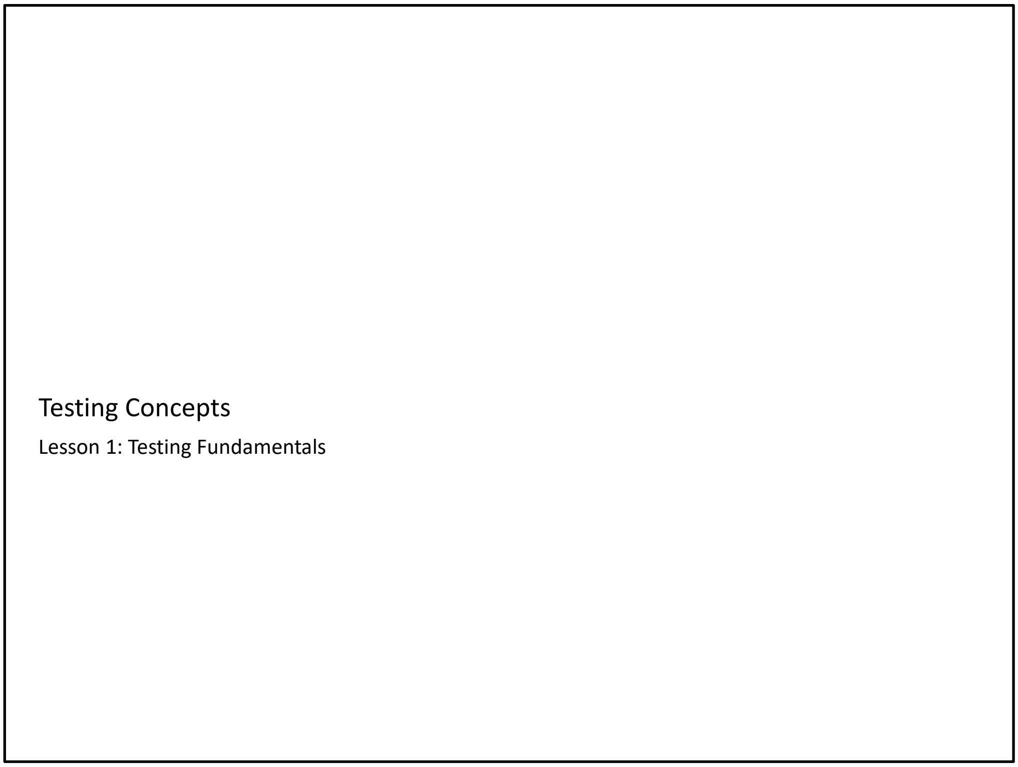


Testing Concepts

Lesson 1: Testing Fundamentals

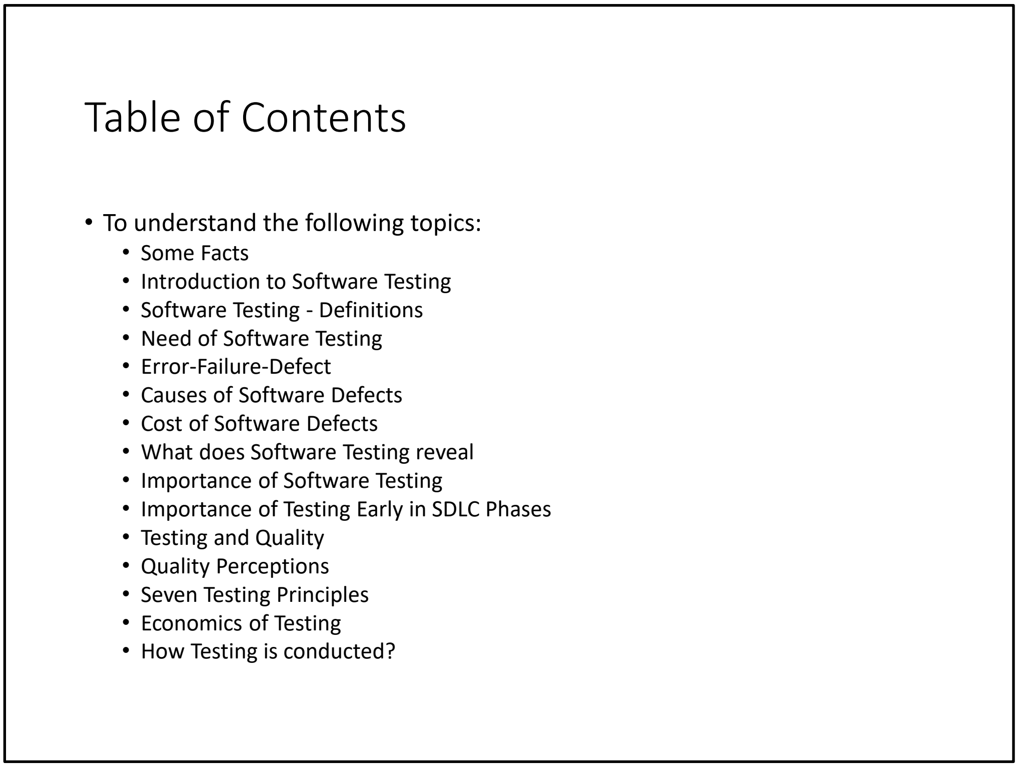


Table of Contents

- To understand the following topics:
  - Some Facts
  - Introduction to Software Testing
  - Software Testing - Definitions
  - Need of Software Testing
  - Error-Failure-Defect
  - Causes of Software Defects
  - Cost of Software Defects
  - What does Software Testing reveal
  - Importance of Software Testing
  - Importance of Testing Early in SDLC Phases
  - Testing and Quality
  - Quality Perceptions
  - Seven Testing Principles
  - Economics of Testing
  - How Testing is conducted?

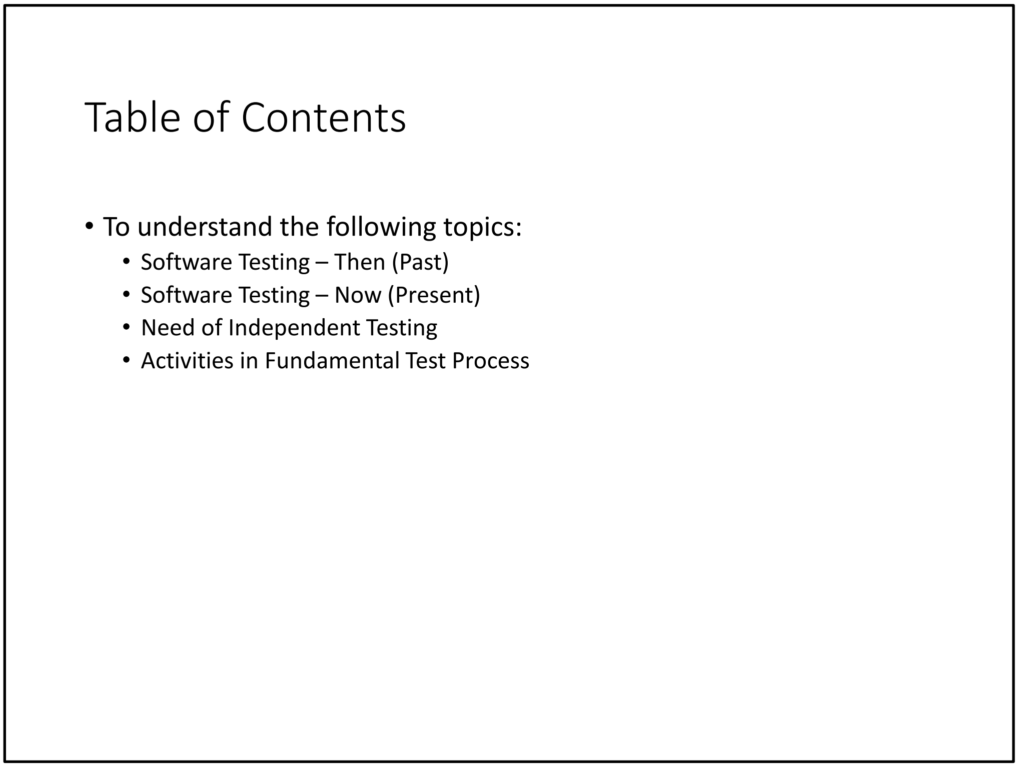
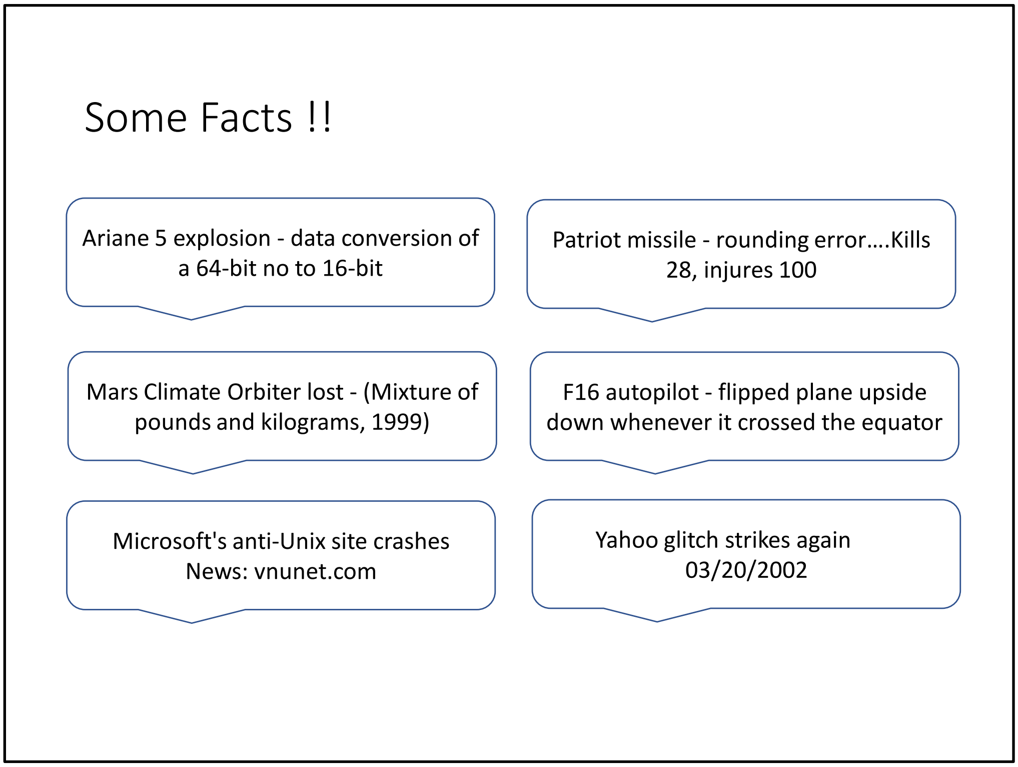


Table of Contents

- To understand the following topics:
  - Software Testing – Then (Past)
  - Software Testing – Now (Present)
  - Need of Independent Testing
  - Activities in Fundamental Test Process



Some Facts !!

Ariane 5 explosion - data conversion of a 64-bit no to 16-bit

Patriot missile - rounding error….Kills 28, injures 100

Mars Climate Orbiter lost - (Mixture of pounds and kilograms, 1999)

F16 autopilot - flipped plane upside down whenever it crossed the equator

Microsoft's anti-Unix site crashes News: vnunet.com

Yahoo glitch strikes again 03/20/2002

Some Facts

Ariane 5 explosion - data conversion of a 64-bit no to 16-bit

On 4 June 1996, the maiden flight of the Ariane 5 launcher ended in a failure.

Only about 40 seconds after initiation of the flight sequence, at an altitude of about 3700 m, the launcher veered off its flight path, broke up and exploded.

The internal SRI* software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer

Patriot missile - rounding error, kills 28, injures 100

A report of the General Accounting office, GAO/IMTEC-92-26, entitled Patriot Missile Defense: Software Problem Led to System Failure at Dhahran,

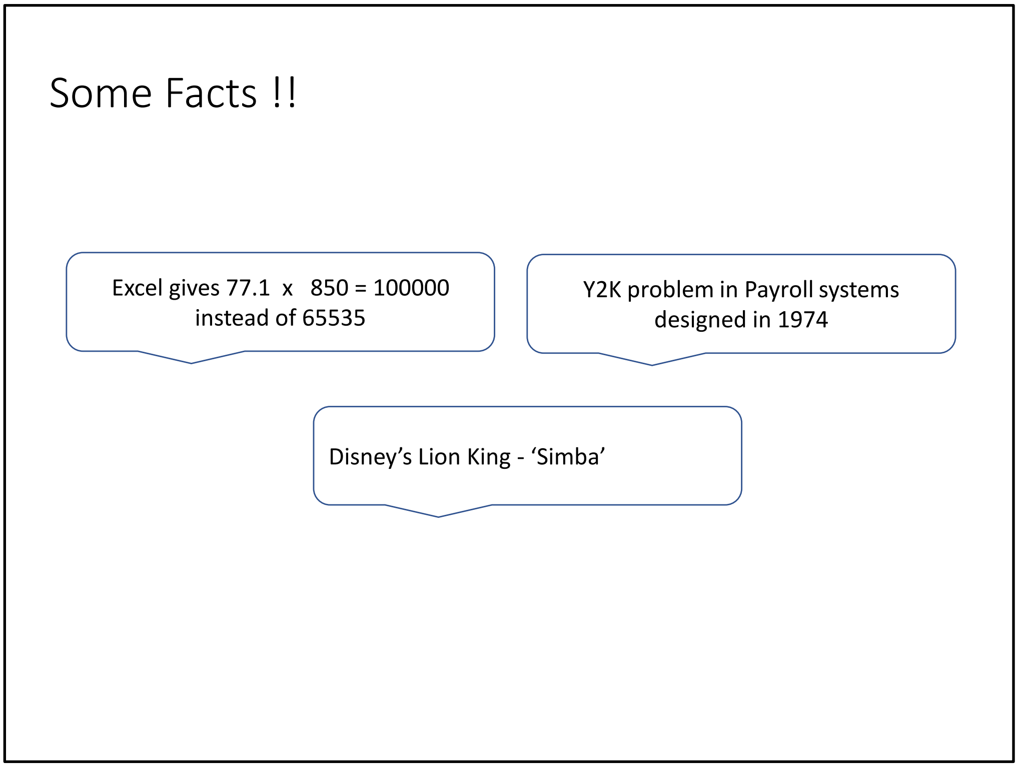
Saudi Arabia reported on the cause of the failure. It turns out that the cause was an inaccurate calculation of the time since boot due to computer arithmetic errors.

Mars Climate Orbiter lost - (Mixture of pounds and kilograms, 1999)

The peer review preliminary findings indicate that one team used English units

(e.g. inches, feet and pounds) while the other used metric units for a key spacecraft operation

F16 autopilot - flipped plane upside down whenever it crossed the equator



Some Facts !!

Excel gives 77.1 x 850 = 100000 instead of 65535

Y2K problem in Payroll systems designed in 1974

Disney's Lion King - 'Simba'

Microsoft's anti-Unix site crashes - News: vnunet.com

The Web site launched by Microsoft and Unisys to lure customers away from Unix has turned into a major embarrassment in more ways than one. First it was revealed that the site was powered by an open-source version of Unix and was running on the Apache Web server. So Microsoft switched to its Internet Information Server software and the site crashed.

Yahoo glitch strikes again - 03/20/2002

Users report irregular and missing content Parts of Yahoo were shut down on Tuesday following software problems encountered in the merging of Yahoo Groups and Yahoo Clubs.

Excel gives 77.1 x 850 = 100000 instead of 65535

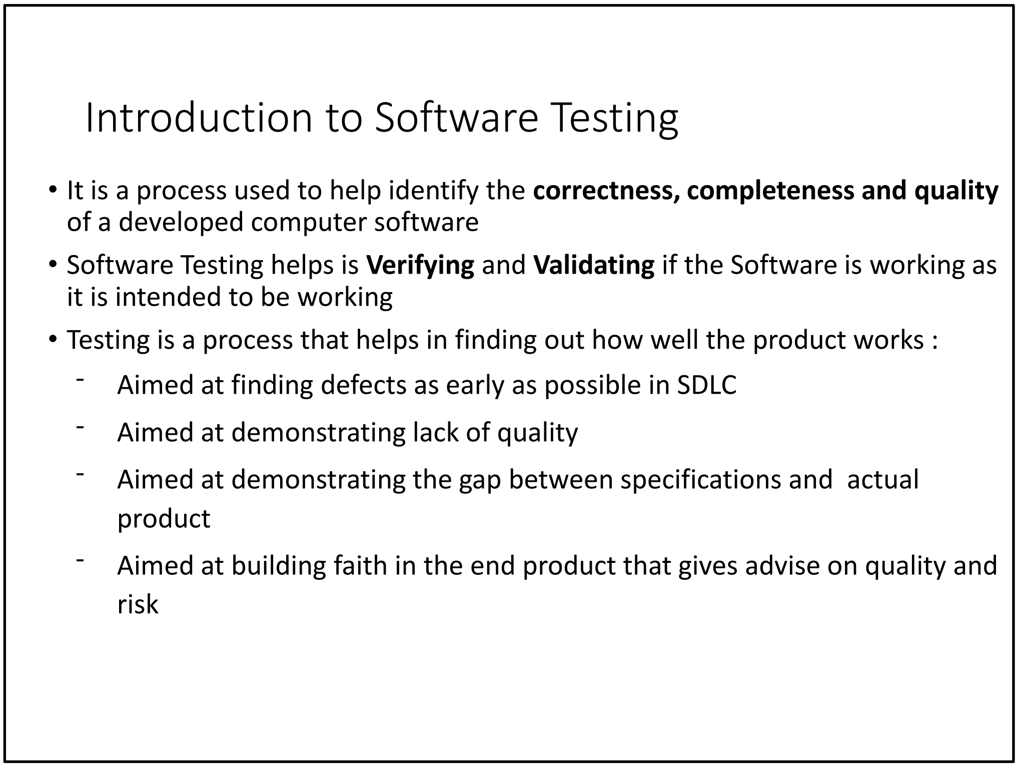
Excel gives 77.1 x 850 = 100000 instead of 65535 : While multiplying two numbers in MS-Excel and if the product equals 65535, it always gives the result 100000.

Y2K problem in Payroll systems designed in 1974

In 1974, when the first payroll was developed, to minimize the utilization of memory space, the year of the dates were stored in two digits instead of four digits i.e. 00 instead of 1900. But after 25 years in the yr. 2000, the question arose that how to store this. Will it be considered 1900 or 2000?

9. Disney's Lion King – Simba

At the fall of 1994 Christmas, the Disney company came up with its first venture – a CDROM game for children with animated story of 'The Lion King Simba'. The sale was very huge. However it was a great loss to the Disney – it failed to work on most of the systems available in the market. The very next day on December 26th, Disney's customer care phones began to ring with calls from angry parents and crying children. Disney failed to properly test the game software on different PC models available in the market and as a result, the software worked only on few systems that were just like the one used by Disney programmers.

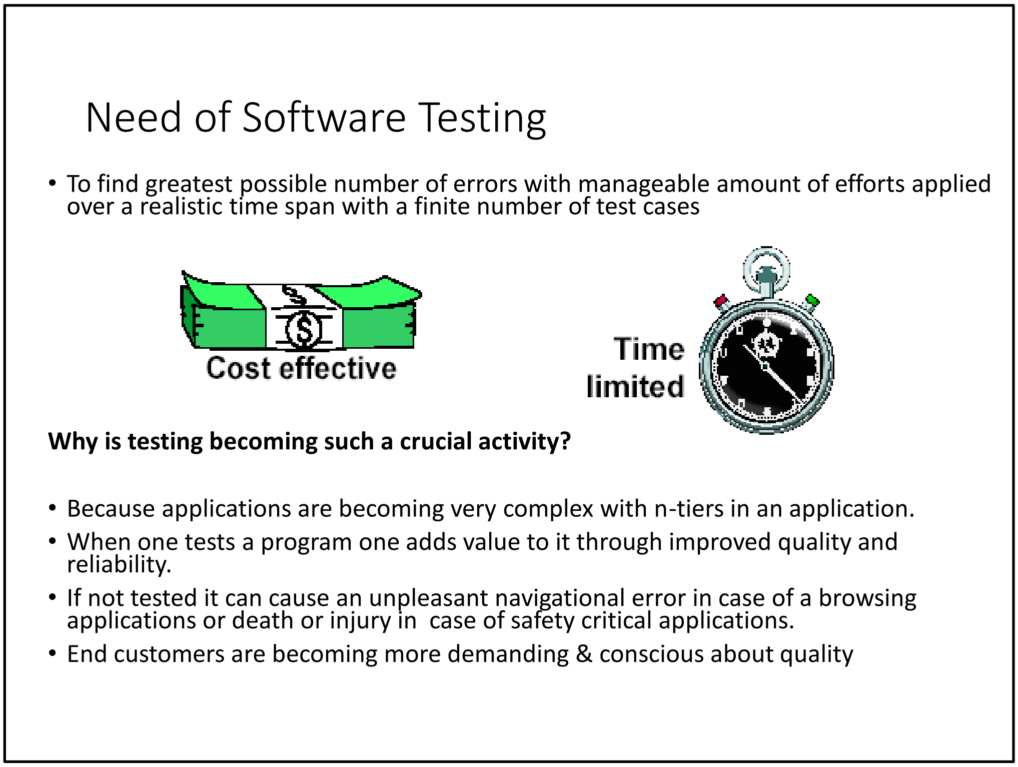


Introduction to Software Testing

- It is a process used to help identify the **correctness, completeness and quality** of a developed computer software
- Software Testing helps is **Verifying** and **Validating** if the Software is working as it is intended to be working
- Testing is a process that helps in finding out how well the product works :
  - Aimed at finding defects as early as possible in SDLC
  - Aimed at demonstrating lack of quality
  - Aimed at demonstrating the gap between specifications and actual product
  - Aimed at building faith in the end product that gives advise on quality and risk

Software Testing Definition :
The process of executing a program (or part of a program) with the intention of finding errors - G. J. Myers
Software testing is the process of testing the functionality and correctness of software by running it
Testing is the process of exercising or evaluating a system or system component by manual or automated means to verify that it satisfies specified requirements - IEEE 83a
The process of analyzing a system to detect the difference between existing and required conditions and to evaluate the feature of the system - IEEE/ANSI, 1983 [Std 829-1983]

Quality: The degree to which a component system or process meets specified requirements and/or user/customer needs and expectation.



Need of Software Testing

- To find greatest possible number of errors with manageable amount of efforts applied over a realistic time span with a finite number of test cases

Cost effective

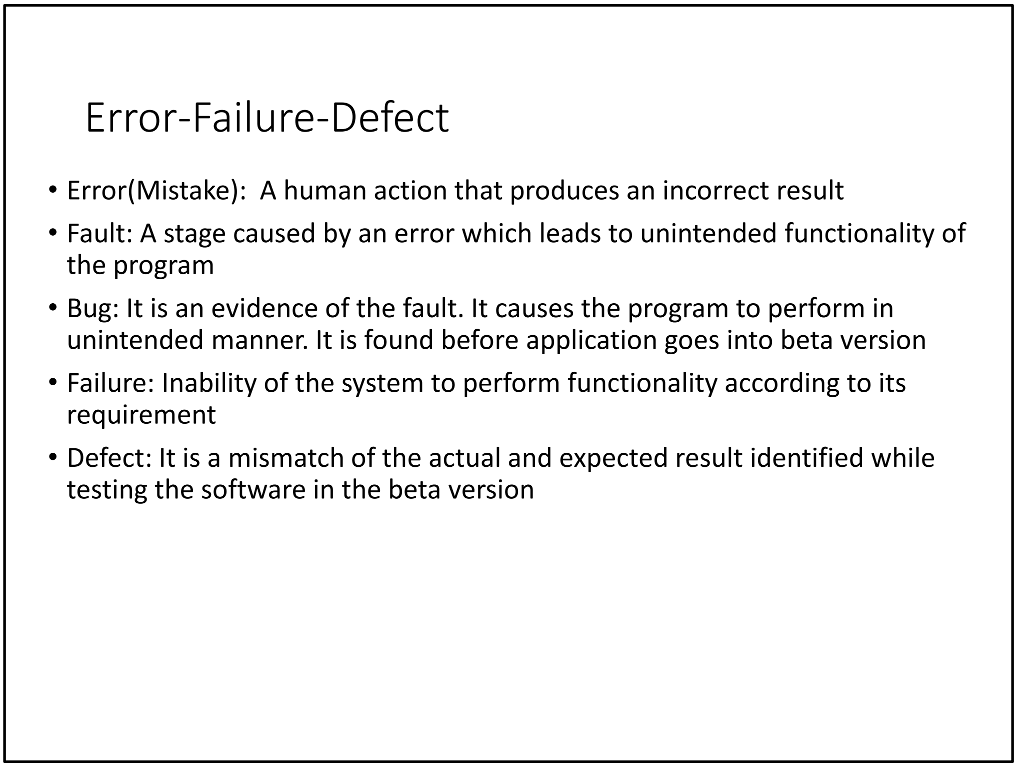
Time limited

Why is testing becoming such a crucial activity?

- Because applications are becoming very complex with n-tiers in an application.
- When one tests a program one adds value to it through improved quality and reliability.
- If not tested it can cause an unpleasant navigational error in case of a browsing applications or death or injury in case of safety critical applications.
- End customers are becoming more demanding & conscious about quality

Why are company's outsourcing the testing phase?

It is being realized that testing is an extremely important phase, customers today are conscious of quality as they need to be more competitive in the market.
It is being realized that the best people to test an application are the ones who have not developed the application. The testers would have the approach of a user and have an unbiased mind.



Error-Failure-Defect

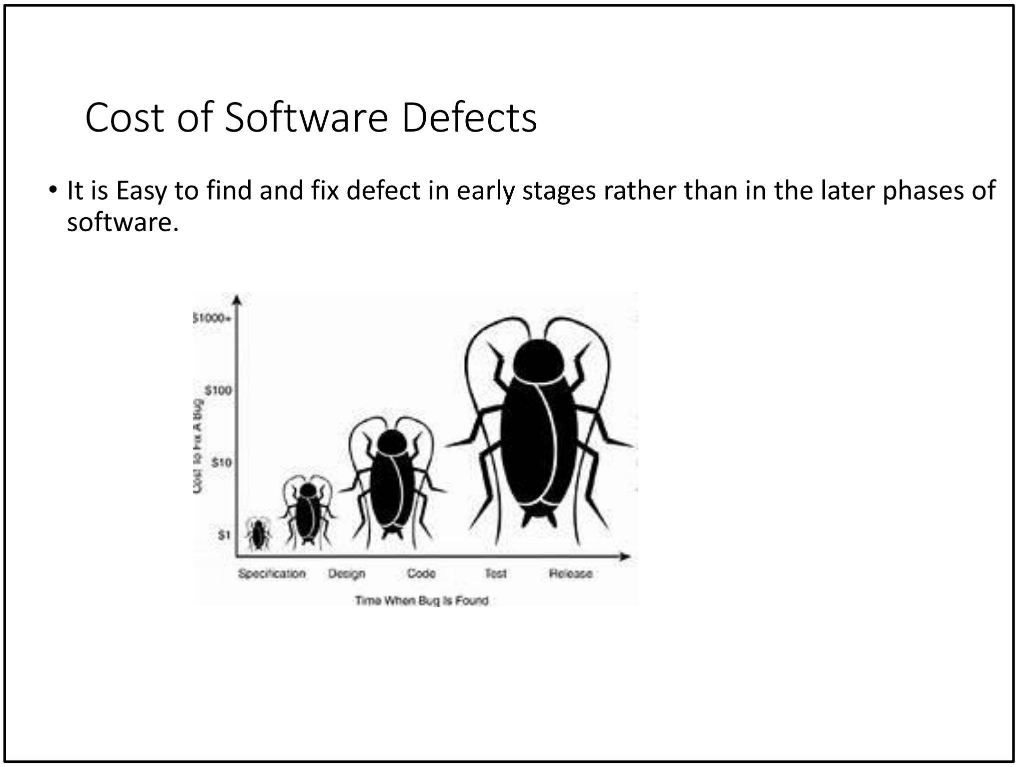
- Error(Mistake): A human action that produces an incorrect result
- Fault: A stage caused by an error which leads to unintended functionality of the program
- Bug: It is an evidence of the fault. It causes the program to perform in unintended manner. It is found before application goes into beta version
- Failure: Inability of the system to perform functionality according to its requirement
- Defect: It is a mismatch of the actual and expected result identified while testing the software in the beta version

Error-Failure-Defect : Example

Consider the below program for addition of two integers;

```
#include<stdio.h>
1.   int main ()
2.   {
3.        int num1, num2, sum;
4.        num1 = 6;
5.        num2 = 4;
6.        sum = num1 – num2;
7.        printf(“6 + 4 = %d", sum);
8.   }
Output : 6 + 4 = 2
```

After compiling and running this program we see that the program has failed to do what it was supposed to do. The program was supposed to add two numbers but it did not. The result of 4 + 6 should be 10, but the result is 2. For now we have detected a ***failure***. The ***fault*** in the program is the line 7 has ‘-’ sign instead of ‘+’ sign which led to a failure (deviation from the required functionality). In this case we can also say we have found the ***bug***. ***Error*** is the mistake programmer made by typing ‘-’ instead of ‘+’ sign. The tester is testing the functionality of the program and realizes the output is faulty and will raise a ***defect***.



Cost of Software Defects

- It is Easy to find and fix defect in early stages rather than in the later phases of software.

Importance of Testing Early in SDLC Phases

Many problems raise during planning or design. Requirements testing can prevent future problems thus lowering the cost.

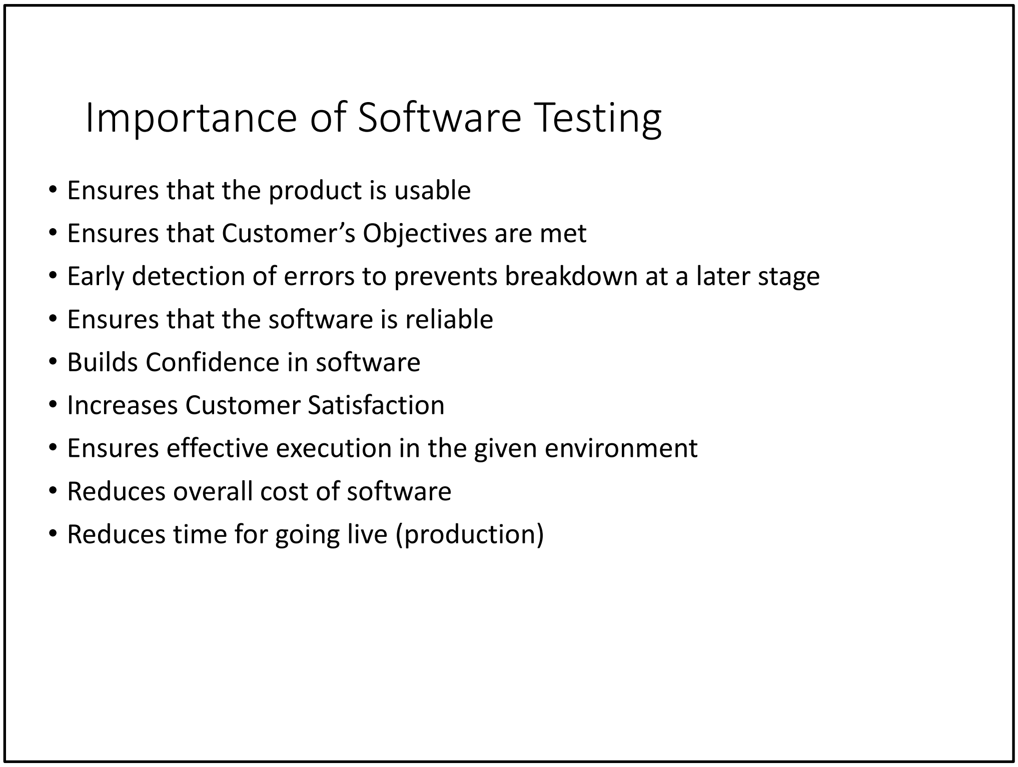
Since the testing process is involved with all phases of the SDLC, Management will not feel as if testing is a bottleneck to release the product.

Test cases written during requirements and shared with the Dev. team before the construction phase can help developers to reduce the chances of failure

The test environment can be prepared in advance

The risk of having a short time for testing is greatly reduced

Involving quality assurance in all phases of the SDLC helps creating a ‘quality culture’ inside the organization.



Importance of Software Testing

- Ensures that the product is usable
- Ensures that Customer’s Objectives are met
- Early detection of errors to prevents breakdown at a later stage
- Ensures that the software is reliable
- Builds Confidence in software
- Increases Customer Satisfaction
- Ensures effective execution in the given environment
- Reduces overall cost of software
- Reduces time for going live (production)

Causes of Software Defects :

Software is written by human beings

Under increasing pressure to deliver to strict deadlines

No time to check but assumptions may be wrong

Environmental conditions

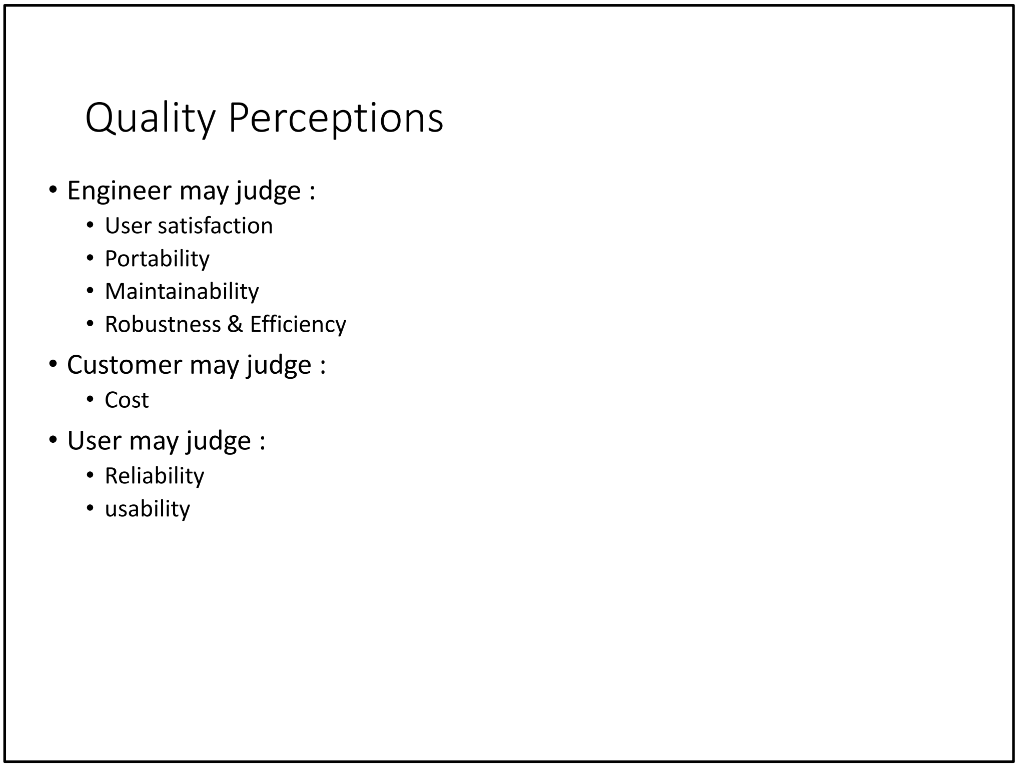
Radiations, Magnetism, Electronic fields and pollution can cause faults in firmware or influence execution of software by changing hardware conditions

Minimal or no proper documentation of Business Requirements

Insufficient time window for development

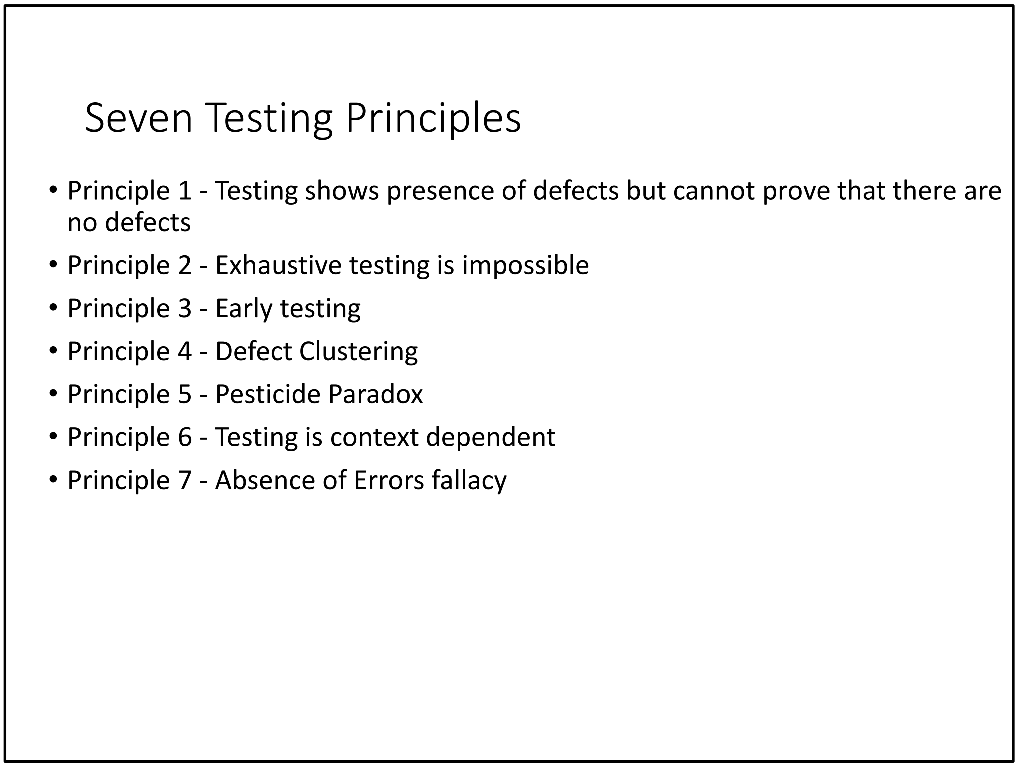
Lack of domain knowledge

Programming Language constraints



Quality Perceptions

- Engineer may judge :
    - User satisfaction
    - Portability
    - Maintainability
    - Robustness & Efficiency
- Customer may judge :
    - Cost
- User may judge :
    - Reliability
    - usability



Seven Testing Principles

- Principle 1 - Testing shows presence of defects but cannot prove that there are no 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

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.

Principle 2: Exhaustive testing is impossible
Testing everything (all combinations of inputs and preconditions) is not feasible except for trivial cases. Instead of exhaustive testing, we use risks and priorities to focus testing efforts.

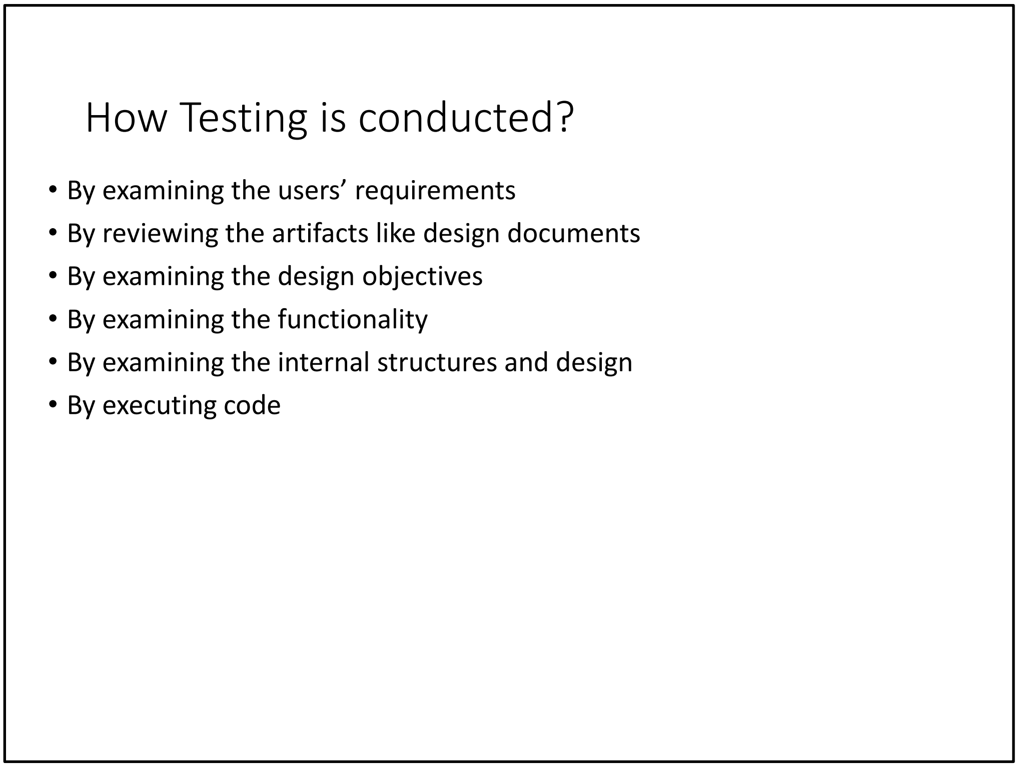
Principle 3: Early testing
Testing activities should start as early as possible in the software or system development life cycle and should be focused on defined objectives.

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 of test cases will no longer find any new bugs. To overcome this 'pesticide paradox', the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

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.

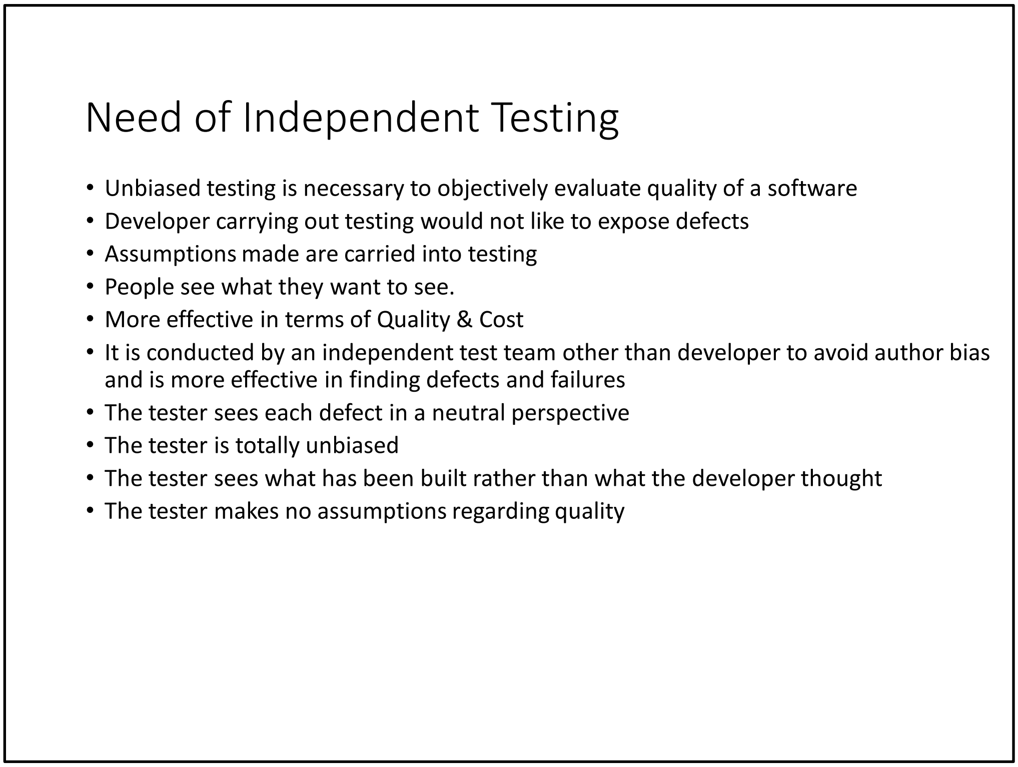


How Testing is conducted?

- By examining the users' requirements
- By reviewing the artifacts like design documents
- By examining the design objectives
- By examining the functionality
- By examining the internal structures and design
- By executing code

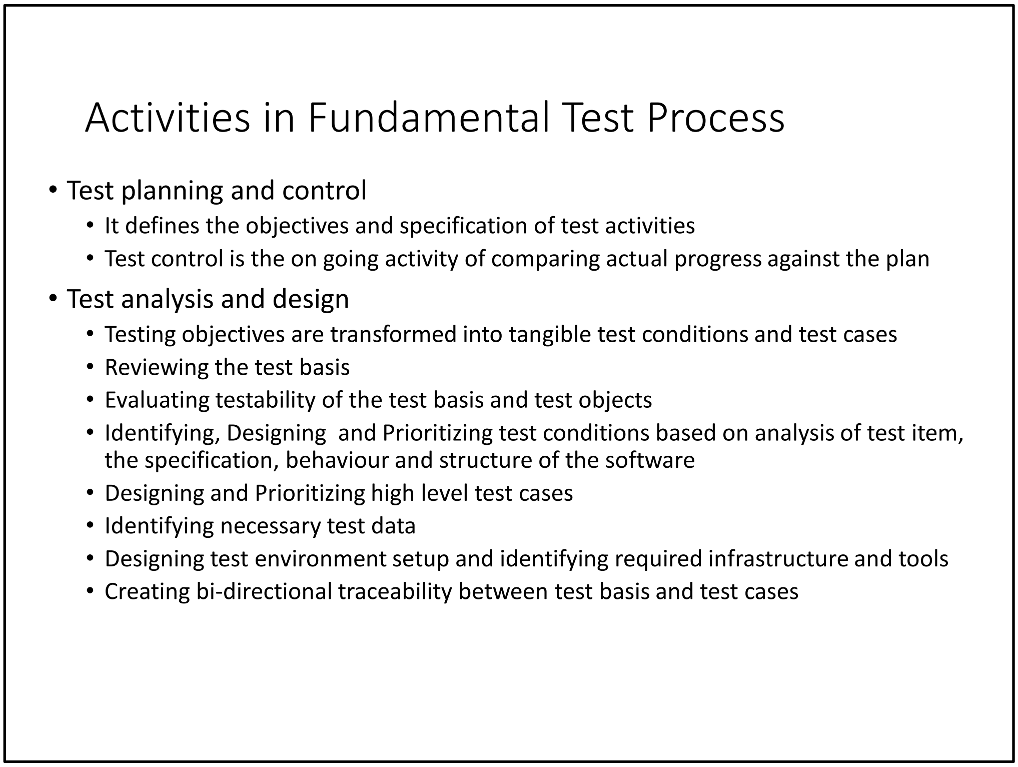
How testing is conducted?

Testing is conducted with the help of users requirements, design documents, functionality, internal structures & design, by executing code.



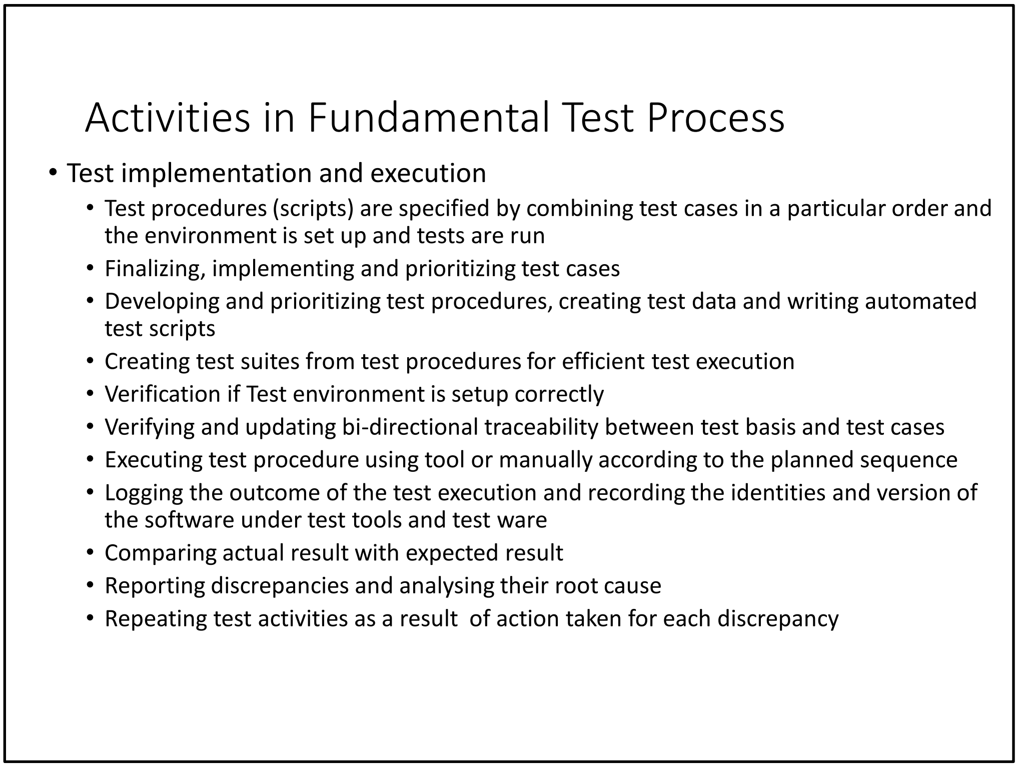
Need of Independent Testing

- Unbiased testing is necessary to objectively evaluate quality of a software
- Developer carrying out testing would not like to expose defects
- Assumptions made are carried into testing
- People see what they want to see.
- More effective in terms of Quality & Cost
- It is conducted by an independent test team other than developer to avoid author bias and is more effective in finding defects and failures
- The tester sees each defect in a neutral perspective
- The tester is totally unbiased
- The tester sees what has been built rather than what the developer thought
- The tester makes no assumptions regarding quality



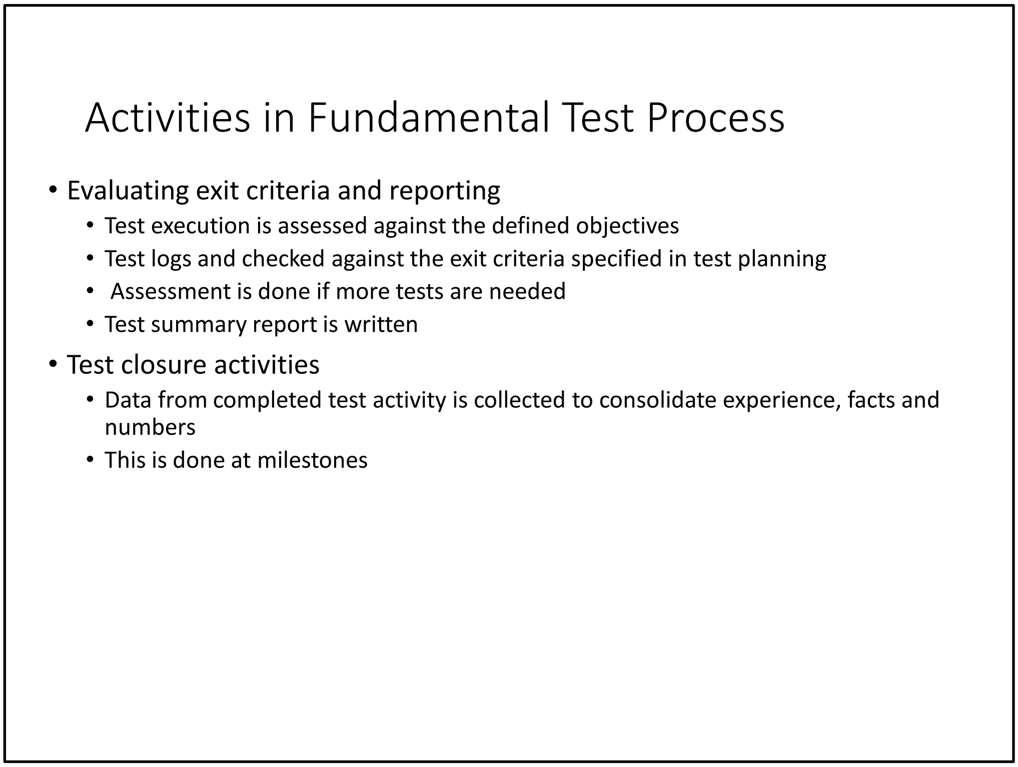
Activities in Fundamental Test Process

- Test planning and control
  - It defines the objectives and specification of test activities
  - Test control is the on going activity of comparing actual progress against the plan
- Test analysis and design
  - Testing objectives are transformed into tangible test conditions and test cases
  - Reviewing the test basis
  - Evaluating testability of the test basis and test objects
  - Identifying, Designing and Prioritizing test conditions based on analysis of test item, the specification, behaviour and structure of the software
  - Designing and Prioritizing high level test cases
  - Identifying necessary test data
  - Designing test environment setup and identifying required infrastructure and tools
  - Creating bi-directional traceability between test basis and test cases



Activities in Fundamental Test Process

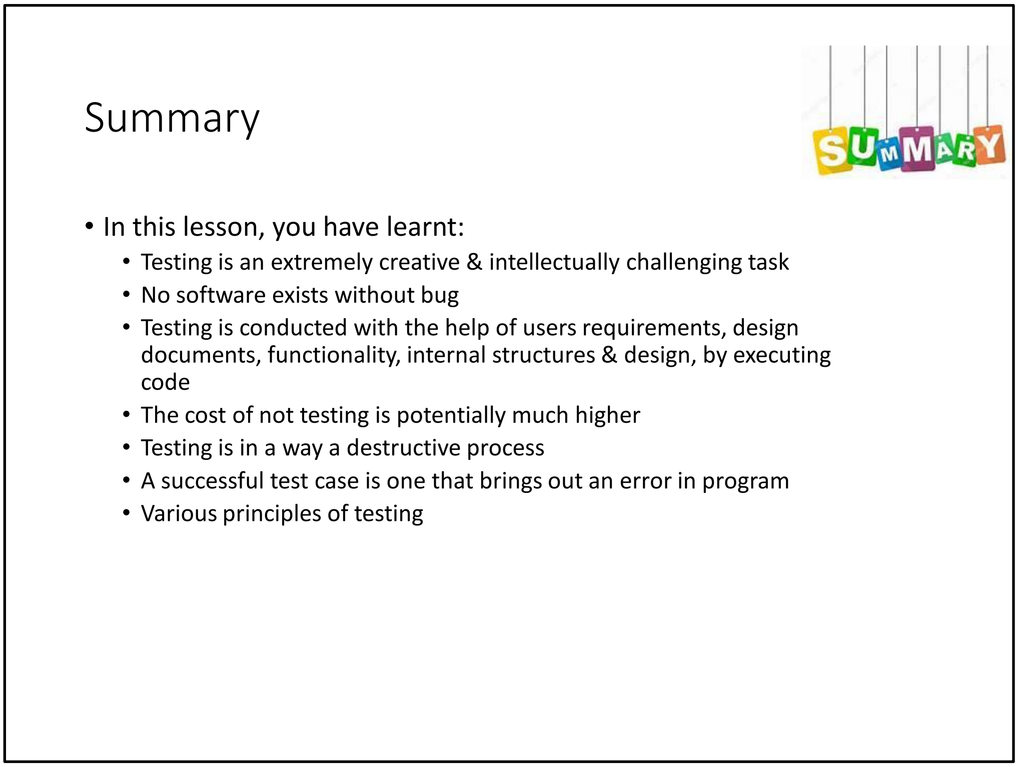
- Test implementation and execution
  - Test procedures (scripts) are specified by combining test cases in a particular order and the environment is set up and tests are run
  - Finalizing, implementing and prioritizing test cases
  - Developing and prioritizing test procedures, creating test data and writing automated test scripts
  - Creating test suites from test procedures for efficient test execution
  - Verification if Test environment is setup correctly
  - Verifying and updating bi-directional traceability between test basis and test cases
  - Executing test procedure using tool or manually according to the planned sequence
  - Logging the outcome of the test execution and recording the identities and version of the software under test tools and test ware
  - Comparing actual result with expected result
  - Reporting discrepancies and analysing their root cause
  - Repeating test activities as a result of action taken for each discrepancy



Activities in Fundamental Test Process

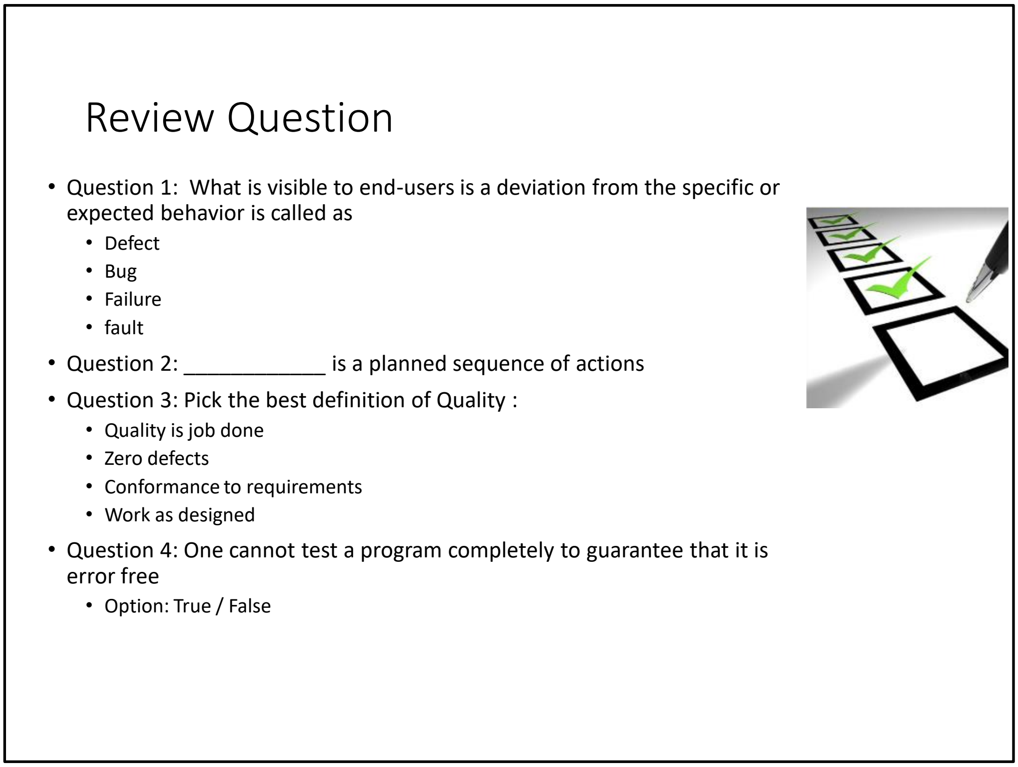
- Evaluating exit criteria and reporting
  - Test execution is assessed against the defined objectives
  - Test logs and checked against the exit criteria specified in test planning
  - Assessment is done if more tests are needed
  - Test summary report is written
- Test closure activities
  - Data from completed test activity is collected to consolidate experience, facts and numbers
  - This is done at milestones

Testware: Artifacts produced during the test process required to plan, design and execute test and any additional software or utilities used in testing.



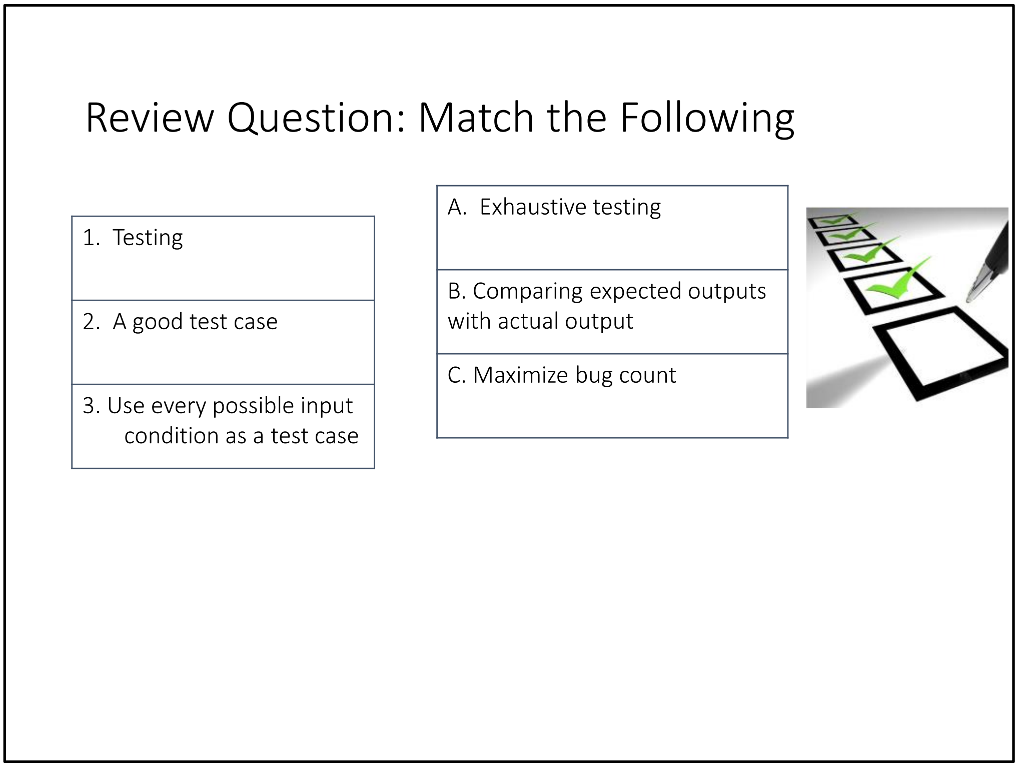
Summary

- In this lesson, you have learnt:
  - Testing is an extremely creative & intellectually challenging task
  - No software exists without bug
  - Testing is conducted with the help of users requirements, design documents, functionality, internal structures & design, by executing code
  - The cost of not testing is potentially much higher
  - Testing is in a way a destructive process
  - A successful test case is one that brings out an error in program
  - Various principles of testing



Review Question

- Question 1: What is visible to end-users is a deviation from the specific or expected behavior is called as
  - Defect
  - Bug
  - Failure
  - fault
- Question 2: _____________ is a planned sequence of actions
- Question 3: Pick the best definition of Quality :
  - Quality is job done
  - Zero defects
  - Conformance to requirements
  - Work as designed
- Question 4: One cannot test a program completely to guarantee that it is error free
  - Option: True / False



Review Question: Match the Following

1. Testing

2. A good test case

3. Use every possible input condition as a test case

A. Exhaustive testing

B. Comparing expected outputs with actual output

C. Maximize bug count