



# SOFTWARE TESTING

## CH01: Fundamentals of Software Testing

# Content

- What is Testing?
- Why is Testing Necessary?
- Quality vs Testing
- Testing General Vocabulary
- Testing Objectives
- General Testing Principles
- Test Process

# What is TESTING?

A white laptop is positioned on the left side of the slide, showing its screen and keyboard. The laptop's screen displays a colorful Windows desktop background with various icons like the Start button, File Explorer, and Task View. The keyboard is visible below the screen.

Most people have an experience with software that did not work as expected. Software that does not work correctly can lead to many problems, including:

- Loss of Money
- Time
- Business Reputation
- Injury or Death

“Software testing is a way to assess the quality of the software and to reduce the risk of software failure in operation.”

# What is TESTING?

- **1979:** Software testing is a process of executing a program or application with the intent of finding the errors.
- **1983:** Testing is an activity that measures the Software Quality.
- **2002:** Testing is a concurrent lifecycle process of engineering, using and maintaining testware in order to measure and improve quality of software being tested.

# What is TESTING?

## Modern Definition:

The **process** of verifying and validating that a software program, application or product:

- Meets the business and technical requirements.
- Works as expected(To Meet Expectation).

# Why is Testing Necessary?



**Software Testing** is necessary because we all make mistakes. Some of those mistakes are unimportant, but some of them are expensive and dangerous.

“We need to check everything and anything we produce because things can always go wrong – **humans make mistakes all the time.**”

# Quality VS Testing



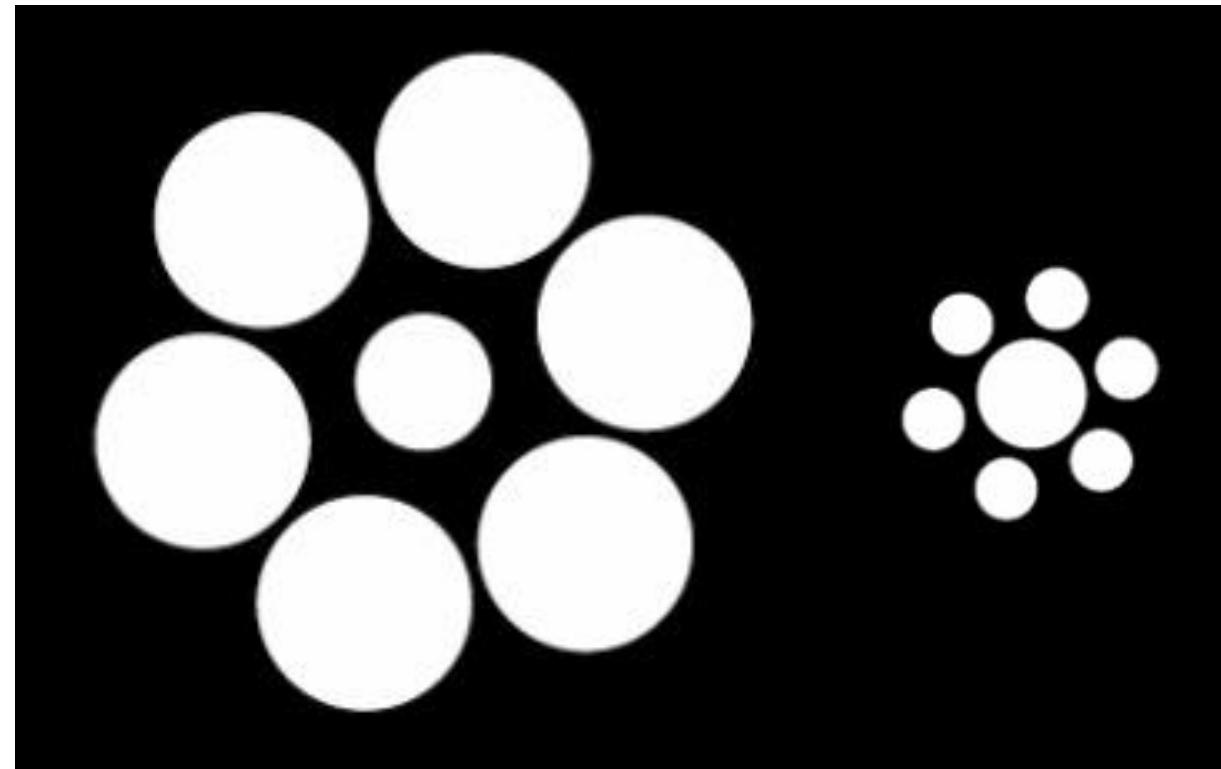
“Quality software is reasonably bug or defect free, delivered on time and within budget, meets requirements and expectations, and is maintainable.”

## ISO Definition of Quality :

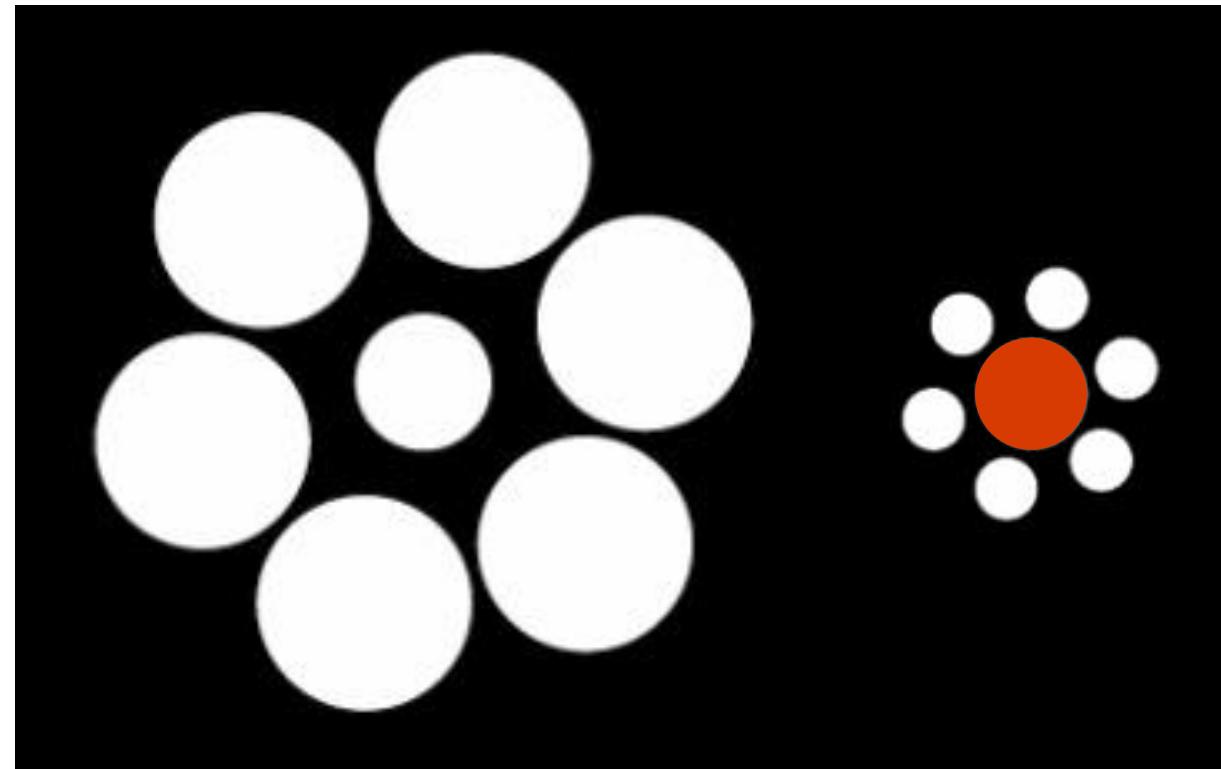
ISO 8402-1986 standard defines quality as:

“**the totality of features and characteristics of a product or service that bears its ability to satisfy stated or implied needs.**”

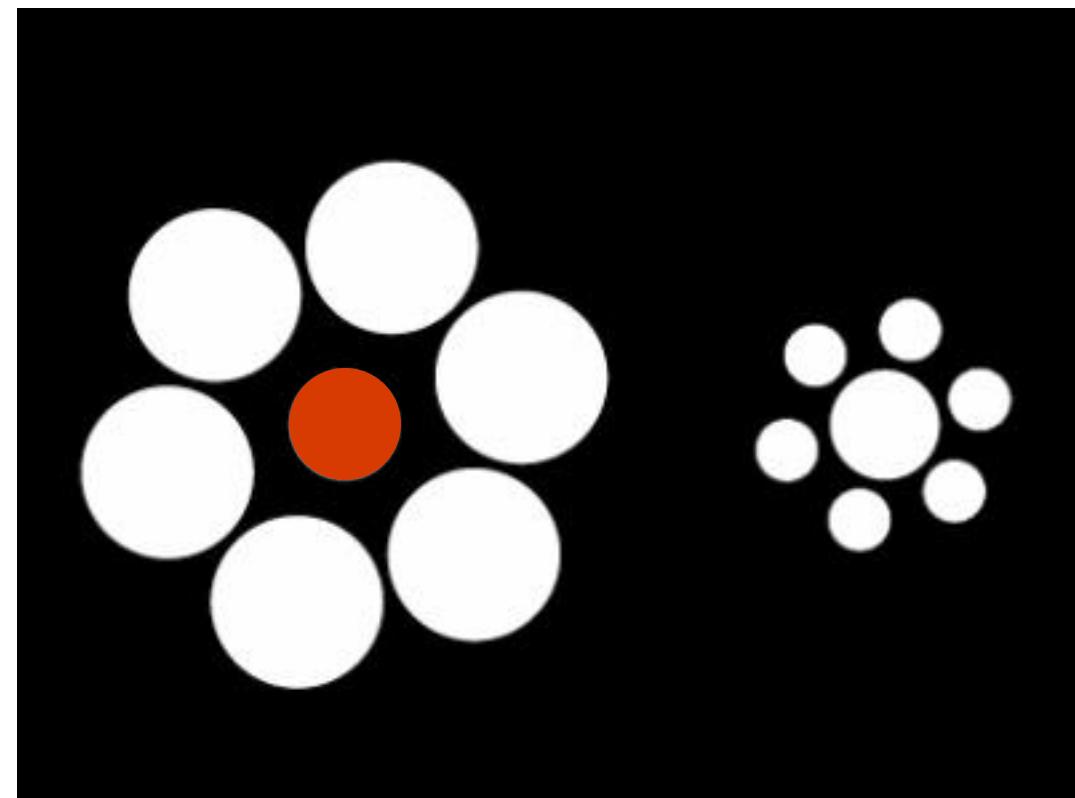
# Cognition in Testing



# Cognition in Testing



# Cognition in Testing



Find N and E?



# Max Information Bad Design



23 Butikker

Teknologi & Gadgets



Se alle våre el-Kjøretøy til Barn og Ungdom.  
Norges største utvalg

el-biler til barn

Nyhet!

12V  
24V  
48V  
fra kr. 1.798

4WD



Ingen  
alders-grense  
el-scooter



el-biler til barn  
( Fra 07.11.2004)  
< Frithjof i sin Fly-Bil

Index

Search

el-retur

Slik Betaler du



Drone-Bike  
Løfteevne: 8kg  
kr. 19.998,-

Index

Akvarium

Alarm

Alkotester

ATV (el.)

Bildefisler

Bil (elektrisk) gas

Bilbane

Conrad-elektronikk

Digital-Kikkert

Disko-Lys

Dummy-Kamera

DVD-Spiller

Elektronikk & DAB

EL-Scooter & Bil

Figurer

Golf-biler (m/skilt)

Hobby & RC

Hoverpod

HP-Måler (Bil)

Isbitmaskin

Kamera (trådløst)

Kino (bærbar)

Kompass (Bil/Båt)

Laser-Jamer (Bil)

Luft-Jekk

Lykt (oppladbar)

Mobil-telefon-1, 2

Motorsykkel-Mini

Omformer (110V)

Oversetter (44 sprk)

PC-mini (9"), 2,

Rakett-Fly

RC Produkter

Robot-Hund

Robot-Støvsuger

Roboter

Solcelle-Ryggsæk

Star Wars

6.998,-

Solcelle-pakker  
Nyhet!

el-biler til barn

Nyhet!

12V  
24V  
48V  
fra kr. 1.798

4WD



Ingen  
alders-grense  
el-scooter



el-biler til barn  
( Fra 07.11.2004)  
< Frithjof i sin Fly-Bil

Index

Search

el-retur

Slik Betaler du



Drone-Bike  
Løfteevne: 8kg  
kr. 19.998,-

Kjøpsloven  
Angreskjema

Ava  
este

Ny

kr.

Elek

3-Hjuls el-sykkel



Ele

Fotball-Trener  
fra kr. 2.598,-

Ta alltid ut 230Vac Adapteren  
når du ikke er tilstede, eller sover



RC Fly-Båt  
70 cm lang  
fra kr. 1299,-

Alle Produktene på  
denne siden lagerføres  
hos ARNGREN i Oslo.  
Se Lagerkoden etter  
Prisene (lev. 2 - 5 dager) :

- Er på Lager
- Kommer før 3 uker
- Lengre enn 3 uker

elsykkel



Fatbike-1500w



el-bil kr. 89.998,-



el-ATV



el-ATV



- Elektriske-Kjøretøy

- Elektrisk-ATV

- Roboter

- el-biler til barn/ungdom

- Solcelle-produkter

- Forbruker Elektronikk

- Batterier & Ladere, etc

- Fjernstyrt produkter

- Disko-Lys - Rakett

- el-scooter

- BMW-i8

el-sykkel



el-ATV



el-bil kr. 89.998,-



el-ATV



el-ATV



el-ATV

el-ATV



el-ATV



el-ATV



el-ATV



el-ATV



el-ATV

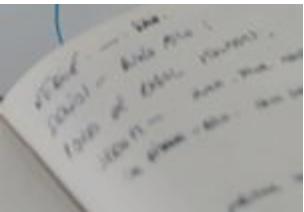


el-ATV

# Less Information Good Design



Gmail Images



# HOME DECORATION

The Best Home Decoration Store

LEARN MORE



# Find Lion?



Not me!!!  
Find **LION...**



## The New Essential Ingredient **#CognitiveQA**



**Cognitive QA** connects quality to desired business outcomes; minimizing risk and rapidly delivering high quality software and products to market with optimal cost savings for an enhanced customer experience and a stronger brand reputation

# Good and Bad Design



# Good and Bad Design

RESPONSIVE



NONRESPONSIVE





?



**GOOD QUALITY**  
**MERCEDES-BENZ OR SUZUKI MEHRAN**

## Quality VS Testing

### Two Views of Quality Definition

- **Popular View:** Quality is directly related to CLASS.
- **Technical View:** To meets Customer Level of Satisfaction within Time Budget and Scope.

## Quality VS Testing

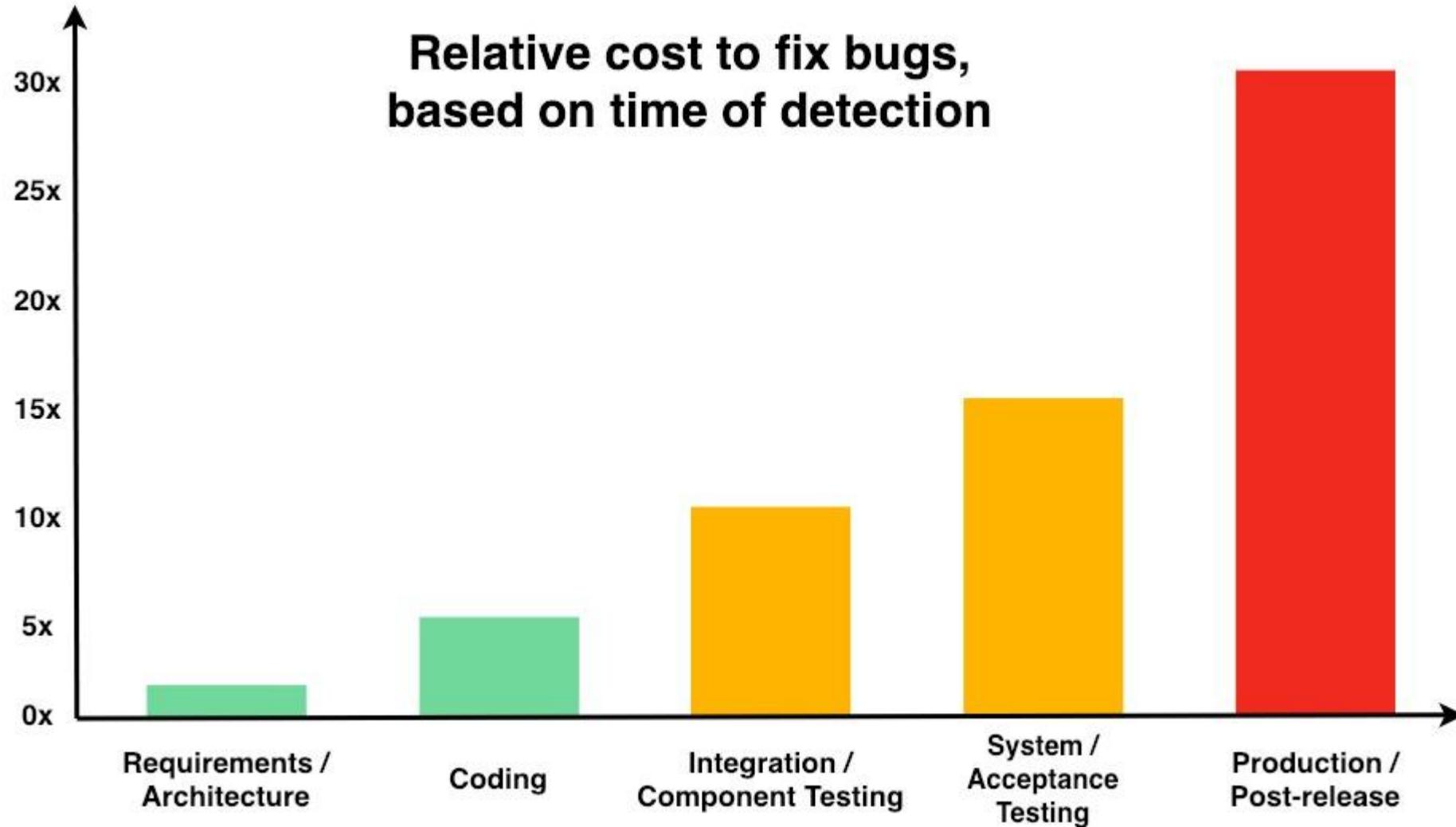
**“TESTING is the measurement of  
QUALITY”**

# Quality VS Testing

## Key Aspects of Quality:

- **Good design** – looks and style
- **Good functionality** – it does the job well
- **Reliable** – acceptable level of breakdowns or failure
- **Consistent** – remain the same
- **Durable** – lasts as long as it should
- **Good after-sales service** - warranty
- **Value for money** – cost-effectiveness

# Cost of Defect



# General Testing Terms

Bug

Defect

Error



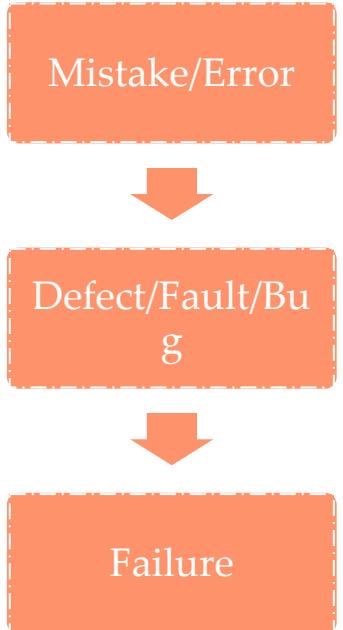
Mistake

Fault

Failure

# General Testing Terms

“A human being can make an error (mistake), which produces a defect(fault, bug) in the program, code or in a document. If a defect in code is executed, the system may fail to do what is should do (or do something it shouldn’t), causing a failure.”



**Note:** Defects in software, systems or documents may result in failures, but not all defects do so.

# Testing

## 7 Principles

### 1 Testing Shows Presence of Defects

Testing always uncovers defects but never tell us how many more are still undiscovered (absence of defects).

### 2 Exhaustive Testing is Impossible

It is impossible to test with every possible input combinations.

### 3 Absence of Errors Fallacy

It is possible that software which is 99% bug-free is still unusable; requirement or expectation may not be met.

### 4 Early Testing Saves Time & Money

Testing should start as early as possible. So that defects in the requirements or design phase are captured in early stages (defects prevention).

### 5 Defects Clustering

Defects present in the form of clusters; approximately 80% of the problems are found in 20% of the modules.

### 6 Testing is Context Dependent

Testing a banking application is different than testing any e-commerce or gaming application.

### 7 Pesticide Paradox

If the same set of repetitive tests are executed, then it will be useless for discovering of new defects.

# General Testing Principle

1. Testing shows the presence of defects, not their absence
2. Exhaustive testing is impossible
3. Early testing saves time and money
4. Defects cluster together
5. Beware of the pesticide paradox
6. Testing is context dependent
7. Absence-of-errors is a fallacy

# Seven Testing Principles

Numbers of testing principles have been suggested over the past 50 years and offer general guidelines common for all testing.

## 1. Testing shows the presence of defects, not their absence:

Testing can show that defects are present but cannot prove that there are no defects. Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, testing is not a proof of correctness.

## 2. Exhaustive testing is impossible

Testing everything (all combinations of inputs and preconditions) is not feasible except for trivial cases. Rather than attempting to test exhaustively, risk analysis, test techniques, and priorities should be used to focus test efforts.

## 3. Early testing saves time and money

To find defects early, both static and dynamic test activities should be started as early as possible in the software development lifecycle. Early testing is sometimes referred to as *shift left*. Testing early in the software development lifecycle helps reduce or eliminate costly changes.

# Seven Testing Principles

## 4. Defects cluster together

A small number of modules usually contains most of the defects discovered during pre-release testing or is responsible for most of the operational failures. Predicted defect clusters, and the actual observed defect clusters in test or operation, are an important input into a risk analysis used to focus the test effort (as mentioned in principle 2).

## 5. Beware of the pesticide paradox

If the same tests are repeated over and over again, eventually these tests no longer find any new defects. To detect new defects, existing tests and test data may need changing, and new tests may need to be written. (Tests are no longer effective at finding defects, just as pesticides are no longer effective at killing insects after a while.) In some cases, such as automated regression testing, the pesticide paradox has a beneficial outcome, which is the relatively low number of regression defects.

# Testing Objectives

- To evaluate work products such as requirements, user stories, design, and code.
- To verify whether all specified requirements have been fulfilled.
- To validate whether the test object is completed and works as the users/stakeholders expect.
- To build confidence in the level of quality of the test object.
- To prevent defects.
- To find failures and defects.
- To reduce the level of risk of inadequate software quality (undetected failures in operation).
- To comply with contractual, legal, or regulatory requirements or standards.
- To provide sufficient information to stakeholders to allow them to make informed decisions, especially regarding the level of quality of the test object.

# Testing Vocab

- ✓ **Test Process:** The set of interrelated activities comprising of test planning, test monitoring and control, test analysis, test design, test implementation, test execution, and test completion.
- ✓ **Test Policy:** A high-level document describing the principles, approach and major objectives of the organization regarding testing.
- ✓ **Test Strategy:** Documentation that expresses the generic requirements for testing one or more projects, providing detail on how testing is to be performed, and is aligned with the test policy.
- ✓ **Test Approach:** The implementation of the test strategy for a specific project.
- ✓ **Test Plan:** Documentation describing the test objectives to be achieved and the means and the schedule for achieving them, organized to coordinate testing activities.
- ✓ **Test Estimation:** The calculated approximation of a result related to various aspects of testing (e.g., effort spent, completion date, costs involved, number of test cases, etc.).

# Testing Vocab

- ✓ **Test Design:** The activity of deriving and specifying test cases from test conditions.
- ✓ **Test Basis:** The body of knowledge used as the basis for test analysis and design.
- ✓ **Test Condition:** A feature, piece of functionality, or anything that you intend to verify.
- ✓ **Test Analysis:** The activity that identifies test conditions by analyzing the test basis.
- ✓ **Test Cases:** A set of preconditions, inputs, actions, expected results and postconditions, based on test conditions.
- ✓ **Test Script:** A sequence of instructions for the execution of a test (Automated Code).
- ✓ **Test Procedure:** A sequence of test cases in execution order, and any associated actions that may be required to set up the initial preconditions and any wrap up activities post execution.
- ✓ **Test Suite:** A set of test cases or test procedures to be executed in a specific test cycle.
- ✓ **Test Data:** Data created or selected to satisfy the execution preconditions and inputs to execute one or more test cases.

# Testing Vocab

- ✓ **Test Object:** The component or system to be tested.
- ✓ **Test Item:** A part of a test object used in the test process.
- ✓ **Test Objective:** A reason or purpose for designing and executing a test.
- ✓ **Test Oracle:** A source to determine expected results to compare with the actual result of the system under test.
- ✓ **Test Automation:** The use of software/script to perform or support test activities.
- ✓ **Test Environment:** An environment containing hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test.

# Testing Vocab

- ✓ **Testware:** Work products produced during the test process for use.
- ✓ **Test Input:** The data received from an external source by the test object during test execution. The external source can be hardware, software or human.
- ✓ **Test Implementation:** The activity that prepares the testware for test execution based on test analysis and design.
- ✓ **Test Execution:** The process of running a test on the component or system under test, producing actual result(s).
- ✓ **Test Harness:** A test environment comprised of stubs and drivers needed to execute a test.
- ✓ **Test Progress Report:** A test report produced at regular intervals about the progress of test activities against a baseline, risks, and alternatives requiring a decision. and/or times, and interdependencies.
- ✓ **Test Summary Report:** A test report that provides an evaluation of the corresponding test items against exit criteria.

# Testing Vocab

Test Oracle

Test Basis □ Test Condition □ Test Case □ Test Procedure □ Test Implement □ Test Execution □

Test Progress Report □ Test Summary Report

Test Analysis      Test Design

Test SUITE = Test Procedure + Test Case

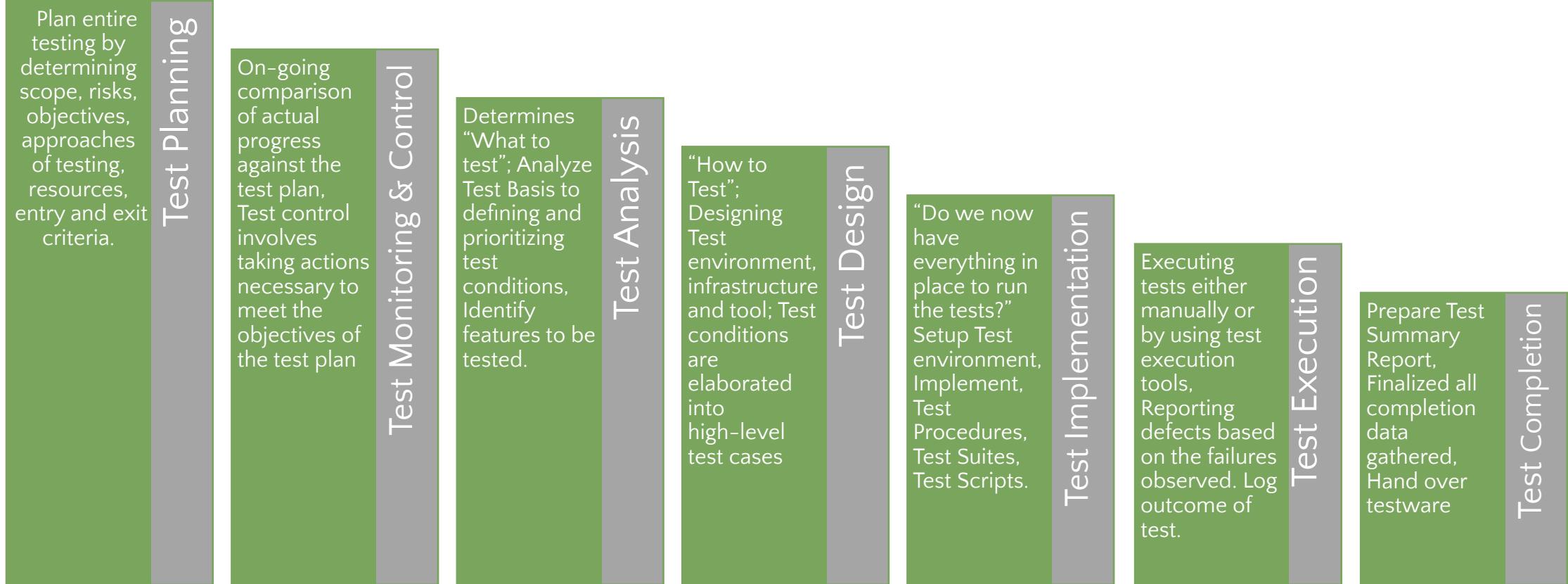
# Test Process

A test process consists of the following main groups of activities:

1. Test Planning
2. Test Monitoring and Control
3. Test Analysis
4. Test Design
5. Test Implementation
6. Test Execution
7. Test Completion

Group  Activities  Sub Activities  Tasks (which would vary from one project or release to another)

# Fundamental Test Process



## Test Planning:

- To determine the scope and risks.
- To Identify the objectives of testing.
- To determine the test approach.
- To implement the test policy and/or the test strategy.
- To determine the required test resources like people, test environments, PCs, etc.
- To schedule test analysis, test design, test implementation, execution & evaluation.
- To determine the Entry and Exit criteria

**ARTIFACT | WORK PRODUCT: TEST PLAN DOCUMENT**

# Test Process

## Test Monitoring and Control:

- Test monitoring involves the on-going comparison of actual progress against the test plan.
- Test control involves taking actions necessary to meet the objectives of the test plan.
- Test monitoring and control are supported by the evaluation of exit criteria.
- Checking test results and logs against specified coverage criteria
- Assessing the level of component or system quality based on test results and logs
- Determining if more tests are needed.
- Test progress against the plan is communicated to stakeholders in test progress reports, including deviations from the plan and information to support any decision to stop testing.

# Test Process

## Test Analysis:

- Determines “**what to test**”.
- Analyze Test Basis to defining and prioritizing test conditions for each feature considering functional, non-functional, and structural characteristics, other business and technical factors, and levels of risks
- Identifying features and sets of features to be tested
- Capturing traceability between test basis and the associated test conditions
- Test analysis may also result in the discovery and reporting of defects in the test basis.

# Test Process

## Test Design:

- Test design answers the question “**how to test?**”
- Test conditions are elaborated into high-level test cases, sets of high-level test cases, and other Testware.
- Designing and prioritizing test cases and sets of test cases
- Identifying necessary test data to support test conditions and test cases
- Designing the test environment and identifying any required infrastructure and tools
- Capturing bi-directional traceability between the test basis, test conditions, test cases.

# Test Process

## Test Implementation:

- Answers the question “**do we now have everything in place to run the tests?**”
- Implementation of testware necessary for test execution.
- Implementation of Test Procedures and Test Suites.
- Creation of automated Test Scripts.
- Prioritize and Schedule Test Suite for Test execution.
- Setup the test environment and verifying that required things has been set up correctly.
- Preparing test data and ensuring it is properly loaded in the test environment.
- Verifying and updating traceability between the test basis, test conditions, test cases, test procedures, and test suites.

# Test Process

## Test Execution

- Test suites are run in accordance with the test execution schedule.
- Recording the IDs and versions of the test item(s) or test object, test tool(s), and testware.
- Executing tests either manually or by using test execution tools
- Comparing actual results with expected results
- Analyzing anomalies to establish their likely causes.
- Reporting defects based on the failures observed.
- Logging the outcome of test execution (e.g., pass, fail, blocked).
- Re-execute test activities (Confirmation Testing, and/or Regression Testing).

# Test Process

## Test Completion:

- Test completion activities collect data from completed test activities.
- Test completion activities occur at project milestones.
- Checking whether all defect reports are closed.
- Creating a Test Summary Report to be communicated to stakeholders.
- Finalizing and archiving the test environment, the test data, the test infrastructure for later reuse.
- Handing over the testware to the maintenance teams, other project teams, and/or other stakeholders who could benefit from its use
- Analyzing lessons learned from the completed test activities to determine changes needed for future iterations, releases, and projects
- Using the information gathered to improve test process maturity

# Cost of Quality

*Money spent during  
and after the project*

## Cost of Non Conformance (Bad Quality)

*Money spent during and after  
the project because of failures*

## Cost of Conformance (Good Quality)

*Money spent during the project  
to avoid failures*

### Prevention Cost (build a quality product)

Training QC staff  
Document the process  
Testing Equipment  
Time required to do it right

### Appraisal Cost (assess a quality product)

Running the Test  
Destructive Testing Loss  
Inspecting Deliverables

### Internal Failure Cost (failure found by the project)

Rework  
Scrap

### External Failure Cost (failure found by customer)

Liabilities, Law Suites, Product recalls  
Warranty Work  
Lost Business or Credibility

Any Question?

End!