MSOE

UP Password → SE2832MSOE

Test Activities

SE 2800

Lecture Objectives:

Discipline

1) List the activities conducted by test engineers
2) Draw the software V Model and explain the relationships between testing activities and design activities.
3) Explain the relationship between acceptance testing, system testing, integration testing, module testing, and unit testing.
4) Explain Beizer's Testing Levels and Test Process Maturity
5) Define the relationship between testing and debugging.
6) Explain the relationship between the cost of fixing a defect and the phase in which the defect is discovered.

Test Activities

Lecture Objectives:

1) List the activities conducted by test engineers
2) Draw the software V Model and explain the relationships between testing activities and design activities.
3) Explain the relationship between acceptance testing, system testing, integration testing, module testing, and unit testing.
4) Explain Beizer's Testing Levels and Test Process Maturity
5) Define the relationship between testing and debugging.
6) Explain the relationship between the cost of fixing a defect and the phase in which the defect is discovered.

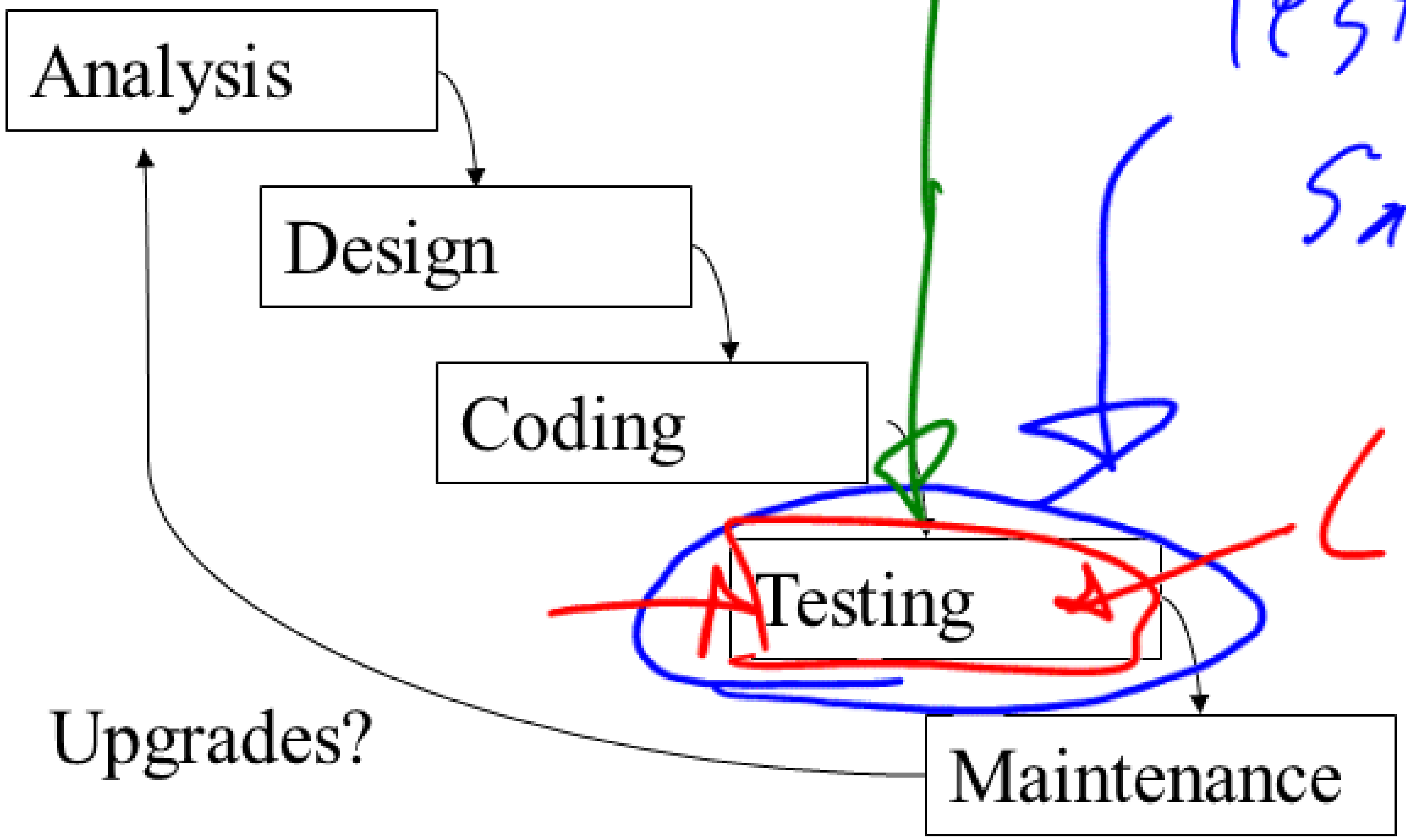
Waterfall Model

All
- Traditional approach

All problems of Ly OF all

Late to start testing.

Testing/coding @ same time.



Late

Test Design in Context

Figuring out the right values to provide to the SW.

- *Test Design* is the process of designing input values that will effectively test software

- Test design is one of several activities for testing software
  - Most mathematical
  - Most technically challenging

Skills → Discrete Math

Reduce # of test cases w/o reducing coverage.

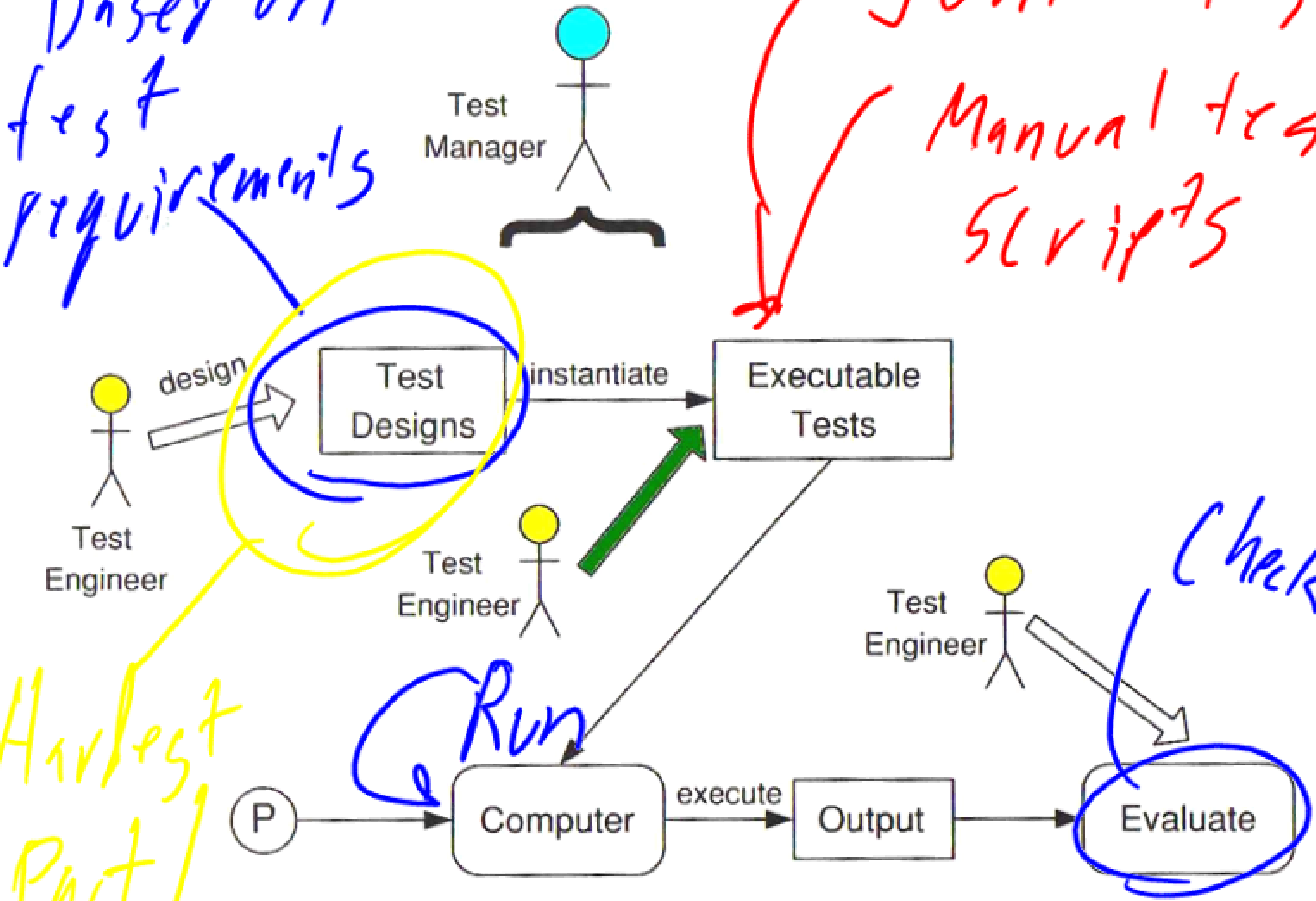
Test Engineer & Test Managers

- Test Engineer : An software professional who is in charge of one or more technical test activities
  - designing test inputs ← Test cases/approaches?
  - producing test values
  - running test scripts
  - analyzing results ← How bad is the failure?
  - reporting results to developers and managers

Boss

- Test Manager : In charge of one or more test engineers
  - sets test policies and processes
  - interacts with other managers on the project
  - otherwise helps the engineers do their work – Aids

Activities of Test Engineers



Based on test requirements

JUnit tests?, Manual test scripts

Run

Checked

Hardest Part
Most time consuming.

1. Test Design – (a) Criteria-Based

theory

- This is the most technical job in software testing
- Requires knowledge of :
  - Discrete math
  - Programming
  - Testing

Mathematical Coverage.

- Requires much of a traditional CS degree
- This is intellectually stimulating, rewarding, and challenging
- Test design is analogous to software architecture on the development side
- Using people who are not qualified to design tests is a sure way to get ineffective tests

Very easily.

Engineering approach.

1. Test Design – (b) Human-Based

Hard

- This is much harder than it may seem to developers
- Criteria-based approaches can be blind to special situations

Problem domain

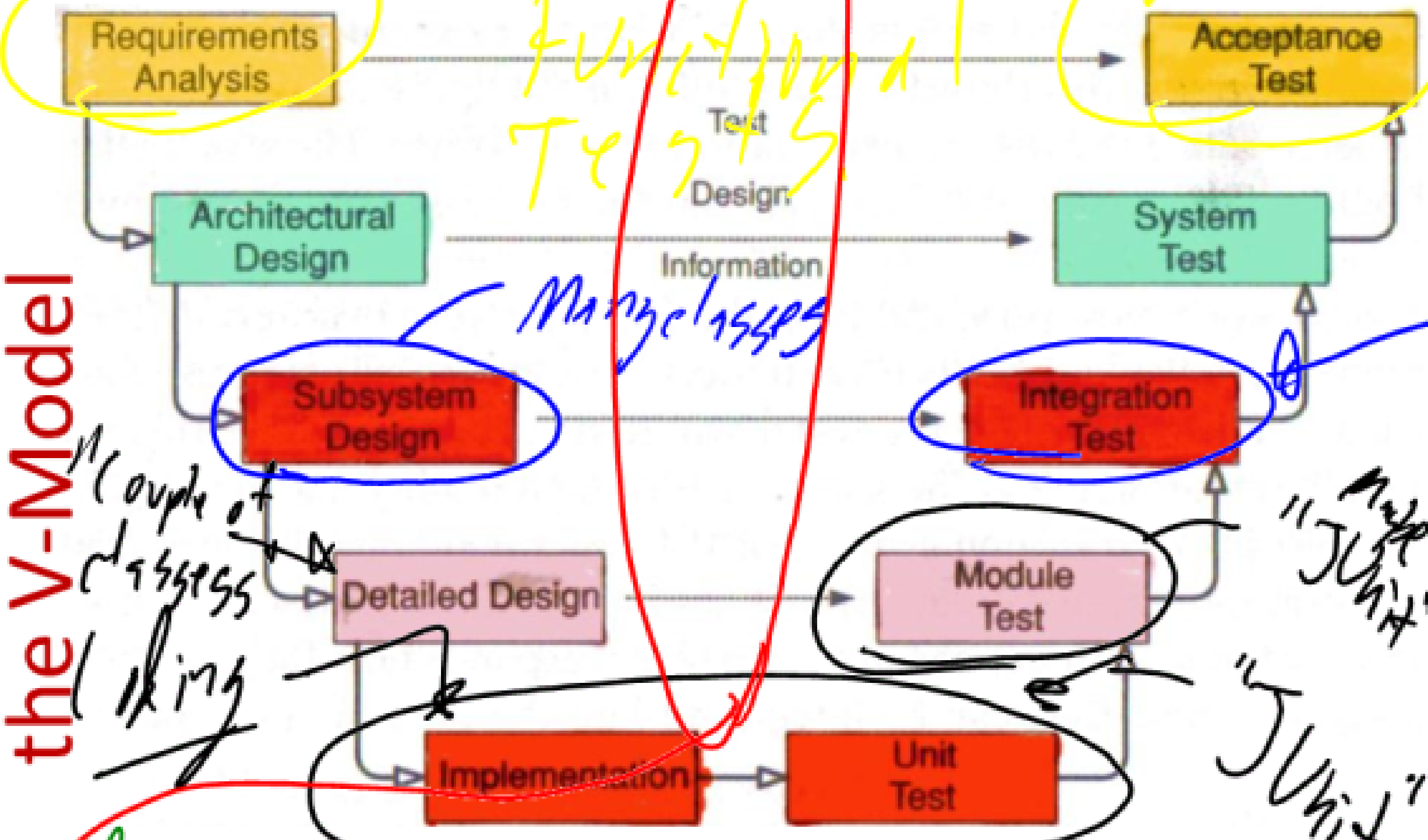
- Requires knowledge of :
  - Domain, testing, and user interfaces

Usability

- Requires almost no traditional CS
  - A background in the domain of the software is essential
  - An empirical background is very helpful (biology, psychology, …) ⇒ English
  - A logic background is very helpful (law, philosophy, math, …)

Lawyers

Software Development and the V-Model



Functional Tests

Many classes

"Couple of classes talking"

Make "JUnit"

"JUnit"

"Sequences"

Testing the interaction of multiple classes.

Relationship between testing and development.

Test design.

Testing Activities

- Acceptance Testing
  - Assess the software with regards to the requirements
- System Testing
  - Assess the software with respect to the architectural design
- Integration Testing
  - Assess software with regards to subsystem design
- Module Testing
  - Assess the software with respect to detailed design

Definitions

- Unit
  - One or more contiguous program statements with an appropriate identifier
    - i.e. procedure, function, method

in code

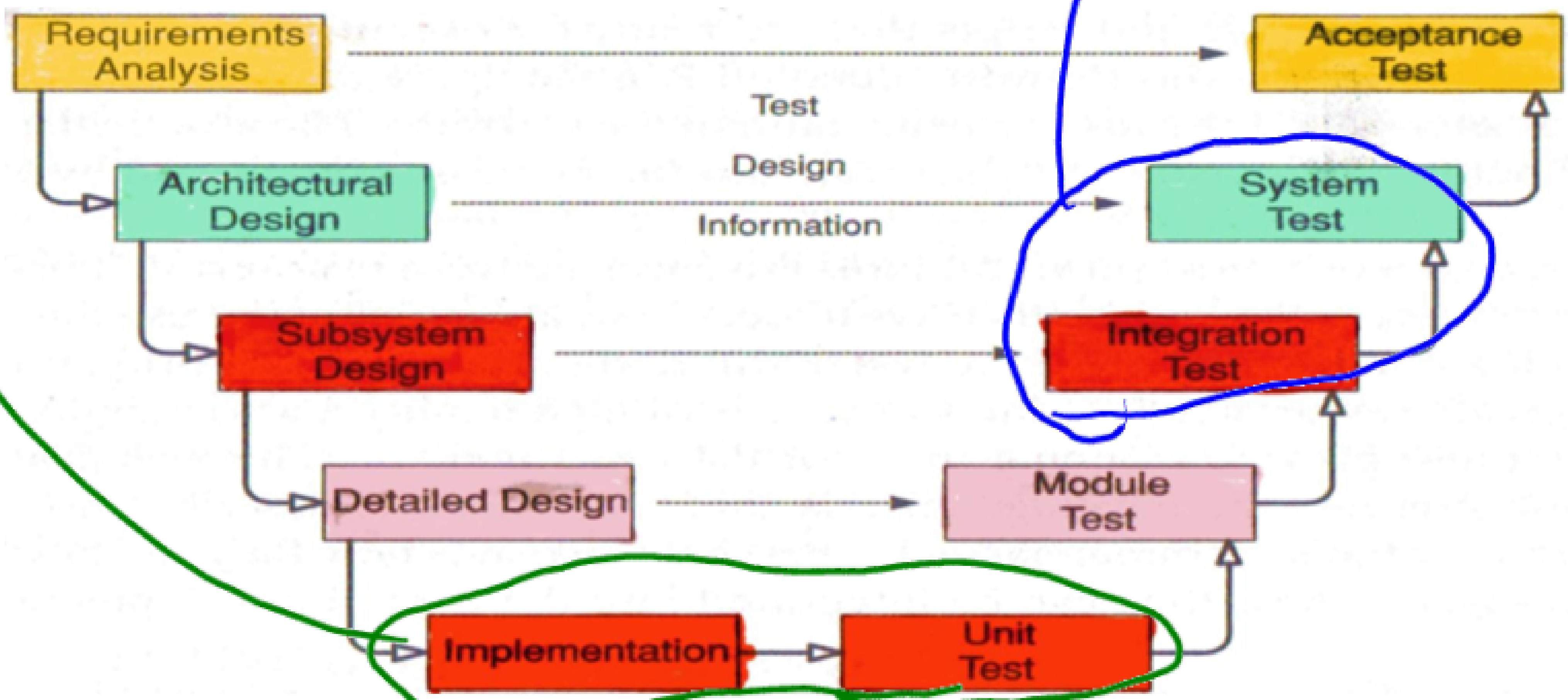
- Module → Higher Level.
  - A collection of related units assembled into a file, package, or class

Unit and module testing is done by the developer.

One set. May be other slightly different ones.

Using different levels of testing

- Different levels of testing should look for different types of faults
  - Case study: Mars lander of September 1999
    - One module used English units
    - One module used Metric units
    - When should this problem have been caught?
  - Zune failure
    - When should this problem have been caught?



Requirements Analysis

Acceptance Test

Test

Design

Information

Architectural Design

System Test

Subsystem Design

Integration Test

Detailed Design

Module Test

Implementation

Unit Test

Why have you tested your programs?

Requirement to test them.

Make sure it does what its is supposed to do.

Testing to make certain nothing broke that worked before.

Discussion: Why have you tested your programs?

Using Ubiquitous presenter, select one of the following which best represents why you have tested your programs.

To aid in removing your mistakes.

A. To debug my program

B. To prove that the software works

I want to show functionality

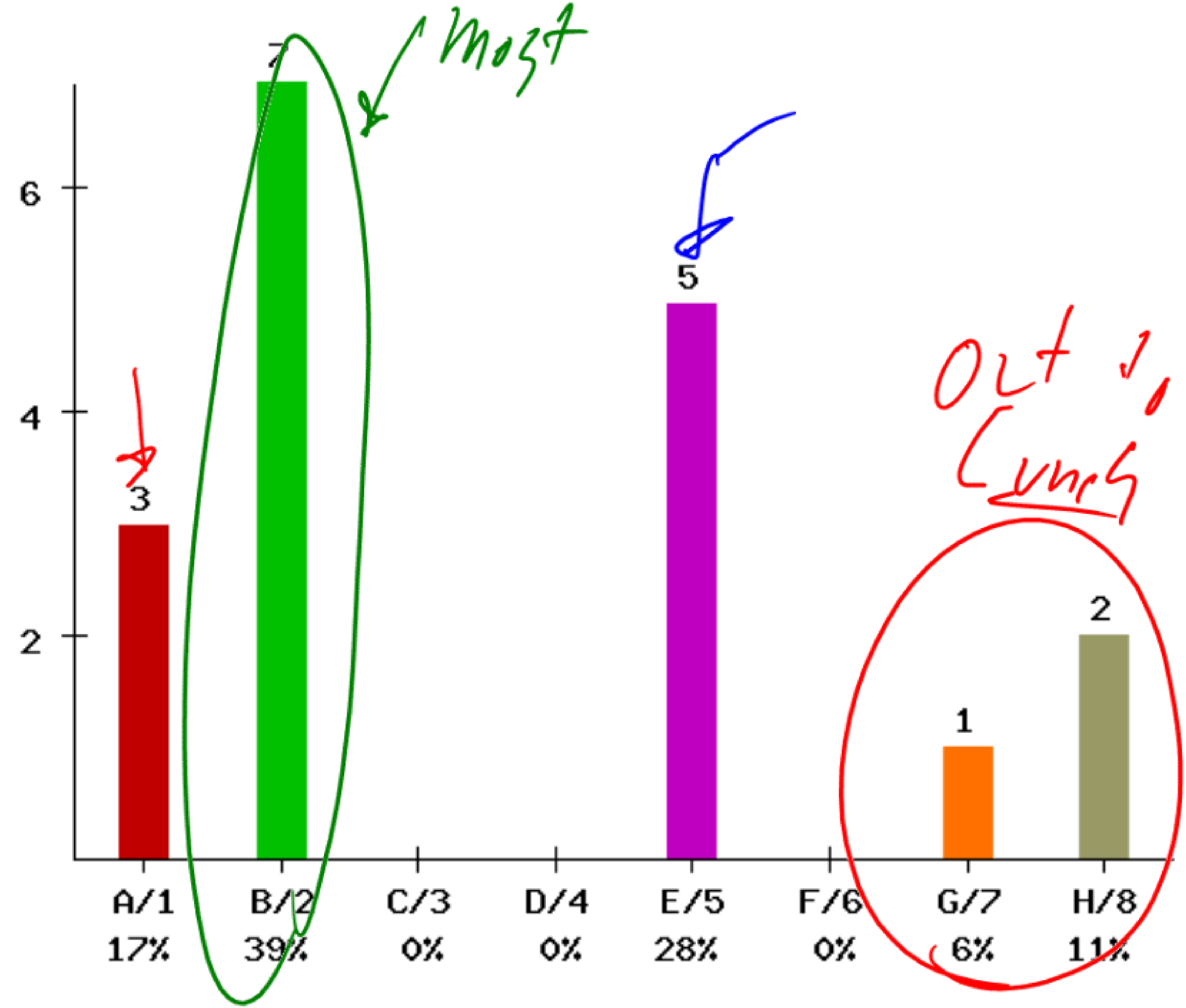
C. To prove that the software doesn’t work.

D. To reduce the risk of using the software.

Stable SW.

E. To develop higher quality software.

We want good SW.


Most
6
4
2
7
3
5
1
2
A/1
17%
B/2
39%
C/3
0%
D/4
0%
E/5
28%
F/6
0%
G/7
6%
H/8
11%
Outliers

Test Process Maturity

More Sophisticated

Thoughts/thinking

L4: Testing is a mental discipline that helps all IT professionals develop higher quality software.

L3: The purpose of testing is not to prove anything specific, but to reduce the risk of using the software.

Working

L2: The purpose of testing is to show that the software doesn't work.

L1: The purpose of testing is to show that the software works.

L0: There is not different between testing and debugging.

Definitions

- Testing

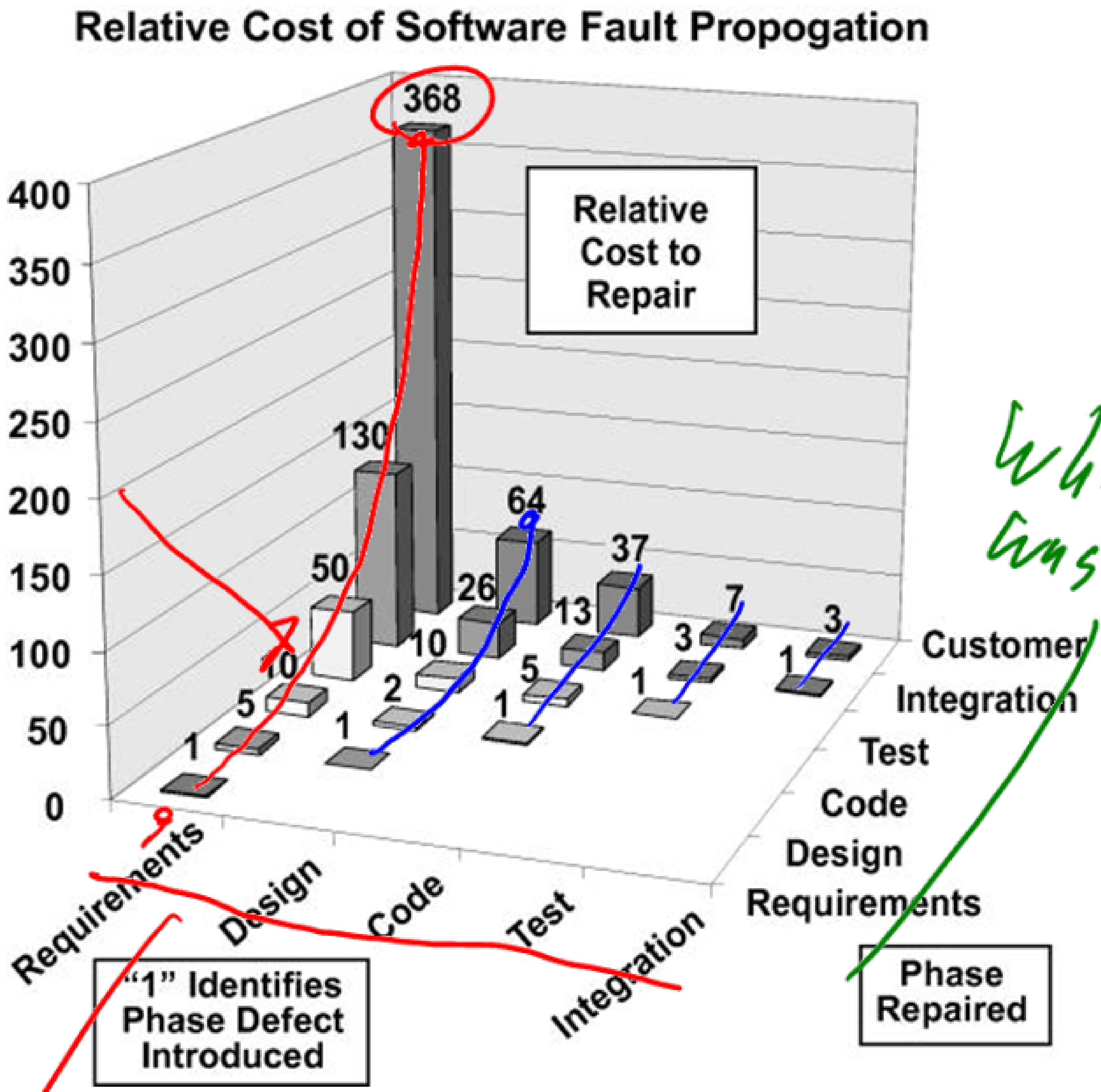
Values/ Conditions

  - Finding inputs that cause the software to fail

- Debugging
  - The process of finding a fault given a failure

the faults which caused a failure.

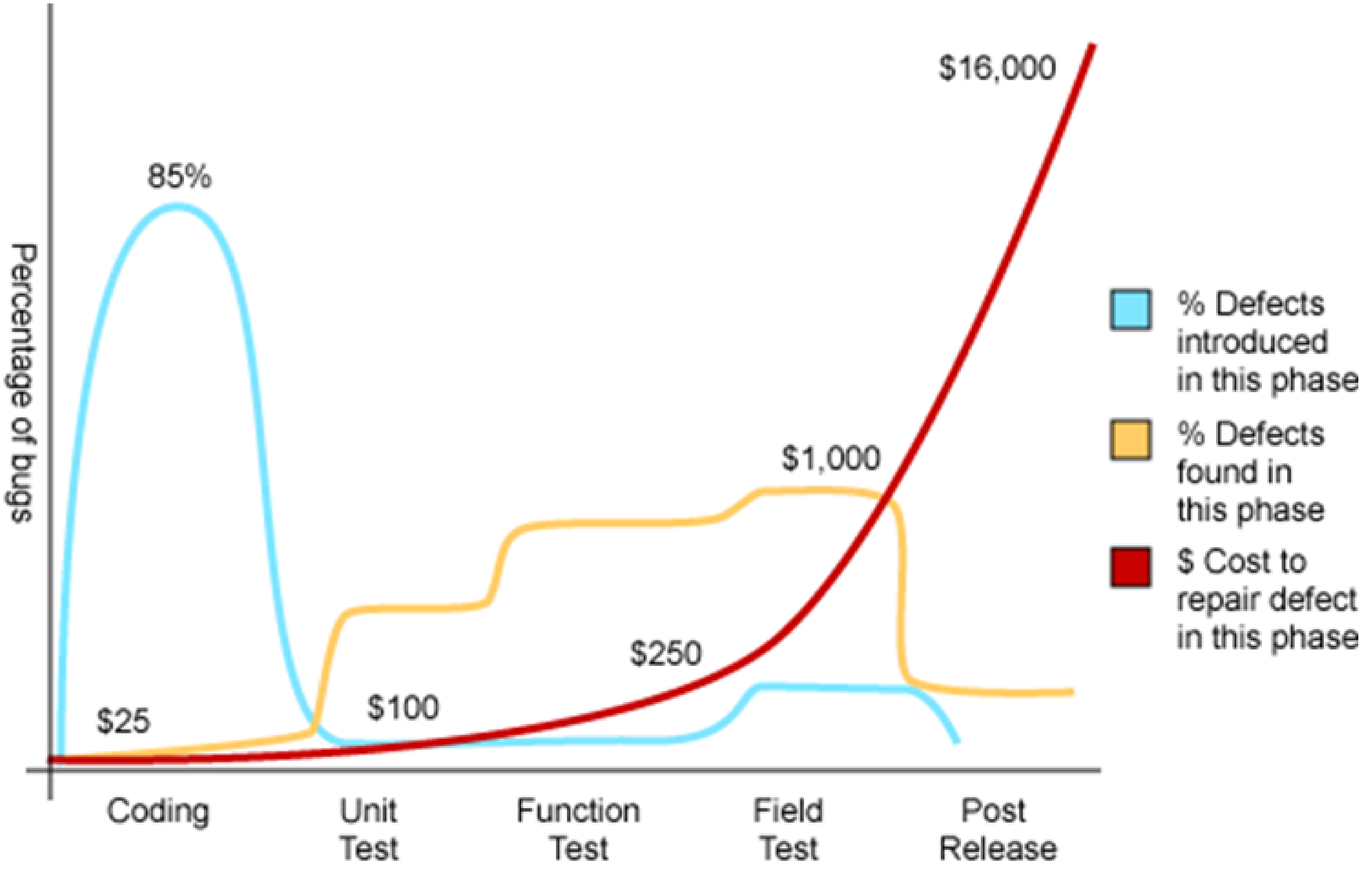
Cost of Software Fault Propagation



when was it fixed

when was the mistake made

Software Fault Propagation – Alternate Data



Source: Applied Software Measurement, Capers Jones, 1996