# MSLW Certification in Manual Testing

- Fundamentals of Testing
- Testing Throughout SDLC
- Static Testing Techniques
- Test Design Techniques
- Test Management
- Tool Support for Testing
- Test Management Tool – Quality Center

Mahindra Satyam

# Fundamentals of Testing

# Learning Objectives

- Introduction
- Fundamentals of Testing
  - Why is Testing Necessary?
  - Role of Testing in Software Development
  - Testing Principles
  - Fundamental Test Process
  - Testing Vocabulary

# Introduction

Software testing is a process used to help identify correctness, completeness, security and quality of developed computer software.

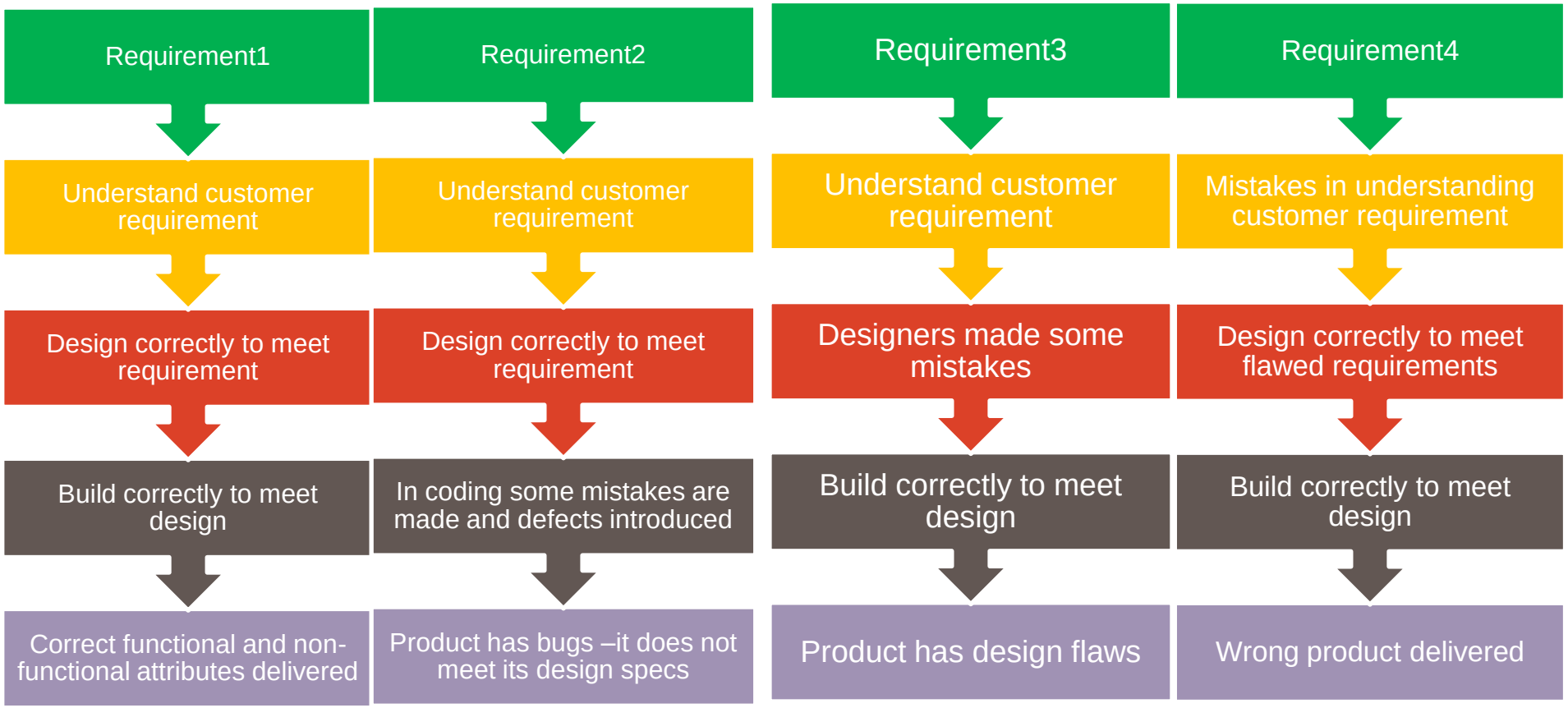
- It proves presence of defects not their absence.

- It is constructive destructive process.
  - Well constructed and executed test is successful when it finds defects that can be fixed.
  - An unsuccessful test is the one that found no defects.
    [the concept of a program without errors is unrealistic]

- It is program execution with the intent of finding defects.
  - We tend to select test data that have low probability of causing the program to fail.

# Why is testing necessary?

- When do defects arise? {When We test the product meets its requirements}



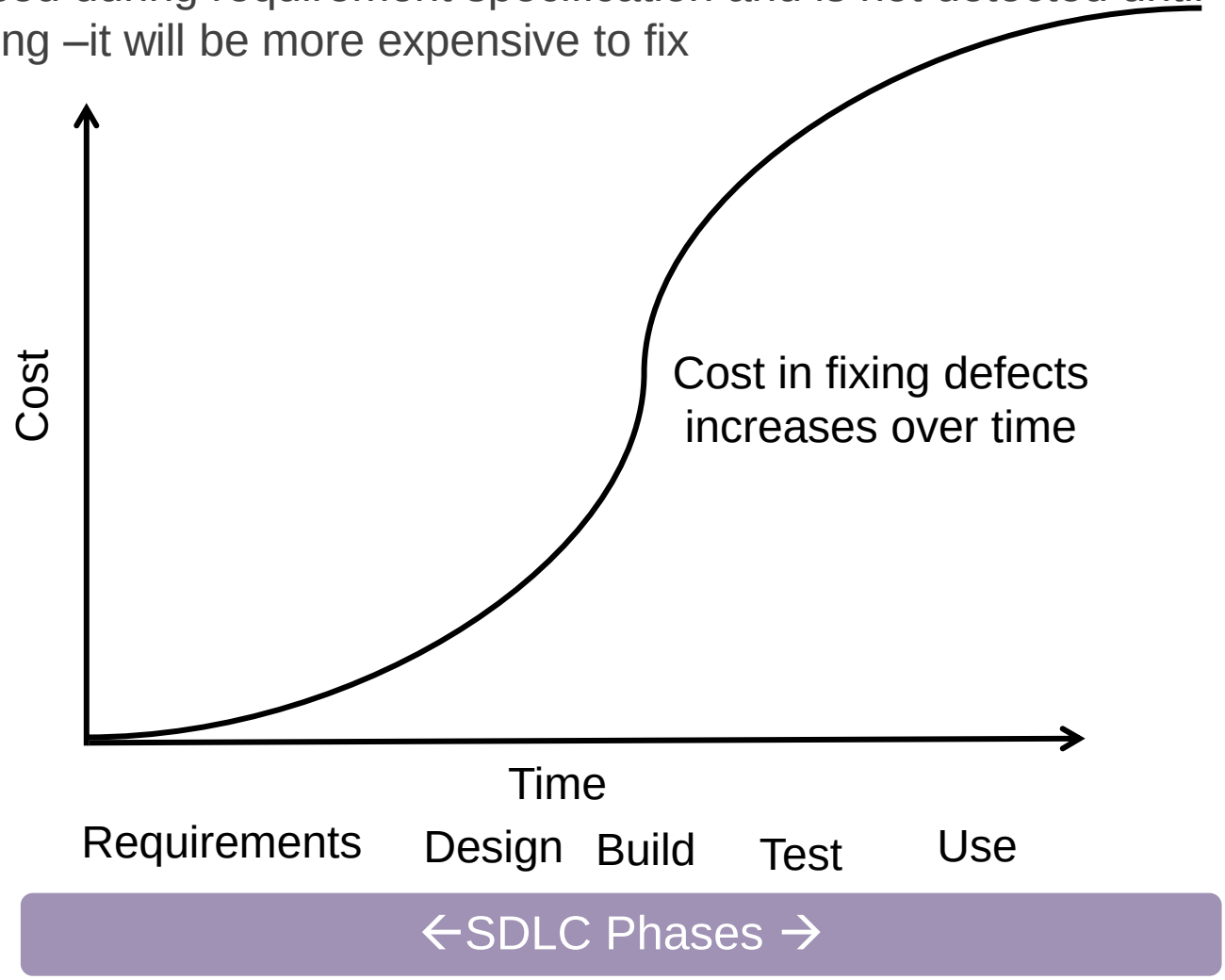

Defect correction easy –tester will fix

Hard to correct Because design changes required

Acceptance testing failure; very costly defects

# Contd..

- Impact of defects
- A defect introduced during requirement specification and is not detected until acceptance testing –it will be more expensive to fix



# Contd..

| Phase | Errors passed in | Errors amplified | Errors introduced | % of errors found in testing |
|---|---|---|---|---|
| Analysis | 0 | 0 | 10 | 0% |

let 10 errors introduced in analysis phase; 6 are crept into design phase remaining 4 of them get amplified by 50%.

| Phase | Errors passed in | Errors amplified | Errors introduced | % of errors found in testing |
|---|---|---|---|---|
| Design | 6 | 6 | 25 | 0% |

out of 37 passed to coding phase, 10 are passed as it is; remaining 27 of them get amplified by 300%.

| Phase | Errors passed in | Errors amplified | Errors introduced | % of errors found in testing |
|---|---|---|---|---|
| Coding | 10 | 81 | 25 | 20% |

out of 116, 20% are detected by testing and rest of 93 passed to integration phase as is;
Integration phase will not have any amplification.

| Phase; | Errors passed in | Errors amplified | Errors introduced | % of errors found in testing |
|---|---|---|---|---|
| Integration | 93 | 0 | 0 | 50% |

No of errors passed to system test phase are 50% of 93; no amplification of errors; 24 defects delevered to customer

| Phase; | Errors passed in | Errors amplified | Errors introduced | % of errors found in testing |
|---|---|---|---|---|
| Systst | 47 | 0 | 0 | 50% |

# Contd..

- We use testing to help us find faults and potential failures during software development, maintenance and operations.
- We do testing to reduce risk of failures occurring in an operational environment
- Fixing a defect has some chance of introducing another defect/being done incorrectly /incompletely ( due to work pressure)
- Work pressure
  - Time, budget, pressure to deliver a technical solution that meets customer's needs
- **How much testing is enough?**
  - We use risks and priorities to focus testing efforts
    - Testing all combination of inputs, pre-conditions is not feasible
    - Risk assessment help us to find how much of testing to do
      - Risk for the customers, the stakeholders, the project and the software

# Seven Principles of Testing

## Principle1: Testing shows presence of defects

- “Testing can show the presence of bugs. It can never show absence of bugs” –Edger W. Dijkstra

## Principle 2: Exhaustive testing is impossible

- Instead of exhaustive testing we use risks and priorities to focus testing efforts

## Principle 3: Early testing

- Test planning can occur as soon as SW requirements have been established
- Test case design happens as soon as SW design has been defined.
- Mistake –Thinking of testing only after completion of coding

## Principle 4: Defect clustering

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

## Principle 5: Pesticide paradox

- If the same tests are repeated over and over again, eventually the same set o test cases will no longer find any new bugs
- Test cases need to be reviewed and revised, and new and different tests need to be written to exercise the different parts of the SW, potentially to find more errors

## Principle 6: Testing is context dependent

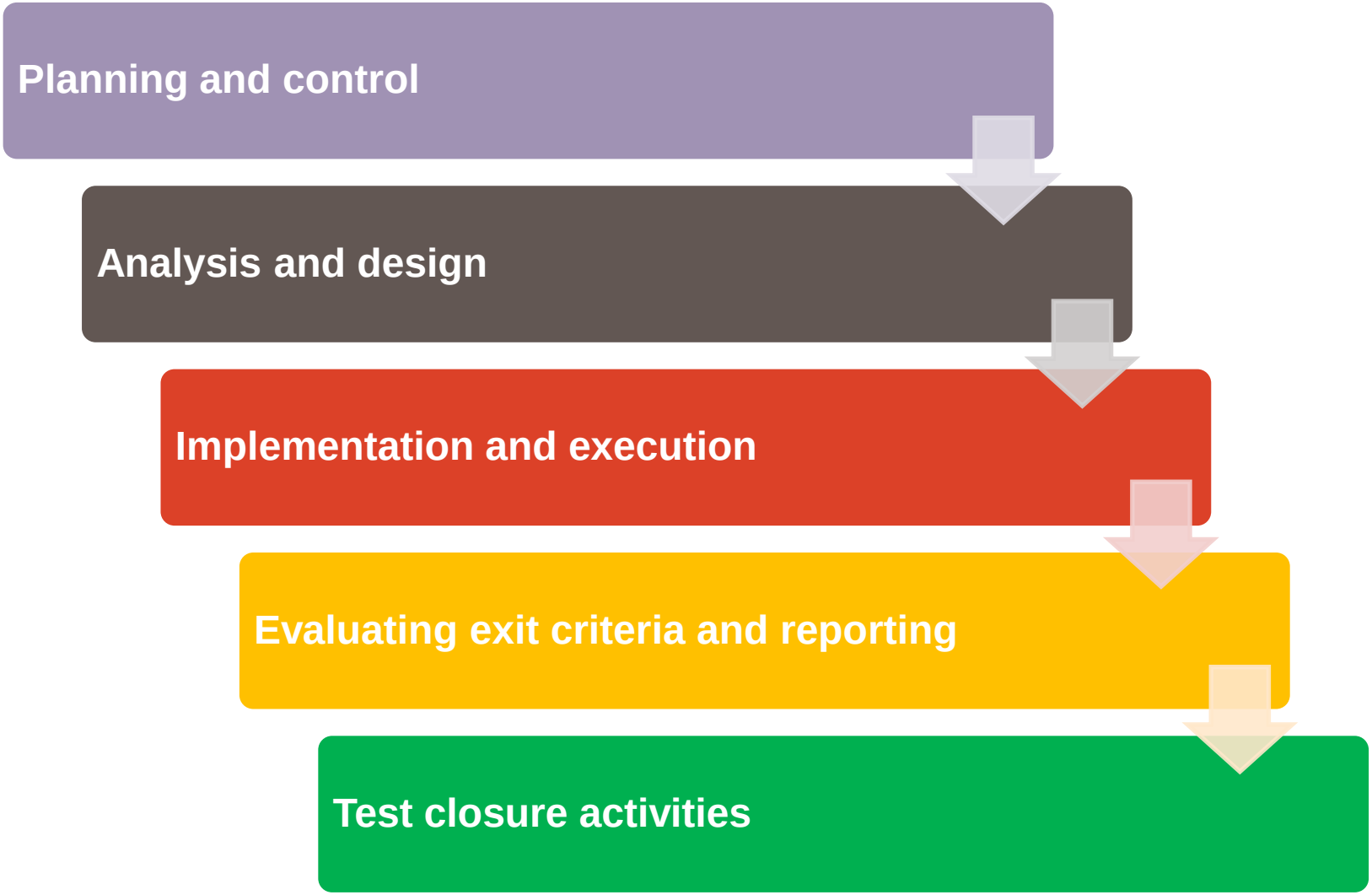
- Safety critical software is tested differently from an e-commerce site

## Principle 7: Absence of errors fallacy

- Finding and fixing defects does not help if the system built is un-usable/does not fulfill the users’ needs and expectations

# Fundamental Test Process

Sequential steps of the fundamental test process are:



# Test planning and control

## Test Planning

- Test Planning is the activity of defining the objectives of testing and the specification of test activities in order to meet the objectives and mission.
- Test plan is a document detailing a systematic approach to testing a system.
- A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements.


Test Plan

## Test Control

- Test control is the ongoing activity of comparing actual progress against the plan, and reporting the status, including deviations from the plan.
- It involves taking actions necessary to meet the mission and objectives of the project.
- In order to control testing, the testing activities should be monitored throughout the project.
- Test planning takes into account the feedback from monitoring and control activities

# Test Process – Test Analysis and Design

- Test analysis and design is the activity during which general testing objectives are transformed into tangible **test conditions** and **test cases**.
- It has the following major tasks:
  - ✓ Reviewing the test basis (such as requirements, risk level, risk analysis reports, architecture, design, interface specifications).
  - ✓ Evaluating testability of the test basis and test objects.
  - ✓ Identifying and prioritizing test conditions based on analysis of test items, the specification, behavior and structure of the software.
  - ✓ Designing and prioritizing high level test cases.
  - ✓ Identifying necessary test data to support the test conditions and test cases.
  - ✓ Designing the test environment setup and identifying and required infrastructure and tools.
  - ✓ Creating bi-directional traceability between test basis and test cases.

# Test Process – Test Implementation and Execution

- Test implementation and execution is the activity where test procedures or scripts are specified by combining the test cases in a particular order for execution.
- It has the following major tasks:
  - ✓ Finalizing, implementing and prioritizing test cases (including the identification of test data).
  - ✓ Developing and prioritizing test procedures, creating test data and, optionally preparing test harnesses and writing automated test scripts.
  - ✓ Creating test suites from the test procedures for efficient test execution.
  - ✓ Verifying that the test environment has been set up correctly.
  - ✓ Verifying and updating bi-directional traceability between test basis and test cases.
  - ✓ Executing test procedures either manually or by using test execution tools, according to the planned sequence.
  - ✓ Logging the outcome of test execution and recording the identities and versions of the software under test, test tools and testware.
  - ✓ Comparing actual results with expected results.
  - ✓ Reporting discrepancies as incidents and analyzing them in order to establish their cause.
  - ✓ Repeating test activities as a result of action taken for each discrepancy.

    **Example:** Re-execution of a test that previously failed in order to confirm a fix.

# Test Process – Evaluating Exit Criteria and Reporting

- Evaluating exit criteria is the activity where test execution is assessed against the defined objectives.
- This should be done for each test level.
- It has the following major tasks:
  - ✓ Checking test logs against the exit criteria specified in test planning.
  - ✓ Assessing if more tests are needed or if the exit criteria specified should be changed.
  - ✓ Writing a test summary report for stakeholders.

# Test Closure Activities

- Test closure activities collect data from completed test activities to consolidate experience, testware, facts and numbers.
- Test closure activities occur at project milestones such as when a software system is released, a test project is completed, a milestone has been achieved, or a maintenance release has been completed.
- It has the following major tasks:
  - ✓ Checking which planned deliverables have been delivered.
  - ✓ Closing incident reports or raising change records for any that remain open.
  - ✓ Documenting the acceptance of the system.
  - ✓ Finalizing and archiving testware, the test environment and the test infrastructure for later reuse.
  - ✓ Handing over the testware to the maintenance organization.
  - ✓ Analyzing lessons learned to determine changes needed for future releases and projects.
  - ✓ Using the information gathered to improve test maturity.

# Testing Vocabulary

- **Error:** Human action that produces an incorrect result
- **Defect:** A flaw in component/system that can cause component or system to fail perform its required function –When a defect is encountered during execution, it may cause a failure of component/system
- **Failure:** Deviation of component/system from its expected delivery, service or result.
  - Failures need not necessarily be caused by defects; can also be caused due to
    - Environmental conditions{ex: radiation burst, pollution etc..}
    - Intentional damages{ ex: misinterpretations of outputs; wrong input}
    - In ability of our brains to handle complexity –brain can only deal with a reasonable amount of complexity
- **Quality:** The degree to which a component/system/process meets specified requirements and/or user/customer needs and expectations
  - Quality of SW is measured in terms of No. of defects found, test runs, system coverage

❖ Metric to measure efficiency of testing is **Defect Removal Efficiency (DRE)** factor $=N_d/(N_r+N_d)$ where, $N_d$ is number of defects detected and $N_r$ is number of defects remaining.

# Thank you

## mahindrasatyam.com

**Safe Harbor**

This document contains forward-looking statements within the meaning of Section 27A of the Securities Act of 1933, as amended, and Section 21E of the Securities Exchange Act of 1934, as amended. The forward-looking statements contained herein are subject to certain risks and uncertainties that could cause actual results to differ materially from those reflected in the forward-looking statements. Mahindra Satyam undertakes no duty to update any forward-looking statements.