# Unit -7(Verification and validation) covers

- Planning verification and validation
- Software inspections
- Verifiation and formal methods
- Critical system verification and validation

# Verification and Validation

# Verification and Validation

- Software testing is incomplete until it undergoes verification and validation processes.
- Verification and validation are the main elements of software testing workflow because they:

1. Ensure that the end product meets the design requirements.
2. Reduce the chances of defects and product failure.
3. Ensures that the product meets the quality standards and expectations of all stakeholders involved.

# Verification and Validation

- Verification is a process of determining if the software is designed and developed as per the specified requirements.
- Depending on the complexity and scope of the software application, the software testing team uses different methods of verification, including inspection, code reviews, technical reviews, and walkthroughs.
- Further, verification checks if the software team is building the product right.?

# Verification and Validation

- The main advantages of the verification are:

1. It acts as a quality gateway at every stage of the software development process.
2. It enables software teams to develop products that meet design specifications and customer needs.
3. It saves time by detecting the defects at the early stage of software development.
4. It reduces or eliminates defects that may arise at the later stage of the software development process.

# Verification and Validation

- **A walkthrough of verification of a mobile application**
- There are three phases in the verification testing of a mobile application development:

1. Requirements Verification
2. Design Verification
3. Code Verification

# Verification and Validation

**Requirements verification** is the process of verifying and confirming that the requirements are complete, clear, and correct. Before the mobile application goes for design, the testing team verifies business requirements or customer requirements for their correctness and completeness

**Design verification** is a process of checking if the design of the software meets the design specifications by providing evidence. Here, the testing team checks if layouts, prototypes, navigational charts, architectural designs, and database logical models of the mobile application meet the functional and non-functional requirements specifications.

# Verification and Validation

**Code verification** is a process of checking the code for its completeness, correctness, and consistency. Here, the testing team checks if construction artifacts such as source code, user interfaces, and database physical model of the mobile application meet the design specification.

# Verification and Validation

- Validation is the process of checking if the software (end product) has met the client’s true needs and expectations.
- Validation focuses only on the output; it does not concern itself about the internal processes and technical intricacies of the development process.
- Validation helps to determine if the software team has built the right product.

# Verification and Validation

The main advantages of validation processes are:

1. It ensures that the expectations of all stakeholders are fulfilled.
2. It enables software teams to take corrective action if there is a mismatch between the actual product and the anticipated product.
3. It improves the reliability of the end-product.

# Verification and Validation

- **A walkthrough of validation of a mobile application**
- Validation emphasizes checking the functionality, usability, and performance of the mobile application.
- **Functionality testing** checks if the mobile application is working as expected. For instance, while testing the functionality of a ticket-booking application, the testing team tries to validate it through:

1. Installing, running, and updating the application from distribution channels like Google Play and the App Store
2. Booking tickets in the real-time environment (fields testing)
3. Interruptions testing

# Verification and Validation

- **Usability testing** checks if the application offers a convenient browsing experience. User interface and navigations are validated based on various criteria which include satisfaction, efficiency, and effectiveness.
- **Performance testing** enables testers to validate the application by checking its reaction and speed under the specific workload. Software testing teams often use techniques such as load testing, stress testing, and volume testing to validate the performance of the mobile application.

# Verification and Validation

Main differences between verification and validation:

|  | Verification | validation |
|---|---|---|
| **Definition** | It is a process of checking if a product is developed as per the specifications. | It is a process of ensuring that the product meets the needs and expectations of stakeholders. |
| **What it tests or checks for** | It tests the requirements, architecture, design, and code of the software product. | It tests the usability, functionalities, and reliability of the end product. |
| **Coding requirement** | It does not require executing the code. | It emphasizes executing the code to test the usability and functionality of the end |

| | | |
|---|---|---|
| **ties include** | A few activities involved in verification testing are requirements verification, design verification, and code verification. | The commonly-used validation activities in software testing are usability testing, performance testing, system testing, security testing, and functionality testing. |
| **Types of testing methods** | A few verification methods are inspection, code review, desk-checking, and walkthroughs. | A few widely-used validation methods are black box testing, white box testing, integration testing, and acceptance testing. |
| **Teams or persons involved** | The quality assurance (QA) team would be engaged in the verification process. | The software testing team along with the QA team would be engaged in the validation process. |
| **Target of test** | It targets internal aspects such as requirements, design, software architecture, database, and code. | It targets the end product that is ready to be deployed. |

# PLANNING VERIFICATION AND VALIDATION

- The development of a V and V plan is essential to the success of a project.
- The plan must be developed early in the project.
- Careful planning is required to get the most out of testing and inspection process.
- Effects of v and v plan requires many considerations that are:

1. **Identification of V and V Goals:**

- V and V goals must be identified from the requirements and specifications.
- These goals must address those attributes of the product that correspond to its user expectations.

# Planning Verification and Validation

**2**. **Selection of V and V techiques:**

Specific techniques must be selected for each of the projects evolving products.

**3. Organizational Responsibilities:**

- The organizational structure of a project is a key planning considerations for project managers.
- An important aspect of this structure is a delegation of V and V activities to various organizations.

# Planning Verification and Validation

4**. Integrating V and V Approaches:**

- Once a set of V and V objectives has been identified, an overall integrated V and V approach must be determined.
- This approach involves the integration of techniques applicable to the various life cycle phases at a delegation of these tasks among the projects organizations Traditional intergrated V and V approaches have followed the “Waterfall model”.

**5. Problem tracking**

Software V and V plan to develop a mechanism for documenting problems:

- When the problem occurred
- Where the problem occurred
- Evidence of the problem
- Priority for solving problem

# SOFTWARE INSPECTIONS

- Inspection in software engineering refers to peer review of any work product by trained individuals who look for defects using a well-defined process.
- It is a manual, static technique that can be applied early in the development cycle.
- It is the most formal review type.
- It is led by the trained moderators.
- During inspection the documents are prepared and checked thoroughly by the reviewers before the meeting.
- It involves peers to examine the product.
- A separate preparation is carried out during which the product is examined and the defects are found.
- The defects found are documented in a logging list or issue log

# SOFTWARE INSPECTIONS

**The inspection process:**

- The inspection process should have entry criteria that determine if the inspection process is ready to begin.
- This prevents unfinished work products from entering the inspection process.

The stages in the inspections process are:

**Planning:** The inspection is planned by the moderator

**Overview meeting:** the author describes the background of the work process.

**Preparation:** each inspector examines the work product to identify possible defects.

# SOFTWARE INSPECTIONS

**Rework:** The author makes changes to the work product according to the action plans from the inspection meeting.

**Follow-up:** the changes by the author are checked to make sure everything is correct.

- The process is ended by the moderator when it satisfies some predefined exit criteria.
- The term inspection refers to one of the most important elements of the entire process that surrounds the execution and successful completion of a software engineering project.

# SOFTWARE INSPECTIONS

**Inspection team and role:**

During an inspection the following roles are used.

**Author**: The person who created the work product being inspected.

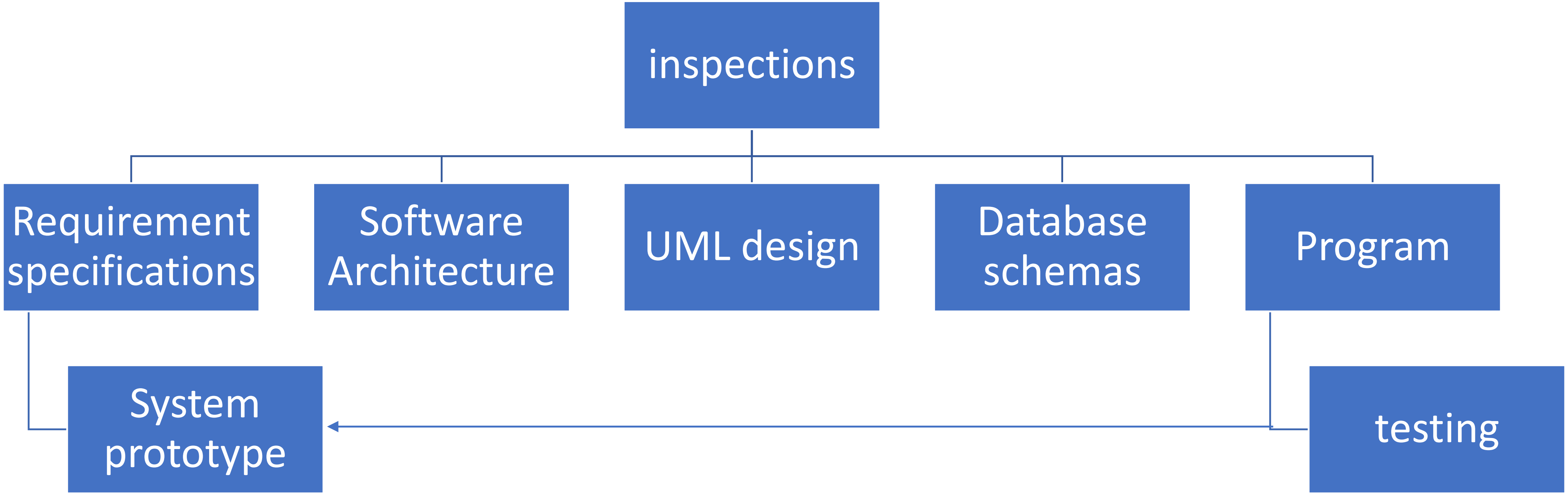
**Moderator:** This is the leader of the inspection. The moderator plans the inspection and coordinates it

**Reader**: The person reading through the documents, one item at a time. The other inspectors then point out defects.

**Recorder/scribe:** The person that documents the defects that are found during the inspection.

**Inspector:** The person that examines the work product to identify possible defects.

# Software inspections



Inspections and testing

# Verification and Formal Methods

- Formal verification is the act of proving or disproving the correctness of intended algorithms underlying a system with respect to a certain formal specification or property, using formal methods of mathematics.
- Helpful in proving the correctness of systems and such as cryptographic protocals, and combinational circuits.
- Verification of these systems is done by providing a formal proof on an abstract mathematical model of the system.
- Can be used when a mathematical specification of the system is produced.
- They are the ultimate static verification techniques.

# Verification and Formal Methods

**Arguments for FM**

- Producing a mathematical specification requires a detailed analysis of the requirements and this is likely to uncover errors.
- They can detect implementation errors before testing when a programs is analyzed alongside the specifications.

**Arguments against FM**

- Requires specialized notations that are not understood by domain experts.
- It is very expensive to develop a specification and even more expensive to show that a program meets that specification.
- It may be possible to reach the same level of confidence in a program more cheaply using other V and V techniques.

# Critical System Verification and Validation

Critical system verification and validation includes.

**Reliability validation:**

Exercising the programs to access whether or not it has reached the required level of reliability.

It typically includes testing and executing the system under different scenarios, inputs, and conditions to observe its behavior and identify any potential reliability issues or failures.

**Safety assurance:**

Concerned with establishing confidence labels in the system.

It involves analyzing and mitigating potential safety risks and hazards associated with the system's operation.

**Security assessment:**

Intended to demonstrate that the system can’t enter some state rather than demonstrate the system can do something.

It involves analyzing the system's architecture, design, and implementation to identify vulnerabilities, potential threats, and attack

**Safety and dependability cases:**

These are structured documents that set out detailed argument and evidence that a required level if safety or dependability has been achieved.

This may involve describing the system architecture, verification results, risk assessments, safety measures, and other relevant information.