

SOFTWARE TESTING & QUALITY ASSURANCE

Prof. Yassir Farooqui, Assistant Professor
Computer Science Engineering





CHAPTER-1

SOFTWARE TESTING



Topics to be Covered

- Testing, Verification and Validation
- Test Strategies for Conventional and OO types
- Unit Testing
- Integration Testing
- Validation Testing
- System Testing
- Security Testing, Stress Testing, Performance Testing
- Metrics for Source Code, Metrics for Testing
- Debugging Process, Debugging Strategies





Testing, Verification and Validation

- Testing is the practise of running a programme with the goal of discovering mistakes.
- The existence of bugs but never their absence can be show by research.
- The design of a programme to discover its faults.
- Method of proving the correctness of the programmes is verification.
- Validation is the method of discovering mistake by running software in a specific world.
- Debugging is diagnosing the bug and fixing it.



Testing, Verification and Validation

- Verification is the software system or component evaluation process to determine.
- whether the products of a given development phase satisfy conditions imposed at the beginning of that phase
- Testing is a series of procedures conducted to test any feature of a piece of software.
- Testing is a tool used to reveal software defects and to assess that the software has achieved a specified degree of quality with regard to selected attributes.





Difference between Verification & Validation

- We check whether we are developing the right product or not.
- Verification is also known as static testing.
- It is a process of checking the work-products whether the product meets the specified requirements.
- Quality assurance comes under verification testing.
- Verification is done before the validation testing.
- We check whether the developed product is right.
- Validation is also known as dynamic testing.
- It is a process of checking the software during or at the end of the development cycle to decide whether the software follow the specified business requirements.
- Quality control comes under validation testing.
- After verification testing, validation testing takes place.



Debugging

Debugging, or localization of faults, is the method of

- (1) The fault or flaw is located,
- (2) code upgrade
- (3) the code is re-tested

- Errors: An error on the part of a software developer is a mistake, misconception, or misunderstanding.
- Faults: As a consequence of an mistake, a fault (defect) is inserted into the programme.

It is a software phenomenon that can cause it to behave inappropriately, not as per its specification.

- Failures: A failure is the inability of a software system or component under specified performance specifications to perform its necessary functions.



Test Case

In a practical sense, a test case is a test-related item which contains the following data:

1. A selection of inputs for experiments.
There are data items that the code under test has obtained from an external source.
Hardware , software, or humans may be the external source.
2. Conditions for execution. These are the conditions needed to run the test, such as a certain database state, or a hardware system config.
3. Expected outputs. These are the results stated to be provided by the code being evaluated.



Test or Series of test

- A test is a collection of associated test cases or a set of associated test cases and test procedures.
- Test Suite: Group of associated tests that are database-related and typically run together is often referred to as test suite
- Oracle test: A test oracle is a document or piece of software that helps testers to evaluate whether or not a test has been passed.
- Test Bed: A test bed is an area containing all the hardware and software used to test a part of the software or software system.





Performance of Software

- Quality relates to the degree to which specified specifications are met by a system , system part, or process.
- Quality relates to the extent to which a system, part of the system or process meets customer need or specifications
- Metric: A metric is a quantitative measure of the extent to which a given attribute is possessed by a system , system part, or method.
- Service metric: scale of software, LOC
- Metric of process: prices, time
- Quality metric: A quality metric is a quantitative measure of the degree to which a given quality attribute is present in an object.



Software Testing Principle

A definition can be described as:

1. A theory, or inference, general or fundamental.
2. A behavioural law or code.
3. The rules or facts of nature that underlie an part of the system or process that meets customer need or specifications

The Principles are as follows:

1. Testing is the method of using a selected set of test cases to exercise a software component with the goal of (i) exposing defects and (ii) assessing consistency. **(Testing shows presence of Defect)**
2. If the test target is to detect defects, then a successful test case is one that has a high degree of defects. probability of revealing a yet undetected defect(s). **(Defect Clustering)**



Software Testing Principle

3. Test findings should be meticulously inspected. **(Absence of Error)**
4. The predicted performance or outcome must be contained in a test case.
5. For both valid and invalid input conditions, test cases should be created. **(Exhaustive testing is impossible)**
6. A community that is independent of the production group should carry out the research.
7. Tests have to be reusable and repeatable. **(Pesticide Paradox)**
8. It should be prepared for early testing. **(Early testing)**
9. In the software life cycle, testing activities should be incorporated.
10. A innovative and demanding job is research. **(Testing is context Dependent)**





Test Strategies for Conventional and OO Software

- Conventional research techniques for applications are-testing unit testing.
- Integration checks, device tests and tests for validation are used.
- The first one focuses on any feature of software that is implemented in the source code.
- The second one based on the design of software architecture.
- The third one checks as a whole the other system and applications.
- The validation of specifications is specified by the fourth one against the built software.
- Objective methods' input and output are not basis of testing.



Unit Testing

- It is a testing technique using which individual modules are tested.
- It is concerned with functional correctness of the standalone modules..
- To focus on a relatively small segment of code and aim to exercise a high percentage of the internal path.
- The main aim is to isolate each unit of the system to identify, analyze and fix the defects.
- It is used to find out the defect at the early stage of the testing.
- It is normally performed by software developers themselves or their peers.



Advantages of Unit Testing

- Reduces Defects in the Newly developed features or reduces bugs when changing the existing functionality.
- Reduces Cost of Testing as defects are captured in very early phase.
- Improves design and allows better refactoring of code.
- Unit Tests, when integrated with build gives the quality of the build as well.
- Codes are more reusable. In order to make unit testing possible.
- Unit testing increases confidence in changing / maintaining code.



Integration Testing

- Integration testing is the second level of the software testing process comes after unit testing.
- units or individual components of the software are tested in a group.
- The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.
- Once all the components or modules are working independently, then we need to check the data flow between the dependent modules.
- It consist of two types Top down and Bottom up Integration testing.



Top-down Integration Test

- It is a method in which integration testing takes place from top to bottom.
- The higher level modules are tested first and then lower level modules are tested and integrated in order to check the software functionality.
- The control program is tested first.
- Modules are integrated one at a time.
- Emphasize on interface testing.

Advantages: No test drivers needed.

- Interface errors are discovered early
- Modular features aid debugging.

Disadvantages: Test stubs are needed

- Errors in critical modules at low levels are found late.



Bottom-up Integration Test

- It is a strategy in which the lower level modules are tested first.
 - These tested modules are then further used to facilitate the testing of higher level modules.
 - The process continues until all modules at top level are tested.
 - Once the lower level modules are tested and integrated, then the next level of modules are formed.
 - Allow early testing aimed at proving feasibility
 - Emphasize on module functionality and performance
- Advantages: No test stubs are needed
- Errors in critical modules are found early
- Disadvantages: Test drivers are needed
- Interface errors are discovered late



Drivers and Stubs

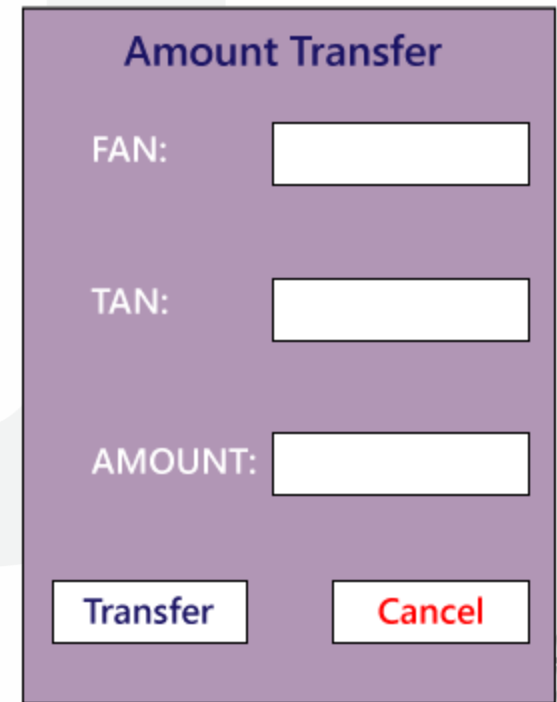
- Stubs and Drivers are the dummy programs in Integration testing used to facilitate the software testing activity.
- These programs act as substitutes for the missing models in the testing.
- They do not implement the entire programming logic of the software module.
- But they simulate data communication with the calling module while testing.
- Stub: Is called by the Module under Test.
- Driver: Calls the Module to be tested.



Example

Let us see one sample example of a banking application, as we can see in the below image of amount transfer.

- First, we will login as a user **P** to amount transfer and send Rs200 amount, the confirmation message should be displayed on the screen as amount transfer successfully.
- Now logout as P and login as user **Q** and go to amount balance page and check for a balance in that account = Present balance + Received Balance. So the integration test is successful.
- Also, we check if the amount of balance has reduced by Rs200 in P user account.

A purple-themed form titled "Amount Transfer". It contains three input fields: "FAN:" (First Account Number), "TAN:" (Transaction Amount Number), and "AMOUNT:". At the bottom, there are two buttons: "Transfer" and "Cancel".

Amount Transfer

FAN:

TAN:

AMOUNT:

Transfer **Cancel**



Validation Testing

- Validation testing is testing where tester performed functional and non-functional testing.
- Functional testing includes Unit Testing (UT), Integration Testing (IT) and System Testing (ST).
- Non-functional testing includes User acceptance testing (UAT).
- Validation testing is also known as dynamic testing, where we are ensuring that **"we have developed the product right."**
- It also checks that the software meets the business needs of the client.





Validation Testing

- As a tester, you need to evaluate if the test execution results comply with that mentioned in the requirements document.
- Any deviation should be reported immediately and that deviation is thus called a bug.
- Tools like HP quality Centre, Selenium, Appium, etc are used to perform validation test.
- we can store the test results there.
- A proper test plan, test execution runs, defect reports, reports & metrics are the important deliverables to be submitted.





Advantage of Validation Testing

- During verification if some defects are missed then during validation process it can be caught as failures.
- If during verification some specification is misunderstood and development had happened then during validation process.
- while executing that functionality the difference between the actual result and expected result can be understood.
- Validation is done during testing like unit testing, integration testing, system testing, load testing, compatibility testing, stress testing, etc.
- Validation helps in building the right product as per the customer's requirement and helps in satisfying their needs.





Alpha Testing

- It is done when product development is on the verge of completion.
- Slight changes in design can be made after conducting the alpha test.
- This testing methodology is performed in lab surroundings by the developers.
- Here developers see things in the software from users point and try to detect the problems.
- These testers are internal company or organization's employees or may be a part of the testing team.
- It is done early at the end of software development before beta testing.



How to do Alpha Testing

- Firstly need to first review the design specification and functional requirements.
- Then develop a comprehensive test plan and test cases.
- After that execute the test cases in order to find log defects and fix those defects.
- Finally retest once the issues are solved for smooth functioning of software.



Advantage of Alpha Testing

- One of the benefits of alpha testing is it reduces the delivery time of the project.
- It provides a complete test plan and test cases.
- Free the team member for another project.
- Every feedback helps to improve software quality.
- It provides a better observation of the software's reliability and accountability.





Beta Testing

- Beta Testing can be defined as the second stage of testing any product before release.
- A sample of the released product with minimum features and characteristics.
- It is being given to the intended audience for trying out or temporarily using the product.
- The beta test is being carried out by real users in the real environment.
- This allows the targeted customers to dive into the product's design, working, interface, functionality, etc.



Features of Beta Testing

- Beta testing used in a real environment at the user's site.
- Beta testing helps in providing the actual position of the quality.
- Testing performed by the client, stakeholder, and end-user.
- Beta testing always is done after the alpha testing, and before releasing it into the market.
- Beta testing is black-box testing.
- Beta testing performs in the absence of tester and the presence of real users
- Beta testing generally is done for testing software products like utilities, operating systems, and applications, etc.





Advantage of Beta Testing

- Beta testing offers numerous benefits to testers, software developer, as well as the users.
- In the assistance of this type of testing, it enables developers, testers to test the product before its release in the market.
- The Beta testing focuses on the customer's satisfaction.
- It helps to reduce the risk of product failure via user validations.
- Beta testing helps to get direct feedback from users.
- It helps to detect the defect and issues in the system, which is overlooked and undetected by the team of software testers.
- Beta testing helps the user to install, test, and send feedback regarding the developed software.





Disadvantage of Beta Testing

- Software engineer has no control over the process of the testing, as the users in the real-world environment perform it.
- This testing can be a time-consuming process and can delay the final release of the product.
- Beta testing does not test the functionality of the software in depth as software still in development.
- It is a waste of time and money to work on the feedback of the users who do not use the software themselves properly.



System Testing

- System Testing is a type of software testing that is performed on a complete integrated system
- It is used to evaluate the compliance of the system with the corresponding requirements.
- In this testing, integration testing passed components are taken as input.
- System testing detects defects within both the integrated units and the whole system.
- The result of system testing is the observed behaviour of a component or a system when it is tested.



System Testing

- System Testing is carried out on the whole system in the context of either system requirement specifications.
- It also checks for functional requirement specifications or in the context of both.
- System testing tests the design and behaviour of the system and also the expectations of the customer.
- System Testing is basically performed by a testing team that is independent of the development team.
- It has both functional and non-functional testing.





System Testing Example

- A car manufacturer does not produce the car as a whole car. Each component of the car is manufactured separately, like seats, steering, mirror, break, cable, engine, car frame, wheels etc.
- After manufacturing each item, it is tested independently whether it is working the way it is supposed to work and that is called Unit testing.
- Now, when each part is assembled with another part, that assembled combination is checked if assembling has not produced any side effect to the functionality of each component and whether both components are working together as expected and that is called integration testing.
- Once all the parts are assembled and the car is ready, it is not ready actually.

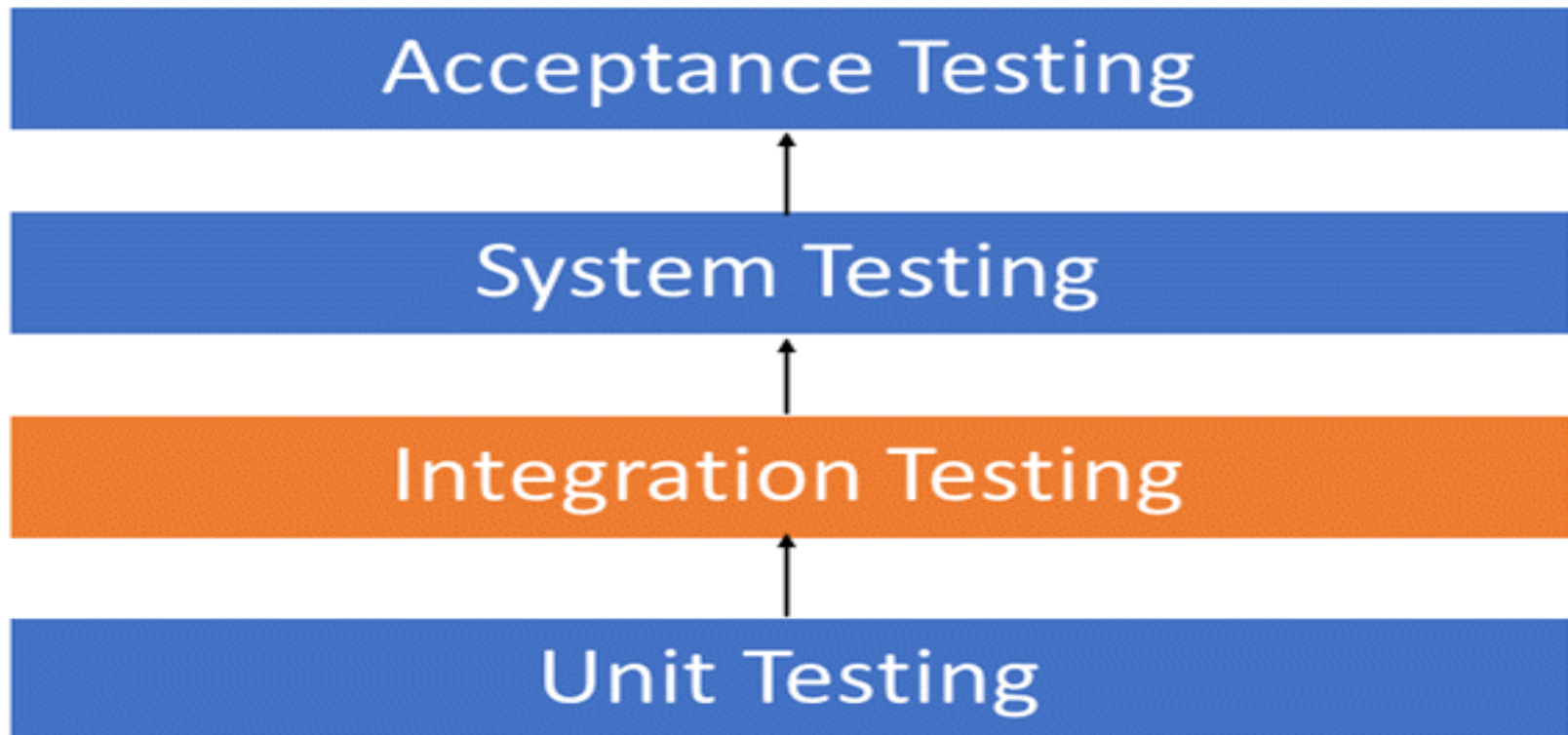


System Testing Example

- The whole car needs to be checked for different aspects as per the requirements defined like if car can be driven smoothly, breaks, gears, and other functionality working properly, car does not show any sign of tiredness after being driven for 2500 miles continuously, color of car is generally accepted and liked, car can be driven on any kind of roads like smooth and rough, sloppy and straight, etc and this whole effort of testing is called System Testing and it has nothing to do with integration testing.
- The example worked the way it was expected and the client was convinced about the efforts required for the system test.



Types of Testing



System Testing Process

Test Environment Setup:

Create testing environment for the better quality testing.

Create Test Case:

Generate test case for the testing process.

Create Test Data:

Generate the data that is to be tested.

Execute Test Case:

After the generation of the test case & test data, test cases are executed.

Defect Reporting:

Defects in the system are detected.

Log Defects:

Defects are fixed in this step.

Retest: If the test is not successful then again test is performed.



Recovery Testing

- Recovery testing a type of system testing which aims at testing whether a system can recover from failures or not.
- The technique involves failing the system and then verifying that the system recovery is performed properly.
- To ensure that a system is fault-tolerant and can recover well from failures, recovery testing is important to perform.
- A system is expected to recover from faults and resume its work within a pre-specified time period.
- Recovery testing is essential for any mission-critical system.



Recovery Testing

A system or software should be recovery tested for failures like :

- Power supply failure
- The external server is unreachable
- Wireless network signal loss
- Physical conditions
- The external device not responding
- The external device is not responding as expected, etc.





Security Testing

- Security testing is a type of software testing that intends to uncover vulnerabilities of the system.
- It determine that its data and resources are protected from possible intruders.
- It falls under non functional testing.
- Security testing is a testing technique to determine if an information system protects data and maintains functionality as intended.
- It also aims at verifying 6 basic principles such as:
- Confidentiality



Security Testing

- Integrity
- Authentication.
- Authorization.
- Availability
- Non-repudiation.





Security Testing

There are four main focus areas to be considered in security testing (Especially for web sites/applications):

- **Network security:** This involves looking for vulnerabilities in the network infrastructure.
- **System software security:** This involves assessing weaknesses in the various software the application depends on.
- **Client-side application security:** This deals with ensuring that the client cannot be manipulated.
- **Server-side application security:** This involves making sure that the server code and its technologies are robust enough to fend off any intrusion.



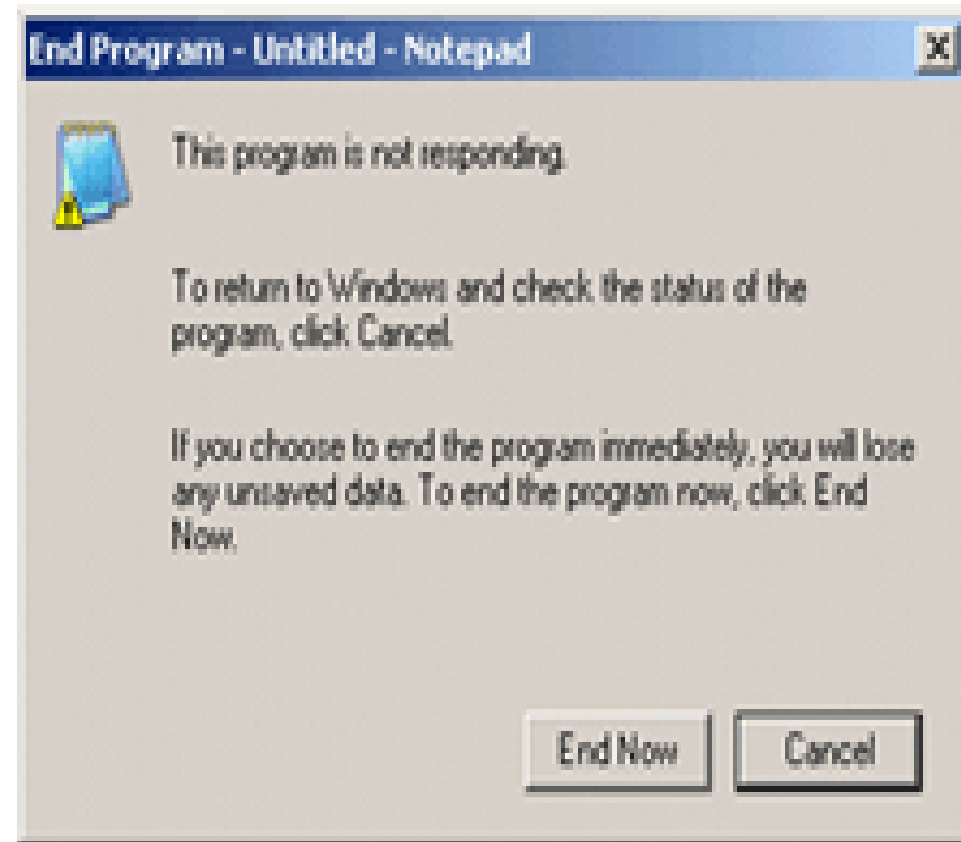
Stress Testing

- Stress Testing is a type of software testing that verifies stability & reliability of software application.
- The goal of Stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions.
- It ensures that software doesn't crash under crunch situations.
- It even tests beyond normal operating points and evaluates how software works under extreme conditions.
- Stress testing is to determine the limit, at which the system or software or hardware breaks.



Stress Testing

- It also checks whether the system demonstrates effective error management under extreme conditions.
- The application under testing will be stressed when 5GB data is copied from the website and pasted in notepad.
- Notepad is under stress and gives 'Not Responded' error.
- Eg: During festival time, an online shopping site may witness a spike in traffic.



Stress Testing

Stress testing is also extremely valuable for the following reasons:

- To check whether the system works under abnormal conditions.
- Displaying appropriate error message when the system is under stress.
- System failure under extreme conditions could result in enormous revenue loss.
- It is better to be prepared for extreme conditions by executing Stress Testing.



Performance Testing

- Performance Testing is a software testing process used for testing the speed, response time.
- Stability, reliability, scalability and resource usage of a software application under particular workload.
- The main purpose of performance testing is to identify and eliminate the performance bottlenecks in the software application.
- uncovers what needs to be improved before the product goes to market.
- Poor performance testing are likely to gain a bad reputation and fail to meet expected sales goals.





Types of Performance Testing

- **Load testing** - checks the application's ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live.
- **Stress testing** - involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of an application.
- **Endurance testing** - is done to make sure the software can handle the expected load over a long period of time.
- **Spike testing** - Tests the software's reaction to sudden large spikes in the load generated by users.
- **Volume testing** - Under Volume Testing large no. of. Data is populated in a database and the overall software system's behavior is monitored.
- **Scalability testing** - To determine the software application's effectiveness in "scaling up" to support an increase in user load.

capacity or size of some attributes of a product or system is the act of determining a measure.

is a quantitative measure of the degree to which a product or system possesses a given attribute.

Failure measure is the number of failures experienced during a given time interval.

Failure count is the way of recording such failures

- d per 

Metrics for Source Code

- These are measurements of the source code that make up all your software.
- Source code is the fundamental building block of which your software is made.
- Measuring it is key to making sure your code is high-caliber.
- The number of lines of code must be looked while measuring source code quality.
- Equally important is to track the percentage of comments within the code, which will tell you how much maintenance the program will require.



Development Metrics

- These metrics measure the custom software development process itself.
- Gather development metrics to look for ways to make your operations more efficient and reduce incidents of software errors.
- Measuring number of defects within the code and time to fix them tells you a lot about the development process itself
- Start by tallying up the number of defects that appear in the code and note the time it takes to fix them.
- If any defects have to be fixed multiple time then there might be a misunderstanding of requirements or a skills gap.



Software Testing Metrics

- Software Testing Metrics are the quantitative measures used to estimate the progress, quality, productivity and health of the software testing process.
- The goal of software testing metrics is to improve the efficiency and effectiveness in the software testing process.
- To help make better decisions for further testing process by providing reliable data about the testing process.
- A Metric defines in quantitative terms the degree to which a system, system component, or process possesses a given attribute.



Debugging Strategies

- The important technique to find and remove the number of errors or bugs or defects in a program is called Debugging.
- It involves identifying the bug, finding the source of the bug and correcting the problem to make the program error-free.
- This process should be done before releasing them into the market.
- The developer can locate the code error in the program and remove it using this process.
- It plays a vital role in the entire software development lifecycle.



Debugging Process

- The steps involved in this process are:
- Identifying the error
- Identifying the error location
- The developer can locate the code error in the program and remove it using this process.
- Retest the code to find if its get repeated. i.e the error that have encountered.
- It plays a vital role in the entire software development lifecycle.



References

- [1]. Roger Pressman, Software engineering- A practitioner's Approach, McGraw-Hill International Editions
- [2]. Yogesh Singh, Software Testing, Cambridge University Press.
- [3]. William E. Perry, Effective Methods for Software Testing, Second Edition, John Wiley & Sons
- [4]. Ron Paton, Software Testing, second edition, Pearson education.
- [5] Dohorthy Graham foundation of Software Testing, ISTQB Certification
- [6] Softwaretesting. Tutorialspoint.
<https://www.tutorialspoint.com/Software testing>



× DIGITAL LEARNING CONTENT

○



Parul[®] University



www.paruluniversity.ac.in

