit or CUnit?
If so, how did you find it?
If not, what sort of things would you like such a tool to be able to do?

Did you find any bugs in your test code?

How good are your tests?
How do you judge the quality of the tests?

Did your tests help you to find bugs in your solution code?