Introduction to Software Testing

Content

- What is software testing
- Why do we need software testing
- How do we do software testing

Software Quality Assurance

- Software testing is one of the methods to assure software quality.
- Software quality assurance involves validation and verification of software.

Validation: specification $\xrightarrow[\text{consistent}]{}$ requirements

Are we build the right software?

Verification: implementation $\xrightarrow[\text{consistent}]{}$ specification

Are we build the software right?

Software Verification

- Software can be verified statically or dynamically.
- Static verification techniques contain review, inspection, walkthrough, and analysis.
- Dynamic verification techniques contain various testing techniques.

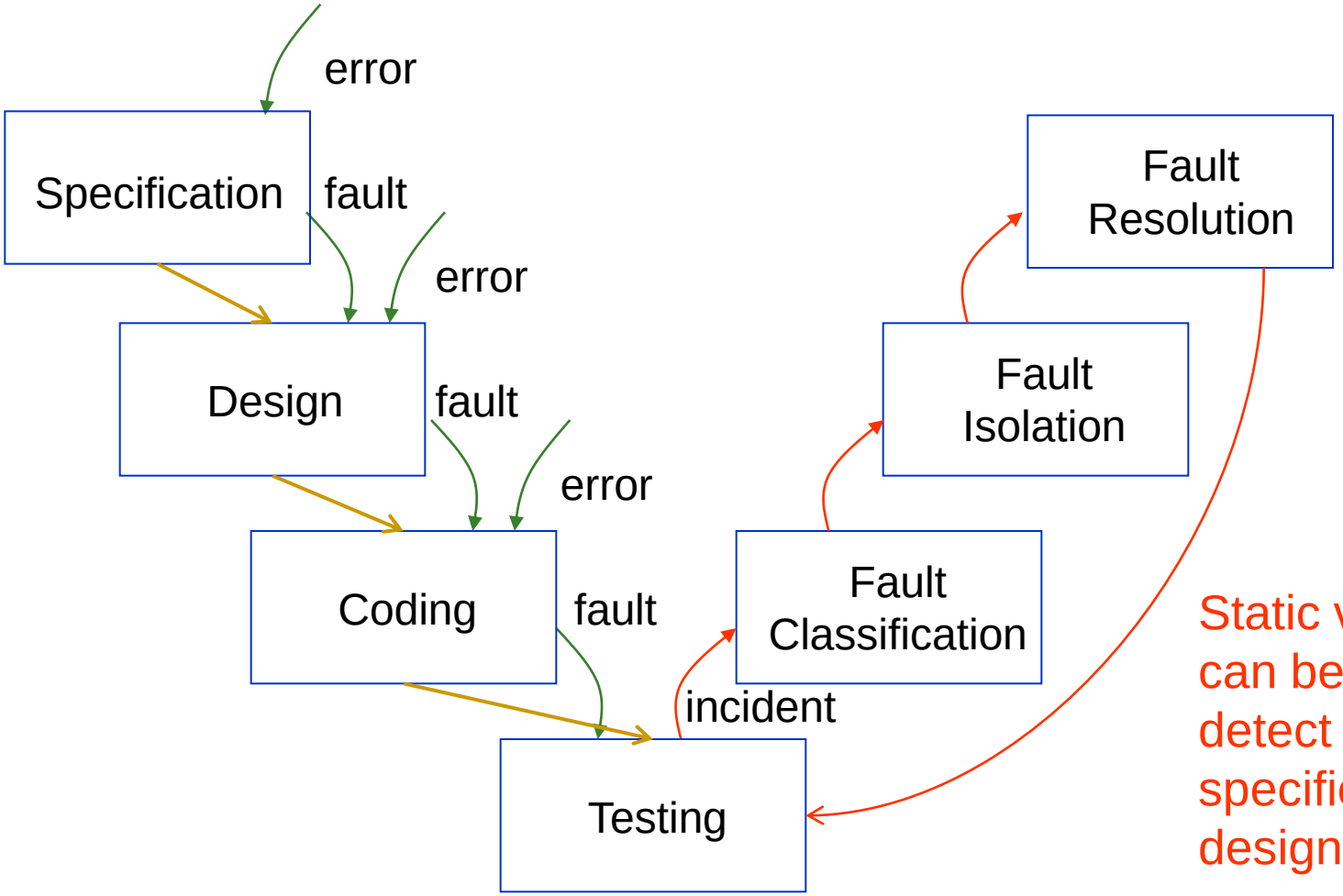
Error, Fault, Failure, and Incident

- People make errors.
- A fault in the documents or programs is the result of an error.
- A failure occurs when a fault executes.
- When a failure occurs, it may or may not be readily apparent to the user. An incident alerts the user to the occurrence of a failure.

What Is Software Testing

- Software testing is the act of executing software with a suite of test cases so that it can either find faults in the program or demonstrate the program is correct.
- Each test case is associated with a specific program behavior. A test case contains a list of test inputs and a list of corresponding expected outputs.
- It is difficult to design a suite of test cases that can prove a program is correct.

The Testing Life Cycle



Static verification can be used to detect faults in specification and design early.

Why Do We Need Software Testing

- Software prevails in our living environment. Quality of software significantly influences our quality of life.
- Software faults in critical software systems may cause dramatic damages on our lives and finance.

Example: THERAC-25 Radiation Therapy

- In 1986 two cancer patients at the East Texas cancer Center received fatal radiation overdoses from the computer-controlled radiation therapy machine.
- A software bug on mishandled race condition.

Example: Shooting down Airbus 320

- In 1988 USA troops mistook an Airbus 320 as an F-14 and shot down the Airbus 320.
- 290 people dead.
- A software bug in the tracking software.

Example: London Ambulance Service

- London Ambulance Service Computer Aided Dispatch automates many of the human-intensive processes of manual dispatch systems associated with ambulance services in the UK.
- Failure of the London Ambulance Service on 26 and 27 November 1992.
- At 23:00 on October 28 the LAS eventually instigated a backup procedure, after the death of at least 20 patients.

Example: Mars Climate Orbiter

- In 1999, the Mars Climate Orbiter is supposed to relay signals from the Mars Polar Lander once it reached the surface of the planet.
- The Mars Climate Orbiter smashed into the planet instead of reaching a safe orbit.
- A software bug that fails to convert English measures to metric values.
- This incident costs $165 million.

Economic Impact

- In 2002 a study commissioned by the National Institute of Standards and Technology found that software errors cost the U.S. economy about $59.5 billion annually, or about 0.6 percent of the gross domestic product.

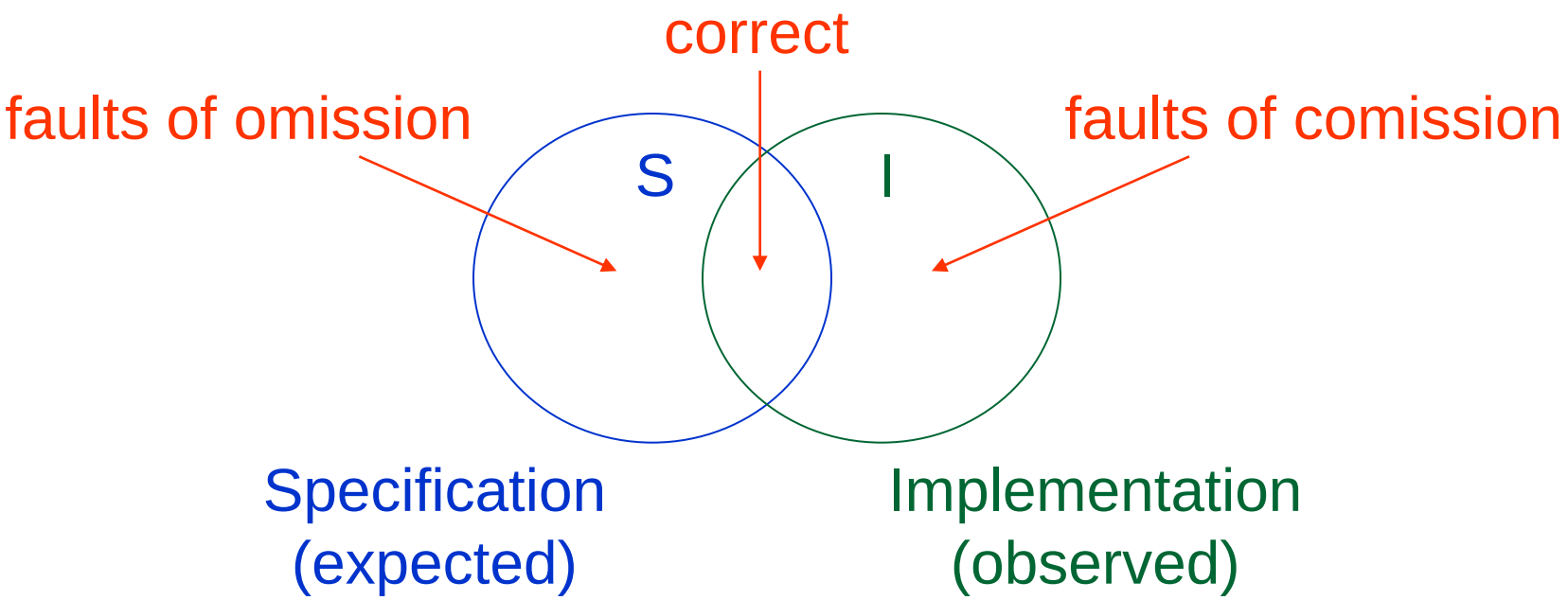
Current Status

- Carefully made programs have 5 faults per 1000 lines of code (LOC).
- Windows XP has 45M LOC, so it may have 225000 faults.

Comparison with Automobile Industry

- If the automobile industry had developed like the software industry, we would all be driving $25 cars.
- If car were like software, they would crash twice a day for no reason, and when you called for service, they would tell you to reinstall the engine.

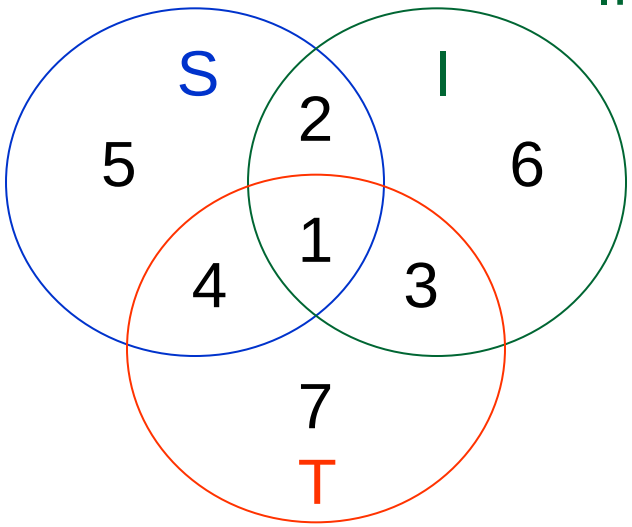
Specified and Implemented Program Behaviors



Specified, Implemented, and Tested Program Behaviors

Specification (expected)

Implementation (observed)

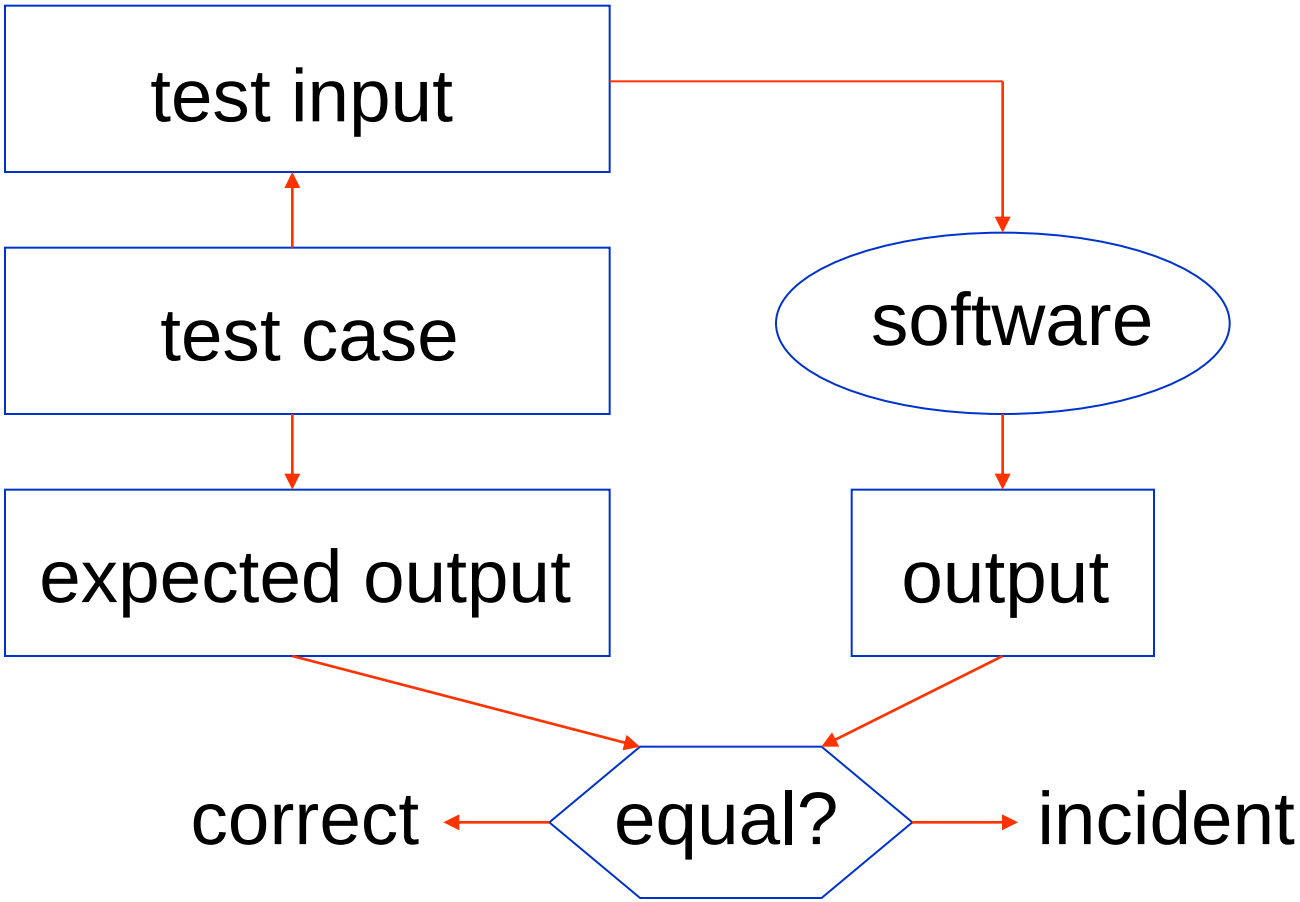


Test Cases (verified)

Test Cases

- Test case generation
- Test case execution

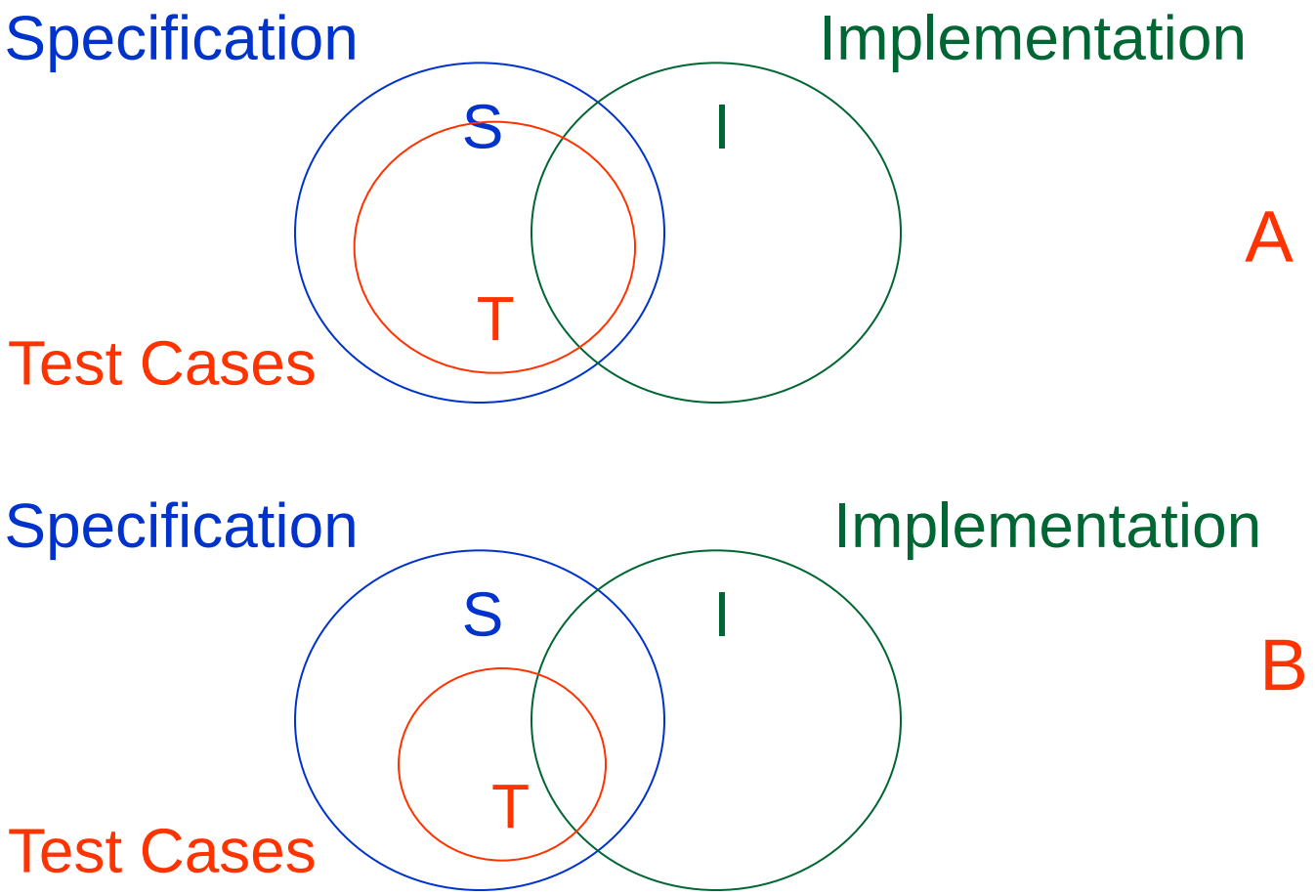
Test Case Execution



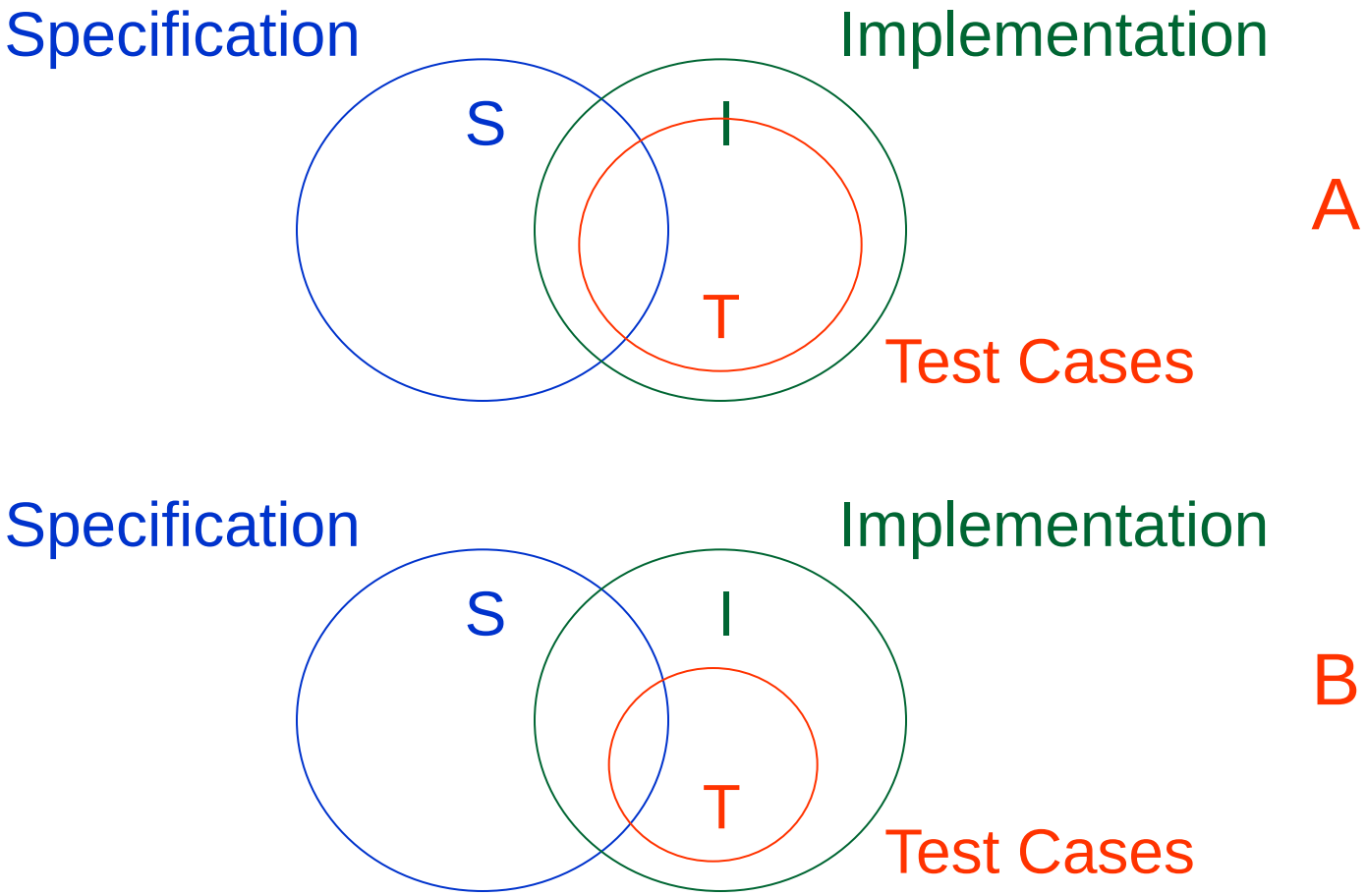
Test Case Generation

- There are two approaches to determining test cases: functional testing (or black box testing) and structural testing (or white box testing).
- Functional testing: the software is viewed as a black box and test cases are determined from the functions described in the specification.
- Structural testing: the software is viewed as a white box and test cases are determined from the structure designed in the implementation.

Functional Testing



Structural Testing



Functional Testing v.s. Structural Testing

- It is hard for functional testing to identify behaviors that are not specified (faults of comission).
- It is hard for structural testing to identify behaviors that are not implemented (faults of omission).
- Neither approach is sufficient, both approaches are needed.

Tracking of Incidents

- An incident tracking system keeps track of the incidents that should be fixed so that all incidents are properly resolved.

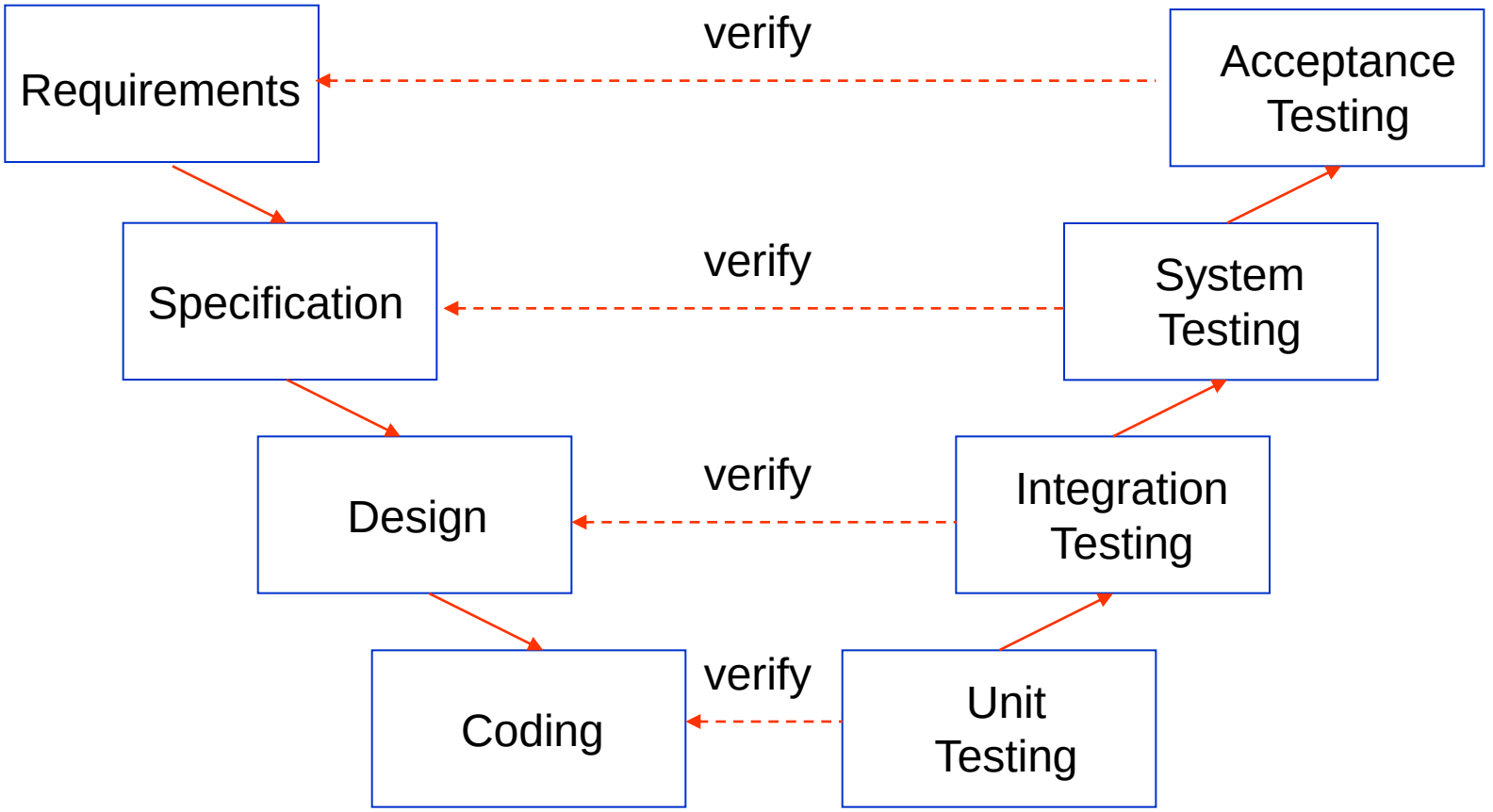
Tracking of Incidents

- Anyone who needs to know about an incident should learn of it soon after it's reported.
- No incident will go unfixed merely because someone forgot about it.
- No incident will go unfixed on the whim of a single programmer.
- A minimum of incidents will go unfixed merely because of poor communication.

Regression Testing

- Regression testing reuses the test cases to test the fixed (or a new version of) software to make sure that the fixing does not introduce new faults into the software.

Levels of Testing (V-Model)



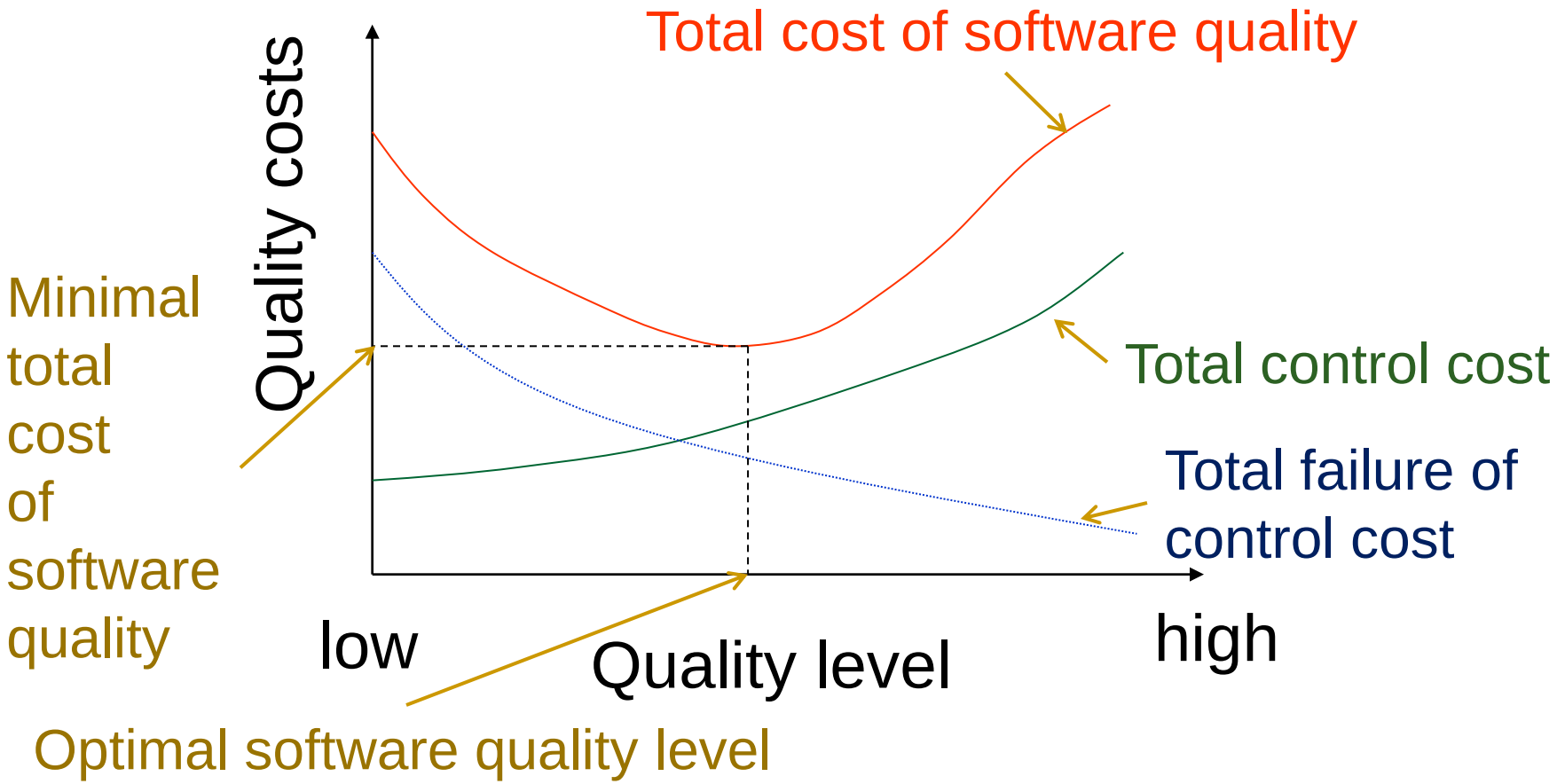
Costs of Software Quality (Control Costs)

- Prevention costs include investments in quality infrastructure and quality activities that are not directed to a specific project or system, being general to the organization.
- Appraisal costs include the costs of activities performed for a specific project or system for the purpose of detecting software errors.

Costs of Software Quality (Failure of Control Costs)

- Internal failure costs include costs of correcting errors that have been detected by design reviews, software tests and acceptance tests and completed before the software is installed at customer sites.
- External failure costs include all costs of correcting failures detected by customers or the maintenance team after the software system has been installed.

Software Quality Levels v.s. Costs



Test Case Coverage

- Software quality costs affects software quality levels.
- We can determine when to stop the software testing based on available software quality resources.
- The coverage of test cases is used to determine the termination of software testing.