Introduction to Software Testing

Ver. 1.0

Disusun oleh: Salman El Farisi

Introduction to Software Testing, Paul Ammann & Jeff Offutt, Cambridge University Press, 2008.

Activities of Test Engineer

- Test Engineer is an IT professional who is in charge of one or more technical test activities including:
    1. Designing testing inputs
    2. Producing test case values
    3. Running test scripts
    4. Analyzing results
    5. Reporting results to developer and manager

Activities of Test Engineer (2)

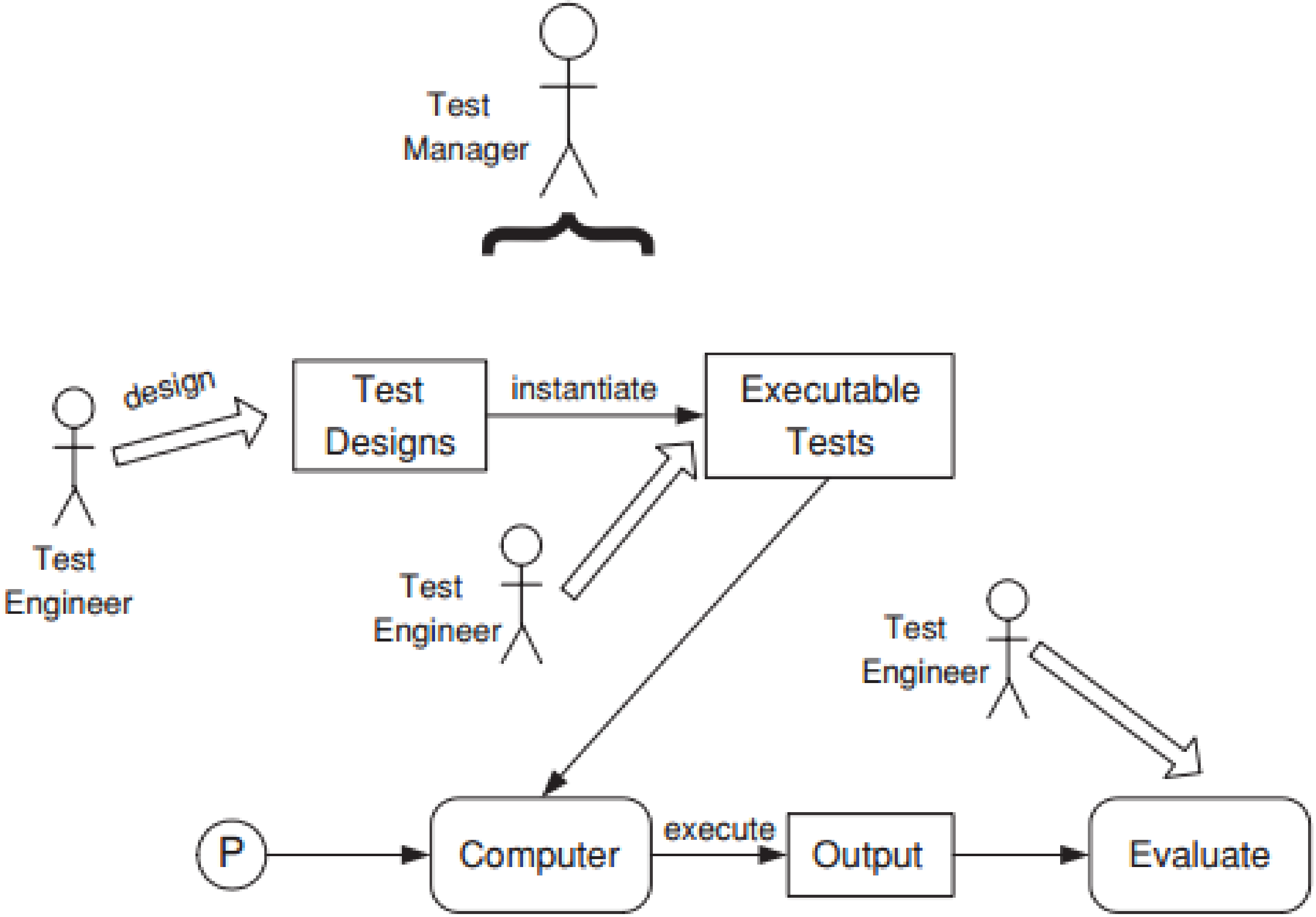


Figure 1.1. Activities of test engineers.

Testing Levels Based on Software Activity

- Acceptance Testing -- Requirement
- System Testing -- Architectural design
- Integration Testing -- Subsystem design
- Module Testing -- Detailed design
- Unit Testing -- Implementation

Testing Levels Based on Software Activity (2)

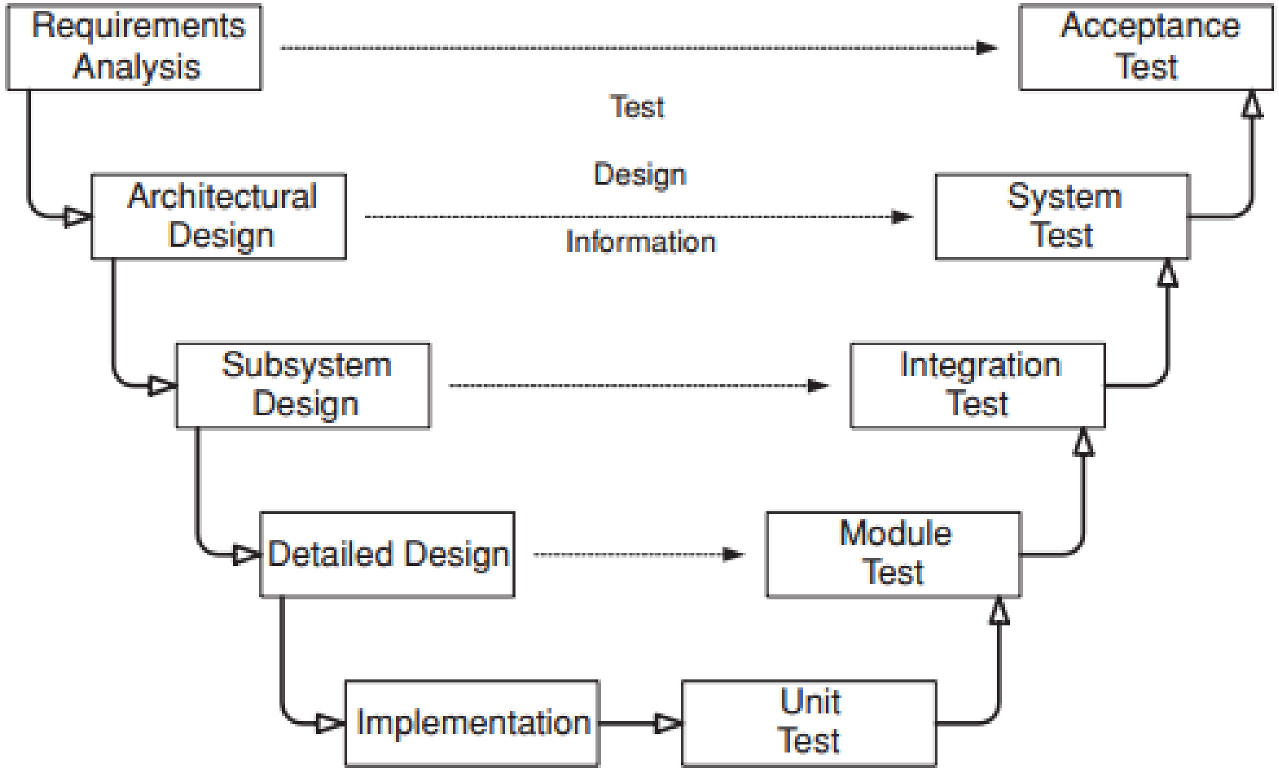


Figure 1.2. Software development activities and testing levels – the "V Model".

Beizer’s Testing Levels Based on Test Process Maturity

- **Level 0:** no difference between testing & debugging.
- **Level 1:** The purpose of testing is to show that the software works.
- **Level 2:** The purpose of testing is to show that the software doesn’t works.
- **Level 3:** The purpose of testing is not to prove anything specific, but to reduce the risk of using software.
- **Level 4:** Testing is mental discipline that help All IT professional develop higher quality software.

Automation of Test Activities

- Software testing is expensive and labor intensive.
- Software testing requires up to 50% software development costs, and even more for safety-critical applications.
- One of the goals of software testing is to automate as much as possible.

Software Testing Limitations

- Testing can show only the presence of failures, not their absence.
- Tester often call a successful (or effective) test one that finds an error.

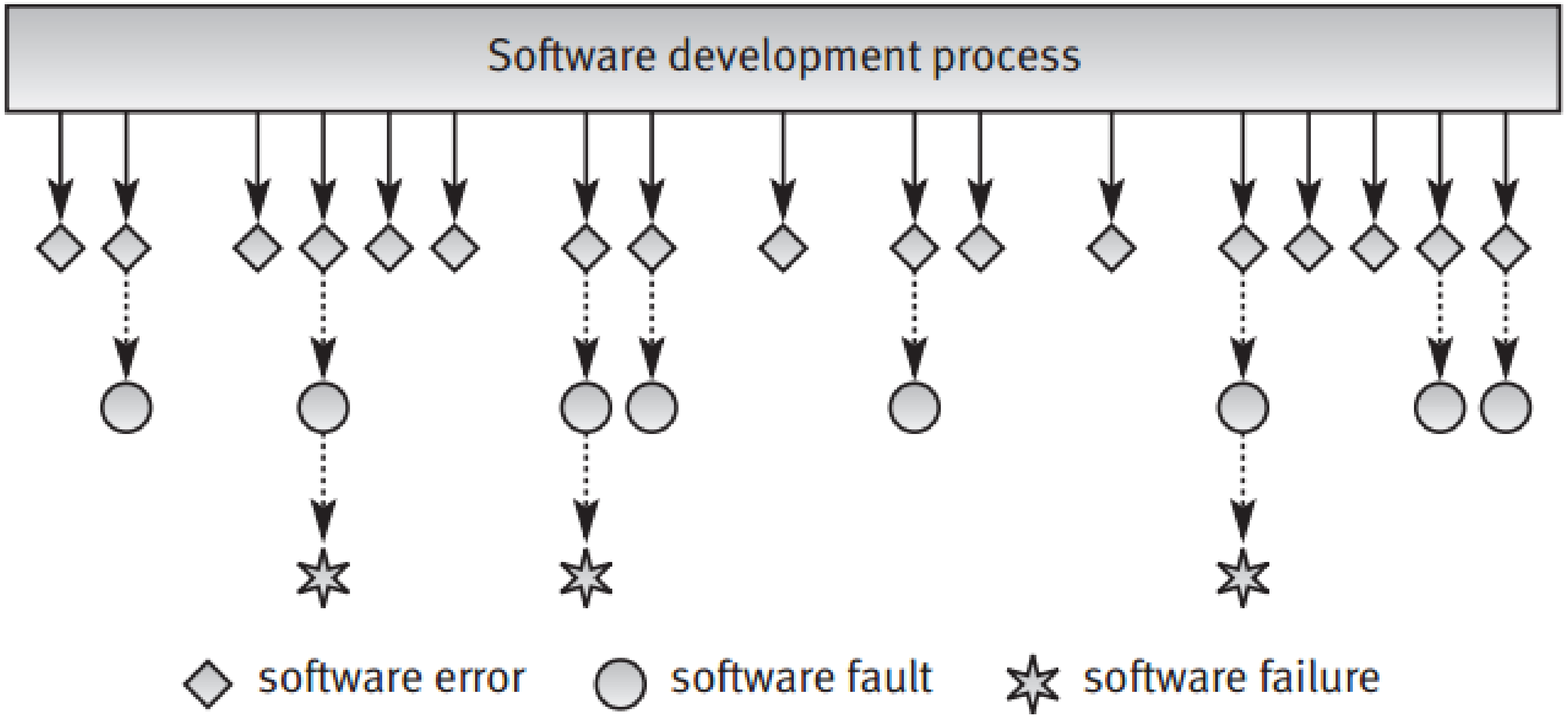
Software Testing Terminology

- **Validation**: The process of evaluating software at the end of software development to ensure compliance with intended usage.

- **Verification**: The process of determining whether the products of given phase of the software development process fulfill the requirements established during the previous phase.

Software Testing Terminology (2)

- **Software Fault:** A static defect in the software
- **Software Error:** An incorrect internal state that is the manifestation of some fault.
- **Software Failure:** External, incorrect behavior with respect to requirement.
- **Testing:** Evaluating software by observing its execution
- **Test Failure:** Execution that results in failure



Sumber: Software Quality Assurance from Theory to Impelementation,
Daniel Galin, Pearson Education, 2004.
Hal. 18

Software Testing Terminology (3)

- **Debugging:** The proses of finding a fault given a failure
- **Test Case Value:** The input values necessary to complete some execution of the software under test.
- **Expected Result:** The result that will be produced when executing the test if and only if the program satisfies its intended behavior.

Software Testing Terminology (4)

- Software Observability: how easy it is to observe the behavior of a program in terms of its outputs, effect on the environment, and other hardware and software component.
- Software Controllability: How easy it is to provide a program with needed inputs, in term of values, operations, and behaviors.

Older Software Testing Terminology

- **Black-box Testing:** Deriving tests from external descriptions of the software, including specifications, requirements, and designs.
- **White-box Testing:** Deriving tests from the source code internals of the software, specifically including branches, individual conditions, and statements.

Older Software Testing Terminology (2)

- **Top-Down Testing:** Test the main procedure, then go down through procedures it calls, and so on.
- **Bottom-Up Testing:** Test the leaves in the tree (procedures that make no calls), and move up to the root. Each procedure is tested only if all of its children have been tested.

Older Software Testing Terminology (3)

- **Static Testing:** Testing without executing the program. This includes software inspections and some forms of analysis.
- **Dynamic Testing:** Testing by executing the program with real inputs.

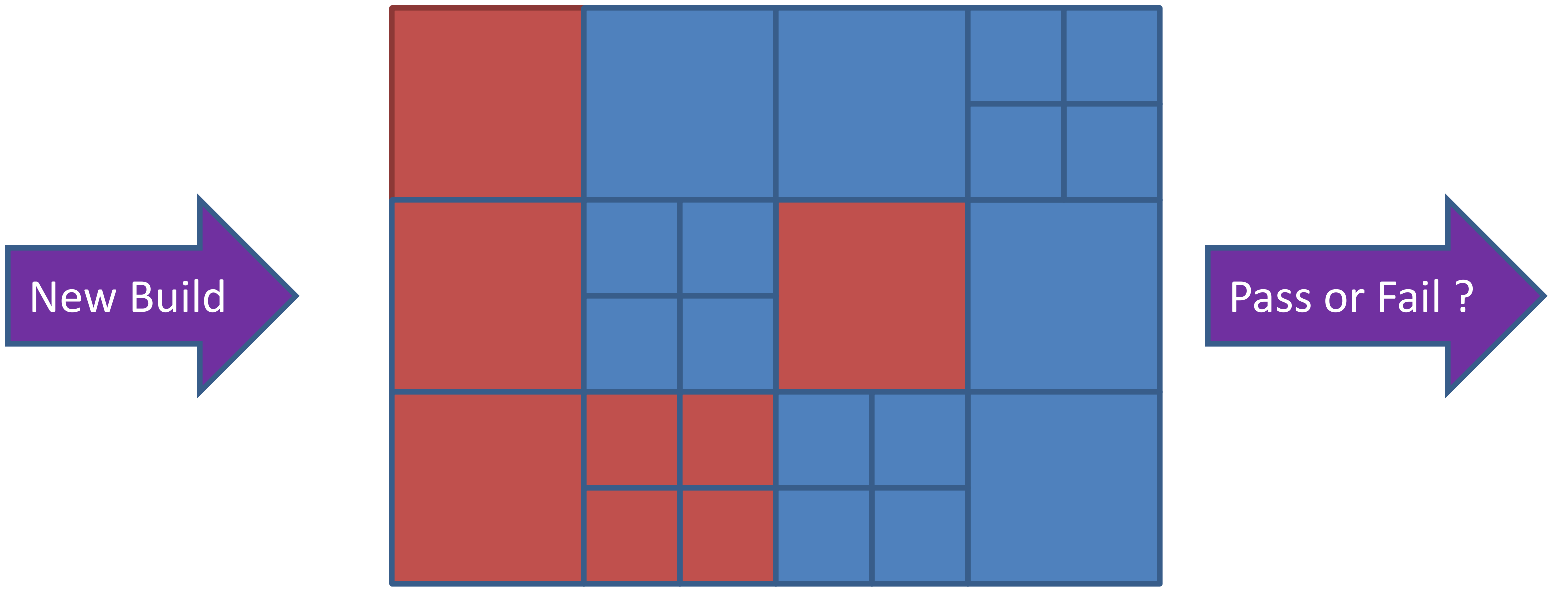
Modern testing terminology

- **Smoke tests** are a subset of test cases that cover the most important functionality of a component or system, used to aid assessment of whether main functions of the software appear to work correctly.[1]
- **Regression testing** is re-running functional and non-functional tests to ensure that previously developed and tested software still performs after a change. [2]

[1] https://en.wikipedia.org/wiki/Smoke_testing_%28software%29
[2] https://en.wikipedia.org/wiki/Regression_testing

Smoke Testing

SOFTWARE



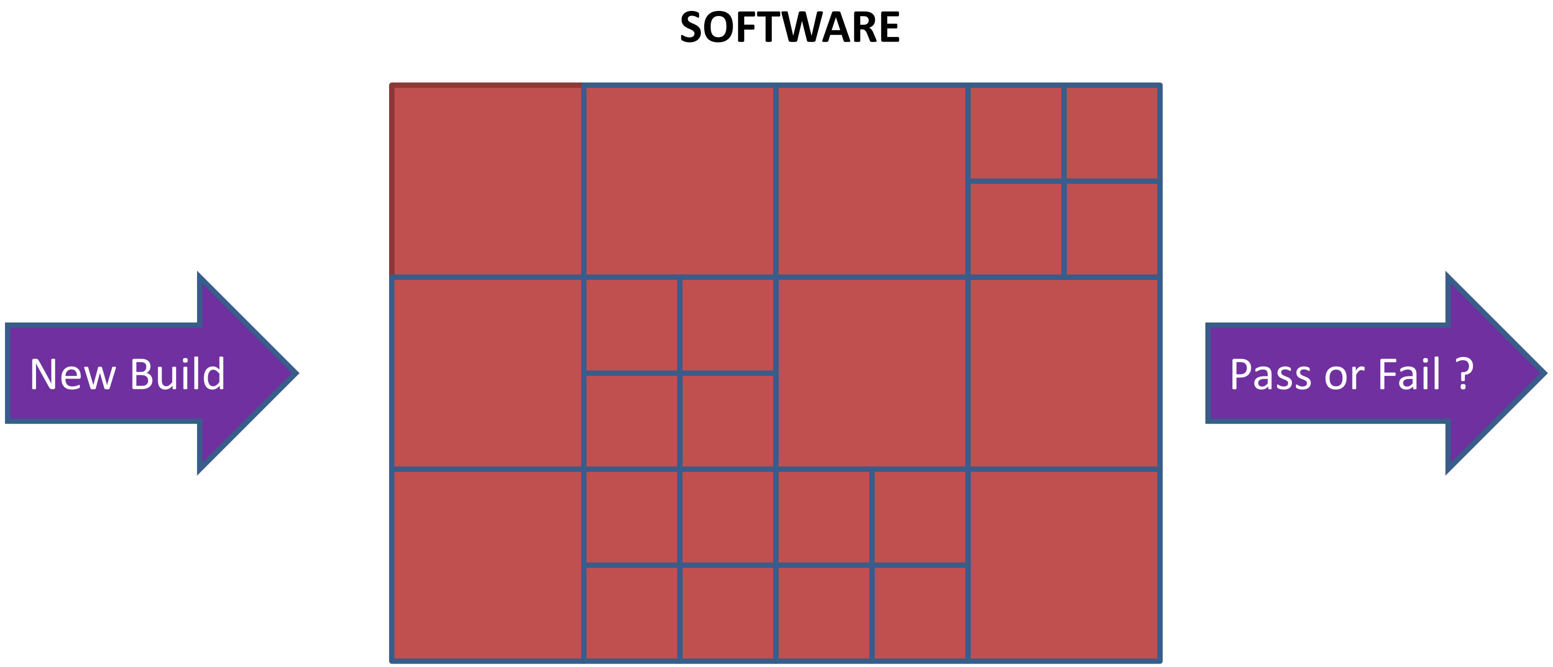
New Build

Pass or Fail ?

🟦 **Software Module**

🟥 **Tested Software Module**

Regression Testing



SOFTWARE

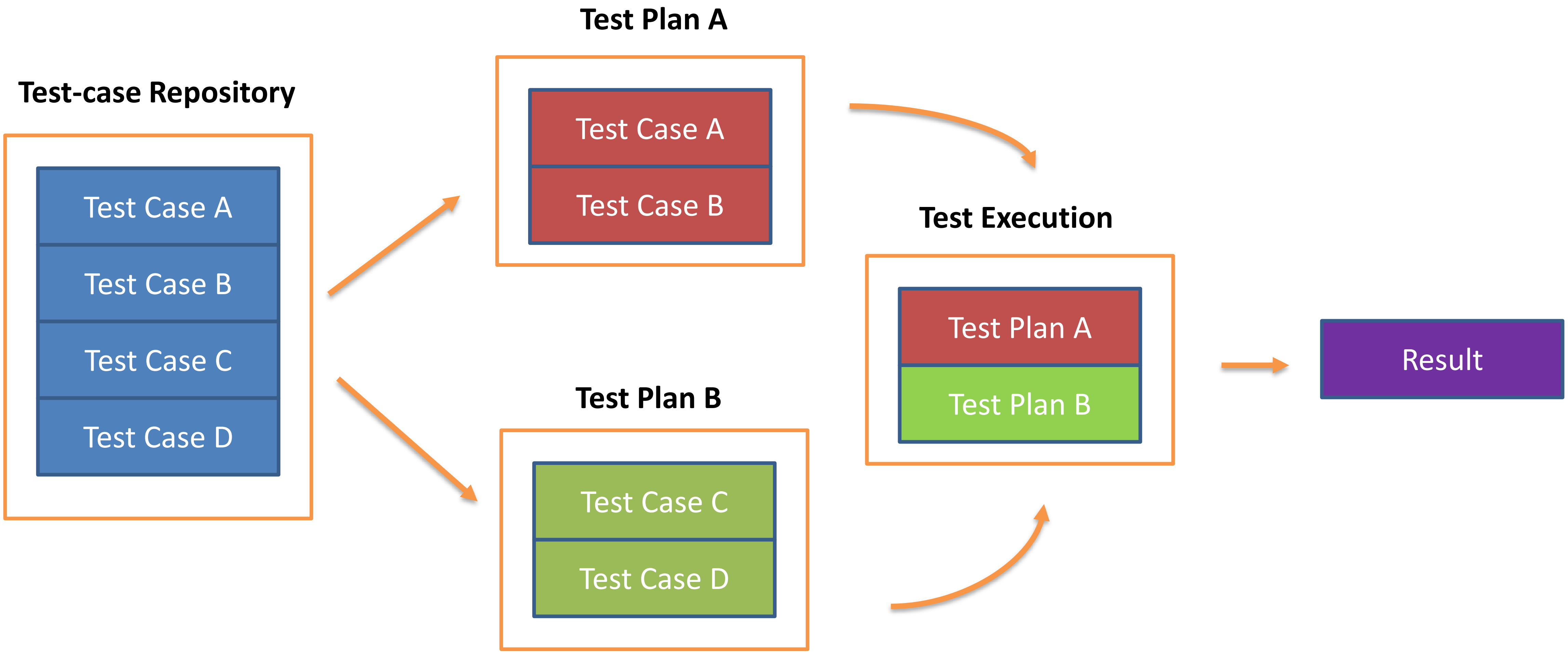
New Build

Pass or Fail ?

🟦 **Software Module**

🟥 **Tested Software Module**

Software Testing Activity



Test-case Repository

Test Case A

Test Case B

Test Case C

Test Case D

Test Plan A

Test Case A

Test Case B

Test Plan B

Test Case C

Test Case D

Test Execution

Test Plan A

Test Plan B

Result