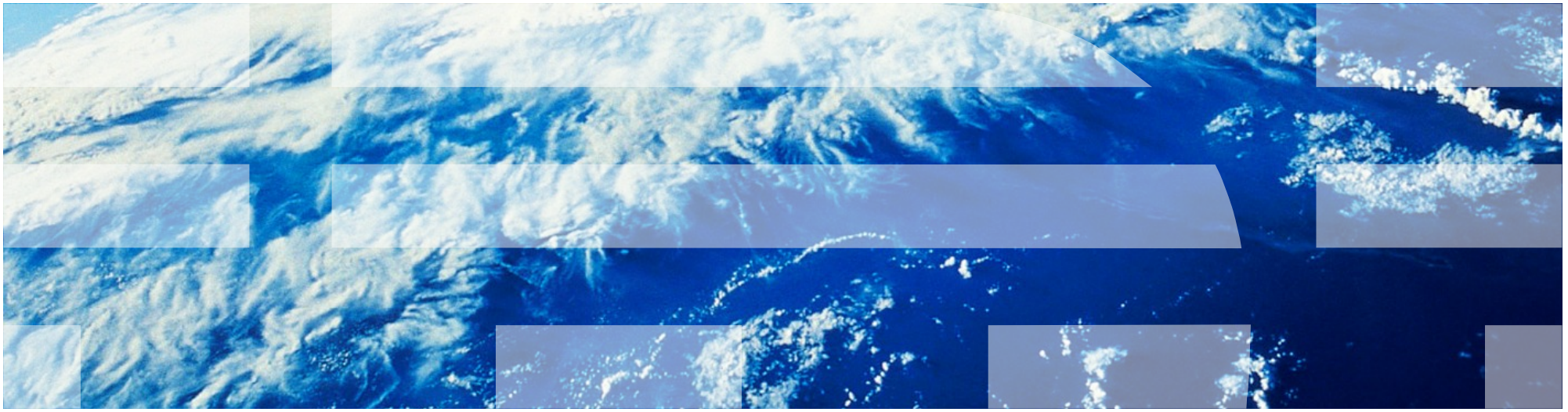
Software Testing
Theory, Practise and Reality



Who am I?

- David Vines (dvines@uk.ibm.com)
- Degree in Computer Science and Operational Research
- Joined IBM in 1984
- Been involved in product development ever since

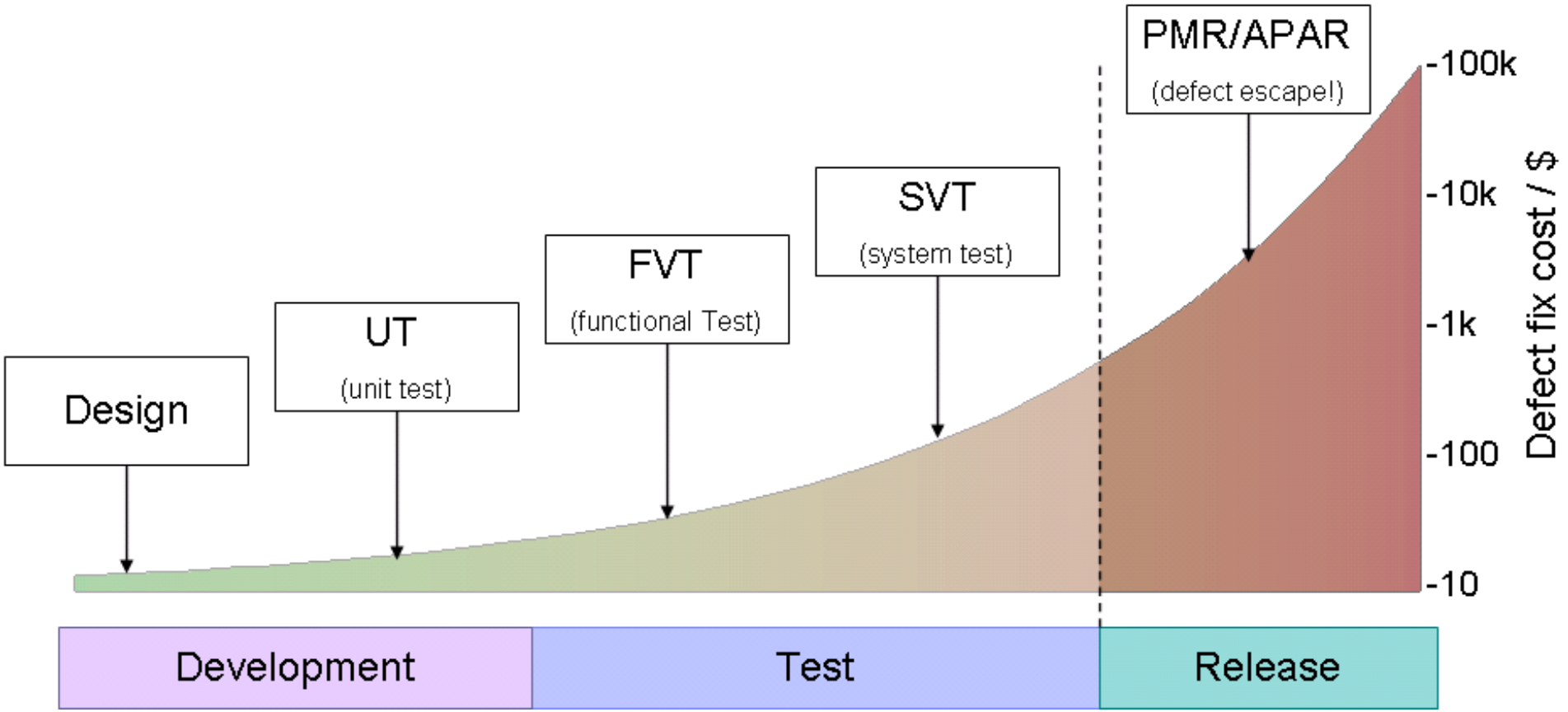
A test problem

- Given the following specification:

  The program reads three integer values. The three values are interpreted as the lengths of the sides of a triangle. The program prints a message stating if the triangle is scalene, isosceles or equilateral.

- What's the set of test cases to adequately test this program?

Why test?



Test Case: 3, 4, 5 – Expected output: SCALENE

Why test?

- "If it can go to wrong, it will"
  - (and usually at the worst possible time)
  - NASA's Mars Climate Orbiter - $327.6 million space probe burnt up in the Martian Atmosphere
  - First test flight of the Ariane 5 – more than $370 million rocket destroyed 37 seconds after launch
  - Northeastern US power blackout 2003 – estimated between $6.8 billion and $10.3 billion

Test Case: 5, 5, 5 – Expected output: EQUILATERAL

What is testing?

- The science of detecting failure
- Verifying that a product is fit for purpose
- Evaluating whether a system meets its requirements
- Creatively trying to break the system
- Testing does NOT add quality – it can only reveal the existing lack of quality

Test Case: 5, 5, 3 – Expected output: ISOSCELES

Different types of tests

- How much of the system is tested
  - Unit test
  - Functional Verification
  - System Verification
- What is tested
  - Function
  - “Non-functional”
    - Performance
    - Stress
    - Accessibility
    - Internationalization
    - Installation
    - Security
- Who's involved
  - Development Team
  - Selected Customers
    - Closed Beta
  - Anyone
    - Open Beta
- Repeatability
  - Fully Automated
  - Manual
    - Scripted
    - Exploratory

Test Case: 5, 3, 5 – Expected output: ISOSCELES

Unit Testing

- Tests of individual units (e.g., classes) of source code
  - Typically written by the author of that source code
  - Verify that the code does what the author intended the code to do
  - Running of the unit tests is fully automated

Test Case: 2, 5, 5 – Expected output: ISOSCELES

Functional Verification Test (FVT)

- Black Box Testing

- Main Aspects are:
  - **Coverage**: Ensuring all functions are called
  - **Variation**: Call functions with a variety of arguments
  - **Sequencing**: test a sequence of steps, not just one

- Big Problem: Consider the triangle problem – Just how many possible testcases are there?
  - Answer: Just considering three positive 31-bit integers, $2^{93} \approx 10^{31}$
  - (or, assuming 1nS per test: 313,823,621,387 years (over 20 times the age of the universe!)

Test Case: 5, 2, 4 – Expected output: SCALENE

Functional Verification Test

- Selecting a subset of possible testcases:
  - Random Testing
  - Equivalence class partitioning
  - Testing boundary conditions
  - Model-based testing

Test Case: 5, 1, 3 – Expected output: NOT A TRIANGLE

System Verification Test

- Tests the complete system

- Usually checks for more than just “is the output correct?”
  - Stress/Performance
  - Variations of environment (Hardware/Software)
  - Recovery
  - Security
  - Interoperability
  - Usability

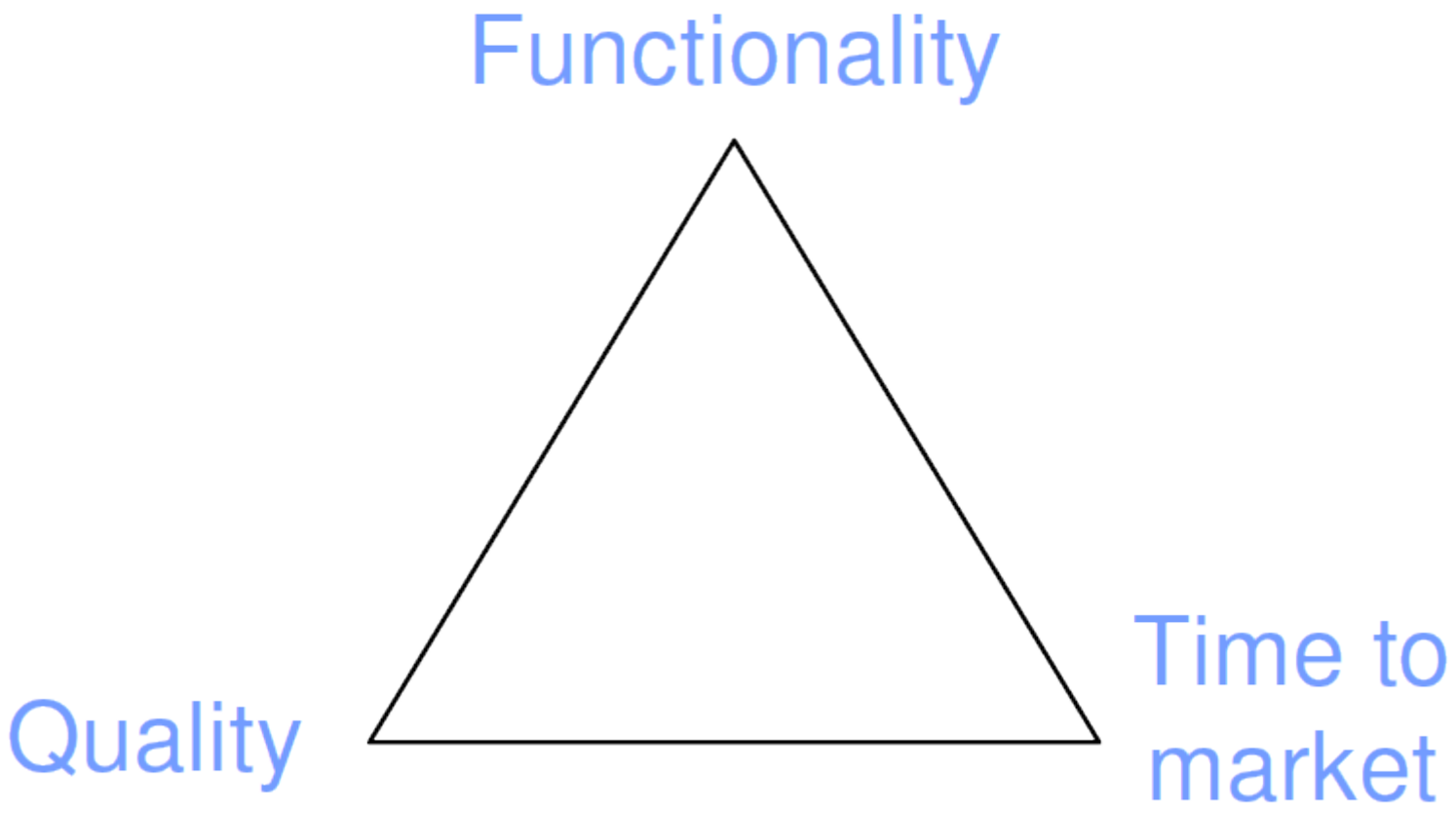
Test Case: 5, 1, 0 – Expected output: NOT A TRIANGLE

System Verification Test

- Need to think/act as the customer will
- Try all those strange things that the customer will try
- Usually done by someone other than the original developers
  - You want/need the different mindset/viewpoint
  - Common developer response to the defect reports is “I didn't expect anyone to do THAT!”
- Use cases can be a good starting point

Test Case: 5, 6, -3 – Expected output: INVALID INPUT

Testing in Reality



Test Case: 0, 0, 0 – Expected output: INVALID INPUT

Testing in Reality

- Need to choose:
  - What to test
  - How to test

- Typically:

  - Focus on testing new functionality

  - Regression testing of old function

  - Automate as much as possible
    - Unit tests are usually run as part of the build process
    - FVT tests also run as part of the build process
    - SVT tests need to use the real product deliverables – but can still be automated

Test Case: 1, 2 – Expected output: INVALID INPUT

Testing in Reality

- How much to test?
  - Testing can expand to occupy the time and money available
- How much to spend?
  - Typically 50% of the resource is spent on testing
- Don't forget that the defect fixes need testing too.....

Test Case: 1.5, 2.5, 3.5 – Expected output: INVALID INPUT

Test cases for the test problem

1) A test for a valid scalene triangle ✓

2) A test for a valid equilateral triangle ✓

3) A test for a valid isosceles triangle ✓

4) Two more tests for isosceles to cover the other two permutations ✓

5) A test in which one side has a value of zero ✓

6) A test where at least one side has a negative value ✓

7) A test with three positive integers where the sum of two of them equals the third (e.g., 1, 2, 3) X

8) Two other tests similar to 6 that cover the other two permutations X

9) A test with three positive integers that do not make a triangle (e.g. 1, 2, 4) ✓

10) Two other tests similar to 9 that cover the other two permutations X

11) A test where all three sides are zero ✓

12) A test where at least one side is not an integer ✓

13) A test where the wrong number of values ✓

14) BONUS: Did you specify the expected output? ✓

Bibliography

- The art of Software testing (1979) – Glenford J. Myers
  - ISBN 0 471 04328-1