



Fundamentals of Testing

01ST FEBRUARY, 2024

(expleo)

Contents

- Overview of Testing
- Why Testing is Necessary?
- Seven Principles of Testing
- Test Process
- Psychology of Testing
- Quiz

Overview of Testing



Overview of Testing

Objective of Testing



Evaluate Work Products

Review requirements, user stories, designs, and code for quality and completeness



Reduce Risk

Minimize the risk of inadequate software quality reaching production



Provide Information

Give stakeholders data to make informed decisions about releases and quality



Objective of Testing



Find Defects

Trigger failures and identify defects to improve software reliability



Verify Requirements

Ensure specified, contractual, legal, and regulatory requirements are fulfilled



Build Confidence

Validate the test object works as stakeholders expect and is complete



Overview of Testing

Objective of Testing

- To comply with contractual, legal, or regulatory requirements or standards, and/or to verify the test object's compliance with provided requirements or standards.
- Another objective may be to increase **code coverage** of the component tests and to give information to stakeholders about the risk of releasing the system at a given time.



What is Testing?

- Testing is process of trying to discover every conceivable **fault or weakness in a work product.**
- Testing is a process of executing a program with the intent of finding an error.
- A good test is one that has a high probability of **finding an undiscovered error.**
- A successful test is one that uncovers an undiscovered error.
- Testing is a process used to **identify the correctness, completeness and quality of developed computer software.**

Testing, apart from finding errors, is also used to test performance, safety, fault-tolerance or security.



What is
Software
Testing?

What is Testing? Let's see some real time Scenario

- Software systems are an integral part of life, from business applications (e.g., banking) to consumer products (e.g., cars).
- Software testing is a broad term that covers a variety of processes designed to ensure that software applications function as intended, can handle the volume required, and integrate correctly with other software applications.
- Software that **does not work correctly** can lead to many problems, including **loss of money, time, or business reputation, and even injury or death**.
- For example, In September 2016, where the Jio SIM was publicly available, at that time it was free of cost, after the release, **bugs or defects is identified by the users** and which is rectified by the JIO Network. In this process we have implemented **Beta testing**.

Overview of Testing

What is Software Testing? Let's see some real time Scenario

- Software testing is a process which includes many different activities; test execution plays a major role which includes checking of results.

Test Execution Process:

- The test process includes activities such as **test planning, analyzing, designing, and implementing tests, reporting test progress and results, and evaluating the quality of a test object.**
- In **Test process, testing includes reviewing work products such as requirements, user stories, and source code.**
- For Example: A credit card company includes a page where a customer can request a credit increase, which is a separate code base from login functionality. Testers might perform integration tests to make sure the system remembers the user after they navigate to the credit increase page, and again after a successful request. Testing of two or more modules together is called **Integration Testing in Software testing.**

Overview of Testing

Software Testing:



FIRST NAME

LAST NAME

STREET ADDRESS (NO P.O. BOXES)

STREET ADDRESS LINE 2

CITY STATE ZIP CODE

Please Select

DATE OF BIRTH SOCIAL SECURITY NUMBER

MM/DD/YYYY Are you under 21 years of age? XXXXXXXX

What card benefit is most important to you?

Travel rewards Cash back Low interest Not sure yet

How do you rate your credit level? [Learn about credit levels](#)

Excellent Average Rebuilding

I understand that this is not a credit card application

[See Your Card Offers](#)

SECURE CREDIT CARD PAYMENT

Subtotal: £ 1000.00
Shipping: £ 0
VAT: £ 175.00
Total: £ 1175.00

Card Holder Details - * indicates a required field.

Name: Roy Leigh *
Email address: RL@edp.co.uk *
Phone number: 0114 200 0000 *
Street: 10 Beauchief Drive *
City/Town: Sheffield *
County: South Yorkshire *
Postcode: S8 3DP *

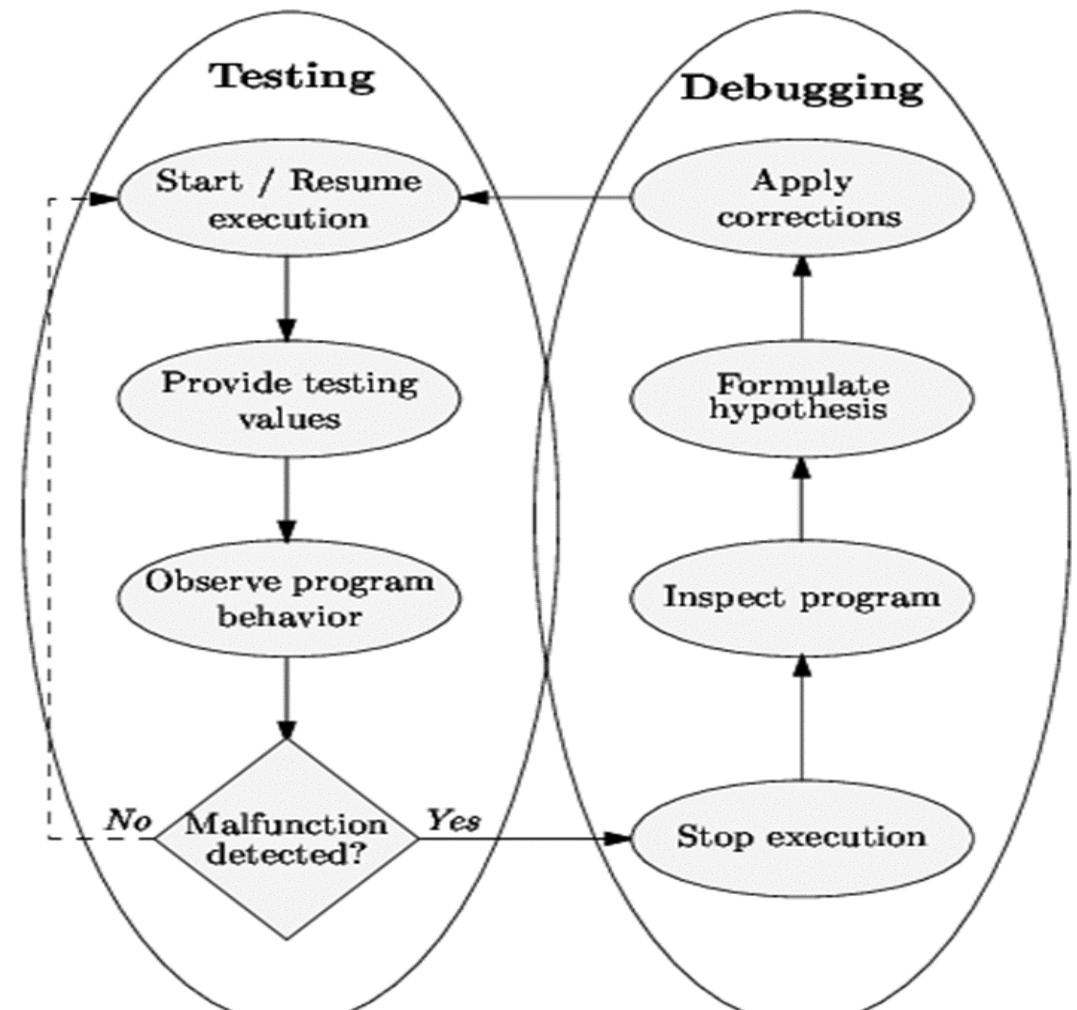
Card type: Visa * Start date: 01 / 05 *
Card number: XXXXXXXXXXXXXXXXXX * Expiry date: 01 / 09 *
Issue number: (Switch cards only) CV2 number: XXX

[Cancel](#)

[Complete Order](#)

Testing and Debugging

- **Executing tests can show failures** that are caused by defects in the software.
- **Debugging** is the development activity that **finds, analyzes, and fixes such defects**. Subsequent confirmation testing checks whether the fixes resolved the defects. In some cases, testers are responsible for **the initial test and the final confirmation test**, while developers do the debugging, associated component and component integration testing.



(expleo)

Overview of Testing

Testing and Debugging



While often confused, testing and debugging are distinct activities with different goals and processes.

1

Testing

Identifies that failures exist (dynamic) or finds defects directly (static)

2

Debugging

Analyzes root causes, diagnoses the defect, and implements fixes

3

Confirmation

Verifies fixes resolved the problem without introducing new issues

Key Insight: Dynamic testing requires debugging (reproduction, diagnosis, fixing), while static testing identifies defects directly without needing reproduction.

Testing and Debugging

- "Testing : answers What is wrong?
- Debugging: answers Why is it wrong and how do we fix it?"

Aspect	Testing	Debugging
Activity	Finds failures and defects	Investigates and removes the root cause of failures
Purpose	To detect issues in the software and assess quality	To identify, analyze, and correct defects that cause failures
Performed	Testers (sometimes developers or users in small teams)	Developers (typically)
Input	Software build, test cases, user stories, test environments	Test reports, logs, stack traces, failing code
Output	Test results, defect reports, logs	Fixed code, updated builds

Dynamic Testing & Debugging

- **Dynamic testing may trigger failures caused by defects.**
- **Debugging follows to:**
 1. Reproduce the failure
 2. Diagnose the root cause\
 3. Fix the defect
- **After fixing:**
 - Confirmation Testing ensures the problem is fixed
 - Regression Testing checks for side-effects elsewhere

Static Testing & Debugging

- Static testing (like reviews or analysis) finds defects without running the code
- No need for failure reproduction or root cause diagnosis
- Debugging here simply means removing the identified defect.

Type	Description	Example
Static	No execution of code	Code reviews, static analysis
Dynamic	Involves running the software	Unit tests, UI tests

Why is testing necessary ?

"Software defects can potentially cause monetary loss or even loss of life."

- **Defective software** can lead to loss of money, time, reputation injury or death .
- Causes of software defects
- Defect Classes and defect repository
- Role of testing in software Development, maintenance and operations .
- **Testing and quality**
- When **defects are detected**, and subsequently fixed, this **contributes to the quality** of the components or systems.

Overview of Testing

Notable Software Bugs



NASA(Motor failure and overheating of engine)



Therac-25 radiation therapy

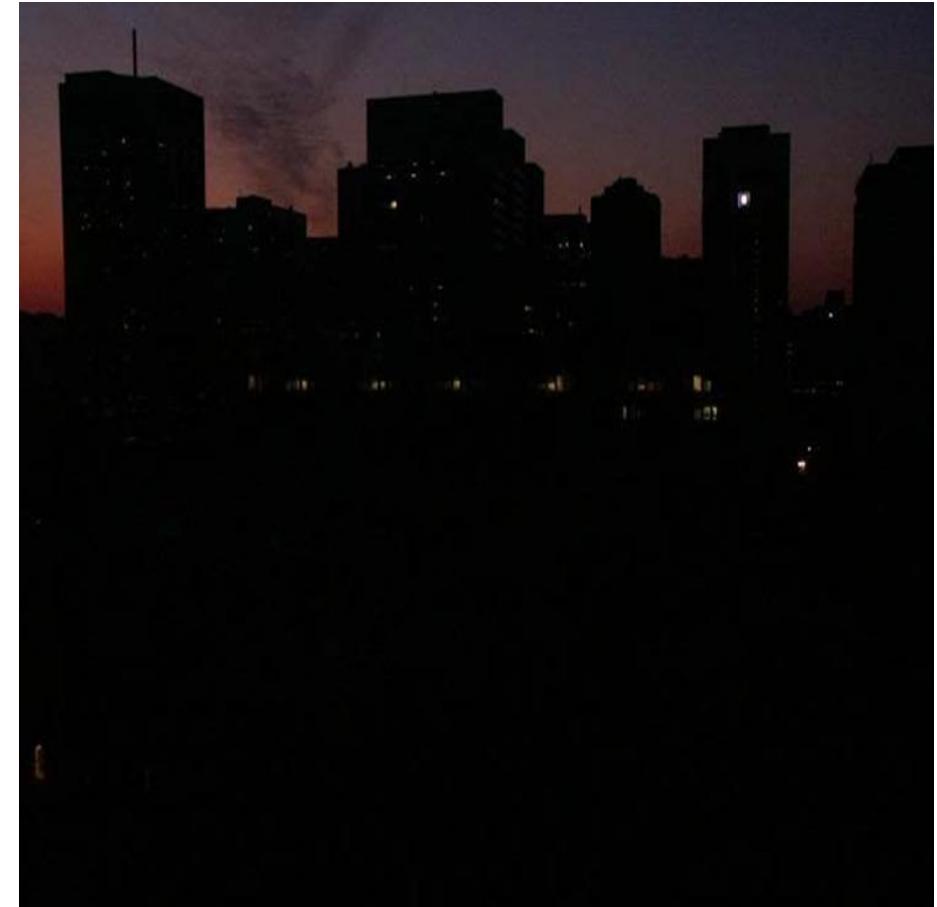


Google's search engine

Overview of Testing

Notable Software Bugs

- **Electricity** :The Northeast blackout of 2003 was a widespread power outage that occurred throughout parts of the Northeastern and Midwestern United States and Ontario, Canada.
 - 508 generating units and 256 power plants shut down.
 - Affected 10 million people in Ontario, Canada.
 - Affected 40 million people in US states.
 - Financial losses of \$6 Billion USD.
 - The alarm system in the energy management system failed due to a software error and operators were not informed of the power overload in the system.



Testing's Contributions to Success

- It is quite common for software and systems to be delivered into operation and, due to the presence of defects, to subsequently cause failures or otherwise not meet the stakeholders' needs.
- However, using **appropriate test techniques** can reduce the frequency of such **problematic deliveries**.
- When those techniques are applied with the appropriate level of test expertise, in the appropriate test levels, and at the appropriate points in the software development lifecycle.

Quality Assurance

- Quality assurance is typically focused on adherence to **proper processes**, in order to provide confidence that the appropriate levels of quality will be achieved.
- When processes are carried out properly, the work products created by those processes are generally of **higher quality**, which contributes to **defect prevention**.
- In addition, the use of **root cause analysis** to detect and **remove the causes of defects**, along with the proper application of the findings of retrospective meetings to improve processes, are important for effective quality assurance.



Quality Assurance

- QA is a set of processes **designed to ensure the developed product satisfies customer requirements in a reliable fashion.**
- QA ensures implementation of standard processes in an organization and their constant improvisation.
- QA is a broader term which expands to industries that deal with **raw materials, assemblies, production, components, services, management, inspection processes** etc.



Quality Control

- QC is a set of procedures designed to ensure a product adheres to the set of quality criteria and meets the client or customer requirements.
- Quality control involves various activities, including test activities, that support the achievement of appropriate levels of quality.
- QC ensures the final product meets the business criteria for success.



Overview of Testing

Difference between Quality Assurance and Quality Control



Errors, Defects, and Failures

- A person can make an **error (mistake)**, which can lead to the introduction of a **defect (fault or bug)** in the software code or in some other related work product.
- **An error** that leads to the introduction of a **defect** in one **work product** can trigger an **error** that leads to the introduction of a **defect** in a **related work product**.
- For example, a requirements elicitation error can lead to a requirements defect, which then results in a programming error that leads to a defect in the code.
- If a defect in the code is executed, this may cause a failure, but not necessarily in all circumstances.
- For example, some defects require very specific inputs or preconditions to trigger a failure, which may occur rarely or never.

Errors, Defects, and Failures - Example

- **Error:** A human mistake made during coding, design, or requirement interpretation.

Example: A developer writes $x = y + 1$ instead of $x = y - 1$.

- **Defect (Bug):** A flaw in the software caused by an error. It exists in the code.

Example: The incorrect formula from the error leads to wrong output.

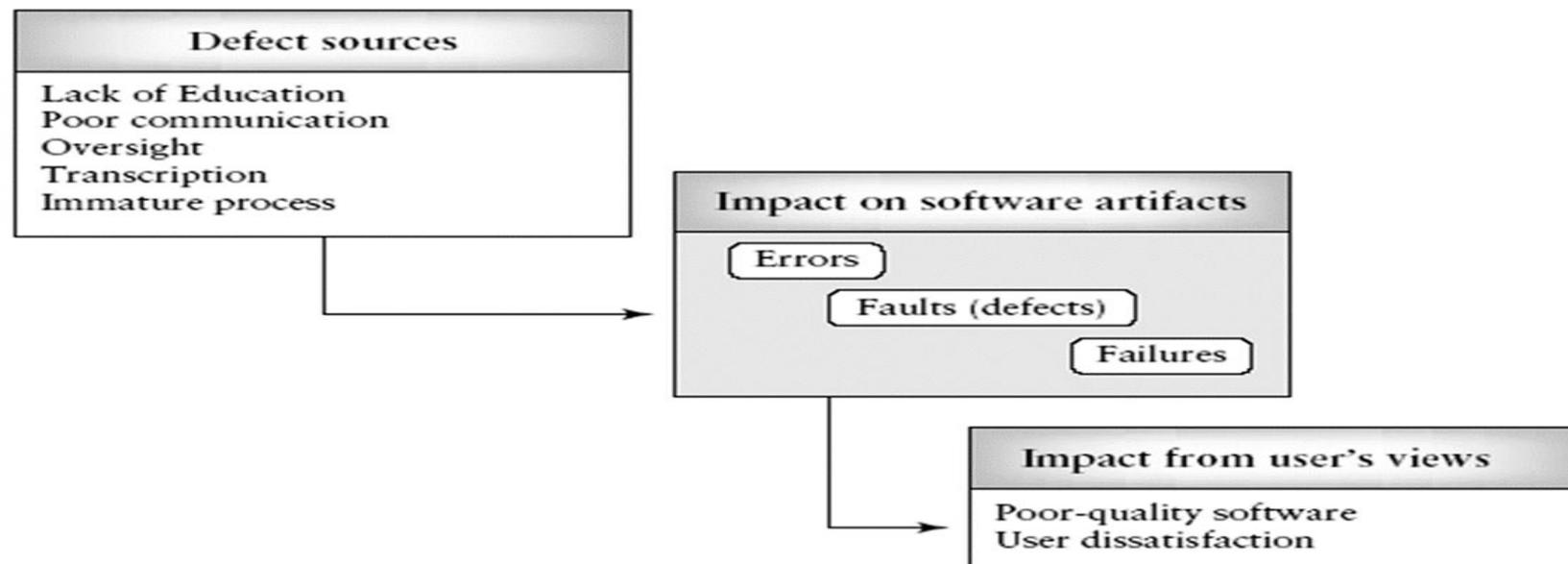
- **Failure:** The system behaves unexpectedly or incorrectly when a defect is executed.

Example: The app crashes or produces wrong results due to the defect.

Overview of Testing

Errors, Defects, and Failures

"A mistake in coding is called **Error**, error found by tester is called **Defect**, defect accepted by development team then it is called Bug, build does not meet the requirements then it Is **Failure**."

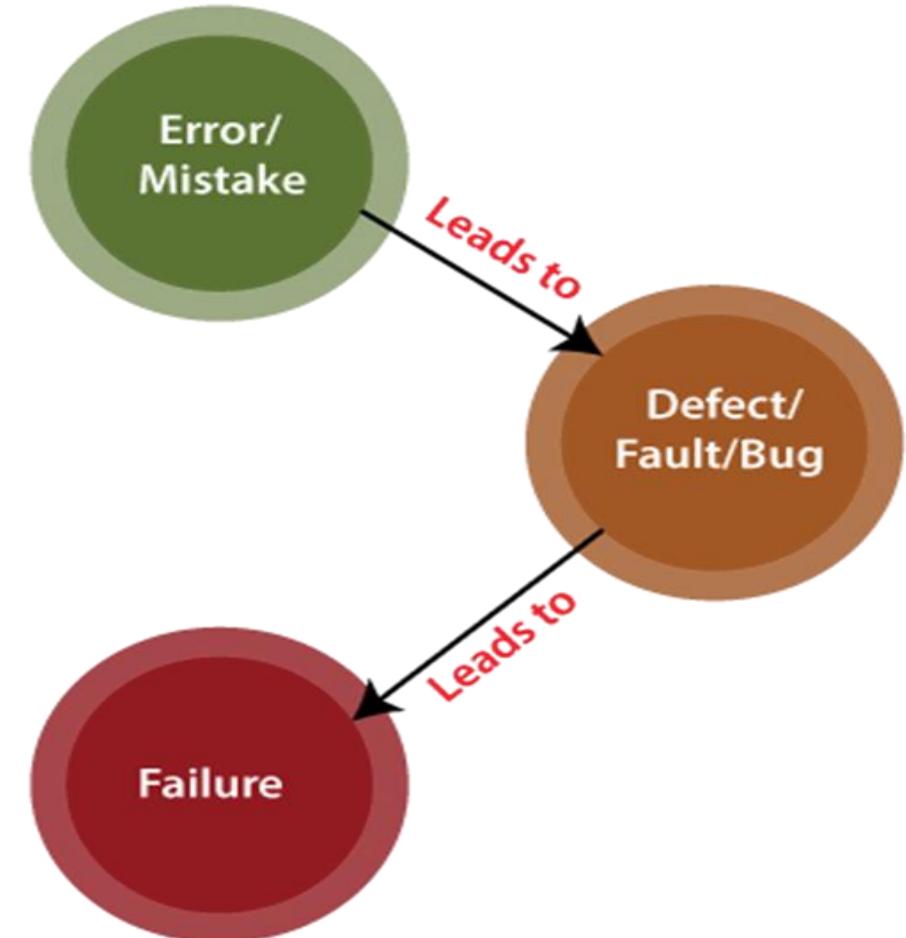


Overview of Testing

Quality Control

Errors may occur for many reasons, such as:

- Time pressure
- Human fallibility
- Inexperienced or insufficiently skilled project participants
- Miscommunication between project participants, including miscommunication about requirements and design



Overview of Testing

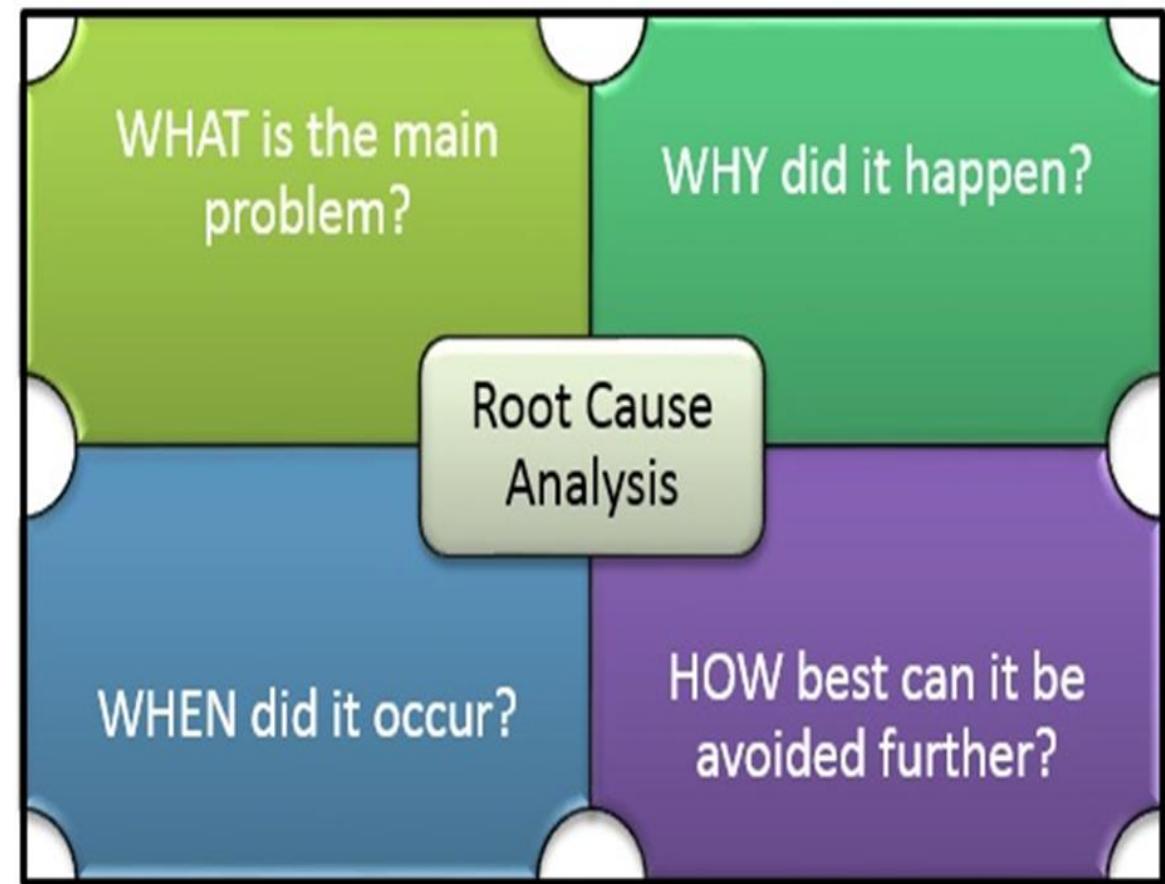
Quality Control

- Complexity of the code, design, architecture, the underlying problem to be solved, and/or the technologies used .
- Misunderstandings about intra-system and inter-system interfaces, especially when such intra system and inter-system interactions are large in number
- New, unfamiliar technologies
- In addition to failures caused due to defects in the code, failures can also be caused by environmental conditions.



Defects, Root Causes and Effects

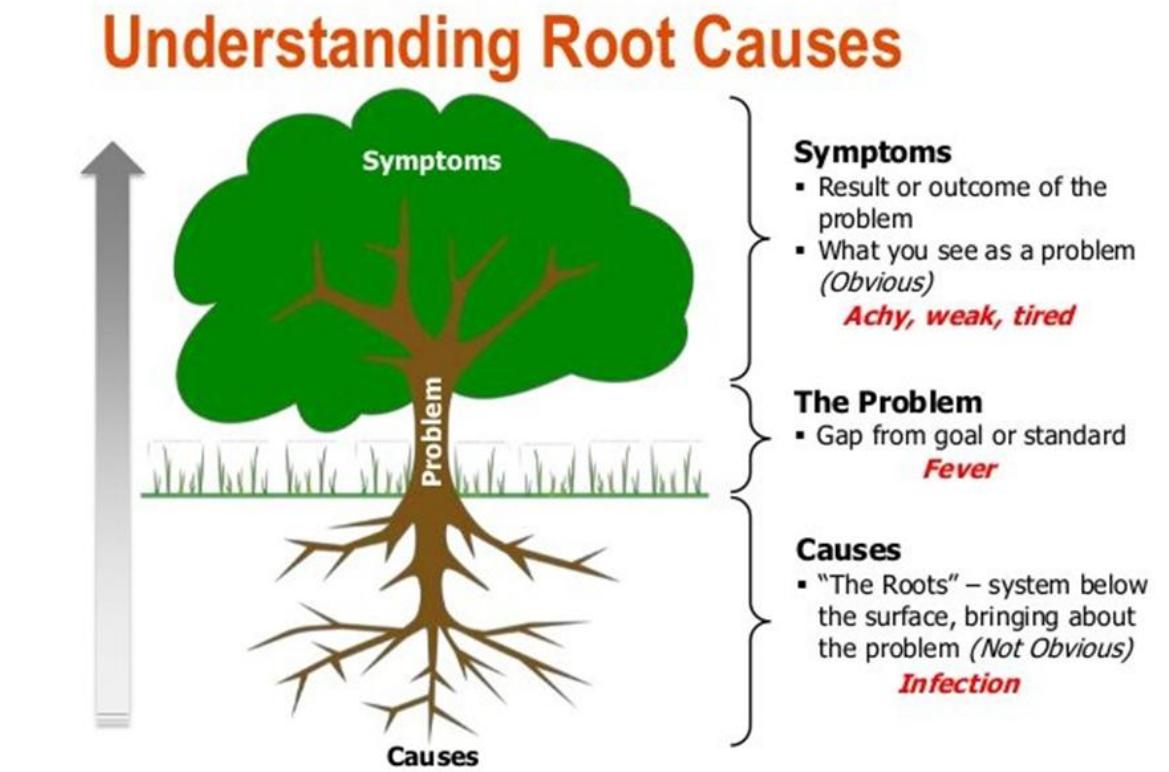
- The root causes of defects are the earliest actions or conditions that contributed to creating the defects.
- **Root Cause Analysis (RCA)** is an approach used in software testing to **identify the root causes of defects or problems** and address them instead of treating the symptoms.
- RCA is done accurately; it helps to prevent defects in the later releases or phases.



Overview of Testing

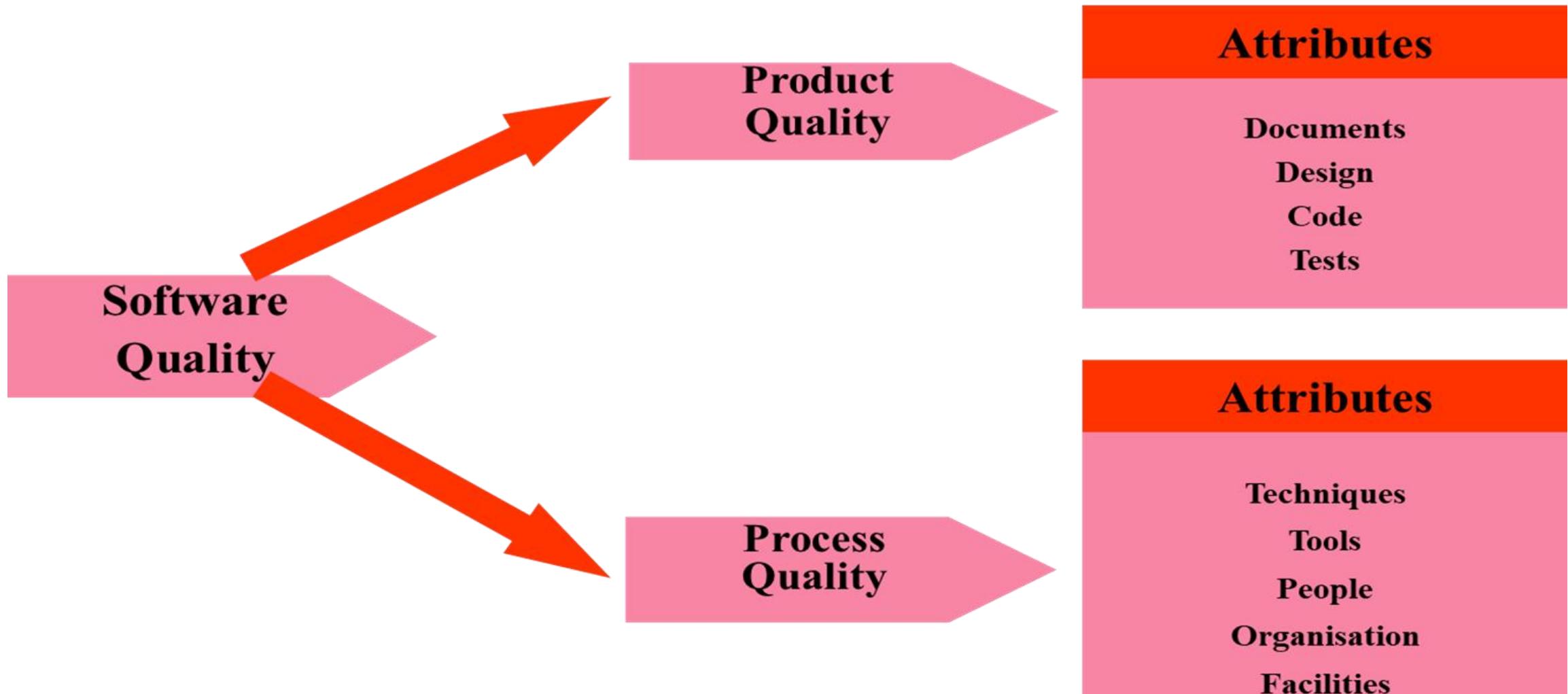
Defects, Root Causes and Effects

- **Defects** can be analyzed to **identify** their **root causes**, to reduce the occurrence of similar defects in the future.
- By focusing on the most significant root causes, **root cause analysis** can lead to process **improvements that prevent a significant number of future defects** from being introduced.



Overview of Testing

Testing & Quality



Overview of Testing

Testing & Quality(QA and QC)

Quality Assurance	Quality Control	Testing
<ul style="list-style-type: none">QA includes activities that ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.	<ul style="list-style-type: none">It includes activities that ensure the verification of a developed software with respect to documented (or not in some cases) requirements.	<ul style="list-style-type: none">It includes activities that ensure the identification of bugs/error/defects in a software.
<ul style="list-style-type: none">Focuses on processes and procedures rather than conducting actual testing on the system.	<ul style="list-style-type: none">Focuses on actual testing by executing the software with an aim to identify bug/defect through implementation of procedures and process.	<ul style="list-style-type: none">Focuses on actual testing.
<ul style="list-style-type: none">Process-oriented activities.	<ul style="list-style-type: none">Product-oriented activities.	<ul style="list-style-type: none">Product-oriented activities.
<ul style="list-style-type: none">Preventive activities.	<ul style="list-style-type: none">It is a corrective process.	<ul style="list-style-type: none">It is a preventive process.
<ul style="list-style-type: none">It is a subset of Software Test Life Cycle (STLC).	<ul style="list-style-type: none">QC can be considered as the subset of Quality Assurance.	<ul style="list-style-type: none">Testing is the subset of Quality Control.

Overview of Testing

Quiz



1) A bug or defect is:

- a) mistake made by a person
- b) A run-time problem experienced by a user
- c) The result of an error or mistake
- d) The result of a failure, which may lead to an error

Answer : Option c)

Overview of Testing

Quiz



2) The effect of testing is to:

- a) Increase software quality
- b) Give an indication of the software quality
- c) Enable those responsible for software failures to be identified
- d) Show there are no problems remaining?

Answer : Option b)

Overview of Testing

Quiz



3. Which of the following is correct?

Debugging is:

- a) Testing/checking whether the software performs correctly**
- b) Checking that a previously reported defect has been corrected**
- c) Identifying the cause of a defect, repairing the code and checking the fix is correct**
- d) Checking that no unintended consequences have occurred as a result of a fix.**

Answer : Option c)

(expleo)

Quiz



4. A phone ringing momentarily distracts a programmer, causing the programmer to improperly program the logic that checks the upper boundary of an input variable. Later, during system testing, a tester notices that this input field accepts invalid input values. The improperly coded logic for the upper boundary check is:

a) The root cause

b) The failure

c) The error

d) The defect

Answer : Option c)

Seven Principles of Testing



Seven Principles of Testing

Seven Testing Principles

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 Principles of Testing

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

- The goal of testing is to **make the software fail**.
- Sufficient testing reduces the presence of defects.
- In case testers are unable to find defects after repeated regression testing doesn't mean that the software is bug-free.
- Testing **talks about the presence of defects and don't talk about the absence of defects**.
- For e.g.: A software tester is testing an android mobile application. He should be working hard to identify potential defects rather than providing defect free product. This will help him to identify more bugs rather than achieving correctness.

Principle 1

Testing shows the presence of bugs



If no defects are found, it is not a proof of correctness

(expleo)

Seven Principles of Testing

2. Exhaustive testing is impossible

- Testing **all combinations of input data values and scenarios are impossible.**
- Testing all combinations would cost more time and money
- Smarter ways of testing should be adopted in order to **complete the project timelines**
- **Prioritize important tests based on risk**
- For E.g.: Assume you are testing the email subscription feature of an application. There are 1000 users subscribed. Testing email subscription is working for 1000 users consumes a lot of time and cost. Instead, identify sample or prioritize the important test based on the requirement.

Principle 2

Exhaustive testing is impossible

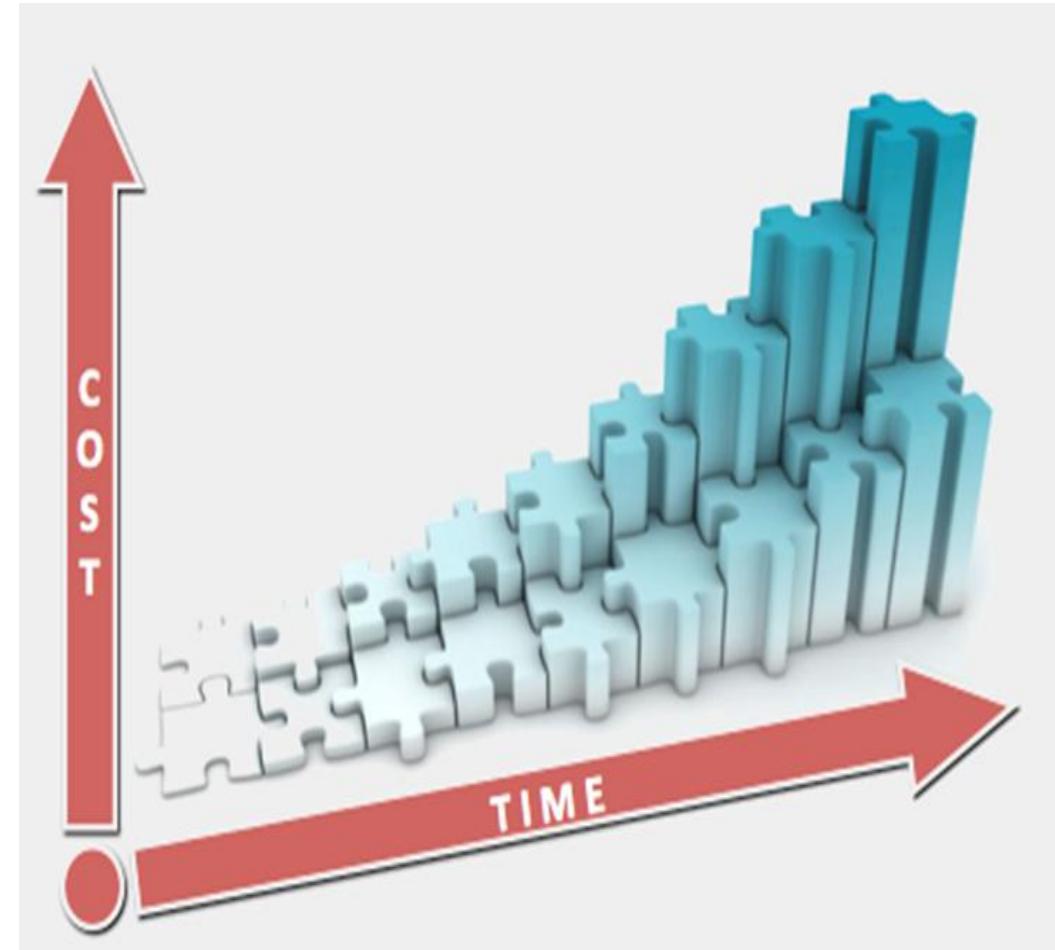


Instead of exhaustive testing, use risks and priorities to focus testing efforts

Seven Principles of Testing

3. Early testing saves time and money

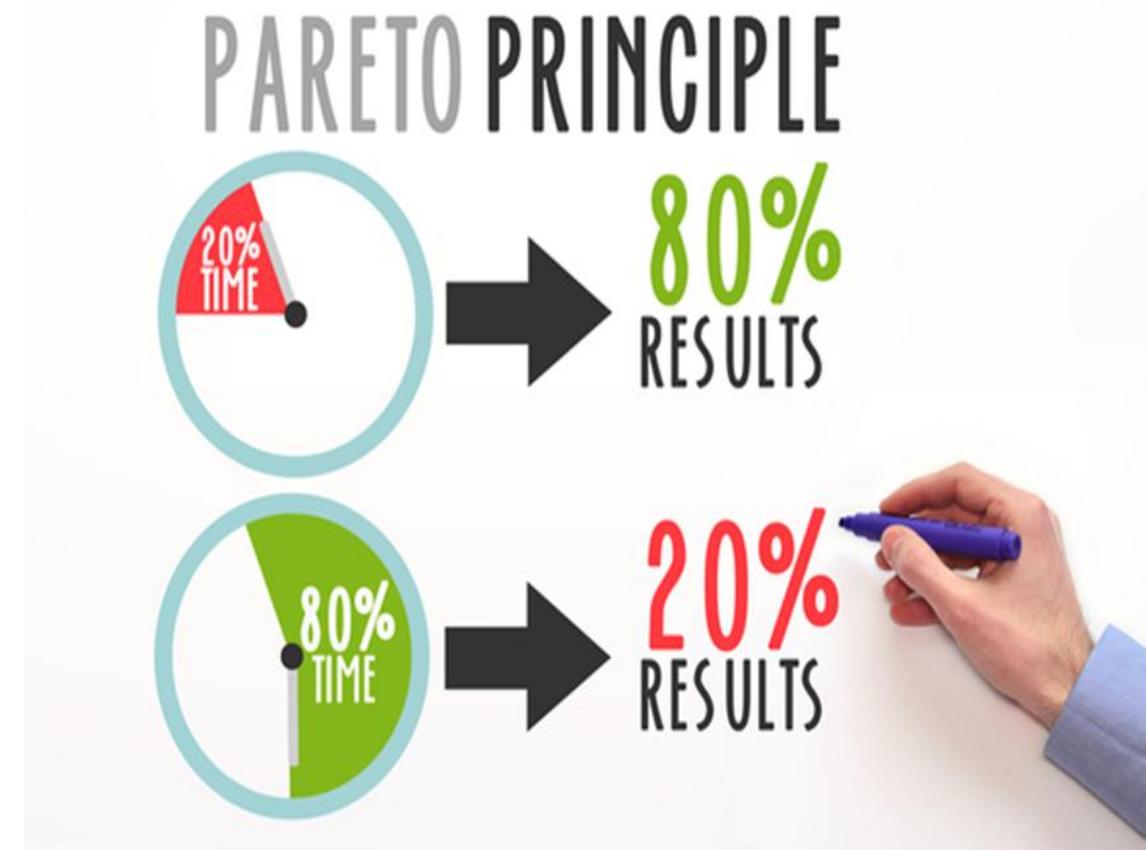
- Software testing starts from requirement gathering.
- Then, testing should be done parallelly along with other product development phases.
- **Finding defects early on will save a lot of money and rework time.**
- For E.g.: Assume two scenarios, first one is you have identified an incorrect requirement in the requirement gathering phase and the second one is you have identified a bug in the fully developed functionality. It is cheaper to change the incorrect requirement compared to fixing the fully developed functionality which is not working as intended.



Seven Principles of Testing

4. Defects cluster together

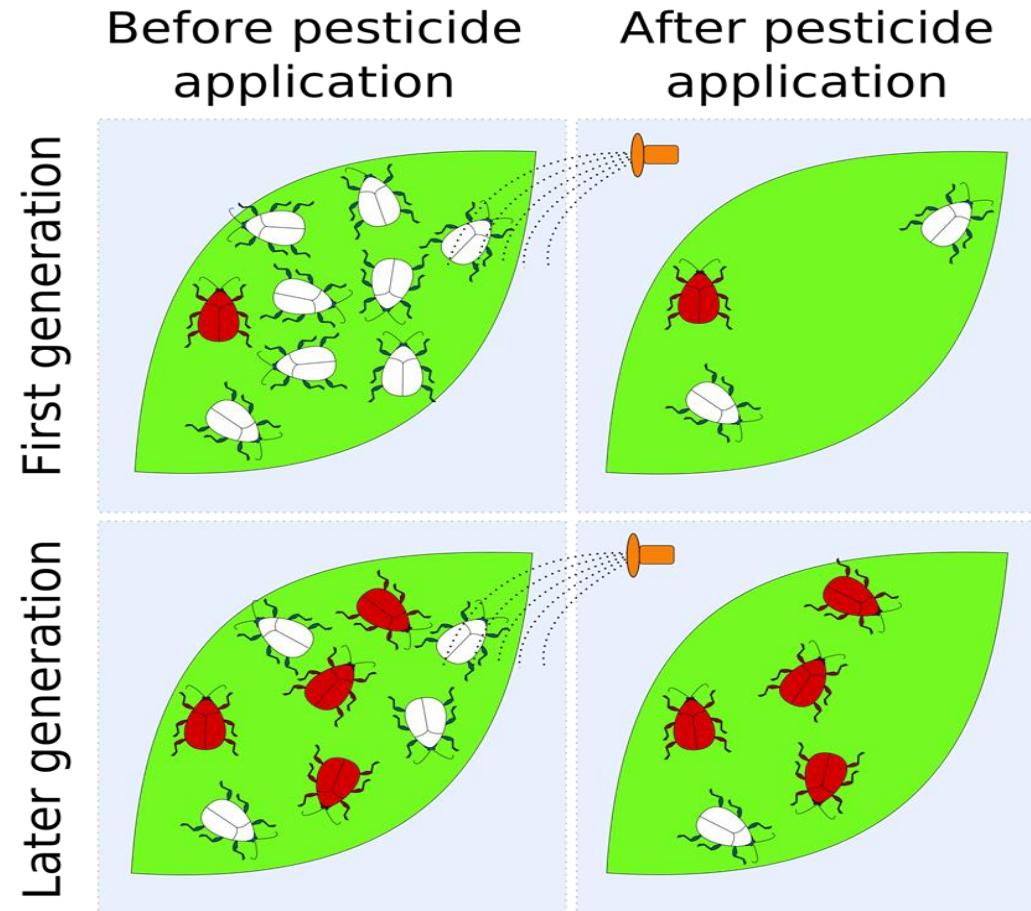
- In a project, a small number of the module can contain most of the defects.
- As per the **Pareto Principle (80-20 Rule)**, **80% of issues comes from 20% of modules and remaining 20% of issues from remaining 80% of modules**. So, we do emphasize testing on the 20% of modules where we face 80% of bugs.



Seven Principles of Testing

5. Test Wear Out

- Repeating the same test cases, again and again, will not find new bugs.
- So, it is necessary to review the test cases and add or update test cases to find new bugs.



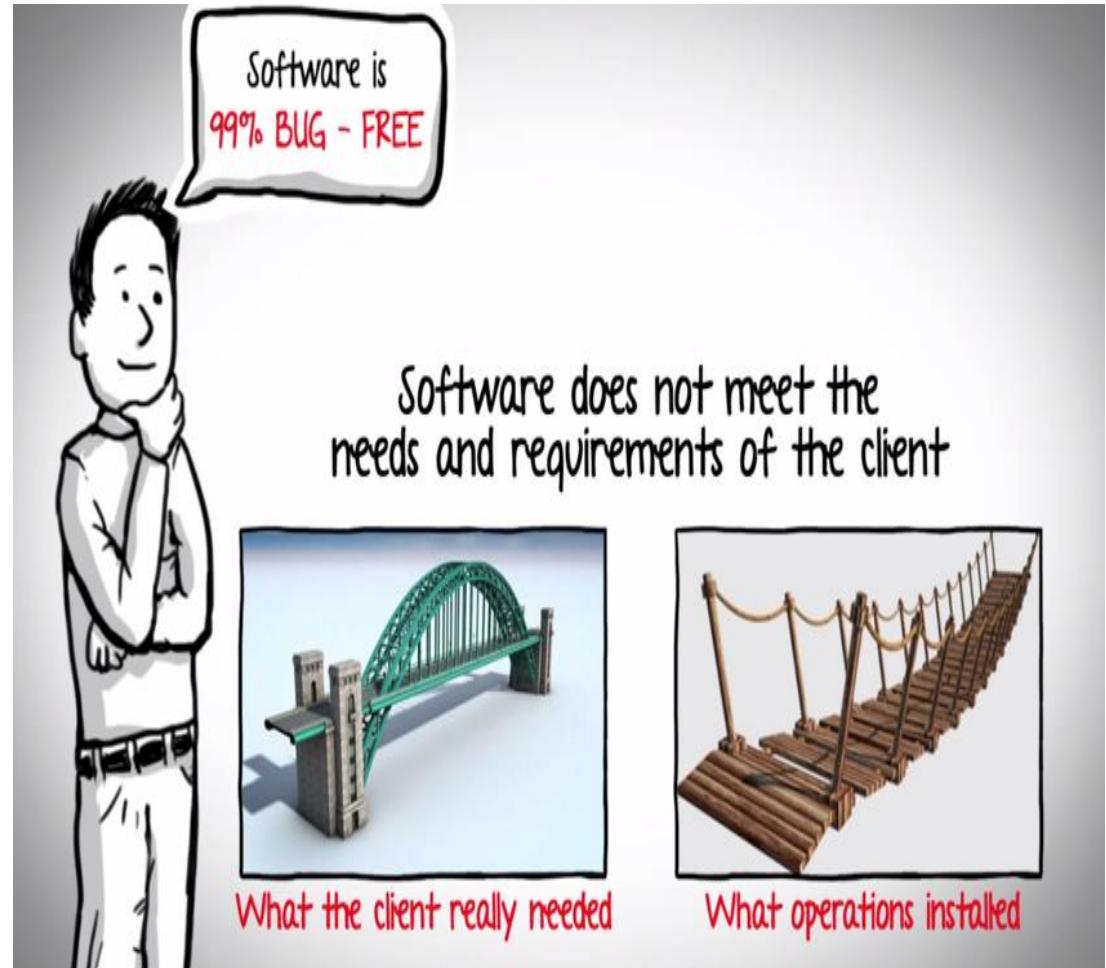
6. Testing is context dependent

- Testing is always context dependent.
- **Software should be tested based on the purpose of the product that has been built.**
- Different software are tested differently.
- Banking site cannot be tested same as the E-commerce website.
- Functionally both are different, so obviously they cannot be tested same.
- For E.g.: Banking site will have functionality features like transactions, interests, savings, etc., the E-Commerce site will have some features like shopping cart, order, payments, etc. So, both are completely different in their functionality. Test scenarios and tests conducted on the software should also be different and depend on the context.

Seven Principles of Testing

7. Absence-of-defects is a fallacy :

- Finding and fixing defects doesn't help if the built software fail to meet user's requirements.
- **Test designed should catch most defects and should cover all client's requirements and specifications before shipping the product.**
- For E.g., Software tester completely tested the product, and the software is 100% defect free. Is that mean the product is ready to be shipped? No! Software tester should verify the test design covers the customer's requirements and specifications. **If the built software fails to meet requirements, then finding and fixing bugs doesn't really help.**



Seven Principles of Testing

Quiz



1. Mr. Test has been testing software applications on mobile devices for a period of 5 years. He has a wealth of experience in testing mobile applications and achieves better results in a shorter time than others. Over several months, Mr. Test did not modify the existing automated test cases and did not create any new test cases. This leads to fewer and fewer defects being found by executing the tests. What principle of testing did Mr. Test not observe?

a) Testing depends on the environment

b) Exhaustive testing is not possible

c) Repeating of same tests will not find new defects

d) Defects cluster together

Answer : Option c)

(expleo)

Seven Principles of Testing

Quiz



2. As a result of risk analysis, more testing is being directed to those areas of the system under test where initial testing found more defects than average. Which of the following testing principles is being applied?

a) Tests Wear Out

b) Testing is context dependent

c) Absence-of-errors is a fallacy

d) Defects cluster together

Answer : Option d)

Seven Principles of Testing

Quiz



3. A product owner says that your role as a tester on an Agile team is to catch all the bugs before the end of each iteration. Which of the following is a testing principle that could be used to respond to this (false) statement?

a) Root cause analysis

b) Testing shows the presence of defects

c) Absence-of-errors is a fallacy

d) Defects cluster together

Answer : Option b)

Seven Principles of Testing

Quiz



4. Which testing principle explains why this approach is unrealistic?

- a) Defect clustering
- b) Exhaustive testing is impossible
- c) Pesticide paradox
- d) Early Testing

Answer : Option b)

Seven Principles of Testing

Quiz



5. Scenario: In a project, testing started only after the coding phase. Several requirement misunderstandings were discovered late. Which principle was violated?

- a) Defect clustering
- b) Exhaustive testing is impossible
- c) Pesticide paradox
- d) Early Testing

Answer : Option d)

Test Process



1. Test Process in Context

- Contextual factors that influence the test process for an organization, include, but are not limited to:
 - Software development lifecycle model and project methodologies being used
 - Test levels and test types being considered
 - Product and project risks
 - Business domain
- Operational constraints, including but not limited to:
 - Budgets and resources
 - Timescales
 - Complexity
 - Contractual and regulatory requirements
 - Organizational policies and practices
 - Required internal and external standards
- It is very useful if the test basis (for any level or type of testing that is being considered) has measurable coverage criteria defined.

1. Test Process in Context

- The coverage criteria can act effectively as **key performance indicators (KPIs)** to drive the activities that demonstrate achievement of software test objectives.
- The following sections describe general aspects of organizational test processes in terms of the following:
 - Test activities and tasks
 - Test work products
 - Traceability between the test basis and test work products



Plan ahead



Adjust the plan



Ask questions

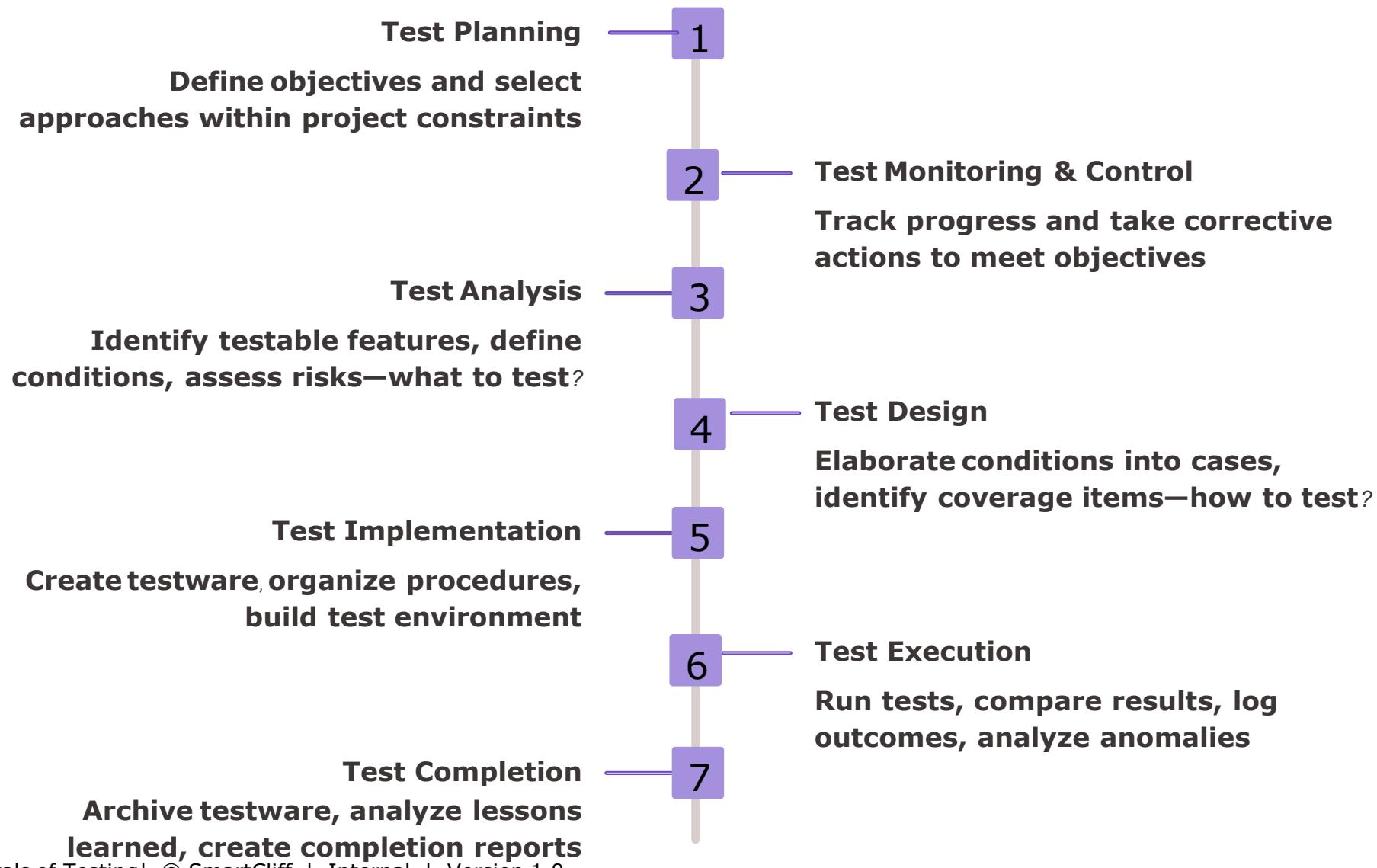


Stakeholders decide project's completion



Avoid applying any practice blindly

2. Test Activities and Tasks



1. Test Planning:

- Test planning involves **producing a document that describes an overall approach and test objectives.**
- It involves reviewing the test basis, identifying the test conditions based on analysis of test items, writing test cases and Designing the test environment.
- **Completion or exit criteria must be specified so that we know when testing (at any stage) is complete.**

Purpose:

- To determine the **scope and risks** and identify the objectives of testing.
- To determine the **required test resources** like people, test environments etc.
- To schedule **test analysis and design tasks, test implementation, execution and evaluation.**

2. Test Monitoring and Control

Test Monitoring:

- Test Monitoring is the process of **evaluating and providing feedback** on the test proceedings that are currently in progress.
- Monitoring works by comparing the **current status of testing-related** tasks against a **previously established plan and evaluating** whether the process is yielding **the expected results**.

Test Control:

- Test Control occurs based on the results of Test Monitoring.
- It refers to taking **corrective action based on test monitoring** reports to **improve quality and efficiency**.
- This is the 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.

3. Analysis and 4. Design:

Test analysis and Test Design has the following major tasks:

- Reviewing the **test basis (such as requirements, software integrity level1(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 any required infrastructure and tools.
- Creating **bi-directional traceability** between test basis and test cases.

5. Implementation and 6. Execution:

- **Test execution** involves running the specified test on a computer system **either manually or by using an automated test tool**. It is a Fundamental Test Process in which actual work is done.
- **Test implementation** has the following major task:
 - To develop and **prioritize test cases** by using techniques and create test data for those tests.
 - To **create test suites** from the test cases for **efficient test execution**.
 - **Test suite is a collection** of test cases that are used to **test a software program**
 - To re-execute the tests that previously failed in order to confirm a fix.
 - To log the outcome of the test execution. A test log is the status of the test case (pass/fail).
 - To compare actual results with expected results.

Evaluating exit criteria and Reporting:

- Evaluating exit criteria is a **process defining when to stop testing**.
- It depends on coverage of code, functionality or risk.
- Basically, it also depends on business risk, cost and time and vary from project to project.

Exit criteria come into picture, when:

➤ **Maximum test cases are executed with certain pass percentage**

➤ **Bug rate falls below certain level**

➤ **When we achieve the deadlines**

- Evaluating exit criteria has the following major tasks:

➤ To assess if more test are needed or if the exit criteria specified should be changed.

➤ To write a test summary report for stakeholders.

7. Test Closure activities:

- As the name suggest, test closure is the last stage of **any testing process and is generally initiated after the delivery of the software product.**
- However, this stage of testing can also be initiated because of various other reasons like the termination of the testing process due to **non-achievement of target, cancellation of the project, when product needs updates, among other things.** This stage of STLC mainly consists of following events:
 - Checking of **actual delivered product with respect to planned deliverables.**
 - Gathering "testware" such as test scripts, test environment, etc. for its reuse and to pass it to maintenance department, that may help them in providing software support in future.
 - Gaining knowledge from the **current testing process**, in order to improve the working, for the **future projects.**

Test Work Products:

- Test work products are created as part of the **test process**.
- Many of the test work products are captured and managed using **test management tools** and **defect management tools**.
- Based on the **test process adopted**, there will be significant **variation in test products**.
 - Test plan (Test planning).
 - Test Progress Report, Test Summary reports (Test Monitoring and Control)
 - Test charters and Test Conditions (Test Analysis)
 - Test cases (Test Design)
 - Test procedure, Test Suite and Test Execution Schedule (Test Implementation)
 - Test case / Test Procedure Status, Defect report (Test Execution)
 - Test Summary Report (Test Completion)

Traceability between the Test Basis and Test Work Products:

- In order to implement effective test monitoring and control, it is important to establish and maintain traceability throughout the test process between each element of the test basis and the various test work products associated with that element.
- Good traceability supports:
 - Analyzing the impact of changes
 - Making testing auditable
 - Improving the understandability of test progress reports and test summary reports to include the status of elements of the test basis
 - Providing information to assess product quality, process capability, and project progress against business goals

Test Work Products:

- Test work products are created as part of the **test process**.
- Many of the test work products are captured and managed using **test management tools** and **defect management tools**.
- Based on the **test process adopted**, there will be significant **variation in test products**.
 - Test plan (Test planning).
 - Test Progress Report, Test Summary reports (Test Monitoring and Control)
 - Test charters and Test Conditions (Test Analysis)
 - Test cases (Test Design)
 - Test procedure, Test Suite and Test Execution Schedule (Test Implementation)
 - Test case / Test Procedure Status, Defect report (Test Execution)
 - Test Summary Report (Test Completion)

Traceability between the Test Basis and Test Work Products:

- In order to implement effective test monitoring and control, it is important to establish and maintain traceability throughout the test process between each element of the test basis and the various test work products associated with that element.
- **Good traceability supports:**
 - Analyzing the impact of changes
 - Making testing auditable
 - Improving the understandability of test progress reports and test summary reports to include the status of elements of the test basis
 - Providing information to assess product quality, process capability, and project progress against business goals

Quiz



1. Which option is part of the implementation and execution area of the fundamental test process?

a) Developing the tests

b) Comparing actual and expected results

c) Writing a test summary.

d) Analyzing lessons learnt for future releases.

Answer : Option b)

(expleo)

Quiz



2. The five parts of the fundamental test process have a broad chronological order. Which of the options gives three different parts in the correct order?

- a) Implementation and execution, planning and control, analysis and design
- b) Analysis and design, evaluating exit criteria and reporting, test closure activities
- c) Evaluating exit criteria and reporting, implementation and execution, analysis and design.
- d) Evaluating exit criteria and reporting, test closure activities, analysis and design.

Answer : Option b)

(expleo)

Quiz



3. Which of the following activities is part of the main activity "test analysis" in the test process?

- a) Identifying any required infrastructure and tools**
- b) Creating test suites from test scripts**
- c) Analyzing lessons learned for process improvement**
- d) Evaluating the test basis for testability**

Answer : Option d)

(expleo)

Verification and Validation



Verification

- **Verification** is the process of **ensuring that work products like requirements, design, and code** are developed according to the **defined specifications and standards**.
- It answers the question: "**Did we build the product correctly?**"
- "Verification is preventive – done early to catch issues before they become expensive."
- No code execution is involved – it's a **static testing technique**.

Activities in Verification:

- Requirements review
- Design inspection
- Code walkthroughs
- Documentation checks
- Peer reviews

Validation

- **Validation** ensures that the **final product meets the user's needs** and works as intended in the real world.
- It answers: "**Did we build the right product?**"
- Validation is detective – it happens when the product is ready or running.
- Requires actual execution of code – **it's a dynamic testing technique.**

Activities in Validation:

- System Testing
- Integration Testing
- User Acceptance Testing (UAT)
- Beta Testing, Functional/Non-functional testing

Verification and Validation

Scenario and Example : Online Banking System

Phase	Verification Example	Validation Example
Requirement Review	Verifying that all requirements are clearly written and testable	Checking if users can log in, view balance, transfer money
Design Review	Ensuring login module uses proper encryption as per design	Testing if login works under different networks/devices

Verification and Validation

Comparison Table

Aspect	Verification	Validation
Definition	Are we building the product right?	Are we building the right product?
Purpose	Ensure the product meets specified requirements/design	Ensure the product meets user needs and expectations
Focus	Process-oriented	Product-oriented
Activities	Reviews, walkthroughs, inspections, static analysis	Testing (system, acceptance), UAT, dynamic execution
Performed During	Early stages: Requirements, Design, Coding	Later stages: Testing, Deployment
Executed By	Developers, Business Analysts, QA	Testers, End users, QA
Input	Documents, plans, architecture, design	Working product/software
Output	Verified documents, specifications	Validated software behavior

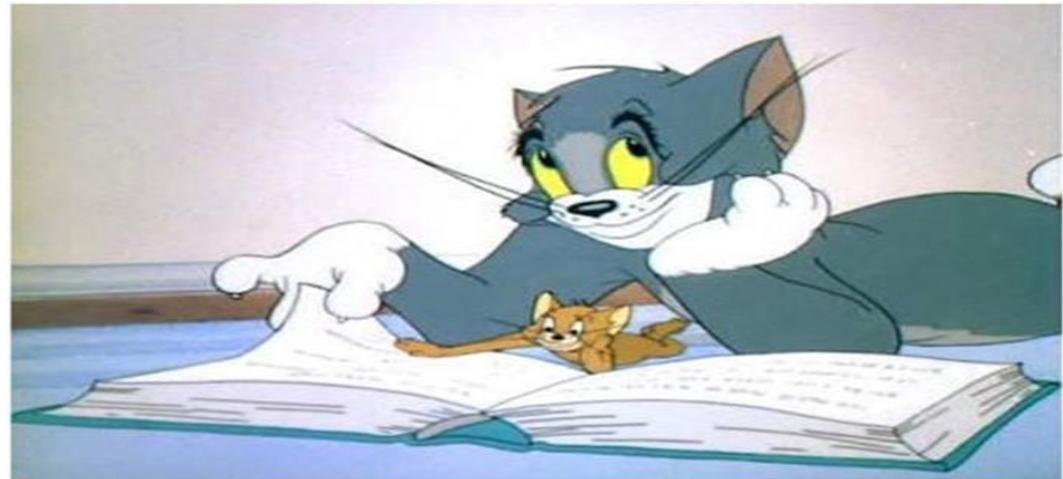
The Psychology of Testing



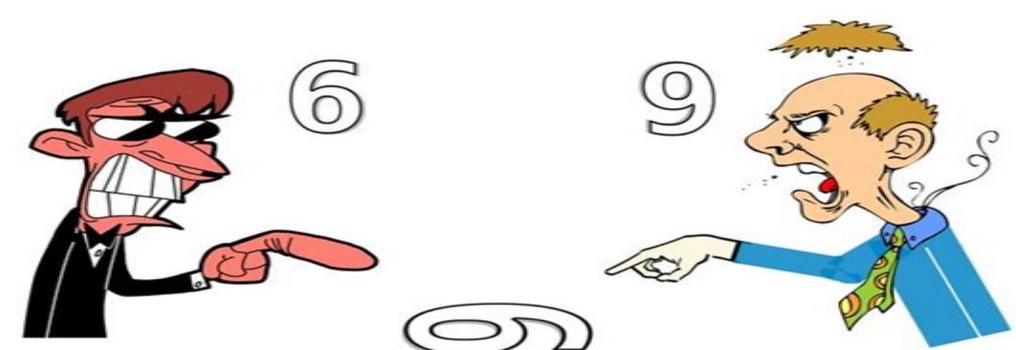
The Psychology of Testing

The Psychology of Testing

- Software development, including software testing, involves human beings. Therefore, human psychology has important effects on software testing.
- The three sections of the psychology of testing are:
 - The mindset of Developers and Testers.
 - Communication in a Constructive Manner.
 - Test Independence.



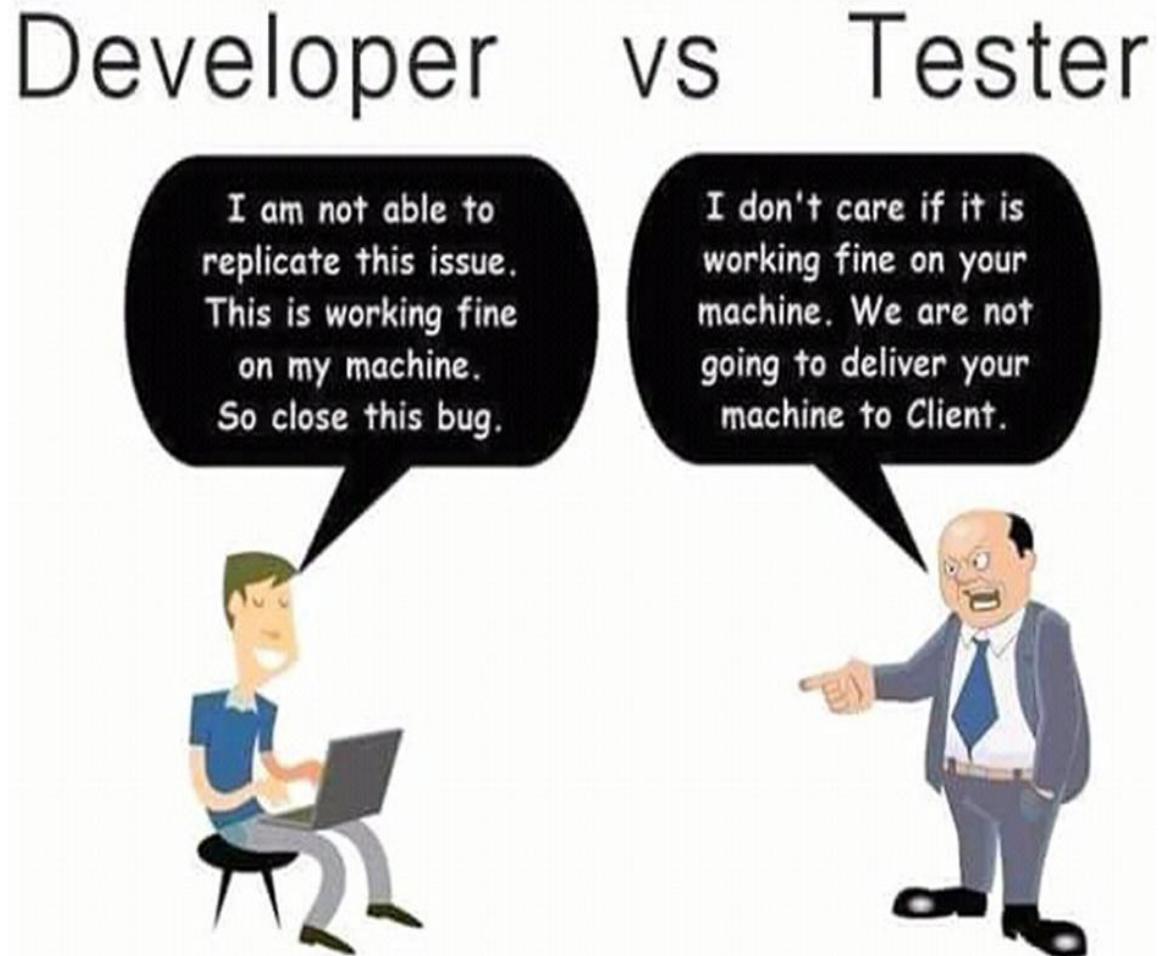
Developer vs Tester



(expleo)

The Psychology of Testing

- The aim of testers and developers is the same i.e., to provide a quality product to the stakeholders. But their way of thinking is different.
- **“Testers and Developers are not different, but they follow a different path to achieve the same goal”.**
- **Developers think :** ‘How can I make the application?’
- **Testers think :** ‘How can I break the application?’



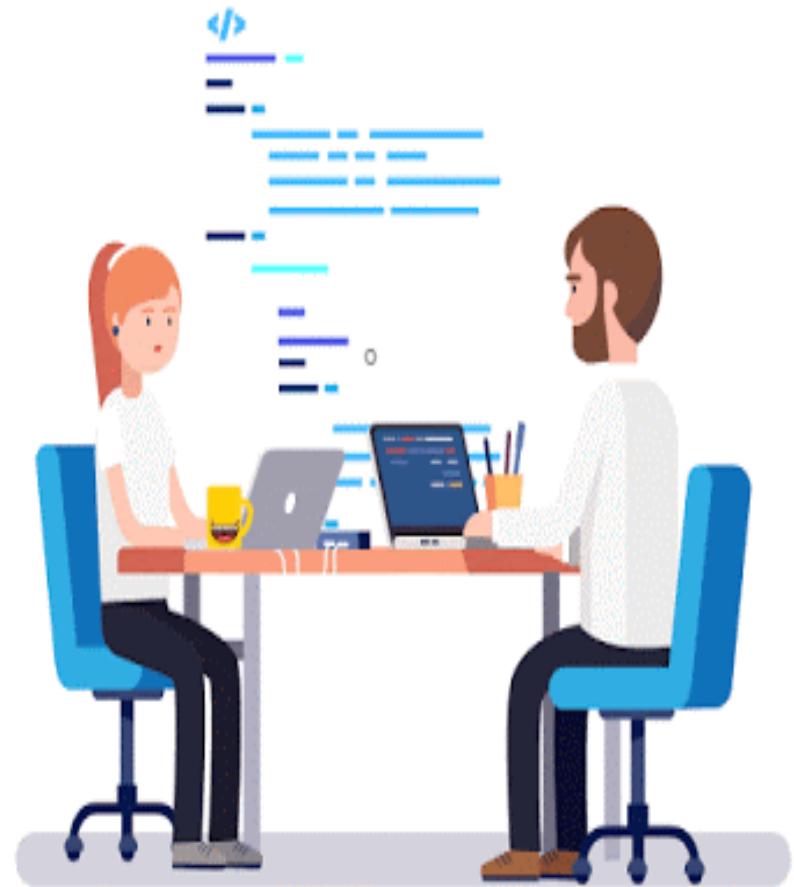
Communication in a Constructive Manner:

- Testers and test managers need to have **good interpersonal skills** to be able to **communicate effectively about the defects, failures, test results, test progress and risks and to build positive relationships among colleagues.**
- As testers are responsible for finding bugs and reporting them, it becomes vitally important for them to **communicate it in a respectful, polite, and suitable way to avoid hurting and disappointing someone.**



Self-testing and Independent testing:

- A **certain degree of independence (avoiding the author bias) is often more effective at finding defects and failures.**
- Tests designed by the person who wrote the software under test (low level of independence).
- Tests designed by another person (e.g., from the development team).
- Tests designed by a person from a different organizational group (e.g., an independent test team) or test specialists (e.g., usability or performance test specialists).
- Tests designed by a person from a different organization or company (i.e., outsourcing or certification by an external body).



(expleo)

The Psychology of Testing

Quiz



1. Which of the following qualities is MORE likely to be found in a tester's mindset rather than in a developers?

- a) A tester's mindset tends to grow and mature as the tester gains experience**
- b) Ability to see what might go wrong**
- c) Good communication with team members**
- d) Attention to detail**

Answer : Option b)

(expleo)

The Psychology of Testing

Quiz



2.Which of the following has highest level of independence in which test cases are :

- a) Designed by persons who write the software under test**
- b) Designed by a person from a different section**
- c) Designed by a person from a different organization**
- d) Designed by another person**

Answer : Option c)

The Psychology of Testing

Quiz



3. Benefits of Independent Testing

- a) Independent testers are much more qualified than Developers
- b) Independent testers see other and different defects and are unbiased
- c) Independent Testers cannot identify defects
- d) Independent Testers can test better than developers

Answer : Option b)

(expleo)