# Definitions

## Anomaly

* Any **condition that** **deviates from expectation** based on requirements specifications, design documents, user documents, standards, etc. or from someone’s perception or experience.

## Error (mistake)

* A human action that produces an **incorrect result.**

## Bug/Defect/Fault/Problem

* A flaw in a component or system that can cause the component or system to **fail to perform its required function.**
* A **defect**, if encountered during execution, may **cause a failure** of the component or system.

## Failure

* **Actual deviation** of the component or system from its expected delivery, service or result

## Defect/fault masking

* An occurrence in which **one defect prevents the detection of another**.

## Software Quality

* The degree to which a system, component, or process **meets specified requirements.**
* The degree to which a system, component, or process **meets customer or user needs or expectations.**

## Role of testing in software development, maintenance and operations

* Reduce the risk of problems
* Reduce long-term defect-related costs
* Contribute to the quality of the software
* Help meeting standards:
  + Contractual or legal requirements
  + Industry-specific standards

## Testing and Quality

* Testing can give confidence in the quality of the software **if it finds few or no defects.**
* If defects are found, the quality increases when those **defects are fixed.**
* Lessons learnt from previous mistakes improve **future performance.**

## Testing

* The process of **exercising** software
  + To verify that it satisfies specified requirements and to detect errors
* The process of **analyzing** a software item
  + To detect the differences **between existing and required conditions** (that is, bugs),
  + To **evaluate the features** of the software item
* Main Test Activities
* Planning and control
* Choosing test conditions
* Designing and executing test cases
* Checking results
* Evaluating exit criteria
* Reporting on the testing process and system under test
* Finalizing or completing closure activities
* Objectives
* Finding defects
* Gaining confidence about the level of quality
* Providing information for decision-making
* Preventing defects
* The activity that **initially finds failures** in a software item

## Debugging

* The development activity that finds, analyses and removes the **cause of the failure.**

## Re-testing

* **Ensures** that the fix does indeed **resolve the failure.**

## Seven Testing Principles

#### Testing shows presence of defects

* Testing can show that defects are present
* Cannot prove that there are no defects
* Appropriate testing reduces the probability for defects

#### Exhaustive testing is impossible

* All combinations of inputs and preconditions are usually **almost infinite number**
* Testing everything is **not feasible**
  + Except for trivial cases
* **Risk analysis and priorities** should be used to focus testing efforts

#### Early testing

* Testing activities shall be started as early as possible
* And shall be focused on defined objectives
* The later a bug is found – the more it costs!

#### Defect clustering

* Testing effort shall be **focused proportionally**
  + To the expected and later observed defect density of modules
* **A small number of modules** usually contains most of the defects discovered
  + Responsible for most of the operational failures

#### Pesticide paradox

* Same tests repeated over and over again tend to lose their effectiveness
  + Previously undetected defects remain undiscovered
* New and modified test cases should be developed

#### Testing is context dependent

* Testing is done differently in different contexts
  + Example: safety-critical software is tested differently from an e-commerce site.

#### Absence-of-errors fallacy

* Finding and fixing defects itself does not help in these cases:
  + The system built is **unusable**
  + Does not fulfill the **users’ needs and expectations**

## Test Planning and Control

* **Starts at the beginning** of the software development project
* Must be **regularly checked**, updated, and adjusted

## Test Control

* **Monitoring** of the test activities
* **Comparing** with the plan
* **Reporting** status of deviations from the plan
* Taking actions for **correction**
* **Updating** the test plan according to the feedback

## Test Prioritization

* Software projects are often run under severe **time pressure**
* Prioritization guarantees that the critical **software parts are tested first**

## Test Plan

* The results from the planning activities should be documented in a **test plan**
* The **test plan** is a formal document that describes how tests will be performed
  + List of test **activities** to be performed to ensure meeting the requirements
  + **Features** to be tested, testing approach, schedule, acceptance criteria