



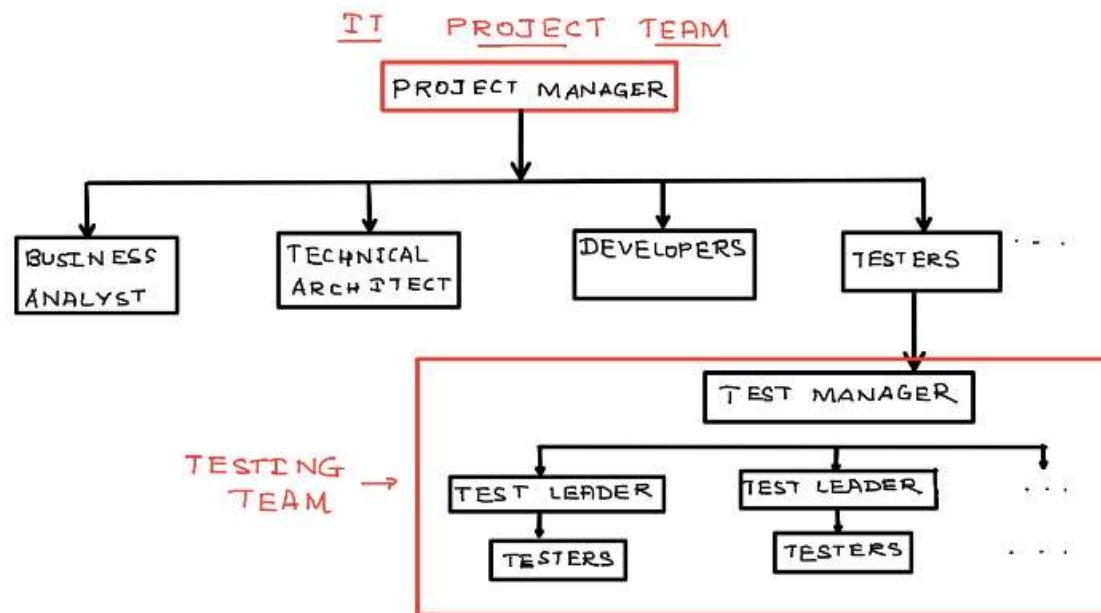
EXAMPLES

- <https://www.cigniti.com/blog/37-software-failures-inadequate-software-testing/>



SOFTWARE TESTING TEAM:

The software testing team comprises of test manager, test leader and tester.



software testing team



ISTQB CH 1

- International Software Testing Qualification Board



TESTING

- Do you test
- Why you test
- What you test
- How you test
- When do you stop testing



SOFTWARE FAILURES

- <https://www.cigniti.com/blog/37-software-failures-inadequate-software-testing/> -need of testing
- Above link listing failures with sources
- <https://www.bugraptors.com/blog/top-software-failures-due-to-lack-of-testing>
- <https://twitter.com/rajaharia/status/1345659284160356352>



- Every Software Failure occurs when it doesn't meet desired requirements
- Requirements can be in terms of any Functionality'
- Calculation , Calibration, Security, Resource Management,
- Example



WHY TO TEST SOFTWARE ?

- Testing is necessary because we all make mistakes.
- Some of those mistakes are unimportant, but some of them are expensive or dangerous.
- We need to check everything that we produce because things can go wrong
- To Err is Human !
- Most important - We should produce QUALITY product !
- What is QUALITY ?



WHAT IS QUALITY ?

- Quality : The degree to which a component, system or process meets specified requirements and/or user/customer **needs and expectations**.
- Quality is Consistently meeting the
 - Requirements
 - Cost
 - Delivery Schedule
 - Service commitments



Quality is not meeting customer expectations
but
beating customer expectations !

Who is interested in Quality of product ?



We should develop Quality Software !

What is the definition of Software ?



DEFINITION OF SOFTWARE

- **Software :**

- Computer programs,
- procedures, associated with
- documentation and
- data pertaining to the operation of a computer system. [IEEE 610]



WHAT IS TESTING ?

○ 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 the software satisfies specified requirements,
 - To demonstrate that it is fit for purpose
 - To detect defects.



TESTING...

- Software testing, when done correctly, can increase overall software quality of conformance by testing that the product conforms to its requirements.
- Testing begins with the software engineer in early stages, but later specialists may be involved in the testing process.
- After generating source code, the software must be tested to uncover and correct as many errors as possible before delivering it to the customer.
- It's important to test software because a customer, after running it many times would finally uncover an error.



SOFTWARE TESTING FUNDAMENTALS

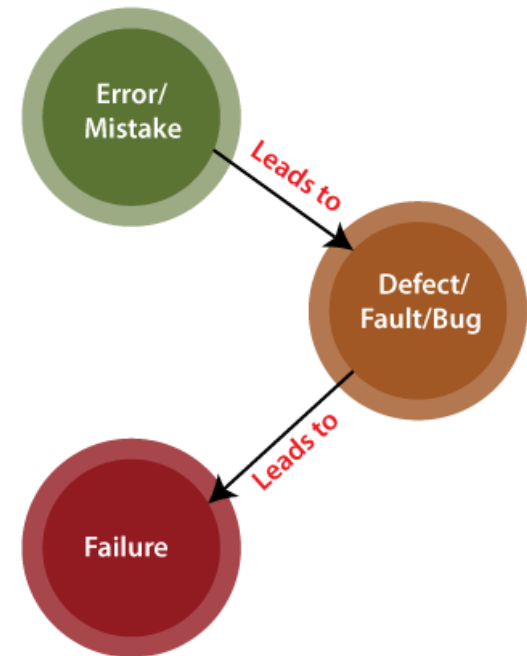
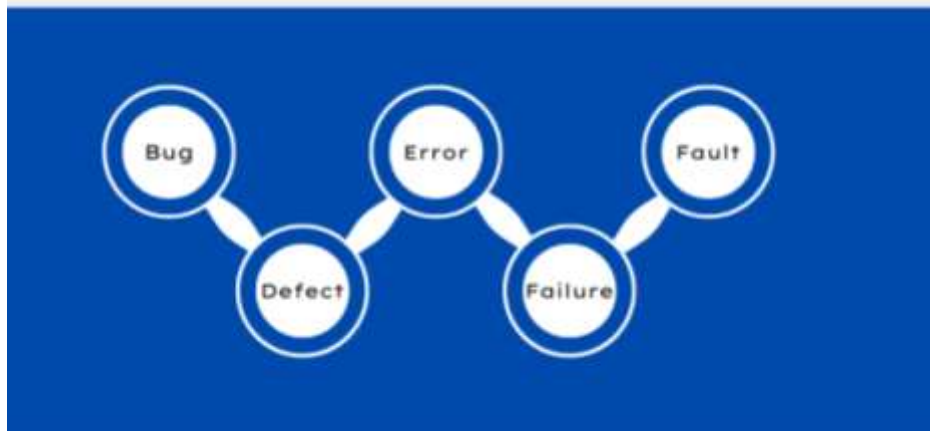
- During software engineering activities, an engineer attempts to build software, where as during testing activities, the intent is to demolish the software.
- Hence testing is psychologically destructive rather constructive.
- A good and successful test is one that uncovers an undiscovered error.
- All the test that are carried out should be traceable to customer requirements at the sometime exhaustive testing is not possible.



WHAT IS THE OBJECTIVE OF TESTING ?

- To check whether the **application works as per requirement specifications.**
- To identify as many defects as possible before the software is delivered
- To provide input for process improvement
- To ensure that best **quality product** is delivered to customer





- Software testing is a process to spot bugs, errors, defects, faults, and failures which are the variance between expected and actual results.
- Whether you test your software manually or with automated procedures, these terms surface when identifying the issues in your coding.
- And by identifying deficiencies, missing requirements, or errors in the software, you are making your software flawless and of high quality for the users.
- <https://geekflare.com/difference-bug-defect-error-failure-fault-testing/>



COMMON TERMS:

- **Error:-** difference between the actual output of a software and the correct output.
 - A mistake made by programmer. OR Anything that stops the execution of application.
- **Fault:** is a condition that cause a system to fail in performing its required function.
- **Bug:** A Bug is a Application Error, in Software doesn't affect the whole application in terms of running. It doesn't stop the execution of application. It produces wrong output.
- **Failure:** occurs when a fault is executed. The inability of a system or component to perform its required functions within specified performance requirements.



ERROR, DEFECT AND FAILURE

- **Error** : A human action that produces an incorrect result. [After IEEE 610]
- **Defect (Bug, Fault)** : A flaw in a component or system that can cause the component or system to fail to perform its required function, e.g. an incorrect statement or data definition.
A defect, if encountered during execution, may cause a failure of the component or system.
- **Failure** : Deviation of the component or system from its expected delivery, service or result

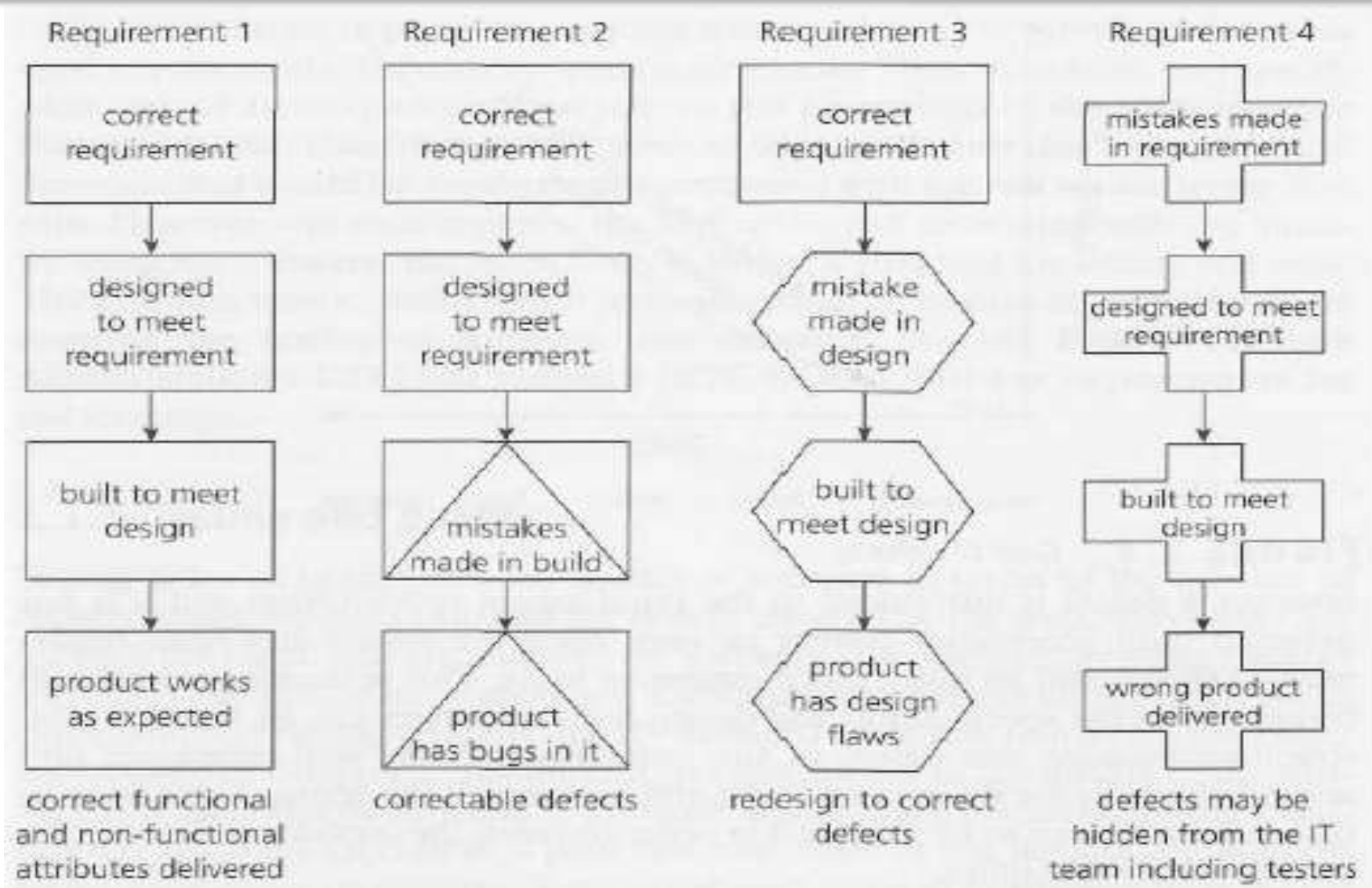


WHAT IS THE CAUSE OF DEFECT ?

- Errors in the specification
- Errors in design
- Errors in implementation of the software and system
- Complex domain and technology
- Errors in use of the system
- Environmental conditions
- Intentional damage
- Potential consequences of earlier errors, intentional damage, defects and failures.

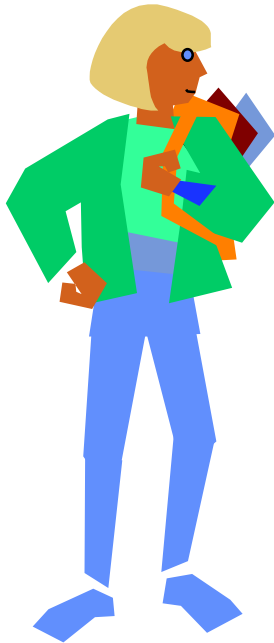


WHEN DEFECTS ARE INTRODUCED ?

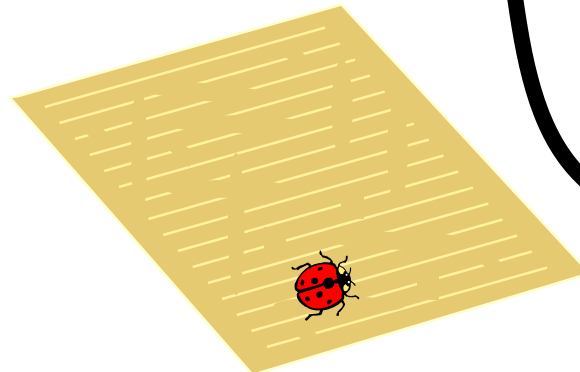


ERROR - FAULT - FAILURE

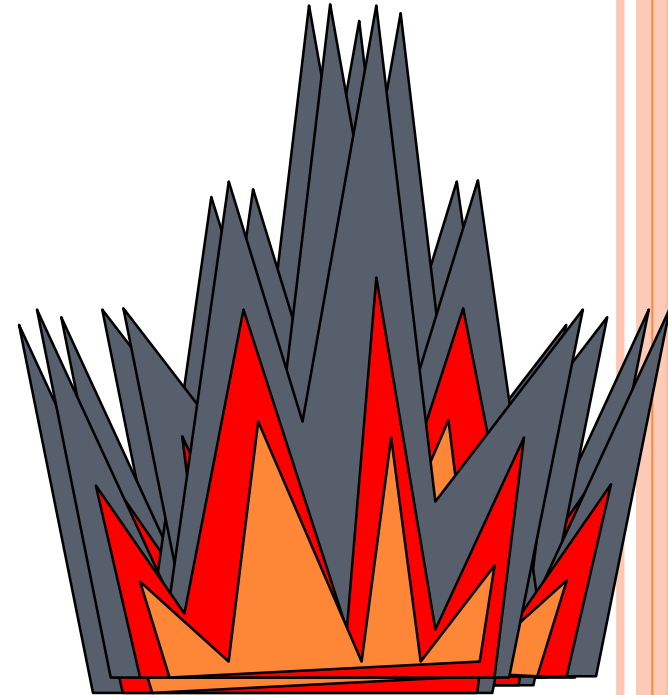
A person makes
an error ...



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



... that can cause
a failure
in operation




Error Prone Construct

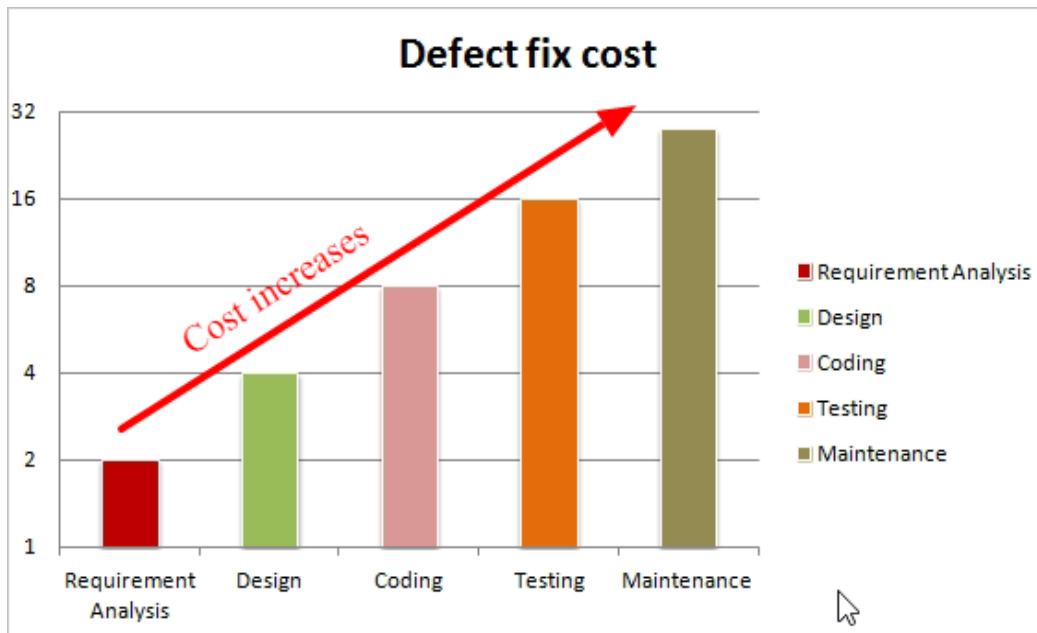
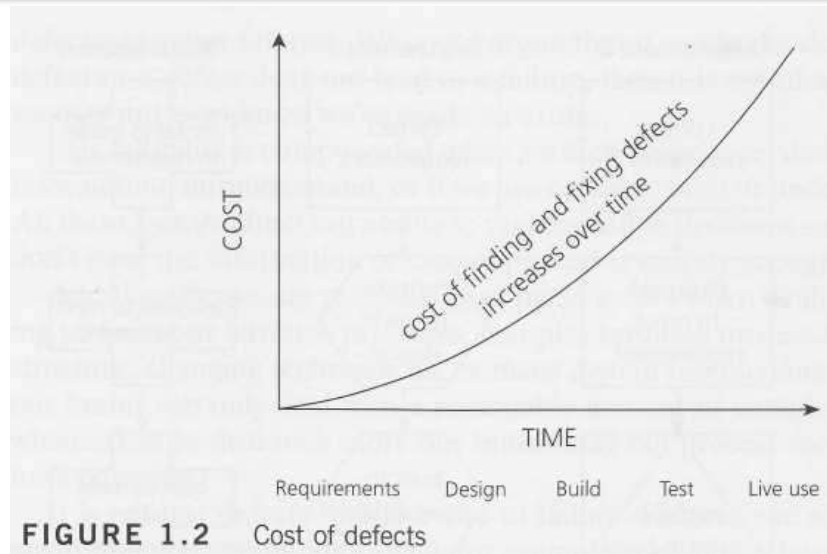
- *Unconditional branch (goto) statements*
- *Floating-point numbers*
- *Pointers*
- *Dynamic memory allocation*
- *Parallelism*
- *Recursion*
- *Interrupts*
- *Inheritance*
- *Aliasing*
- *Unbounded arrays*



WHAT IS THE COST OF DEFECTS ?

- *A stitch in time saves nine !*
 - If an error is made at requirement stage and the defect is detected in the requirement stage itself, then it is relatively cheap to find and fix.
 - The specifications can be corrected and re-issued.
 - If an error is made at requirement stage and the consequent defect is detected at the design stage then both requirements and design should be corrected and re-issued.
 - This is more expensive.
 - The cost of defect increases if it is not detected early.
- 

COST OF DEFECTS



EXHAUSTIVE TESTING

- A test approach in which the test suite comprises of all combinations of input values and preconditions.
- Exhaustive testing requires
 - Infinite time
 - Infinite resources



HOW MUCH TESTING IS ENOUGH ?

- It's never enough
- When you have done what you planned
- When your customer/user is happy and confident to use software
- When you have proved that the system works correctly
- It depends on the **risks** for your system



HOW MUCH TESTING?

- It depends on **RISK**
- What is RISK ?
 - A factor that could result in future negative consequences
 - Usually expressed as **impact and likelihood**.
- There can be a risk of
 - missing important functions
 - incurring failure costs ... losing business
 - releasing untested or under-tested software
 - losing credibility, reputation and market

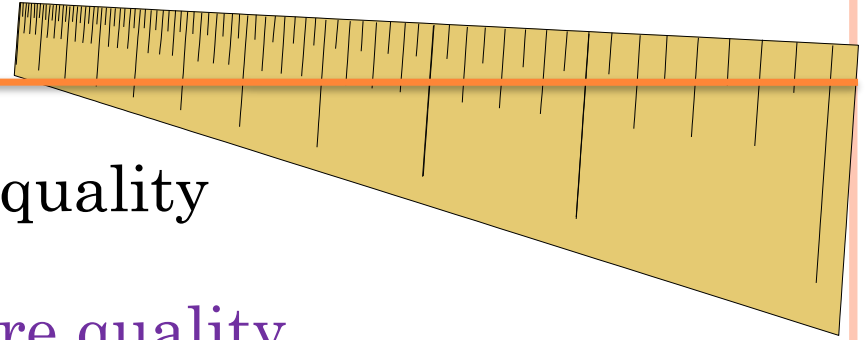


MOST IMPORTANT PRINCIPLE

**Prioritise tests
so that,
whenever you stop testing,
you have done the best testing
in the time available.**



TESTING AND QUALITY



- Testing measures software quality
- Testing can improve software quality
- What does testing include ?
 - system function, correctness of operation
 - non-functional qualities:
 - reliability,
 - usability,
 - maintainability,
 - reusability,
 - security, etc.



TESTING PRINCIPLES

- “All tests should be traceable to customer requirements.” For a customer, the most severe defect are the one that causes the program to fail to meet the requirements.
- “Tests should be planned long before testing begins.” All test can be planned and designed before any code has been generated
- “Testing should begin ‘in the small’ and progress toward testing in the large.” The first test planned and executed generally focus on individual component.



TESTING PRINCIPLES

- “As testing progresses focus shifts on integrated clusters of components and then the entire system.”
- “Exhaustive testing is not possible.” to test exhaustively, even a small program could take years.
- “To be more effective, testing should be conducted by an independent third party.”
 - Software engineer who constricts the software is not always the best person to test it. So a third party deems to be more efficient.



PRINCIPLES OF TESTING

- Principle 1 – Testing shows presence of defects
- Principle 2 – Exhaustive testing is impossible
- Principle 3 – Early testing
- Principle 4 – Defect clustering
- Principle 5 – Pesticide paradox
- Principle 6 – Testing is context dependent
- Principle 7 – Absence-of-errors fallacy
- <https://www.softwaretestinghelp.com/7-principles-of-software-testing/>
- <https://www.practitest.com/blog/the-pesticide-paradox/>



GENERAL TESTING PRINCIPLES

- A number of testing principles have been suggested over the past 40 years and offer general guidelines common for all testing.
- **Principle 1 – Testing shows presence of defects**
 - 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, it is not a proof of correctness.



GENERAL TESTING PRINCIPLES

- **Principle 2 – Exhaustive testing is impossible**
 - Testing everything (all combinations of inputs and preconditions) is **not feasible except for trivial cases(unimpt)**. Instead of exhaustive testing, *risk analysis and priorities should be used to focus testing efforts.*
- **Principle 3 – Early testing**
 - Testing activities should **start as early as possible** in the software or system development life cycle, and should be *focused on defined objectives.*




GENERAL TESTING PRINCIPLES

○ Principle 4 – Defect clustering

- A small **number of modules contain most of the defects** discovered during pre-release testing, or are responsible for the most operational failures.

○ Principle 5 – Pesticide paradox

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

GENERAL TESTING PRINCIPLES

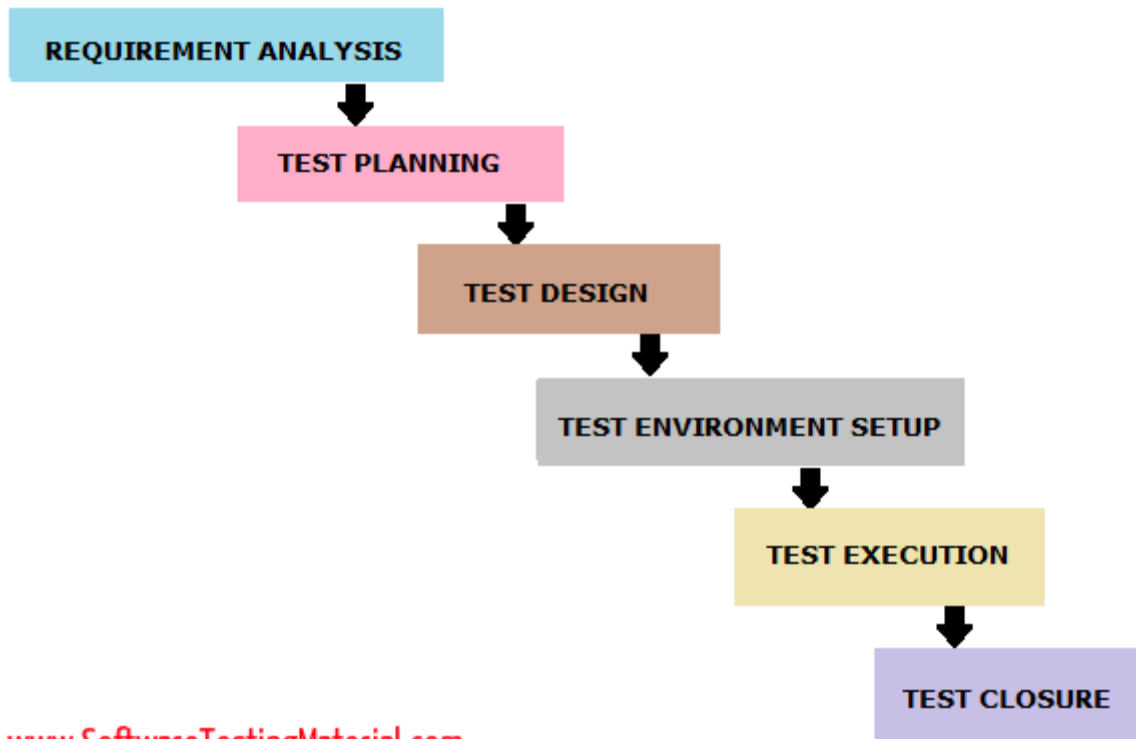
- **Principle 6 – Testing is context dependent**
 - Testing is done **differently in different contexts**.
 - For example, safety-critical software is tested differently from an e-commerce site.
- **Principle 7 – Absence-of-errors fallacy**
 - Finding and fixing defects does not help if the system built is unusable and does not fulfill the users' needs and expectations.



FUNDAMENTAL TEST PROCESS (TEST LIFE CYCLE)

- Test Planning and Control
- Test Analysis and Design
- Test Implementation
- Test Execution
- Defect Reporting
- Test Closure
- <https://www.lambdatest.com/blog/software-testing-life-cycle/>





www.SoftwareTestingMaterial.com



STLC Phase	Entry Criteria	Exit Criteria	Deliverables
Requirement Analysis	Requirements specification document, Acceptance criteria document, Application architectural document	Signed off RTM, Signed off Automation feasibility report	List of all testable requirements, Automation feasibility report (if applicable)
Test Planning	Requirements Documents, Automation feasibility report	Approved Test plan document, Approved Test strategy document, Signed off effort estimation document	Test Strategy, Test Plan, and Test Effort estimation document.
Test Design	Requirements Documents (Updated version of unclear or missing requirement), RTM, Test Plan, Test Estimation Document, Automation Analysis Report	Reviewed and approved test cases, test scripts, test data	Test cases, Test Scripts (if automation), Test data.
Test Environment Setup	Test Plan, Test environment setup plan, Smoke Test cases, Test Data	Working test environment setup, Valid test data setup, Successful smoke test	Test Environment. Smoke Test Results.
Test Execution	Test Plan document, Test cases, Test Scripts, Test data, Test Environment.	Execute all planned test cases, Log all defects found	Test case execution report, Defect reports, RTM
Test Closure	Testing has been completed, Test Case Execution report (make sure there are no high severity defects opened)	Signed off Test Closure report	Test Closure report, Test metrics



- **Requirement Analysis**
- The entry criteria for this phase is the BRS (Business Requirement Specification) document. During this phase, the test team studies and analyzes the requirements from a testing perspective.
- This phase helps to identify whether the requirements are testable or not. If any requirement is not testable, the test team can communicate with various stakeholders (Client, Business Analyst, Technical Leads, System Architects, etc) during this phase so that the mitigation strategy can be planned.
- **Entry Criteria:** BRS (Business Requirement Specification)
- **Deliverables:** List of all testable requirements, Automation feasibility report (if applicable)



- **2. Test Planning:**
- Test planning is the first step in the testing process.
- In this phase typically Test Manager/Test Lead involves determining the effort and cost estimates for the entire project. Preparation of the Test Plan will be done based on the requirement analysis.
- Activities like resource planning, determining roles and responsibilities, tool selection (if automation), training requirements, etc., carried out in this phase.
- The deliverables of this phase are Test Plan & Effort estimation documents.
- **Entry Criteria:** Requirements Documents
- **Deliverables:** Test Strategy, Test Plan, and Test Effort estimation document.
-



- **3. Test Design:**
- The test team starts with test case development activity here in this phase. Testers prepares test cases, test scripts (if automation), and test data.
- Once the test cases are ready then these test cases are reviewed by peer members or team lead.
- Also, the test team prepares the Requirement Traceability Matrix (RTM). RTM traces the requirements to the test cases that are needed to verify whether the requirements are fulfilled. The deliverables of this phase are Test Cases, Test Scripts, Test Data, Requirements Traceability Matrix
- **Entry Criteria:** Requirements Documents (Updated version of unclear or missing requirement)
- **Deliverables:** Test cases, Test Scripts (if automation), Test data.



- **4. Test Environment Setup:**
- This phase can be started in parallel with the Test design phase.
- The test environment setup is done based on the hardware and software requirement list. In some cases, the test team may not be involved in this phase. The development team or customer provides the test environment.
- Meanwhile, the test team should prepare the smoke test cases to check the readiness of the given test environment.
- **Entry Criteria:** Test Plan, Smoke Test cases, Test Data
- **Deliverables:** Test Environment. Smoke Test Results.



- **5. Test Execution:**
- The test team starts executing the test cases based on the planned test cases. If a test case result is Pass/Fail then the same should be updated in the test cases.
- The defect report should be prepared for failed test cases and should be reported to the Development Team through a [bug tracking tool](#) for fixing the defects.
- [Retesting](#) will be performed once the defect was fixed. Click here to see the [Bug Life Cycle](#).
- **Entry Criteria:** Test Plan document, Test cases, Test data, Test Environment.
- **Deliverables:** Test case execution report, Defect report, [RTM](#)



- **6. Test Closure:**
- The final stage where we prepare Test Closure Report, Test Metrics.
- The testing team will be called out for a meeting to evaluate cycle completion criteria based on Test coverage, Quality, Time, Cost, Software, Business objectives.
- The test team analyses the test artifacts (such as Test cases, Defect reports, etc.,) to identify strategies that have to be implemented in the future, which will help to remove process bottlenecks in the upcoming projects.
- Test metrics and Test closure report will be prepared based on the above criteria.
- **Entry Criteria:** Test Case Execution report (make sure there are no high severity defects opened), Defect report
- **Deliverables:** Test Closure report, Test metrics



- **Functional Testing**

- Functional testing is performed to test the functional aspects of a software application. Every function should be tested to verify if the functional requirements are met. There are various types of functional testing.
- Unit testing
- Integration testing
- Regression testing
- Sanity Test
- Smoke test
- Interface testing
- System test
- Acceptance testing/Beta testing



- **Non-Functional Testing**

- Non-functional testing is done to test the performance, reliability, security, and usability of the application. This testing helps to improve the quality of the application. Non-functional testing is performed using certain tools and not done manually.
- Performance Testing
- Load Testing
- Stress Testing
- Volume Testing
- Security Testing
- Compatibility Testing
- Install Testing
- Recovery Testing
- Reliability Testing
- Usability Testing
- Compliance Testing
- Localization Testing
- Alpha testing



WHO SHOULD PARTICIPATE IN TESTING ?

Developer → Unit Testing

Developer / Tester → Integration Testing

Tester → System Testing

Customer → User Acceptance Testing

- ✓ End User
- ✓ Management
- ✓ Auditors
- ✓ Consultants



HOW LONG ONE SHOULD CONTINUE TESTING ?

- There is **no single criteria**
- It depends on many factors
 - Till all the test cases are executed
 - Till **all the requirements** (specified) are working properly
 - Till **budget / time** is over
 - Till **customer** is confident in using the system
- How long one should continue testing depends on the risk factor of application.



QUALITY ASSURANCE Vs. QUALITY CONTROL

Quality Assurance

1. **Process Driven**
Approach : It is a process to monitor and improve existing quality processes.
2. Quality Assurance ensures that the processes designed for the product development and services are **effective** enough to meet the objectives.
3. Quality Assurance focuses on **defect prevention**.

Quality Control

1. **Concerned with Product:** It measures and controls the quality of the software as it is being developed.
2. Quality Control ensures that the final product is **error free and satisfactory**.
3. Quality Control finds product defects.





QA	QC
Strategic Issue	Operational Issue
Prevention(Proactive)	Correction(Reactive)
Prevent Defects	Find Defects



- 1. Proactive (QA) vs. Reactive (QC)
- 2. Process (QA) vs. Product (QC)
- 3. System (QA) vs. Parts (QC)
- 4. Creation (QA) vs. Verification (QC)
- 5. Entire Team (QA) vs. Dedicated Personnel (QC)
- <https://www.qualio.com/blog/quality-assurance-vs-quality-control>
- <https://www.softwaretestinghelp.com/quality-assurance-vs-quality-control/>



	QA	QC	Testing
Purpose	Setting up adequate processes, introducing the standards of quality to prevent the errors and flaws in the product	Making sure that the product corresponds to the requirements and specs before it is released	Detecting and solving software errors and flaws
Focus	Processes	Product as a whole	Source code and design
What	Prevention	Verification	Detection
Who	The team including the stakeholders	The team	Test Engineers, Developers
When	Throughout the process	Before the release	At the testing stage or along with the development process

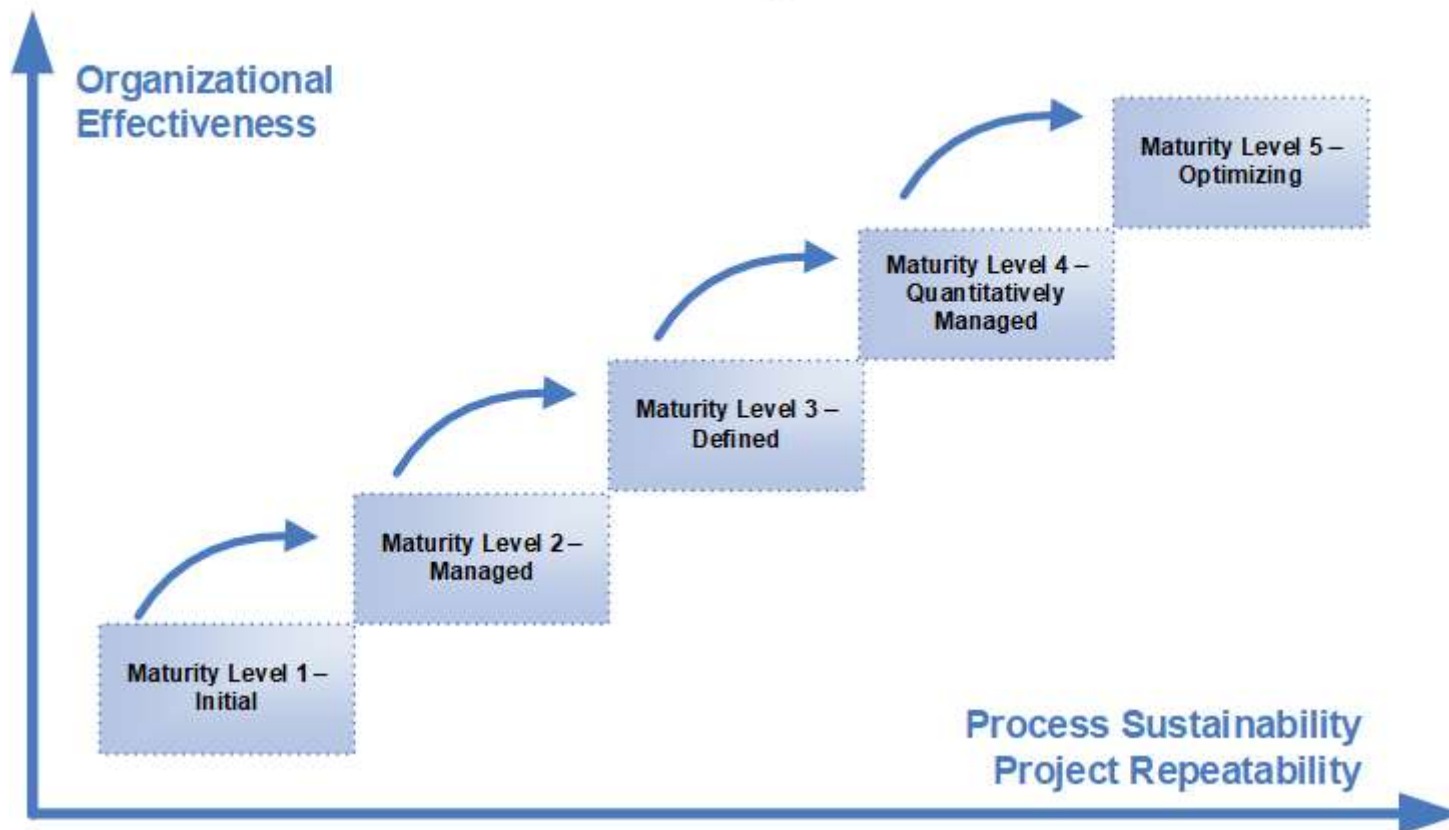


AUDITS

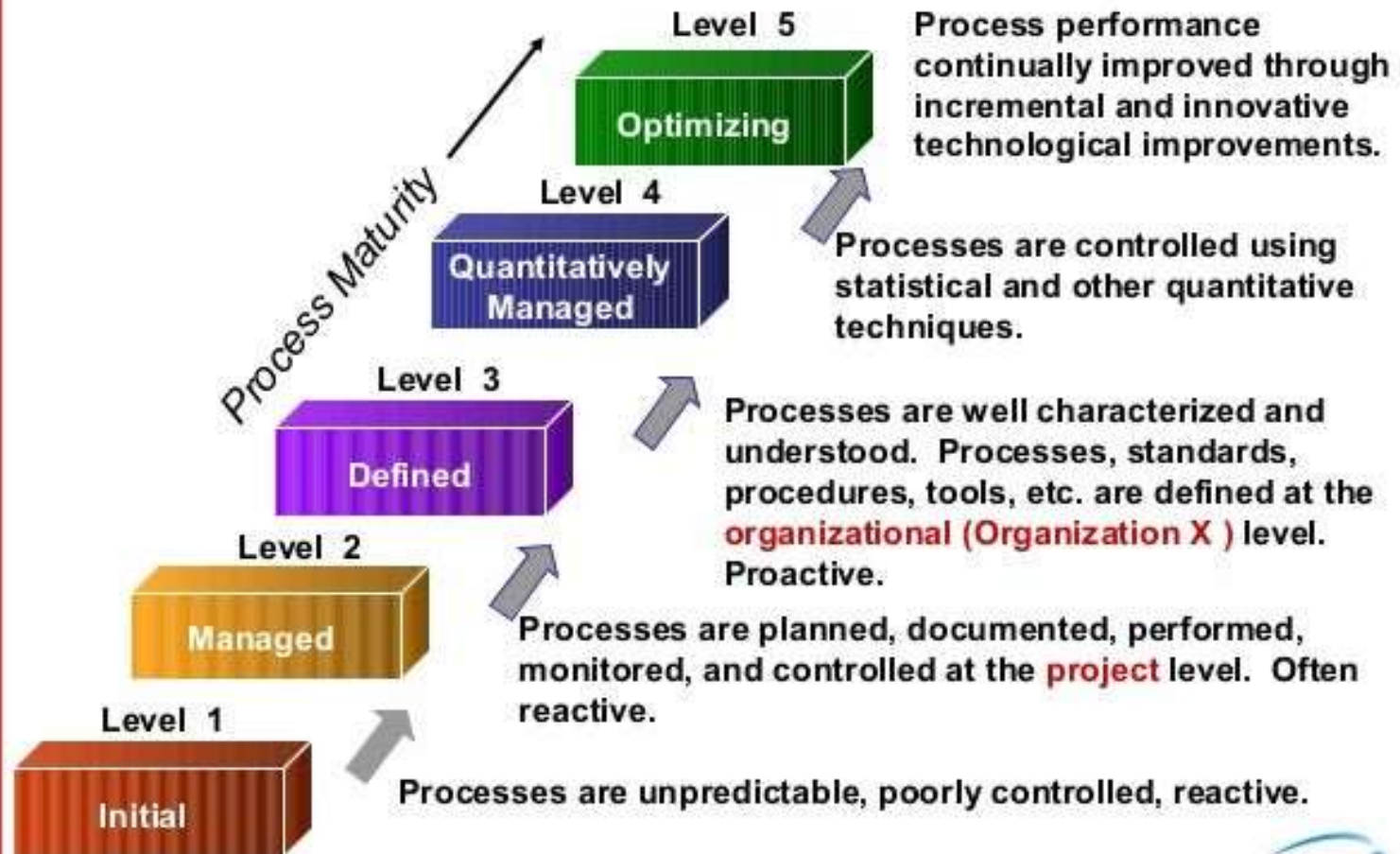
- **1. ISO 9000 Quality Standards :**
- <https://www.geeksforgeeks.org/iso-standards-in-software-engineering>
- **2. CMMI(Capability Maturity Model Integration)-LEVEL-5**
- <https://www.99corporates.com/Company-List/CERTIFICATI>
- <https://www.bmc.com/blogs/cmmi-capability-maturity-model-integration/>



CMMI Maturity Levels



CMMI Staged Representation - 5 Maturity Levels



- This certificates are given to Company Process framework for achieving Quality Sotware Product

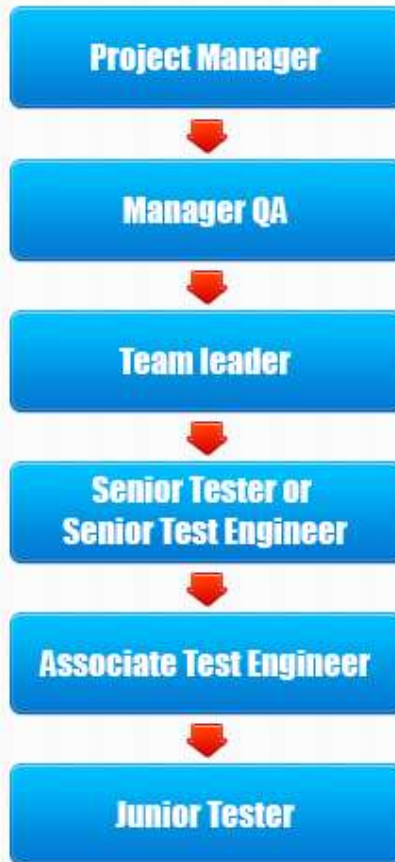


CONCLUSION

- What is a good test for you
- What is your strategy for testing
- What is your definition of Quality



Software Testing Job Hierarchy



hierarchystructure.com



- What is Testing ?
- Testing Principles
- Types of testing – static , dynamic
- Functional - Unit, Integration, System ,Acceptance
- STLC
- What is objective of testing
- Quality – QA, QC
- Defect, bug, error , failure, fault
- White /Black/Grey Box testing, differences



