ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

Software Testing

Chapter 1
Overview

Contents

What is software quality?

❖Software quality – IEEE definition:

The degree to which a system, component, or process meets specified requirements.

Error - Fault - Failure

❖Error

- Also known as mistake.
- An error can be a Syntax (grammatical) error or Logic
- A software error made by human action which produces an incorrect result.

❖Fault

- Also known as a defect or bug.
- A fault is a manifestation of an error in software
- All software errors may **not** cause software faults

❖Failure

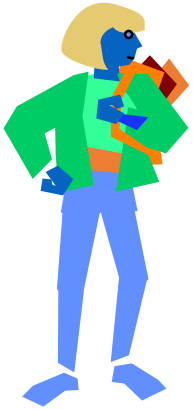
- A fault becomes a failure if it is **activated/executed**.
  - ✓deviation of the software from its expected delivery or service
  - ✓Not all faults result in **failures**; some stay dormant in the code and we may never notice them.

Failure is an event;

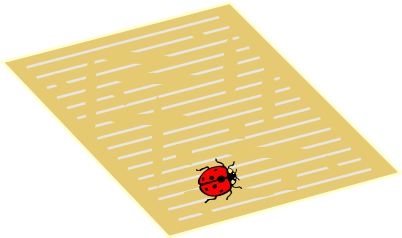
Fault is a state of the software, caused by an error

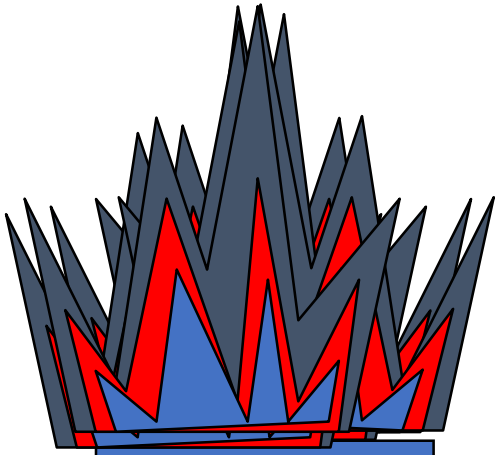
Error - Fault - Failure

A person makes an error ...



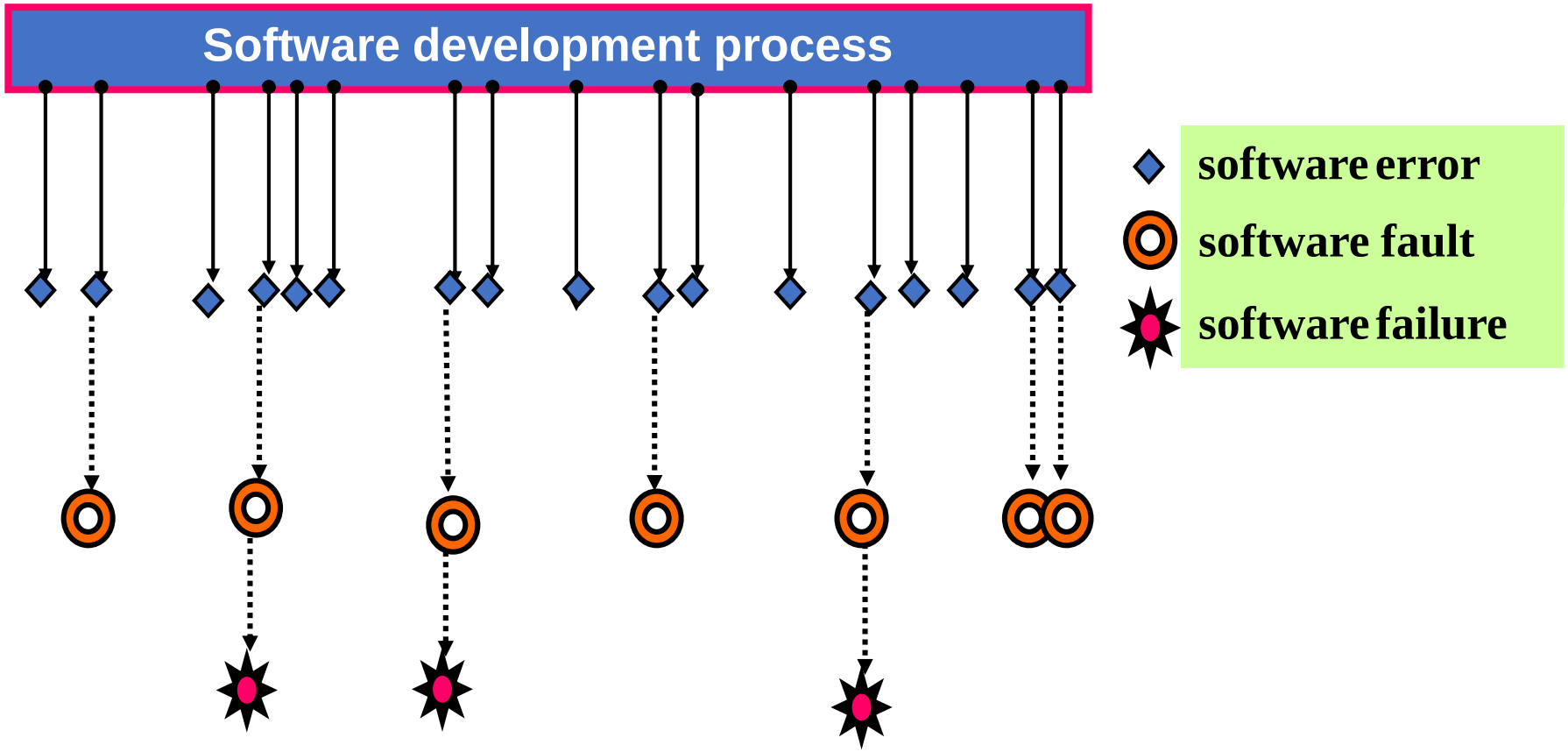
... that creates a fault in the software ...





... that can cause a failure in operation

Error - Fault - Failure



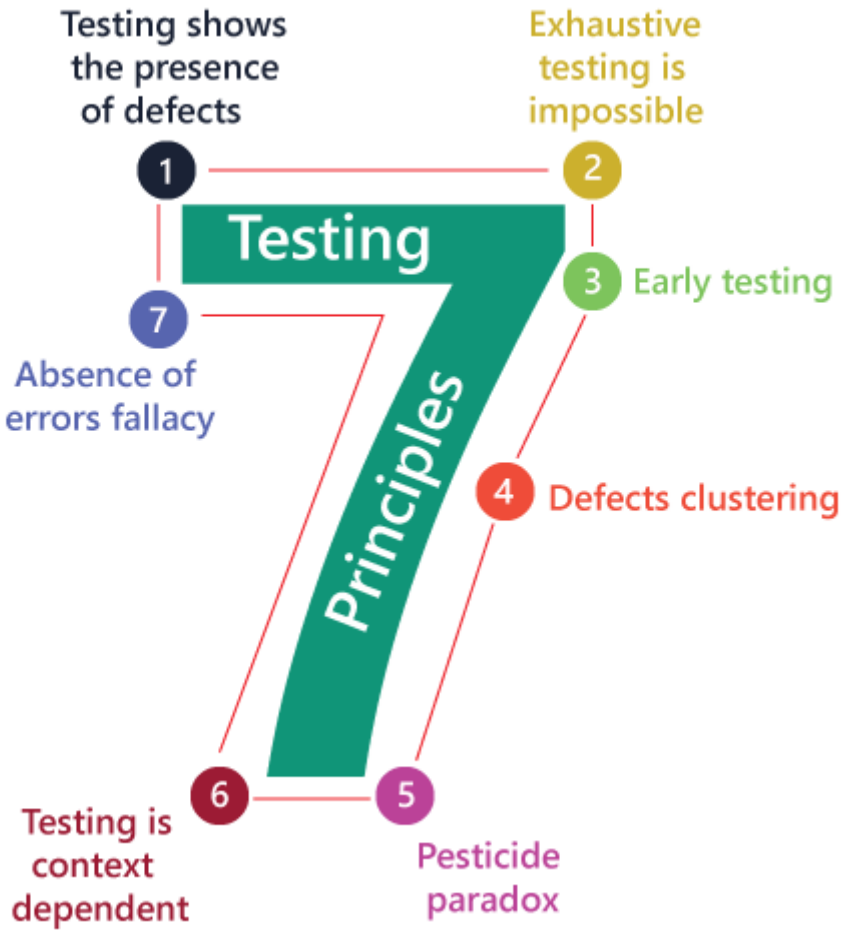
Causes of software errors

- **Errors may occur for many reasons, such as:**
  - Time pressure
  - Human is error prone
  - Inexperienced or insufficiently skilled project participants
  - Miscommunication between project participants, including miscommunication about requirements and design
  - Complexity of the code, design, architecture, the underlying problem to be solved, and the technologies used
  - New, unfamiliar technologies

Definitions of Software Testing?

- Software Testing the **process** consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects.

Principles of Testing



Principles of Testing

❖Principle 1:

- Testing can show that defects are present , but cannot prove that there are no defects.

Principles of Testing

❖Principle 2:

- Exhaustive testing means to test everything, all preconditions and combinations of inputs.
- Testers should apply risk analysis and set priorities to focus testing efforts.

Principles of Testing

❖Principle 3:

- Testing activities should start as early as possible in the software or system development life cycle and should be focused on defined objectives.

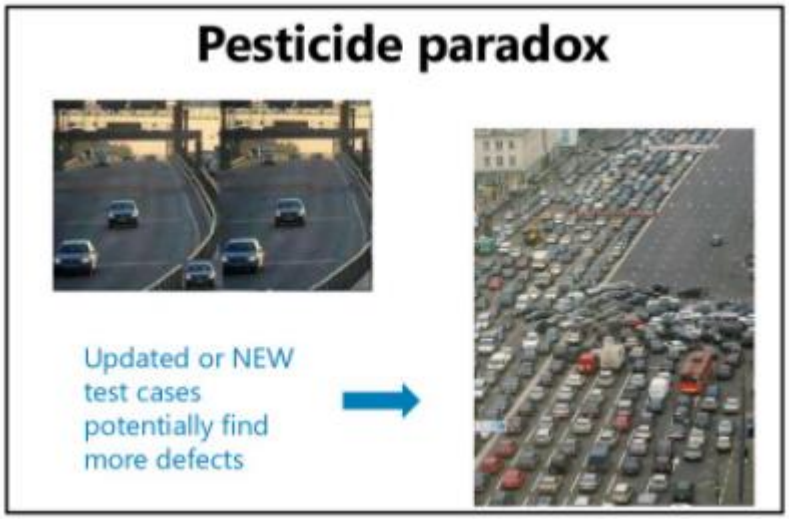
Principles of Testing

❖Principle 4:

- A small number of modules contain most of the defects discovered during pre-release testing or show the most operational failures.

Principles of Testing

❖Principle 5:



- If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new bugs.
- To overcome this 'pesticide paradox', the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

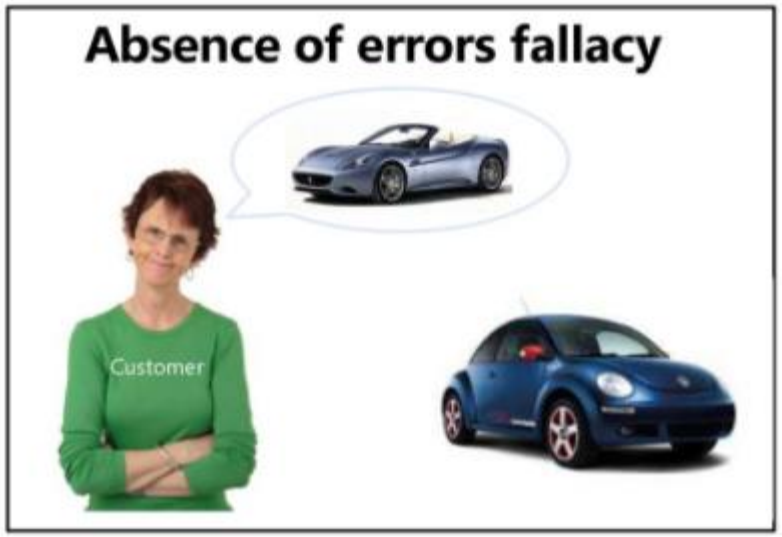
Principles of Testing

❖Principle 6:

- Testing is done differently in different dependent contexts.
- For example, safety-critical software is tested differently from an e-commerce site.

Principles of Testing

❖Principle 7:



- Finding and fixing defects does not help if the system built is unusable and does not fulfill the users' needs and expectations.

Why Testing is necessary?

- All Software has defects (bugs)
- Software products are getting larger and more complicated
- Software is written by people – people make mistakes

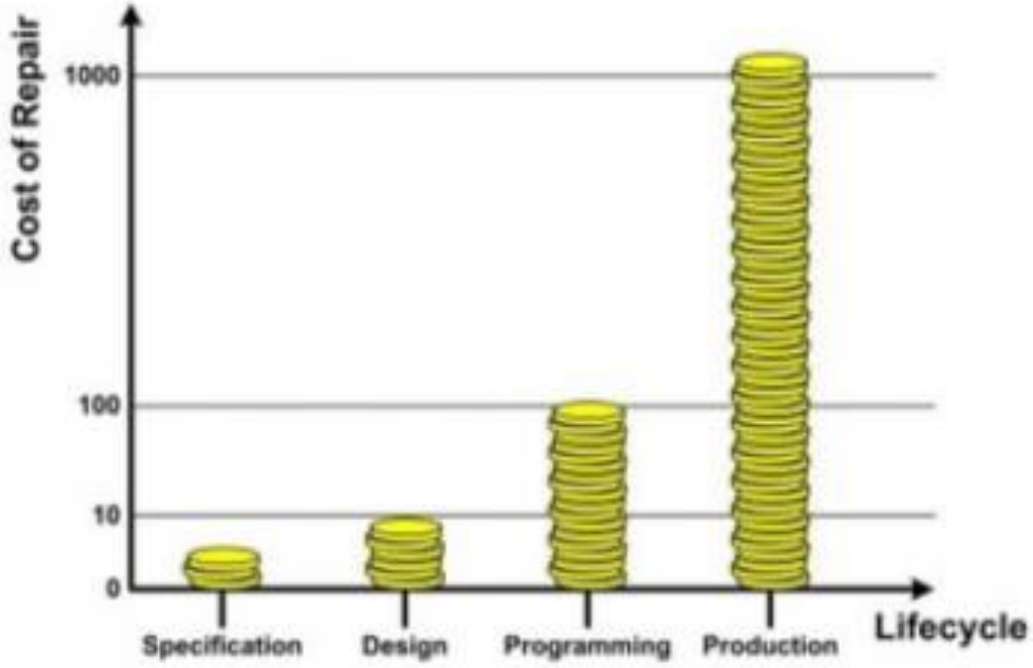
Why Testing is necessary?

- Some of the problems might be trivial, but others can be costly and damaging- with loss of money, time, or business reputation – and even may result in injury or death

❖ Not all software systems carry the same level of risk and not all problems have the same impact when they occur

Why Testing is necessary?

- Cost of defects: the cost to finding and fixing defects rises considerably across the life cycle



Why Testing is necessary?

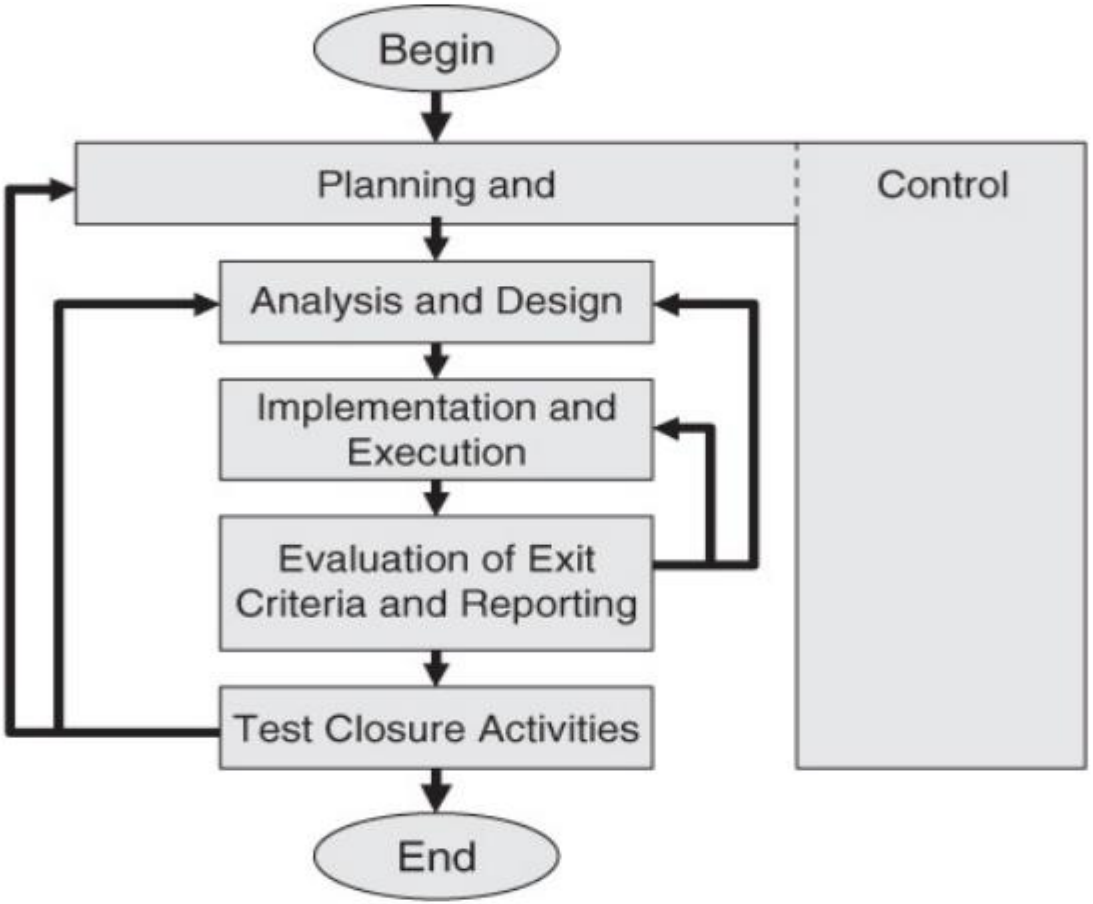
- Software testing looks to find the most important defects as early as possible – increasing confidence that the software meets specification
- Help to measure the quality of software

Software Testing Objectives

- To identify and reveal as many errors as possible
- To gain confidence about the *level of quality*
- To prevent defects
- To provide information for decision-making (stakeholders)
- To reduce the level of risk of inadequate of software quality
- To make sure that the system works as expected, meets the user requirements
- To comply with the contractual, legal or regulatory requirements or standards

Fundamental Test Process

- The five stages of the fundamental test process:



Test Planning

"A goal without a plan is just a wish"

- Major tasks are:
  - Identify the objectives of testing
  - Determine scope
  - Determine the Test Approach
  - Determine the required test resources
  - Implement the test policy and/or the test strategy
  - Schedule test analysis and design tasks
  - Schedule test implementation, execution and evaluation
  - Determine the Exit Criteria

Test Planning

- Test Control:
  - The ongoing activity of comparing actual progress against the plan
  - Reporting status, including deviations from the plan
  - Taking actions necessary to meet the mission and objectives of the project
  - Test Planning takes into account the feedback from monitoring and control activities
  - Major tasks are:
    - ✓Measure and analyze results
    - ✓Monitor and document progress, test coverage and exit criteria
    - ✓Initiate corrective actions
    - ✓Make decisions

Analysis and design

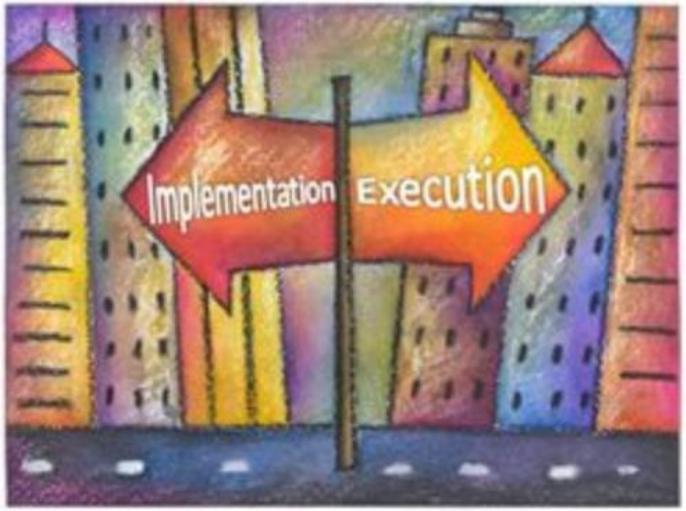
"Analysis gain better understanding"

- Review the Test Basis: - in doing so evaluate testability of Test Basis and Test Objects(s).
- Identify and prioritize Test Conditions and associated Test Data.
- Test Conditions and associated Test Data are documented in a Test Design Specification.
- Design and prioritize the Test Cases
- Identify Test Data required to support Test Cases
- Design the test environment set-up
- Identify any required infrastructure and tools

Implementation and Execution

- Develop, implement and priorities Test case
- Create the Test Scripts
- Create test data
- Write automated test scripts
- Check the environment – Verify that the test environment has been set up correctly

"Ideas are Easy. Implementation is hard"



Evaluating exit criteria and reporting

"Evaluation- Learning tool to Improve"

- Evaluating exit criteria is the activity where test execution is assessed against the defined objectives.
- Exit criteria should be set and evaluated for each test level
- Check test logs against the exit criteria specified in the test planning
- Assess if more tests are needed or if the exit criteria specified should be changed.
- Write a test summary report to stakeholders

Evaluating exit criteria and reporting

- How to measure exit criteria?
  - All the planned requirements must be met
  - All the high Priority bugs should be closed
  - All the test cases should be executed
  - If the scheduled time out is arrived
  - Test manager must sign off the release

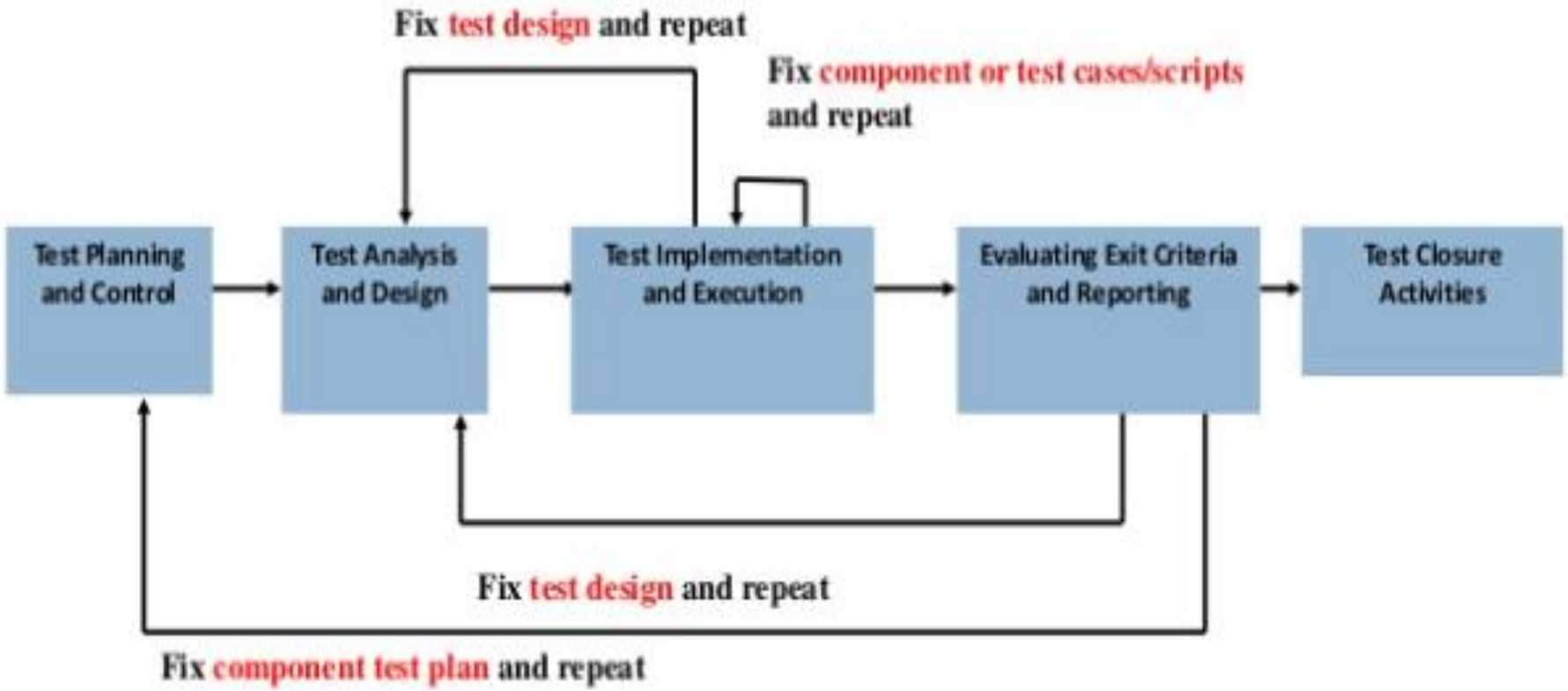
Note: All these parameters can be met by percentages (not 100%)

Test closure activities

"Closure – Way to start new beginning"

- Collect data from complete test activity
- Finalize and archive the test ware
- Test wares such as scripts, test environment etc.
- Evaluate how testing went and analyze lessons learned for future releases and projects

Fundamental Test Process



ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

Thank You !