Bharathidasan University

Centre for Differently Abled Persons

Tiruchirappalli - 620024

Programme Name	:	Bachelor of Computer Applications
Course Code	:	20UCA5CC7
Course Title	:	Software Engineering
Semester	:	V
Unit	:	V
Compiled by	:	Dr. M. Prabavathy Associate Professor Ms.G.MAYA PRAKASH

STRUCTURED CODING TECHNIQUES

STRUCTURED CODING TECHNIQUES

Structured coding techniques in software engineering refer to the process of designing and developing software using a structured approach to writing code.

1. Top-down design: This involves breaking down the problem into smaller, more manageable parts, and then designing and coding each part separately before integrating them into a larger system.

2. Stepwise refinement: This involves gradually refining the design and code of a system, starting with a general outline and adding detail and complexity as the development process progresses.

3. Modular programming:

- This involves breaking down a program into smaller, more manageable modules, each with its own well-defined inputs, outputs, and functionality.

4. Control structures:

- It is used to control the flow of execution within a program, such as loops, conditionals, and subroutines.

5. Data structures:

- It is used to organize and store data within a program, such as arrays, linked lists, and trees.

CODING STYLE

- Programmers have their own style of writing programs.
- Good programming style makes the code easy to understand, not only for other people but also for the programmer himself.

To simplify software maintenance

  - Readability implies clarity and understandability.
  - Programs have to be read to be maintained.
  - Good style can avoid common bugs and typos.
  - Good style can make mistakes clear.

STANDARD AND GUIDELINES

Coding standards are the set of guidelines or norms that are globally set or are set by various software development organizations

The following are the common coding standards and guidelines are:

1. Rules for global variables declaration

Define what type of variables can be declared as global and what cannot.

2. Naming Conventions

It is a set for declaring any sort of entity or variable within the software.

3. Use coding style that is easy to understand and is globally accepted

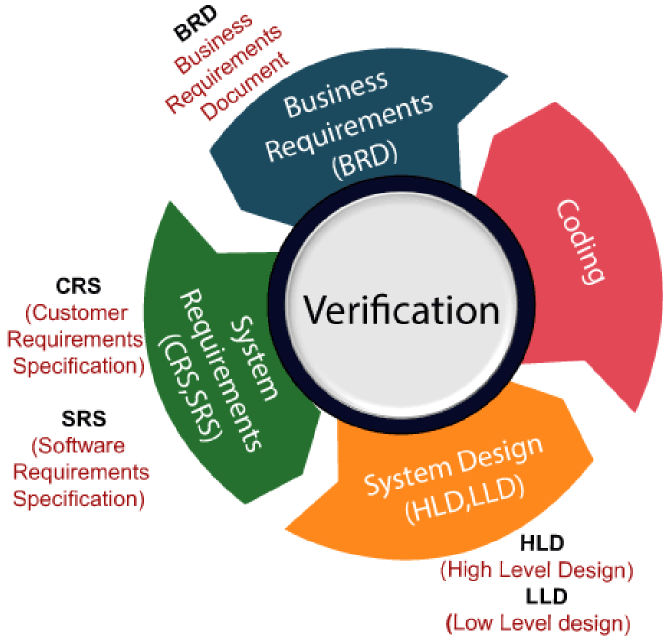
4. The length of functions or methods should be small

Keep the body of the functions small, nearly about 10-12 code lines.

5. Do not unnecessarily use statements that break the control flow of statements

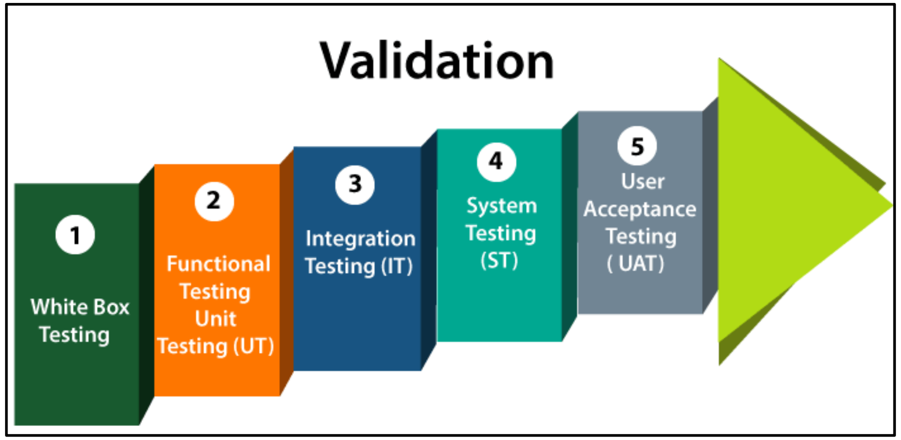
VERIFICATION

- Verification is a process of determining if the software is designed and developed as per the specified requirements.
- Verification: Are we building the product right?
- Verification is Static Testing.


BRD
Business Requirements Document
Business Requirements (BRD)
Coding
Verification
System Requirements (CRS,SRS)
CRS
(Customer Requirements Specification)
SRS
(Software Requirements Specification)
System Design (HLD,LLD)
HLD
(High Level Design)
LLD
(Low Level design)

VALIDATION

- Validation is the process of checking if the software (end product) has met the client's true needs and expectations
- Validation testing is testing where tester performed functional and non-functional testing.
- Validation: Are we building **the right product?**



Verification	Validation
We check whether the developed product is right.	We check whether we are developing the right product or not.
Verification is also known as static testing.	Validation is also known as dynamic testing.
Quality assurance comes under verification testing.	Quality control comes under validation testing.
Verification is done before the validation testing.	After verification testing, validation testing takes place.
In this type of testing, we can verify that the inputs follow the outputs or not.	In this type of testing, we can validate that the user accepts the product or not.

QUALITY ASSURANCE

QUALITY ASSURANCE

- Quality Assurance is also known as QA Testing.
- QA is defined as an activity to ensure that an organization is providing the best product or service to the customers.

The main objective of quality assurance is:

- Proof of fitness of product.
- Inspection of quality of product.
- Quality conformance.
- Customer satisfaction.

- This cycle for quality assurance consists of four steps:

Plan, Do, Check, and Act.

WALKTHROUGHS

- The walkthrough is a review meeting process but it is different from the Inspection.
- It does not involve any formal process i.e. it is a nonformal process

Advantages

- Incorporates multiple perspectives
- Fosters communication between the team

Disadvantages

- Areas where questions are not raised may go unnoticed
- Meetings can be time-consuming

INSPECTION

- It refers to peer review of any work product by trained individuals who look for defects using a well defined process.
- Inspection meetings can be held both physically and virtually.

Steps of Inspection

1. Planning - when the entry criteria for the inspection state are met.

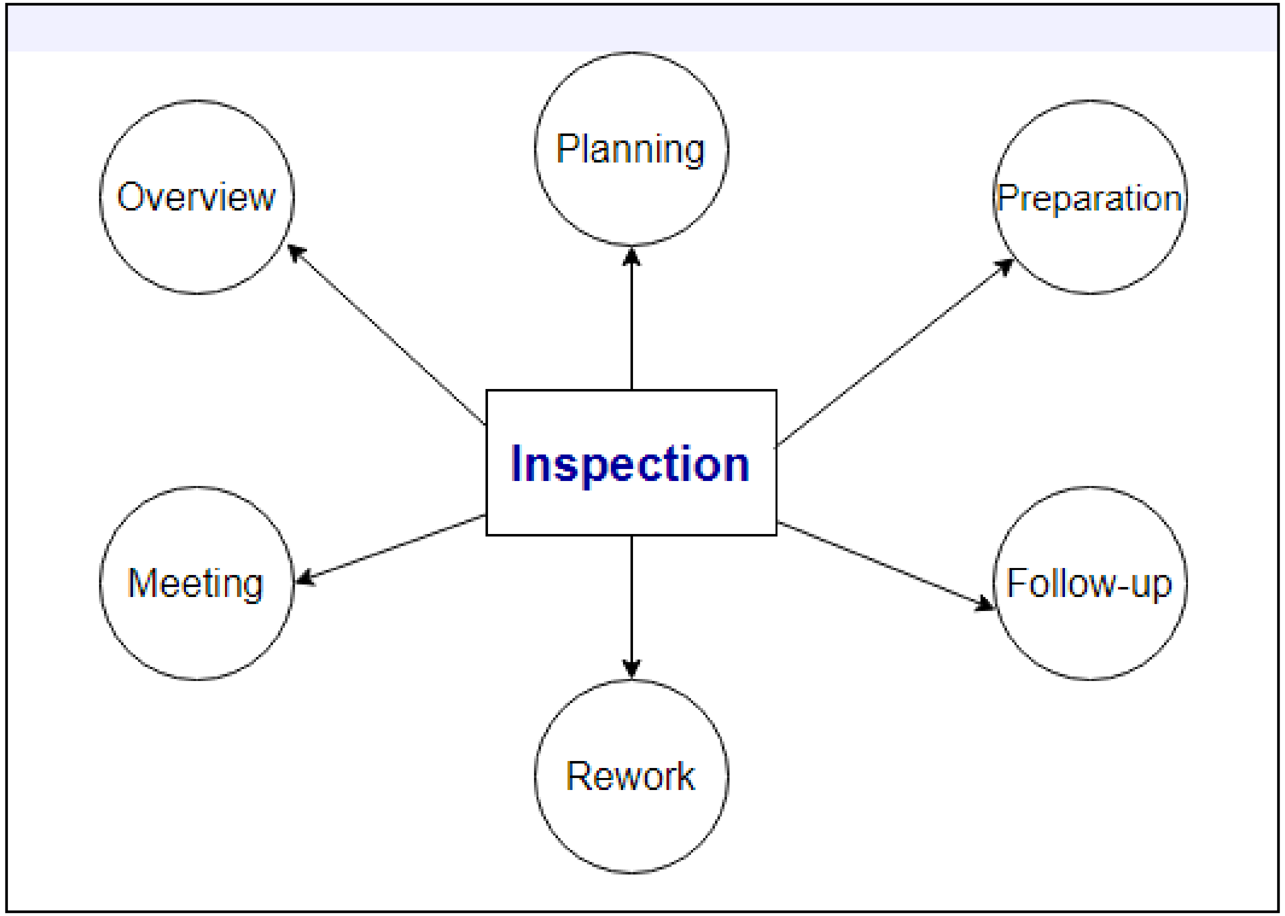
2. Overview - a presentation is given to the inspector with some background information needed to review the software product properly.

3. Preparation - The inspector collects all the materials needed for inspection.

4. Meeting - The moderator conducts the meeting and the defects are collected and reviewed.

5. Rework - To defect disposition determined at the meeting.

6. Follow-up - The moderator makes the corrections and then compiles the inspection management and defects summary report.


Planning
Overview
Preparation
Inspection
Meeting
Follow-up
Rework

DIFFERENCE BETWEEN INSPECTION AND WALKTHROUGH

Inspection	Walkthrough
Formal	Informal
Initiated by the project team	Initiated by the author
Planned meeting with fixed roles assigned to all the members involved	Unplanned.
Reader reads the product code. Everyone inspects it and comes up with defects.	Author reads the product code and his team mate comes up with defects or suggestions
Recorder records the defects	Author makes a note of defects and suggestions offered by team mate

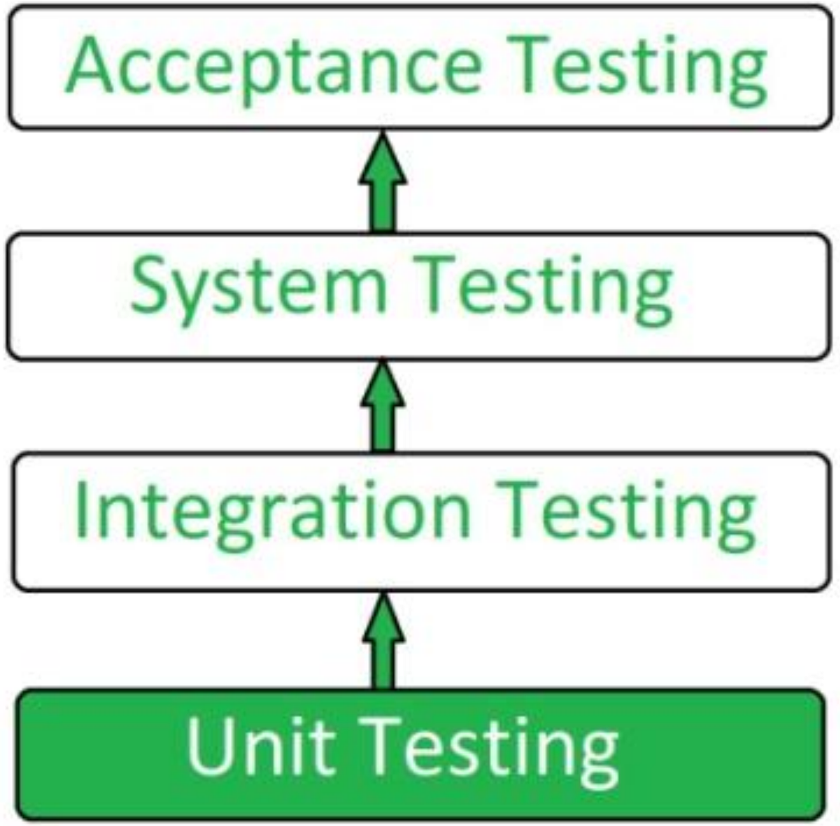
UNIT TESTING

UNIT TESTING

- Unit testing focuses on individual units or components of a software system.
- The purpose of unit testing is to validate that each unit of the software works as intended and meets the requirements.

Objective of Unit Testing:

- To isolate a section of code.
- To verify the correctness of the code.
- To test every function and procedure.
- To fix bugs early in the development cycle and to save costs.
- To help with code reuse.



Acceptance Testing

System Testing

Integration Testing

Unit Testing

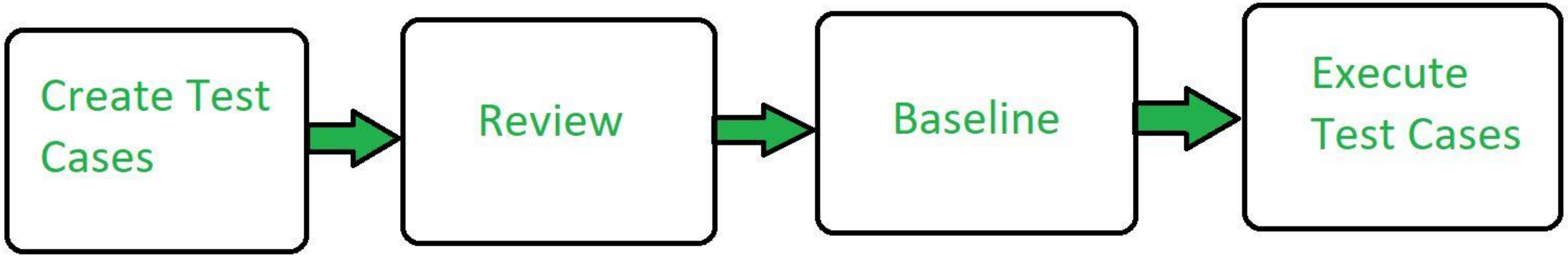
Types of Unit Testing:

There are 2 types of Unit Testing:

Manual

Automated.

Workflow of Unit Testing: Unit Testing Techniques



There are 3 types of Unit Testing Techniques. They are

1. Black Box Testing:

This testing technique is used in covering the unit tests for input, user interface, and output parts.

2. White Box Testing:

This technique is used in testing the functional behavior of the system by giving the input and checking the functionality output.

3. Gray Box Testing:

This technique is used in executing the relevant test cases, test methods, test functions, and analyzing the code performance for the modules.

DEBUGGING

- Debugging is the process of identifying and resolving errors, or bugs, in a software system.

There are several common methods and techniques used in debugging, including:

1.Code Inspection:

reviewing the source code of a software system to identify potential bugs or errors.

2. Debugging Tools:

Tools available for debugging such as debuggers, and profilers that to identify and resolve bugs.

3. Unit Testing:

Testing individual units or components of a software system to identify bugs or errors.

4. Integration Testing:

Interactions between different components of a software system to identify bugs or errors.

5. System Testing:

Testing the entire software system to identify bugs or errors.

6. Monitoring:

Monitoring a software system for unusual behavior or performance issues

7. Logging:

Recording events and messages related to the software system.

SYSTEM TESTING

SYSTEM TESTING

- System testing is a type of software testing that evaluates the overall functionality and performance of a complete and fully integrated software solution.
- It has both functional and non-functional testing.
- System Testing is a black-box testing.
- System Testing is performed after the integration testing and before the acceptance testing.

System Testing is performed in the following steps:

- **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 and the test data, test cases are executed.
- **Defect Reporting:** Defects in the system are detected.
- **Regression Testing:** It is carried out to test the side effects of the testing process.
- **Log Defects:** Defects are fixed in this step.
- **Retest:** If the test is not successful then again test is performed.

Types of System Testing

1. Performance Testing:

It carried out to test the speed, scalability, stability and reliability of the software product or application.

2. Load Testing:

It is carried out to determine the behavior of a system or software product under extreme load.

3. Stress Testing:

It performed to check the robustness of the system under the varying loads.

4. Scalability Testing:

It carried out to check the performance of a software application or system

FORMAL VERIFICATION

- Formal verification is the process of using automatic proof procedures to establish that a computer program will do what it's supposed to

- Formal verification helps confirm that your embedded system software models and code behave correctly.

SOFTWARE MAINTENANCE

- Software maintenance refers to the process of modifying and updating a software system after it has been delivered to the customer.

- The goal of software maintenance is to keep the software system working correctly, efficiently, and securely, and to ensure that it continues to meet the needs of the users

There are several key aspects of software maintenance, including:

Bug fixing: The process of finding and fixing errors and problems in the software.

Enhancements: The process of adding new features or improving existing features to meet the evolving needs of the users.

Performance optimization:

The process of improving the speed, efficiency, and reliability of the software.

Porting and migration:

The process of adapting the software to run on new hardware or software platforms.

Re-engineering:

The process of improving the design and architecture of the software to make it more maintainable and scalable.

Documentation:

The process of creating, updating, and maintaining the documentation for the software, including user manuals, technical specifications, and design documents.

Software Maintenance must be performed in order to:

- Correct faults.
- Improve the design.
- Implement enhancements.
- Migrate legacy software.
- Retire software.
- Requirement of user changes.
- Run the code fast.