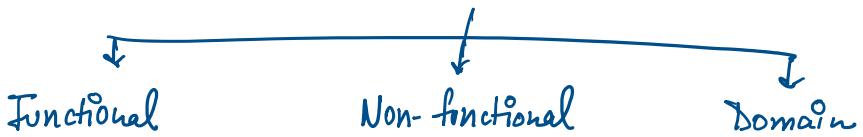


Module 3

17 November 2024 15:11

★ Software Requirements

- Detailed descriptions of the functionalities, constraints & goals of a software app.
- These requirements serve as the foundation for software development, guiding the design, implementation & testing phases to ensure the final product meets the need of stakeholders.



① Functional Requirements

- What the software should do. Functions & features a software must have.
- Basic functionalities that users expect from the software
- Examples E-commerce platform

User Authentication
Search Functionality
Cart Functionality

② Non-functional Requirements

- How the software performs a task rather than what it should do
- Quality attributes, performance criteria & constraints
- Examples : Performance

Usability
Reliability
Security

③ Domain Requirements

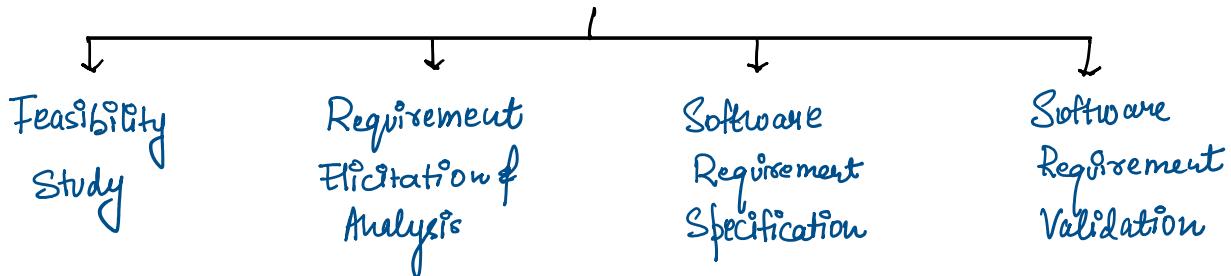
- Specific to the domain or industry in which the software operates
- Terminology, rules of standards relevant to particular domain

- Terminology, rules of standards relevant to particular domain
- Example: E-commerce (Various payment gateways like Paypal, Credit Cards, etc.)

★ Importance of Software Requirements

- Foundation of Development
- Prevent Scope Creep
- Basis for testing
- Improves communication
- Ensures Quality

★ Requirement Engineering (process of gathering & documenting Requirements)



① Feasibility Study

- Assessment of practicality & potential success of a proposed project
- Evaluates whether a project is feasible & worth pursuing
- Types: Technical Feasibility
 - Economic Feasibility (cost-benefit analysis)
 - Legal Feasibility (legal & regulatory requirements)
 - Schedule Feasibility (within desired timeframe)

② Requirement Elicitation & Analysis

→ Requirement Elicitation gathers requirements from stakeholders & other sources to understand what the software must do.

- How to gather?
- Techniques: Interviews, surveys, questionnaires, observation, etc.
 - Participants: Clients, end users, developers, etc.

- Participants : Clients, end users, developers, etc.
- Requirement Analysis Analyze & prioritise requirements, resolve conflicts & ensure feasibility.
- Requirement Classification (functional, nonfunctional, domain)
- Prioritization
- Conflict Resolution
- Modeling of Specification (data flow diagrams, ER diagrams, use case, state diagrams)
- Validation of Verification (walkthroughs, prototyping, inspections, review meetings)
- Requirement Traceability

③ Software Requirement Specification (SRS)

- Comprehensive document that outlines a software project's functional & non-functional requirements.
- Primary purpose : bridge the gap btw the client's expectation & development team's understanding.
- Guiding light ; clearly defining requirements
- Reference for designers, developers, stakeholders & testers; ensuring everyone is on the same page of working towards the same goal.
- Software Analyst ; translates requirements in ordinary language into technical terms.
- Software Requirement Phase Activity
 - Define an Idea
 - Brainstorming Phase
 - Feasibility Analysis
 - Precise Requirement Details
 - Research
 - Cost, Scope & Risk Management
 - Overall Planning
 - Collaboration & Communication
 - Requirement Documentation
 - Requirement Prioritization & Validation

- Requirement Prioritization & Validation
- Regular reviews of updates

Design Phase

- High level structured design document
- begins with a defined set of requirements ; results in detailed blueprint of how the software will be developed.
- transforming the software requirements gathered into structured design document
- Detailed blueprint ; architectural design, UI design & data design, databases, sketches, prototypes, etc.

→ Roles of Responsibilities

- Client (Stakeholder)
- Project Manager
- Business Analyst
- Technical Architect
- Software Developer
- QA Specialist

* for the next
phase -
development



Overall system architecture of how the system will be structured to meet the requirements

Detailed blueprint for development of each module & component

Development Phase

- Actual coding & building of the software application based on previously defined requirements & design specifications

→ Activities

activities required to build a software system

→ Activities

- Coding
- Integration
- Component Testing Phase
- Review of Revision
- Version Control
- Documentation

→ Key Principles

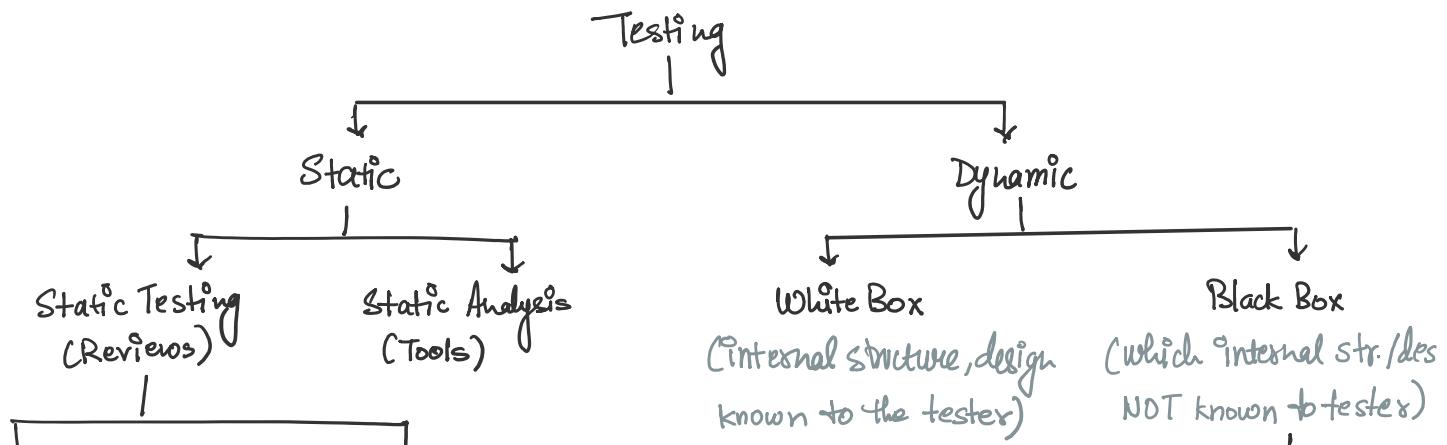
- Modularity of Scalability
- Efficient Problem Solving
- Collaboration of Communication
- Adherence to Requirements of Design
- Code Quality of Cleanliness

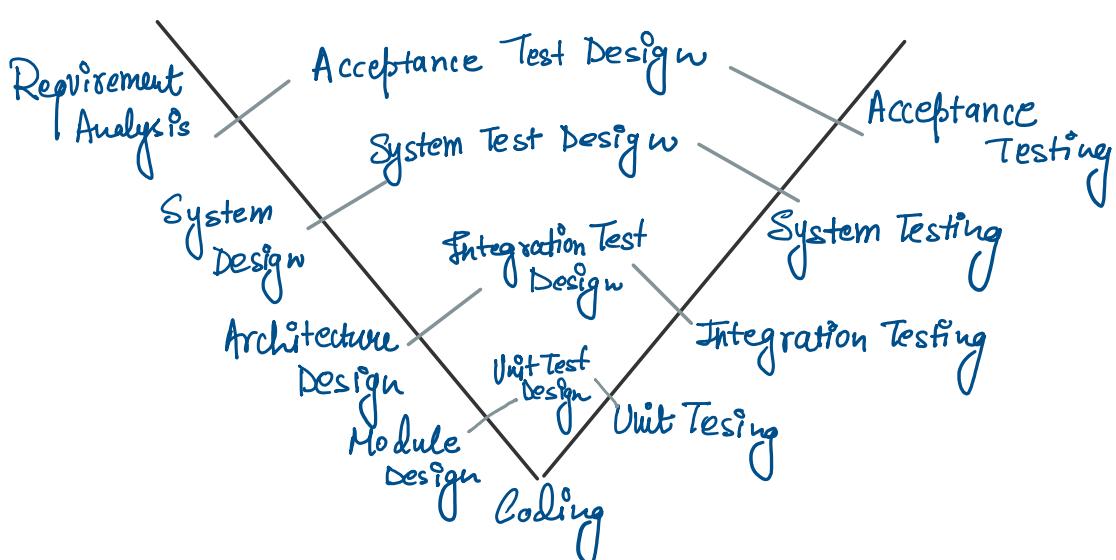
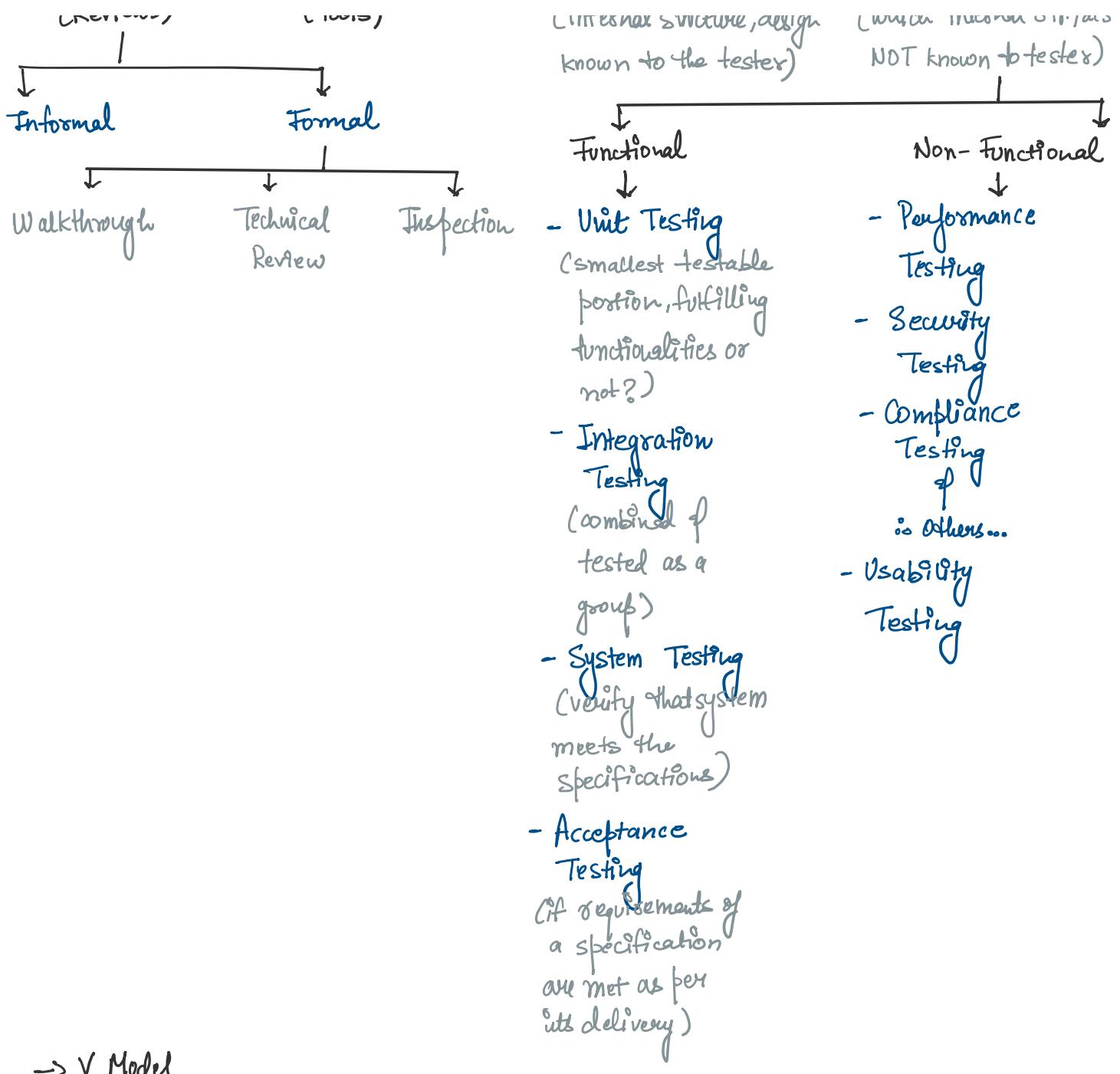
→ Roles of Responsibilities

- Software Developers / Engineers
- Lead Developer / Technical Lead
- Quality Assurance (QA) Engineers
- Project Manager / Scrum Master

Testing Phase

- maintains the quality of a software product
- finding flaws, mistakes, faults & reporting them to the creator so the program is error free & reliable





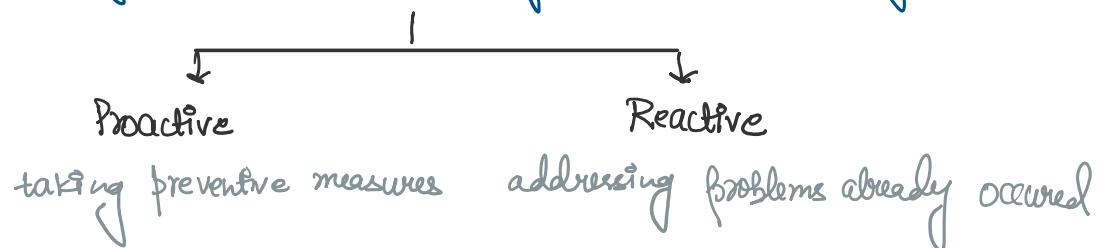
u Coding

→ Key Contributors in Testing -

- Test Lead
- Software Testers
- Test Environment Manager
- Test Manager

Maintenance Phase

- Altering & upgrading software after it has been released
- Fixing bugs, adding new features, improving performance, updating software



→ Key Aspects

- Bug fixing
- Enhancements
- Performance Optimization
- Porting & Migration

