Introduction
Software Testing

Outlines

- Software Testing Introduction,
- What is testing?
- Importance of testing
- Testing terminologies
- Principles of software testing,
- Differences between Manual and Automation Testing.

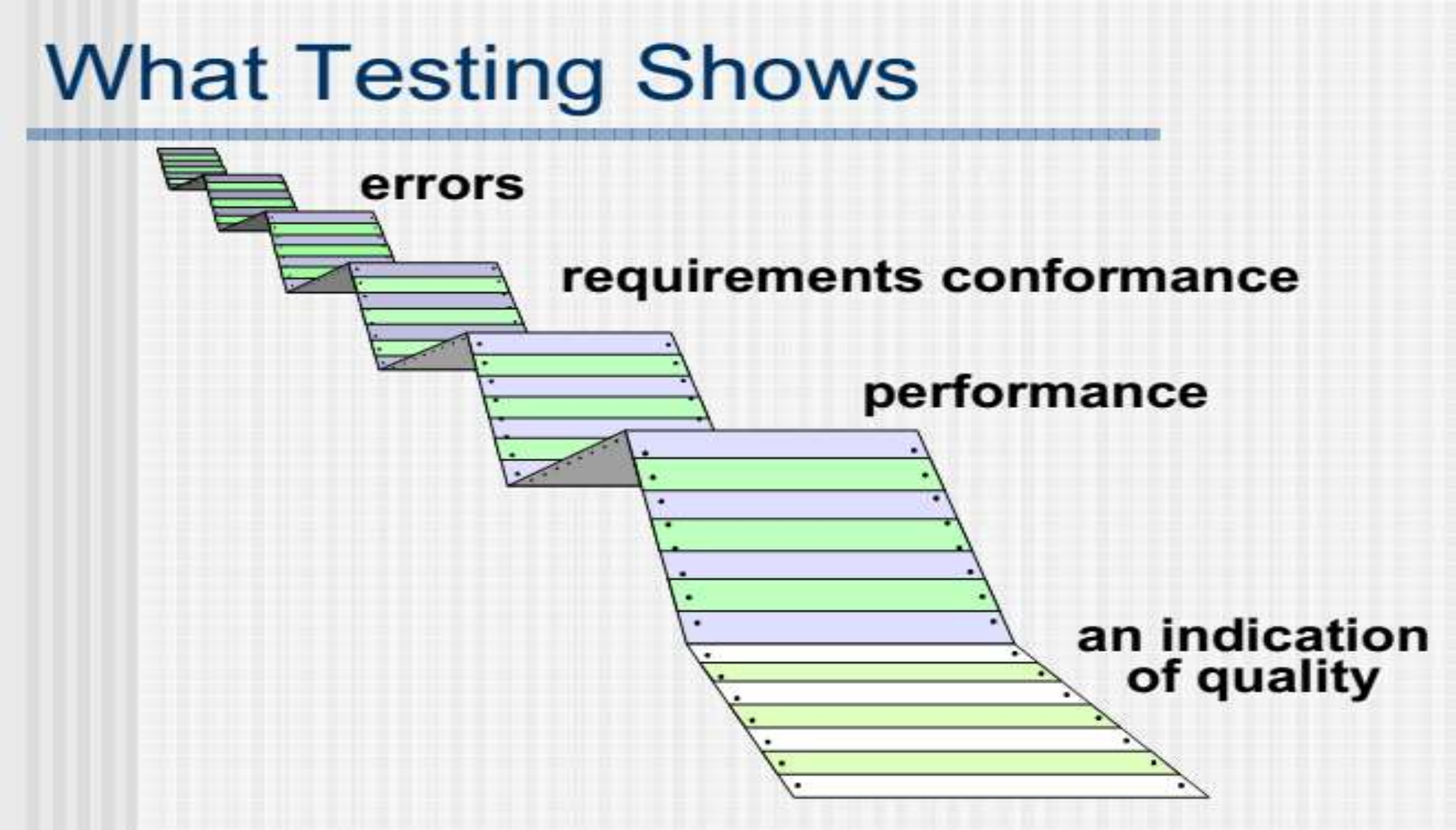
Software Testing

- Software testing is defined as an activity to check whether the **actual results match the expected results** and to ensure that the software system is Defect free.
- Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements.
- It can be either done manually or using automated tools

Purpose-Not only to find errors but more

- Software testing is required to check the **reliability** of the software
- Software testing ensures that the system is free from any **bug** that can cause any kind of failure
- Software testing ensures that the product is in line with the **requirement** of the client- **Requirements conformance**
- It is required to make sure that the final product is **user friendly**
- **No matter how well the software design looks on paper, once the development starts and you start testing the product you will definitely find lots of defects in the design.**

Out put of testing



What Testing Shows

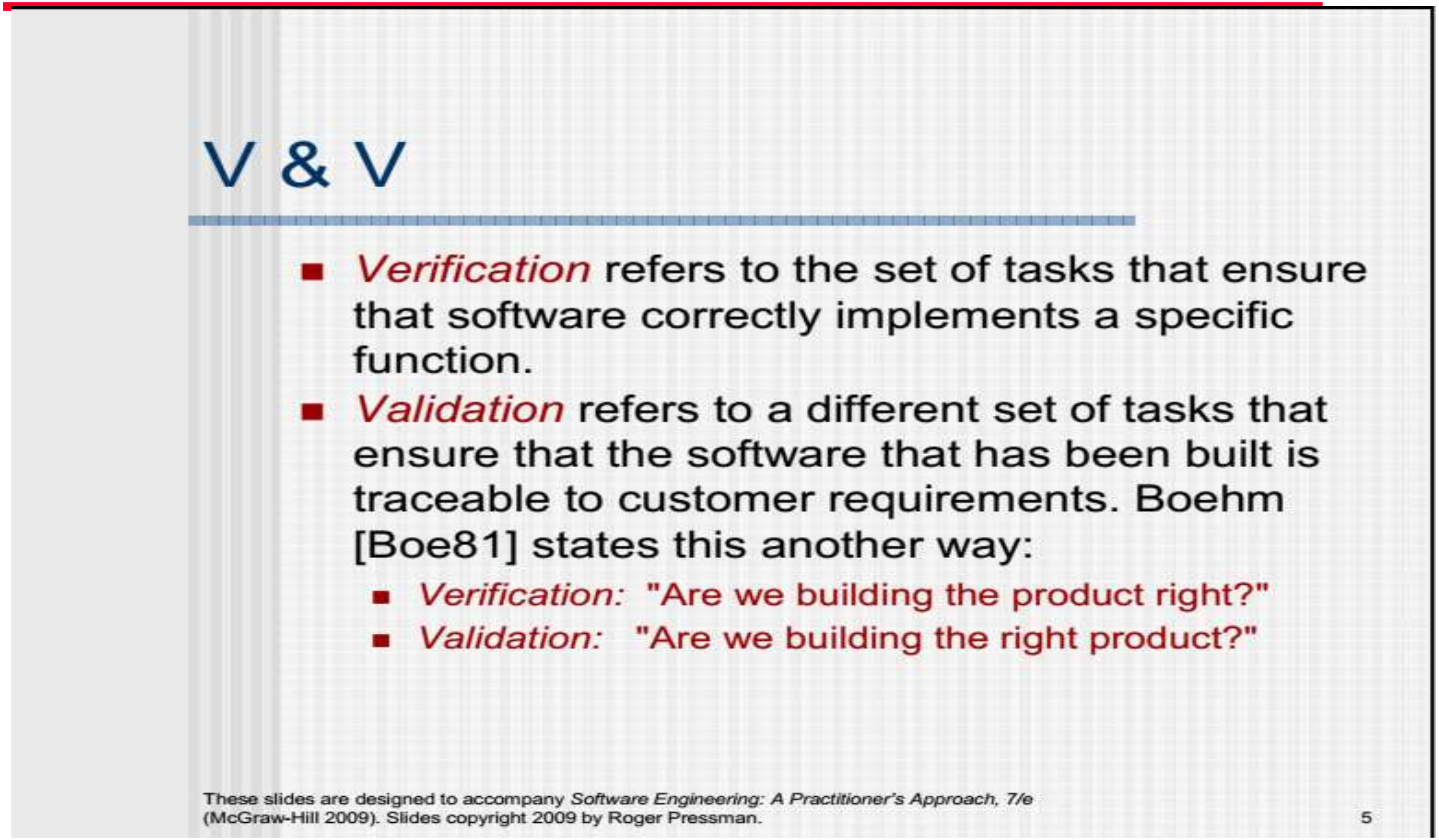
errors

requirements conformance

performance

an indication
of quality

V & V

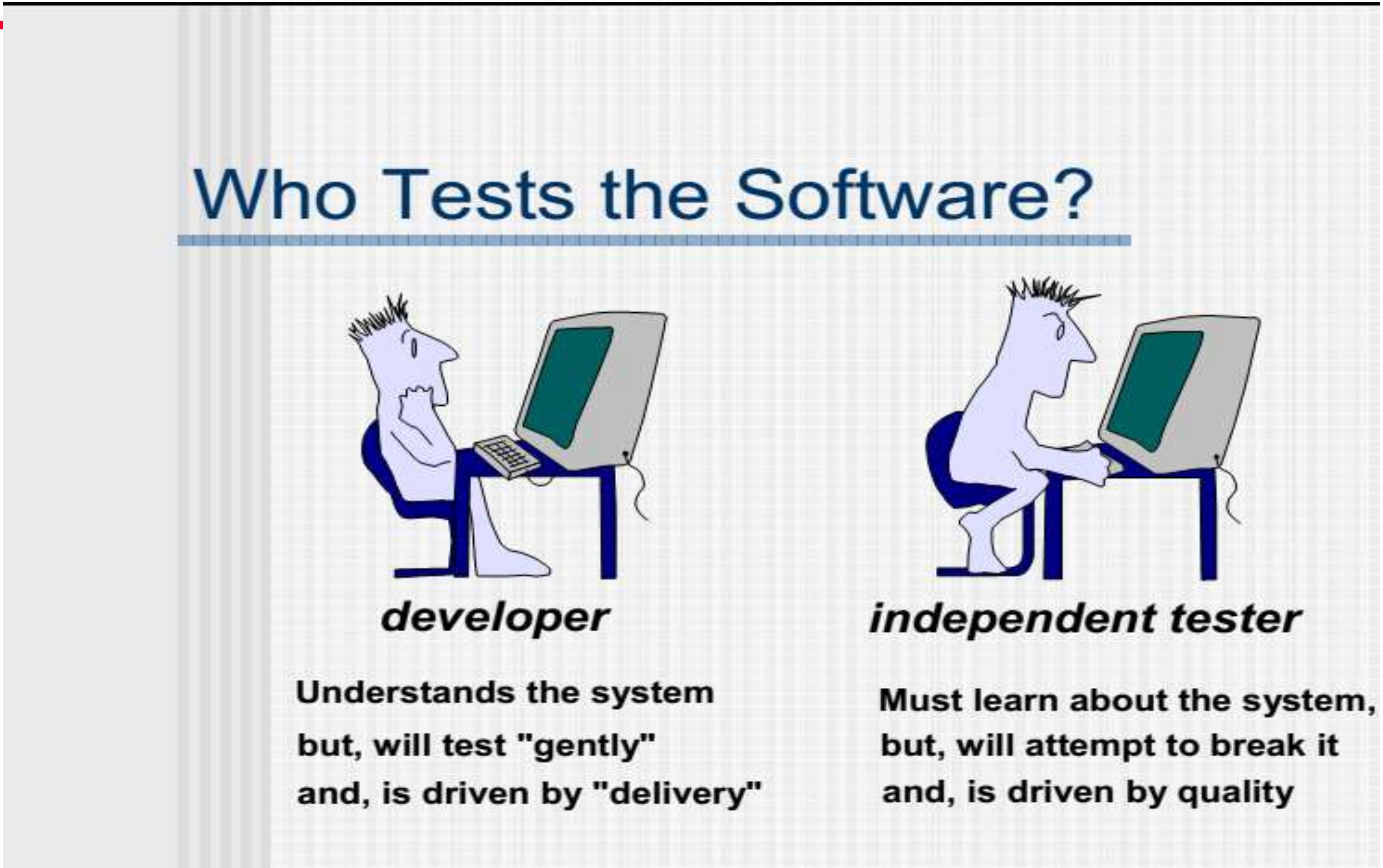


V & V

- *Verification* refers to the set of tasks that ensure that software correctly implements a specific function.
- *Validation* refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements. Boehm [Boe81] states this another way:
  - *Verification:* "Are we building the product right?"
  - *Validation:* "Are we building the right product?"

These slides are designed to accompany *Software Engineering: A Practitioner's Approach, 7/e* (McGraw-Hill 2009). Slides copyright 2009 by Roger Pressman.

Who?



Who Tests the Software?

developer

Understands the system

but, will test "gently"

and, is driven by "delivery"

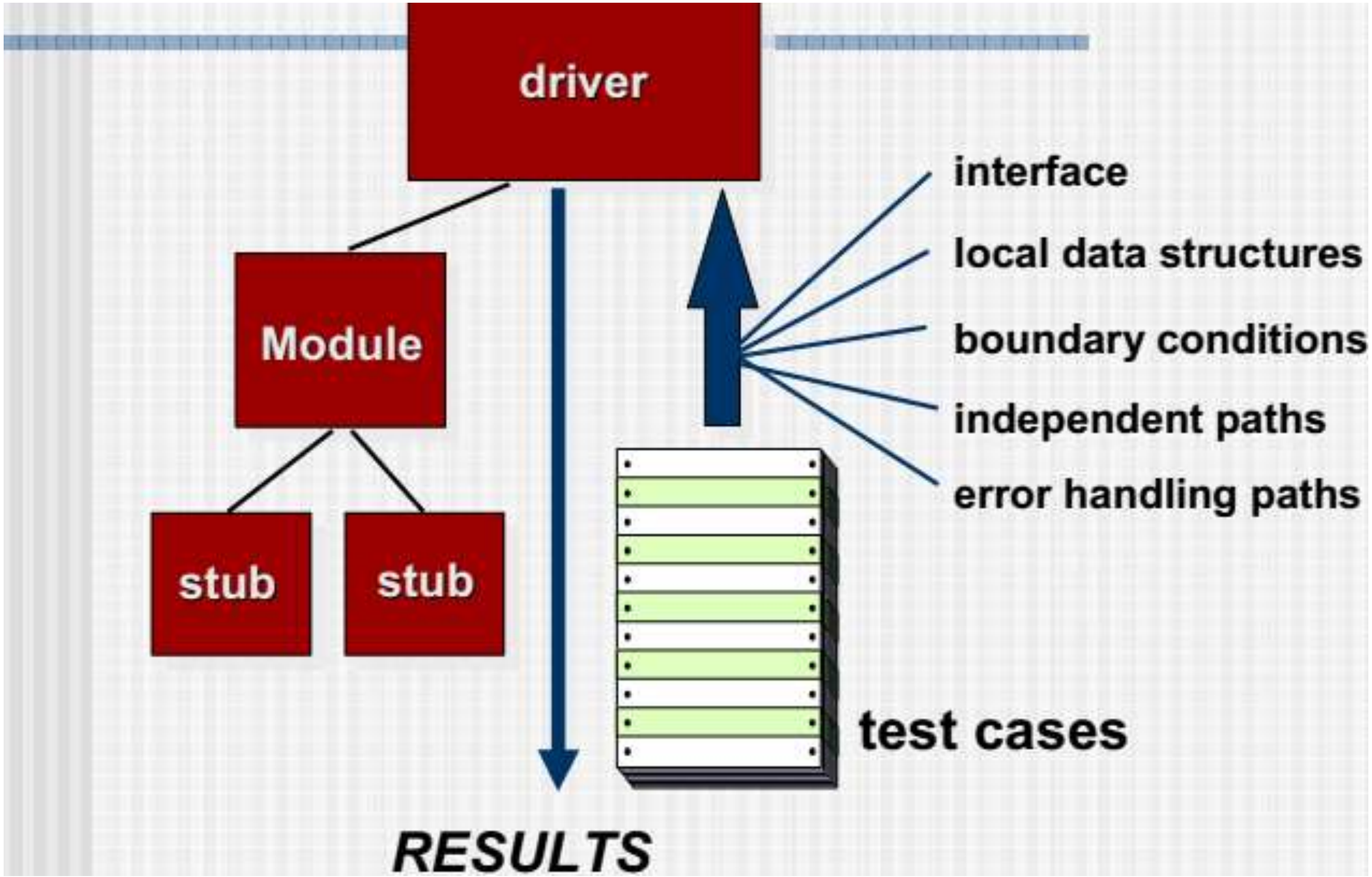
independent tester

Must learn about the system,

but, will attempt to break it

and, is driven by quality

Testing Mechanism



Myths of testing

- ***Myth: Anyone can do manual testing***
- **Fact**: Testing requires many skill sets

- ***Myth: Testing ensures 100% Defect free product***
- **Fact**: Testing attempts to find as many defects as possible. Identifying all possible defects is impossible.

- ***Myth: Automated testing is more powerful than manual testing***
- **Fact**: 100% test automation cannot be done. Manual Testing is also essential.
- ***Myth: Testing is easy***
- **Fact**: Testing can be extremely challenging. Testing an application for possible use cases with minimum test cases requires high analytical skills.

7-Principles of software testing

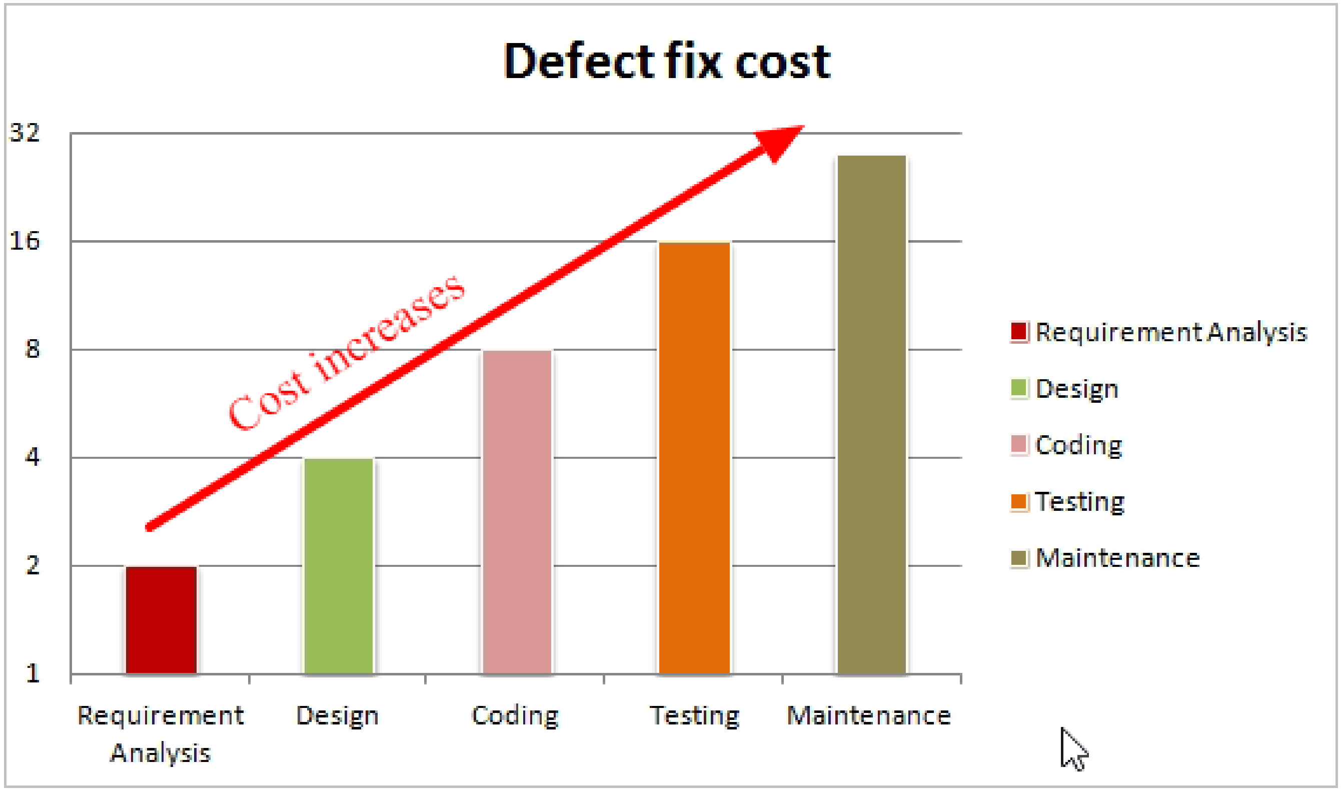
#1) Testing Shows the Presence of Defects

- In the software testing industry, no one will say that there is **no defect** in the software, which is quite true as testing cannot prove that the software is error-free or defect-free.

#2) Early Testing

- Testers need to get involved at an early stage of the Software Development Life Cycle (SDLC).

Cost required for fixing a defect found during the Requirement Analysis is less and it goes on increasing as we move towards the Testing or the Maintenance phase.



7-Principles of software testing

#3) Exhaustive Testing is Not Possible

- It is not possible to test all the functionalities with all valid and invalid combinations of input data during actual testing. Instead of this approach, testing of a few combinations is considered based on priority using different techniques.

#4) Testing is Context-Dependent

- Different domains are tested differently, thus testing is purely based on the context of the domain or application.

#5) Defect Clustering

- During testing, it may happen that most of the defects found are related to a small number of modules.

7-Principles of software testing

#6) Pesticide Paradox

- Pesticide Paradox principle says that if the same set of test cases are executed again and again over the period of time then these set of tests are not capable enough to identify new defects in the system.

#7) Absence of Error

- If the software is tested fully and if no defects are found before release, then we can say that the software is 99% defect free. But what if this software is tested against wrong requirements? In such cases, even finding defects and fixing them on time would not help as testing is performed on wrong requirements which are not as per needs of the end user.

Categories Of Testing

- **Manual Testing**
- Manual testing is testing of the software where tests are executed manually by a QA Analysts. It is performed to discover bugs in software under development.
- In Manual testing, the tester checks all the essential features of the given application or software.
- In this process, the software testers execute the test cases and generate the test reports without the help of any automation software testing tools.

Categories Of Testing

- **Automation Testing**
- In Automated Software Testing, testers write code/test scripts to automate test execution.
- Testers use appropriate automation tools to develop the test scripts and validate the software.
- The goal is to complete test execution in a less amount of time.
- Automated testing entirely relies on the pre-scripted test which runs automatically to compare actual result with the expected results. This helps the tester to determine whether or not an application performs as expected.

Manual VS Automated Testing

Manual Testing	Automated Testing
•Manual testing is not accurate at all times due to human error, hence it is less reliable.	•Automated testing is more reliable, as it is performed by tools and/or scripts.
•Manual testing is time-consuming, taking up human resources.	•Automated testing is executed by software tools, so it is significantly faster than a manual approach.
•Investment is required for human resources.	•Investment is required for testing tools.
•Manual testing is only practical when the test cases are run once or twice, and frequent repetition is not required.	•Automated testing is a practical option when the test cases are run repeatedly over a long time period.
•Manual testing allows for human observation, which may be more useful if the goal is user-friendliness or improved customer experience.	•Automated testing does not entail human observation and cannot guarantee user-friendliness or positive customer experience.

TESTING TERMINOLOGY

- **Error, Bug, Defect**
- **The error is a human mistake.**
- The errors can be committed by anyone in the IT team during the different phases of software development.
- A bug is getting a problem at the time of testing, where as a defect is problem that got by the customer in production time.
- BUG is the issue found in the software during the Testing Lifecycle.
- Defect is an issue which is not found during the Testing Lifecycle

TESTING TERMINOLOGY

- **Errors and how they occur**
- Why do we make *errors* that cause *faults* in computer software leading to potential *failure* of our systems? Well, firstly we are all prone to making simple human errors. This is an unavoidable fact of life. However, this is compounded by the fact that we all operate under real world pressures such as tight deadlines, budget restrictions, conflicting priorities and so on.

Testing VS Debugging

- **Testing**:
  Testing is the process of verifying and validating that a software or application is bug free, meets the technical requirements as guided by its design and development and meets the user requirements effectively and efficiently with handling all the exceptional and boundary cases.

- **Debugging**:
  Debugging is the process of fixing a bug in the software. It can defined as the identifying, analyzing and removing errors.
- This activity begins after the software fails to execute properly and concludes by solving the problem and successfully testing the software.
- It is considered to be an extremely complex and tedious task because errors need to be resolved at all stages of debugging.

Testing VS Debugging

TESTING	DEBUGGING
Testing is the process to find bugs and errors.	Debugging is the process to correct the bugs found during testing.
It is the process to identify the failure of implemented code.	It is the process to give the absolution to code failure.
Testing is the display of errors.	Debugging is a deductive process.
Testing is done by the tester.	Debugging is done by either programmer or developer.
There is no need of design knowledge in the testing process.	Debugging can’t be done without proper design knowledge.
Testing can be done by insider as well as outsider.	Debugging is done only by insider. Outsider can’t do debugging.
Testing can be manual or automated.	Debugging is always manual. Debugging can’t be automated.
It is based on different testing levels i.e. unit testing, integration testing, system testing etc.	Debugging is based on different types of bugs.