**COURSE 2. SOFTWARE ENGINEERING**

**SOFTWARE ENGINEERING FUNDAMENTALS**

**Program:** A Computer program is a sequence of instructions written to perform a specified task on a computer.

**Software:** Software is a set of programs, procedures, and documentation concerned with the operation of a data processing system.

**Software process:**  A Process is a series of definable, repeatable, and measurable tasks leading to a useful result.

**Software Development**

Ad-hoc Software Development(till the 1960s):

Earlier there were no engineering disciplines in software development.

* The software was developed on a Trial & error basis.
* No specific process was followed during the development of the product. No testing was done.
* Defects were detected only after the product was delivered to the external Users.

**Software Engineering**

Software Engineering is the application of a systematic, disciplined, quantifiable approach to the design, development, operation, and maintenance of software.

An establishment and use of sound engineering principles in order to obtain economical software that is reliable and works efficiently on real machines.

**Software Crisis**

*(Difficulty to write correct program within the time element.)*

* Software fails to meet the user requirements.
* Software crashes frequently.
* The development of software became expensive.
* Difficult to alter, debug, and enhance the software.
* The software was often delivered late.
* Software used resources non-optimally.

**Process:**

A Process is a series of definable, repeatable, and measurable tasks leading to a useful result.

The **software development process** involves the transformation of user needs into an effective software solution.

* Value-added business activity and not just technical activity.

**SDLC - Software Development Life Cycle**

*(defines the tasks that need to be carried out at each step)*

*(SDLC ensures the software developed is of high quality)*

SDLC is the process used in a project to develop a software product.

It describes how the development activities are performed and how the development phases follow each other.

* How each phase performs and how they are connected to each other.

**Importance of SDLC**

*(What the customer had actually explained its the product as and what actually was the end product and what actually the customer wanted)*

* Responsible for the development team to develop the solution by bringing clarity and understanding to his needs.
* Only then ~ no deviations from expectations and end products.

**Planning:** Focus on determining the objective and estimation of resources for the project.

**Analysis:** Study and structure of the system requirements.

**Design:** Deliver the actual functionality

**Implementation:** Converting the design into a complete information system.

**A feasibility study** is to identify the issues that may be difficult or impossible to implement in terms of cost, time, and technology.

**Analysis:** All possible requirements of the system to be developed are captured and documented in a document called SRS(Software Requirement Specification).

**Design:** System design is prepared based on the SRS System design. Helps in specifying hardware and system requirements, and also helps in defining overall system architecture.

**Implementation:** Design is converted into a workable solution.

**Testing:** Begins once the coding or implementation phase is over. To find errors in software.

**Maintenance:** After the software is deployed in the client environment, the client can come back with changes or enhancements.

1. **Analysis**

*(The analyst is responsible for defining the requirements of the system without bothering about how these requirements will be built)*

*Main focus: The problem the customer is trying to solve.*

The goal of the system analysis is to define the requirements of the system.

Requirement gathering requires the client as well as the service provider to get the detailed and accurate requirements.

SRS(Software Requirement Specification): *Document produced at the end of the analysis phase.* Artifact of the analysis phase.

**Activities in Analysis Phase:**

* Requirements gathering and analysis
* Preparing Requirements Specification (SRS)

Project requirements are gathered and analyzed by the project managers and various stakeholders. (Who is going to use the system, how the system will be used, what has to be given as an input, and what are the outputs generated by the system)

Prepared SRS serves as a guideline for the next phase.

1. **Design**

*(Converts the requirements document into system specification)*

*(Focuses on identifying the architecture, the screens, components, and module interdependencies)*

*Various design artifacts are functional hierarchy, diagrams, screen layout diagrams, normalized tables, business process diagrams, and pseudocode)*

* Software design deals with transforming customer requirements into a set of documents that is suitable for implementation in a programming language.
* Process of defining the architecture, interface, component, and other characteristics of a system.
* The design stage takes the requirements identified in the approved requirements document(SRS) as its initial input.

**Levels of Design**

Decompose the entire project into units/modules and identify the system architecture, data structure, and processing logic.

DD (Design Document) = HLD(High Level Design) + LLD(Low Level Design)

**HLD:** Focus on what modules require, what each module performs, and how these modules communicate with one another. Architecture design, data design, and interface design are parts of HLD.

**LLD:** Focus on writing a detailed algorithm, module design and data design are parts of LLD.

1. **Construction (Code+Unit Testing)**

*(After the module is coded by the developer, they perform unit testing to ensure functional requirements.*

*The outcome of this phase are the source code, executable and databases applicable.)*

Modular and subsystem programming code will be accomplished during this stage.

**Unit testing/module testing** is done in this stage by the developers.

This stage produces the source, executable, and databases applicable.

1. **Testing**

*(Find Errors. Ensure that the technical and business requirements of the client are met)*

Testing is the process of executing the program with the intent of finding errors.

**Software testing** is a process of verifying and validating that a software application or program meets the business and technical requirements.

**Levels of Testing:** *(Code is tested at various levels)*

* Unit Testing
* Integration Testing
* System Testing

**Verification:** confirms that the software meets its technical specifications. (Ensures are you doing the product right)

**Validation:** Confirms that the software meets the business requirements. *(Ensures you are doing the right product)*

**Defect:** Variance between the expected and actual result.

*Verification is to check whether the software conforms to specifications.*

*Validation is to check whether the software meets the customer's expectations and requirements.*

**Levels of Testing**

**Unit Testing**(or module testing): Done by the developer. Every individual module or unit is tested to check its functional correctness.

**System Testing:** Integrated system is tested as a whole to check if the system is functionally and non-functionally correct.

**Integration Testing:** Done to check errors for the interface errors between the integrated components and finally.

**Acceptance test:** Done by the end user for the system acceptance.

1. **Maintenance**

Changes or enhancements happen everywhere. Software is no exemption. Any change that is made to the software after it is deployed is known as maintenance.

**SDLC Models**

1. Waterfall Model
2. V-model
3. Prototype Model
4. RAD (Rapid Application Development)
5. Incremental Model
6. Spiral Model

1. **Waterfall Model** (Liner sequential Model)

*( Model is well suited when requirements are clear and stable)*

*Each phase has a well-defined starting and ending point. Each phase produces a deliverable, which is given as input to the next phase.  Each phase has to be completed in order to start the next phase.*

The waterfall model derives its name due to the cascading effect from one phase to the other as depicted in the diagram.

Each phase has a well-defined start and end point, with identifiable deliverables to the next phase.

**Advantages**

* Simple and easy to use.
* Easy to manage due to the rigidity of the model - each phase has specific deliverables and a review process
* Phases are processed ad completed one at a time.
* Works well for smaller projects where requirements are very well understood.
* Linear approach
* Equivalent importance to all phases
* Contract-related issues can be addressed effectively.

**Limitations**

* This model is suitable if the requirements are well-defined and stable.
* The user gets a feel of the system only at the later stages of development.
* Backtracking cost is high in case of a problem.
* Increased development of time and cost.
* Systems must be defined upfront
* Rigidity
* Hard to estimate costs and project overruns.

Waterfall Model is best suited when:

* Requirements are clear and stable.
* The new version of the existing software system is created that is when we want to automate a manual process.
* The tools, technologies, and requirements are well known in advance and it is not going to change.

1. **V-Model**

*(The test documents are planned and developed in parallel at each level of the software development.)*

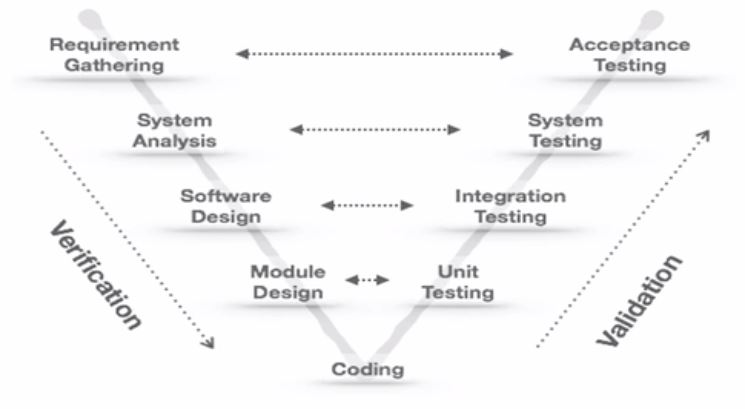
Verification and Validation Model - evolved from the waterfall model.

Each phase must be completed before the next phase begins.

Testing is emphasized in this model more than in the waterfall model.

It is a structured approach to testing.

Testing is done from the earlier stage thereby bringing high quality into the development of our products.



**Advantages:**

* Stage containment mechanisms identify problems that are existing in the product being developed before proceeding to the next stage.
* Ensures defects are not propagated to the next phase - the downward flow of defect
* V & V are done side by side.
* Defects found at an early stage - lower defect resolution cost
* Allows testers to be active from the initial state of the project’s lifecycle

**Limitations**

* Least flexible & rigid coz we can’t go back to the previous model.
* Backtracking cost is high in case of a problem.
* Increased development time and cost.
* Hard to estimate costs & project overruns.

1. **Prototype Model**

*(Concept of developing a model or sample for the system to be built)*

An incomplete version of the software program is being developed.

Simulates only a few aspects of the features of the system to be built.

 Prototyping can be used by end users to point out requirements missed by the developers.

Developers build prototypes during the requirement phase.

Prototype evaluated by the end users to provide corrective feedback.

Developers further refine the prototype based on feedback.

When a user is satisfied, the prototype code is brought up to the standards needed for the final product.

The process of prototyping involves:

* Identify basic requirements.
* Develop initial prototype
* Review
* Revise and enhance the prototype

Two types of prototyping:

* Throw away - used when requirements are unclear
* Evolutionary - used when requirements are unstable

**Advantages**

* Reduced time and cost
* Active user involvement
* The client gets the feel of the project earlier in project life cycle.
* Steady, visible signs of progress produced
* Awareness of additional needed functionality.

**Limitations**

* The user might get confused.
* Documentation is absent
* Expensive
* Developers' attachment to prototype

1. **RAD (Rapid Application Development)**

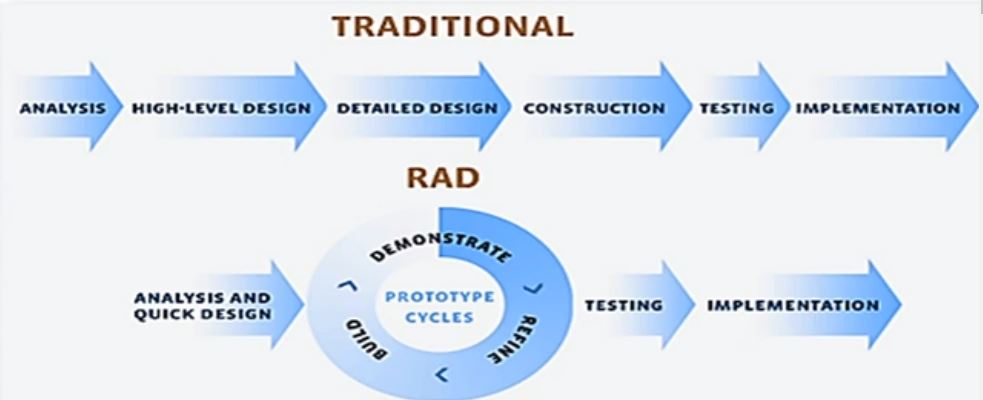
*(Parallely built)*

A high-speed version of the linear sequential model.

Characterized by a very short development life cycle

The RAD model follows a component-based approach.

Individual components are developed by different people and assembled to develop a large software system.



Phases of RAD:

* Business modeling: The information flow is identified between various modules
* Data modeling: Info gathered from business modeling to define data objects.
* Process modeling: Data object converted to achieve business objective
* Application generation: Automated tools are used to process models into the actual system.
* Testing and turnover: new components plus the interfaces are tested.

Advantages:

* Delivery of a fully functional project in a short period.
* Facilitates parallel development.

Limitations:

* Developers and clients must be committed to completing in time
* If either party is indifferent to needs, may run into serious problems.
* Not suitable for large project

1. **Incremental Model**

*Each model is broken into individual units called modules.*

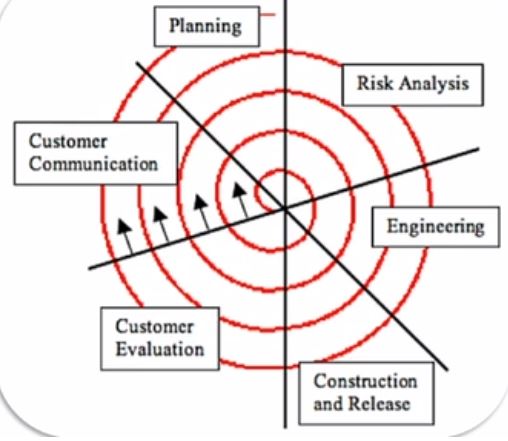
*Each module undergoes all SDLC phases.*

**Advantages:**

* Divide n conquer the breakdown of tasks.
* High-risk or major functions are addressed in the first increment cycles.
* Each release delivers an operational product.
* Customers can respond to each build.
* Customers get important functionality early.

1. **Spiral Model**

*Bested suited projects which are risky.*

**

Risk analysis: alternate solutions

Provides early indication of risk

Critical high-risk functions are developed first

**Limitations**

* Not for small projects
* Relies on risk assessment analysis.

**ETVX Model**

***(Entry-Task-Validate-Exit)***

**Entry** - Inputs required

**Task:** actions

**Validation:** quality needed

**Exit**

**TEST YOUR UNDERSTANDING**

1. Which conforms that the software meets its technical specifications? -  **VERIFICATION**
2. The software is put into operation in the client’s environment. The client comes back to enhance the UI to attract more customers. This phase is called as ­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **MAINTENANCE**
3. Match the appropriate usage of the SDLC process models, based on the nature of the requirements

When the requirements are unclear - **THROW AWAY PROTOTYPE**

When the requirements are unstable - **EVOLUTIONARY PROTOTYPE**

1. Match the phases of the Rapid Application Development model against their objectives

Application generation → Automated tools are used to convert process models into the actual system,

Business modeling → The information flow is identified between various modules, Testing and turnover → New components along with all the interfaces are tested,

Data modeling → Information gathered from business modeling is used to define data objects,

Process modeling → Data objects are converted to achieve the business objective

1. High-risk or major functions are addressed in the first cycles thereby delivering an operational product. Which of the SDLC process models achieves this? **Incremental model**
2. Beta Technologies has undertaken a collision avoidance system software to be implemented for airports. Additional safety measures have to be automated by warning pilots when another aircraft gets closer, otherwise, impacts are huge. Which of the following SDLC process models best suits the requirement? - **SPIRAL MODEL**
3. The student mark processing system software has been developed and deployed at the St. Peters university. The system shows the grade as 0 for all the students rather than the actual grade. Which phrase below was not done properly during the SDLC? - **SOFTWARE TESTING**
4. Which of the below is one of the phases of the prototype model - **QUICK DESIGN**

**POST QUIZ**

1. Which model emphasizes Validation and Verification at each level of stage containment? **V-MODEL**
2. Match the correct E, T, V, X (Entry, Task, Verify, Exit) criteria  for the below scenario

Reviewed design document → **Exit**,

Input containing the reviewed SRS → **Entry**,

Review of the artifacts → **Verification**,

Developing HLD and LLD Artifacts → **Task**

1. Scenario: LIC has manually carried out their process of premium collection procedure for the past 45 years. Now, they have employed the services of another company to automate the above.

Question: Which of the following models would you suggest to the outsourcing company? - **WATERFALL MODEL**

1. Testing performed by the user to ensure that the system meets the agreed upon quality attributes and the specification is called as…. **ACCEPTANCE TESTING**
2. Consider that you have to develop a flight control system. The system is simulated as such that the original system is working. There are many potential hazards with such a system. What model would you suggest to develop the system? **SPIRAL MODEL**
3. Identify the correct phases of the waterfall model -  **Analysis, Design, Coding, Testing, Operation, and Maintenance phase**

**REQUIREMENT ANALYSIS**

**PRE-QUIZ**

1. During which phase  the following activities are identified: Identifying the  major modules of the system, how these modules integrate, the architecture of the system, and describing the pseudocode for each of the identified modules - **Design**
2. The aesthetics of the website is part of the functional requirement. State true or false - **FALSE**
3. \_\_\_\_\_\_\_ describes how the development activities will be performed and how development phases follow each other.  - **Software Development Process**
4. \_\_\_\_\_\_\_\_\_ is the application of a systematic, disciplined, quantifiable approach to the design, development, operation, and maintenance of software. - **Software Engineering**
5. The requirement came to Allen Software company to develop software for military purposes. .00001 second delay in the missile launching software would create a greater loss of human life. What kind of model is best suited for this scenario - **Spiral model**

**INTRODUCTION**

**Types of requirements:**

* **Functional requirements:** behavior of the system (ex-pen to write)
* **Non-functional requirements:** quality attribute (ex-color of the pen)

**REQUIREMENT ANALYSIS IN DETAIL**

**Requirement Engineering:** Process of documenting and maintaining the requirements for a software project to be successful. This phase is very critical.

The requirements are documented in a standard document called **SRS (Software Requirement Specification).**

This document serves as a contract between the customer and the developer.

Requirement Engineering involves the following activities:

* Requirement Elicitation
* Requirement Analysis
* Requirement Documentation
* Requirement Review

1. **Requirement Elicitation**

Process of gathering requirements from the users, customers, and other stakeholders. It can be grouped into 3 categories:

* Address problems of scopes: the boundary of the system is ill-defined. requirements may address too less or too much info. Users may provide unnecessary info, thus making the objective unclear.
* Address problems of understanding: The user himself is not clear about his needs. May specify requirements that are ambiguous. The problem in communication. May omit imp requirements.
* Address problems of volatility (changing requirements): change over time. The user needs to evolve over time. Because of politics, and marketplace changes or the user might have missed some conditions.

1. **Requirement Analysis**

The process of studying and analyzing the customer and the user needs to arrive at a clear, consistent, unambiguous requirement understanding.

**Why is Requirement Analysis difficult?**

Misunderstanding between the customer and the analyst.

The analyst may not understand the domain requirements. The customer doesn't understand alternatives and trade-offs. - lack of proper communication - difficulty in making analysis.

Problem complexity: Inconsistencies, omission, inappropriate or unnecessary details in the problem statement.

**Goals of requirement analysis and specification phase:**

* Understand user’s requirements
* Remove Inconsistencies, anomalies, etc from requirements
* Document requirements properly in an SRS document.

**System Analyst** - a person who performs requirement analysis

* Understands user requirement
* Collects required data
* Prepares analysis document- SRS

**Activities of Requirements Analysis:**

* Requirements gathering
* Analysis of gathered requirements

**Requirements Gathering**

* Observation of existing systems
* Studying existing procedures
* Discussion with the customer and the end-users
* Having questionnaires to understand user requirements
* Conducting discussions with domain experts to understand the system,

**Requirements gathering for automating system**

An analyst can easily obtain:

* Input and output formats
* Accurate details of the operational procedures

**Analyzing the gathered requirement**

* Understanding the user requirements
* Detect and remove inconsistencies, ambiguities, and incompleteness

**Incompleteness and Inconsistencies**

* Resolves further discussions with the end-users and the customers.

**Contradicting Requirements:** One requirement conflicts with another requirement.

**Incomplete Requirement:** Some requirements have been omitted.

Requirement Analysis: what is to be built and not how it is to be built.

*Solutions,complexities,imp requirement*

1. **Requirement Documentation (SRS)**

Requirement documented in a format specific. No implementation details.

* The outcome of the analysis phase
* Info about the requirements gathered.
* Aim of requirements specification: Systematically organize and document requirements

***SRS*** *is the artifact produced at the end of the analysis phase.*

* Concentrates on what needs to be done,
* User needs
* Contract doc
* Reference doc
* Definition for implementation

**Properties of SRS**

* Concise
* Easy to change
* Specify what the system must do
* Consistent
* Complete
* Traceable
* Verifiable

SRS contains 3 parts:

* Functional requirements
* Non-Functional requirements
* Constraints on the system

**Functional requirements** that specify the input the task and the output.

**Non-functional requirements** specify the overall quality attributes and constraints to be imposed on the system

**Constraints** in the system indicate the restrictions

Functional Requirements describe the set of high-level requirements identified as customer requirements. An example of a functional requirement would be that a system must send an email whenever an order is placed.

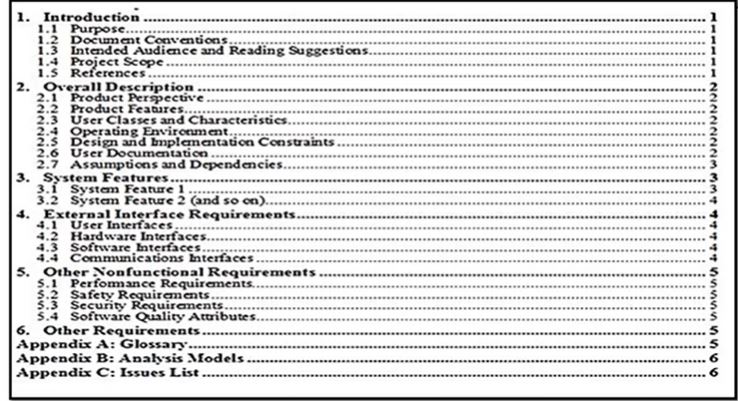
Constraints describe things that the system should or should not do.

Examples of constraints include

* standards compliance
* how fast the system can produce results
* Hardware to be used,
* Operating system
* DBMS to be used
* Capabilities of I/O devices
* Data representations

**SRS Document - IEEE 800.30**

* Introduction
* Function requirements
* Non-Functional requirements
  + External Interface requirements
  + Performance requirements
* Constraints



**ERD - Entity Relationship Diagram**

Database Design

* Represent conceptual level DB system
* Describe things and their relationships at high level

**Entity**: a business object that represents a group or category of data.

**Attribute**: Properties of an entity.

**Relationship**: specifies the relations among entities.

A relationship characterized by Cardinality and Optionality.

**Cardinality**: 1-1, 1-Many, Many-Many, Recursive

**Optionality**: Mandatory, Optional

**TEST YOUR UNDERSTANDING**

1. Which of the following options are the steps involved in Requirements Analysis?

**The requirements Gathering phase is followed by the Analysis of the gathered requirements**

1. Match the correct objectives of each phase involved in Requirements Engineering

**Requirements Specification** → Documents all the requirements properly,

**Requirements Analysis**  → Analyzing the customer and the user/stakeholder to arrive at a definition of software requirements,

**Requirements Elicitation**  → Gathering requirements from the users, customers, and other stakeholders

1. Identify the type of design that helps in transforming the data model created during the requirements analysis phase into the data structures that will be used to implement the software - Data Design
2. Match the objectives of the types of design involved

**Architecture design** → Defines the modules of the system and the functions that each module performs,

**Interface design**  → Describes how the software communicates with itself, and with the users interacting with the software,

**Low-Level Design**  → Focuses on writing a detailed algorithm

1. From the below options, identify the role of the system analyst - **Creates SRS**
2. Identify the correct statements from the below options. - **Analysis is performed followed by High-level design and then Low-level design**

**POST QUIZ**

1. What kind of nonfunctional requirement best suits the below scenario?  Whenever the new offers are published on the online shopping site, an SMS has to be sent to all the registered customers within 10 minutes of publishing. - **Performance Requirement**
2. Which of the following are available in SRS Document? -**Functional Requirements, Non-Functional Requirements, Constraints**
3. Consider the below scenario. A team has many players and the player belongs to one team. Identify the  cardinality between player and team - **M:1**
4. Choose the correct type of testing for the given Scenario

Testing to check if the card is swiped more than three times with the wrong pin, the card has to be blocked → **Security Testing**,

Testing to check if the report is generated within 5 sec on click of the button as agreed in the SRS → **Performance Testing**,

Testing to check if 1000000 users are accessing the website at the same point of time → **Load Testing,** Testing to test the GUI components on the screen → **Usability Testing**

1. Identify the possible entities from the given option - **Customer, Sale**
2. Whenever a new product arrives, the stock needs to be updated. This requirement is an example of a non-functional requirement. State true or False - **FALSE**

**BASICS OF TESTING**

**PRE-QUIZ**

1. An SRS has the following requirement. The stock exchange shall show the stock report for the next 24 hours. What is the issue with this requirement - **Ambiguous**
2. Client Comes to Allen Company for a Banking Solution. Who from the below options would be best suited to gather all the requirements correctly from the client? - **System Analyst**
3. The standard document that describes all the requirements of the system is called............. **Software Requirement Specification**
4. A good SRS should be \_\_\_\_\_\_, \_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_.  **Traceable, Consistent, Complete**
5. In the remote control Car application, in the step-by-step execution of the requirement described,  it is mentioned when the fuel level goes below the minimum level, the application should indicate the user in red color. In the output section of the same process, it is mentioned that the indicator will glow pink. What is the kind of requirement specified in SRS - **Contradicting**
6. Client Comes to Allen Company for a Banking Solution. Which phase of SDLC is best suited to gather what is expected from a client? - **Requirement analysis**

**INTRODUCTION**

The intention of testing is to find as many bugs.

Unit, Integration, and System testing.

**BASICS OF TESTING**

**Software Testing**

Process of executing the program with the intention of finding gaps and errors in the software.

Process of verification and validating of an application or a program so that it meets the technical requirements that guide the design and development artifacts and work as expected.

Testing is the process of evaluating, measuring, and rating a system or system component by manual or automated means to ensure that it satisfies specifies requirements and works as expected.

Else to identify the differences between the expected and actual results.

ST ensures correctness, completeness, and quality of the developed software - Product to be corrected before released to the end users.

Software testing needs planning, test needs specification, and once executed, they need results recording.

**Importance of ST**

* Building confidence in the most software systems
* Detect Faults - to improve quality and reliability

**Testing Profession**

* Understanding the application functionality and coding
* Testers are the one who certifies the software product as a quality and reliable product
* Testers destruct/disrupt the software to give better solutions

Testing is done by **Programmers and testers.**

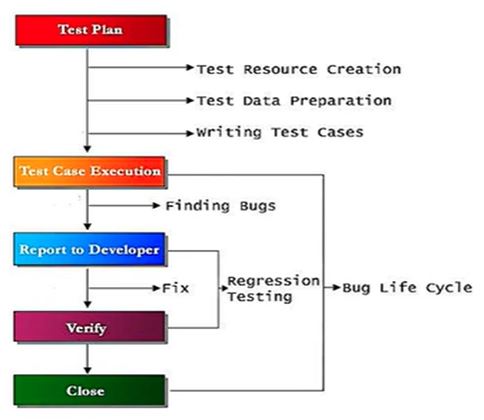
**Good Tester posses:**

* Creative thinking
* Questioning skills
* Never give attitude.

**Testing life cycle**

Software testing life cycle identifies what test activities to carry out and when to accomplish those test activities.

**Phases of Testing life cycle**



* **Test Plan:** How tests should be performed.
* Resource allocation, creation of test environment, test schedule & test functionality in test design,
* **Test Design**
* Test scenarios, test cases, test data, and test scripts are prepared
* **Test Execution**
* Executing the test scripts and finding bugs.
* **Report to Developer**
* Bugs are reported to the developer
* **Verify**
* Regression testing is later carried out to ensure that the bugs are fixed.
* **Close**
* Once the retest succeeds, then the defect is closed. Else life cycle continues

**Methodology of Testing**

* **Manual Testing**
* **Automated Testing**

Example:

* Java - JUnit
* Dot Net - N UNIT

* **Bugzilla -** Defect Tracking Tool

**Levels of Testing**

* Unit Testing
* Integration Testing
* System Testing
* Acceptance Testing

1. **Unit Testing**

Used to check whether a particular module is implementing its specification. Individual components of the software are tested to ensure that it works as expected.

Done by the developer. (In rare cases done by software testers)

The cost of correcting the errors during unit testing is much lesser than correcting it at higher levels.

1. **Integration Testing**

Individual modules that are unit tested are then integrated to perform integration testing.

Done to reveal interface errors between modules - integrated units.

The components are integrated in a iterative way, or all together called Big Bang.

Done ny developers or testers.

Integration Testing Approaches

* Top-down
* Bottom Up

**Top-down Integration:** Testing is conducted from the main module to the sub module step-by-step. First, the top-level modules are tested before the lower-level units.

If lower-level modules are not ready, then a dummy module which is a temporary program called **stub** is used instead of the actual module.

**Bottom-Up Integration:** Testing is done from the sub-module to the main module step-by-step. First, the Bottom level modules are tested before the upper-level units.

If upper-level modules are not ready, then a dummy module called **driver** is used to simulate the actual module.

1. **System Testing**

Performed on the complete integrated system from an end-to-end perspective.

Test conducted by testers to find if there are differences between implementation and specification.

Done by the testers independent of the development team to be unbiased.

**System testing involves:**

* **Function testing:** Each function of the application works as per the requirement specification.
* **Non - Function testing:** checks the performance, accessibility, reliability, and other non-function required of the system as specified in the specification document.

**Types of Performance Testing**

* **Stress Tests:**  tests the system beyond normal operational capacity and evaluates the result. Test the system with maximum stress and ensure that it doesn’t crash
* **Regression Tests:** done when the system is replacing an existing system. This replacement may be bcoz of defect fixing and rectifying the defect and due to enhancement of the original system.
* **Usability test:** Tests the extent to which the system is user-friendly. Done from the end-user perspective to know if the system is easily understandable, easy to use, and impressive to the user

1. **Acceptance Testing**

Done by the client to test whether the system meets the specified requirements and certify if it is acceptable for delivery to the customer.

Two types of acceptance testing:

* **Alpha testing**- Done by the client in the developer’s environment.
* **Beta Testing -** Done by the client in a real-world environment i.e in the client’s environment

**Types of Testing**

* Static
* Dynamic

1. **Static Testing**

Code of the software and the work product that is the associated documents are observed and tested manually to find errors.

Execution of the code is not performed. The sanity of the code is checked. - syntax check and manually review the code and the document to find errors.

Techniques:

* Review:
* Walkthrough
* Inspection

Member of static testing:

**Author**-writer of the doc under review,

**Moderator**-one who leads the review process,

**Reader**-one who presents the doc,

**Recorder/Scribe**-records each defect found and any suggestion or feedback given in the meeting for process improvement,

**Inspector**-inspects the document

**Review**

* Can happen in all phases of software development
* Conducted by individual/groups others than the one who developed it
* Formal or informal
* Formal is conducted and recorded as a document and used to improve the quality of the work product.

**Walkthrough**

* Conducted by the author of the document under review
* Mostly done on code.
* Explains doc step-by-step to the audience and asks for their comments and suggestions
* Only suggestions are given and not how to correct it

**Inspection**

* Done by the quality improvement team
* Inspectors check the artifacts against errors or violations in specifications at the end of the inspection.
* Can be accepted with minor changes or needs to be reworked completely

Artifacts for which reviews are performed are:

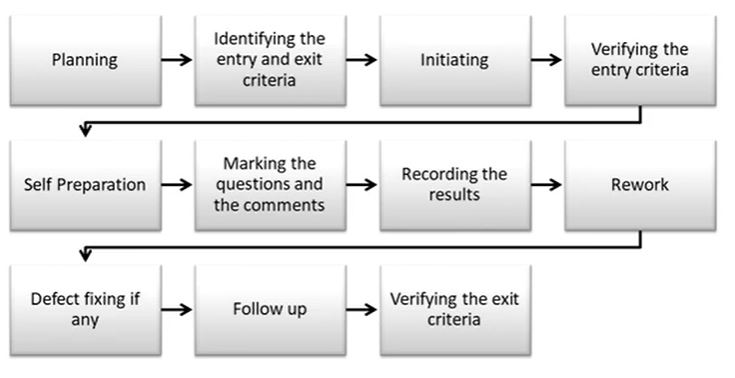
* SRS
* Design specification
* Source code
* Test plans
* Test specification
* Test cases
* Test scripts
* User guides
* Web pages

**Review Advantages:**

* Early defect detection & correction
* Fewer defects
* Helps in identifying omissions
* Save testing cost & time

**Review Process**

* Planning**:** People are identified and roles are assigned to each of the identified persons
* Identifying the entry and exit criteria :
* Initiating during the initiating kick-off - The document to be reviewed is distributed and the objectives are explained
* Verifying the entry criteria
* Self-preparation - individual work is done by each of the participants on their own
* Marking the questions and the comments
* Recording the results
* Rework
* Defect fixing if any
* Follow up
* Verifying the exit criteria - checking if the defects have been addressed

****

Types of review

* **Informal Review:** There is no formal process. Either the pair programming or the technical lead will review the design and the code.
* **Walkthrough:** meeting led by the author - Informal meeting. The author reads documents and his teammates come up with suggestions or defects.
* **Technical Review:** Formal meeting where the team of experts examines the technical quality of the artifact and identifies discrepancies from specifications and standards.
* **Inspection:** Led by a trained moderator who is not the author.  The moderators role is to do a pure examination of a doc. - Formal and driven by checklists and rules. This review process makes use of entry and exit criteria

1. **Dynamic Testing**

Techniques:

* Black Box testing
* White box testing

**Black Box testing**

Functionality of the software is tested without the knowledge of the internal implementation of the code.

Only input and the expected output are known. Also known as **specification-based testing.**

Techniques in Black Box

* Equivalence class partitioning
* Boundary value analysis
* Cause effect analysis
* Error guessing

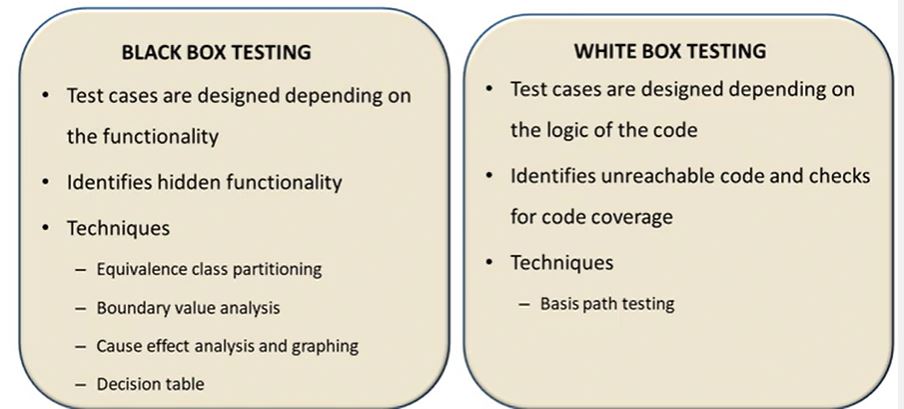
**White box testing(Glass/structural/clear box testing)**

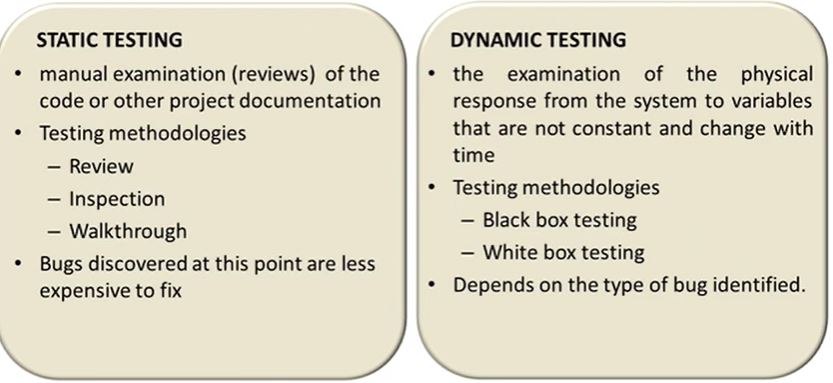
Test internal logic of code. Test cases should be designed

* **Basic path testing** : test case over every statement, condition, and loop.
* (determine independent parts and eliminate infeasible paths)

**Debugging**

Done by the developer to identify bugs and fix the defect.





**TEST YOUR UNDERSTANDING**

1. Identify this technique of dynamic testing where, For a range of input, three values are chosen, One value above the range, One value below the range, and One value within the range - **Equivalence partitioning**
2. What is the difference between the actual output of a software and the correct output?- **Error**
3. Match the objectives of the phases of the Software Testing Life Cycle

**Test Execution**  → Executing test scripts and Finding bugs,

**Test Design**→ Test scenarios, test cases, test data, and test scripts are prepared,

**Test Plan**  → Resource allocation, creation of test environment, test schedule and test functionality

1. Match the roles involved in Static Testing

**Author**   → Writer of the ‘document under review’,

**Inspector**  → Inspecting the document,

**Reader**   → Presents the document,

**Moderator**  → Leads the review process,

**Scribe**  → Records each defect found

1. What is the type of testing in which the tester will know about the input and the expected output details based on the specification document only but no knowledge of implementation? - **Black Box Testing**
2. Match the objectives against the techniques of generating test cases in black box testing

**Cause Effect Analysis**  → It is suitable for applications in which combinations of input conditions are few,

**Cause Effect Graphing** → The causes and effects represent the nodes,

**State Transition Diagram**  → Involves actions as one of its components

1. Which of the following options would basis path testing perform? -  **Statement Coverage,**

**Condition or Branch Coverage, Loop Coverage**

1. Identify the correct phases of the software testing life cycle. - **Requirements Analysis, Test Preparation, Test Case development, Test Environment Set up, Test Execution, Test Cycle closure**

**POST QUIZ**

1. The testing technique that deals with the internal logic and structure of the code is called \_\_\_\_\_\_\_\_. **WhiteBox Testing**
2. In the online shopping portal, for customer registration, the password field can accept only characters in the range of 5 to 25. Derive test cases using Boundary value analysis - **5,25,4,26**
3. Determine the cyclomatic complexity for the following code:

Accept year

if(year mod 4=0 and year mod 100!=0) or(year mod 400 =0)

print year is leap

else

print year is not leap

end if.

**The correct answer is: 4**

1. Boundary value analysis can only be used during white-box testing.  State if True or False. - **FALSE**
2. After the implementation of the Library management system, the tester identified that certain logic is redundantly rewritten by the developers, and the coding standards are violated in a few modules. What type of testing is carried out to identify these errors? - **Static Testing**
3. Walk through is performed by the trained moderator, whereas the Inspection is usually conducted by the author itself to record defects and deviations - **FALSE**
4. The tester is trying to test whether the values in the drop-down are listed properly. What type of testing does the tester perform in this scenario? - **Black box testing**

**SOFTWARE MAINTENANCE**

**PRE-QUIZ**

1. Two Modules ValidateCustomerID and RegisterCustomer are units tested Individually. What kind of testing should be done to check whether RegisterCustomer is calling ValidateCustomerID and ValidateCustomerID is called properly by RegisterCustomer? - **Integration Testing**
2. Which of the following techniques involves the step-by-step reading of the product, with each step checked against a predefined list of criteria? - **Inspection**
3. In the student mark processing system, the HOD wants an analytics report of student performance department-wise. On clicking the VIEW PERFORMANCE, the analytics should be displayed in 2 seconds. What testing should be done under this scenario - **Performance testing**
4. if(a>b && b>c){…}. Identify the Mc cabe's number for the given conditional construct - **3**
5. Allen Software company has designed an application for Alph Client. The application is designed in such a way that the student can either take the test normally and submit or for visual impairment persons the application converts their voice to text and submits the exam. What non-functional testing should be done in this scenario - **Usability testing**

**SOFTWARE MAINTENANCE**

Change to already completed software.

Software Phases:

* Development Phase
* Maintenance Phase

Types of Maintenance

* Adaptive
* Corrective
* Perfective

**Software Evolution**

It is impossible to produce a system of any size that does not need to be changed.

Parts of the software may have to be modified to correct the errors that are found in operation or to improve its performance or other non-functional characteristics.

The client may come with new requirements or changes to the existing requirements.

**Strategies for Software Change:**

* Software Maintenance
* Architectural Transformation
* Software re-engineering

**Changes may be due to:**

* New requirements emerge when the software is used
* The business environment changes
* Errors must be repaired.
* New equipment must be accommodated
* The performance or reliability may have to be improved.

**Software Maintenance:** Changes are made in response to clients' requirements for change but the fundamental software structure is stable. That is the process of modifying a software system or component after delivery to correct faults, improve performance or other attributes or adapt to a changing environment.

**Architectural Transformation:** The architecture of the system is modified. This change is from a centralized to a distributed architecture.

**Software re-engineering:** No new functionality is added to the system, but it is restructured and reorganized to facilitate future changes.

**Types of Maintenance**

* Corrective
* Adaptive
* Perfective
* Preventive

1. **Corrective Maintenance**

Maintenance that includes the repair of defects in an existing system

Defects can stem from

* Requirements specification errors
* Design errors
* About 80% of all problems stem from requirements and design
* Coding errors

For Example, the XYZ Software Company developed a Billing software to Z Department Store. When the biller, puts the bill he finds that for all the customer the bill amount was approximated to 10 decimal points whereas the actual requirement is for 2 decimal point. Customer comes back to XYZ to fix the existing defect. This is corrective maintenance

1. **Adaptive Maintenance**

Maintenance to adapt software to changes in the working environment.

Invoked by:

* Internal needs

External requirements. E.g. changes in law

For Example,

The Government recently changed the VAT rate to x%.  This change means that many organisations had to make alterations to their systems.

1. **Perfective Maintenance**

Making functional enhancements to system which can increase the system’s performance even when the changes have not been suggested by faults.

Includes all efforts to polish or refine the quality of the software or the documentation

Important that the improvement reduces the system maintenance costs.

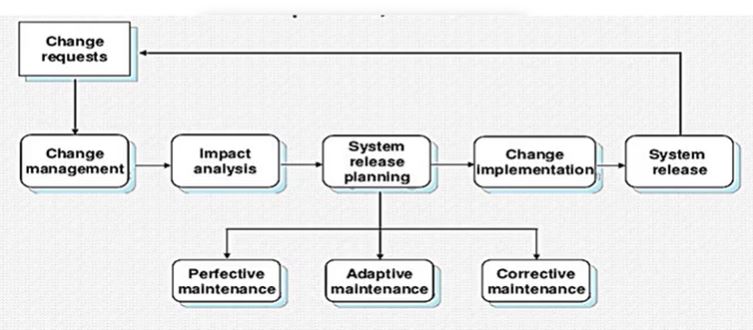
Example: A report is generated for the sales in the excel format.  In addition to it, for better understanding, a report is generated to represent the sales in graphical manner.

1. **Preventive Maintenance**

Changes made to the system to avoid any software fault in the future. Prevent problems before they occur.

An example is the Y2K problem

**Maintenance Process**



If the change is accepted then it moves to the system release planning where change requests is analyzed and mapped to a particular type of maintenance.

The changes are then implemented and released.

Maintenance costs are greater than the development cost((2\* to  100\* depending on the application)

It is affected by both technical and non-technical factors.

 cost increases as software are maintained.

the structure of the software will be corrupted in maintenance. So making further maintenance will be more difficult thereby increasing the cost.

Aging software can have high support costs (e.g. old languages, compilers, etc.).

Relative costs of maintenance

3% Requirements definition

3% Preliminary design

5% Detailed design

7% Implementation

15% Testing

67% Operations and Maintenance

The majority of a software budget in large companies is devoted to maintaining systems

A few Maintenance Examples are

* Y2K
* Anti-Virus Software - don't usually have to update software, but must send virus definitions
* Operating System Patching - Microsoft, Apple, Linux/Unix -OS is core to the use of a computer, so it must be constantly maintained

**TEST YOUR UNDERSTANDING**

1. Which maintenance includes all efforts to refine the quality of the software or the document? - **Perfective Maintenance**
2. Match the scenarios against the types of Maintenance

The UI of the managetrip.com is refined to include more options for Booking tickets and venues → **Perfective Maintenance**,

Client comes back for a change to be made in the Billing report getting generated due to incorrect rounding of amount → **Corrective Maintenance**,

The fare details are updated in the Railway reservation software to reflect the increase in fare → **Adaptive Maintenance**

1. Identify the correct facts about Maintenance

**It is affected by both technical and non technical factors,**

**Aging software can have high support costs**

1. Match the terms involved in Maintenance and its objectives

**Software re-engineering**  → New functionalities are not added to the system but it is restructured and reorganized,

**Architectural transformation**  → The system is modified from a centralized to a distributed platform,

**Software maintenance**  → Changes are made in response to changed requirements but the fundamental software structure is stable

1. Any change/enhancement performed on the developed software is called as - **Maintenance**

**POST QUIZ**

1. Help option provided in the online shopping portal has minimal inputs about the application navigation. Client wants Allen software company to include advanced help options to better serve their customers. What kind of maintenance best suits the above scenario - P**erfective Maintenance**
2. The customer comes back to Allen software company stating that when more than 'n' users simultaneously hit the website at the same time for customer registration , the application failed to generate the customerid correctly.What kind of maintanence best suit the given scenario - **Corrective Maintenance**
3. When a system is restructured and reorganized to facilitate future changes, it is called as \_\_\_\_\_\_\_\_\_. **Software re-engineering**
4. RBI has slashed the interest rate from 8% to 7% on all the loans that banks offer. What kind of maintanence has to be carried out on the banking applications to accommodate this change - **Adaptive Maintenance**
5. State true or false. Well-documented and well-structured software is difficult to maintain - **FALSE**

**CONFIGURATION MANAGEMENT AND VERSION CONTROL**

**PRE-QUIZ**

1. Software maintenance for the change of the platform is an example for **Adaptive maintenance.**
2. In an online shopping application, during customer registration the customer was made to enter his city in a text box. As the site became popular for online shopping, the client came back to include autocomplete feature in the city field to improve user friendliness. What maintenance needs to be carried out in this scenario? - **Perfective**
3. Client has developed an application that allows each of their customers to store 2TB of data. As the number of Customers are increasing client feels the storage space has to be increased for smooth operations to its customers. What type of maintanence is this? - **Preventive Maintanence**
4. Y2K problem is an example for ----**Preventive maintenance**
5. In Software maintenance, changes are implemented by modifying existing components and adding new components to the system. State if True or False. - **True**
6. Any changes done to the software during the operational phase of the software before project wind up is called as maintenance.  State if True or False. - **FALSE**
7. Client wanted to add  a new feature to his existing application  "Discount Offers" for all the existing customers. Whenever a new product comes to the supermarket,  their customer's should be intimated with the week day offer. What kind of maintenance is this? - **Perfective Maintanence**

**SOFTWARE CONFIGURATION MANAGEMENT**

**Configuration:**  The functional and physical characteristics of a hardware or software as set forth in technical documentation or achieved in a product.

Involves:

**Configuration Identification:** Define the product and its configuration documentation identification. Each item with a unique identifier.

**Change Management:** tracking and Controlling changes during the development of a product and its configuration documentation

**Configuration Status Accounting:** Provide status and information about a product and its configuration documentation.

**Configuration Audit:** Responsible for reviewing the items against various specifications for assessing their quality and correctness.

**Configuration Management:** Process of coordinating the software development by tracking, modifying, and controlling changes to the software.

Importance:

New versions of software systems are created as they get changed:

* For different machines/OS
* Different functionalities not present in the previous version
* To adapt for particular user requirements.

Responsible for managing and controlling the costs and efforts involved in making changes to a system.

**SCM defines**

* the type of documents to be managed.
* Who takes responsibility for CM procedures and the creation of baselines?

*(A baseline is a formally reviewed and approved document by the management)*

* Policies for change control and version control.
* CM records must be maintained.
* Describes the tools that should be used to assist the CM process and any limitations on their use.

**Effectiveness of SCM**

Software entities that SCM is expected to manage to include:

* Plans
* Specification(SRS,design)
* User Documentation
* Test Data
* Support Software Tools, Source Code, Executable, and Libraries

SCM is said to be effective when:

* Every work product can be accounted for
* Every work product or change made to it can be tracked and controlled

Also aggregates hardware components such as CPU hard disks to be under control.

Name and organize using the Object Oriented approach.

* **Base Object:** unit of text. Created by a SE during analysis, design, code, or text
* **Aggregate Object: a** collection of basic objects and other aggregate objects.

Ex: design specification, Data Design, Architectural Design, Module Design, Interface Design

**The double-headed arrow** between source code component test specification and design specification specifies the interrelationships.

If a change were made to the source code object, the **arrow** indicates the software engineer to determine what other objects (and SCIs) might be affected

**Baseline**

Any document that is formally reviewed and agreed upon by a QA team can be changed only through formal change control procedures.

Examples of baselines are

* The API has been completely defined; the bodies of the methods are empty
* All data access methods are implemented and tested
* The GUI is implemented.

**Evolution graph**

describes the change history of an object.

**Configuration Repository**

Database is managed by the server that stores all the configuration management information by maintaining all info in a centralized repository.

Helps to understand who has the particular system version.

Queries like above:

* Who has a particular system version?
* What platform is required for a particular version?
* What versions are affected by a change to component X?
* How many reported faults in version T

The Configuration items available in the Configuration Management system will be in the read-only mode by default

**Check in -** An operation used to make a developer’s object version available to other users.

**Check out** - A process that creates a new version of an object from an existing version stored in the database.

Developers check out objects so they can work on them.

Change occurs at any time during software development and change management is the systematic approach to managing change.

**Activities in Change Management:**

* Filtering changes
* Managing changes and the change process
* Reviewing and closing of Requests for Change (RFCs)
* Management reporting and providing management information

**Synchronization control**

When two people in a group try to access the same file and make changes then one would overwrite the other.

This can be solved using the **Synchronization control mechanism of Configuration management.**

It ensures that parallel changes, performed by two different people don’t overwrite one another.

To make this possible, he can obtain a lock on the object in the database.

This will disable others from accessing or updating that object until the currently checked-out version has been replaced thereby releasing the lock.

**Change Control Board (CCB)**

When a project is in the development process, the client may come up with changes.

These changes cannot be directly reported to the concerned person in the development team.

All these change requests will be sent to the **Change Control Board (CCB) or Software Change Control Board (SCCB).**

A committee that accepts the change requests, analyzes them, and takes the decision on whether or not a proposed change to a software project should be implemented.

* Composed of project stakeholders or their representatives.
* The authority of the change control board may vary from project to project. The decision taken by this board will be accepted as final and binding.
* If accepted, the code will be given for the developer to make the relevant changes, else it will be intimated to the client that it is rejected.

**The change control process comprises of three phases :**

* Evaluation phase,
* Resolution phase
* Verification phase.

When a **change is to be made it is identified** and submitted by the client to the Change control board.  It is then evaluated by the CCB.

They analyze the impact of the change and the cost also.

If it's feasible, then they approve the change, mark the change request as opened, and handover it over to the development team.

They plan for the change and implement it and **test for its correctness.**

The **tester** verifies if the changes are implemented correctly.  If yes, the verification process is approved and the change request is closed, if not they reopen the change request and give it back to the dev team.

**Version management:** To maintain the various versions of the file

The version control mechanism can be used to maintain multiple versions of a software project or files or documents.

* tracking the changes is easier.
* providing a lock to that file.
* saved in the repository
* accessed by anyone in the group
* read-only or for making changes.

**Benefits of version management**

* **Automatic backup** is the main advantage of version management.
* If a file is changed and we want to roll back to the previous version, it's possible using this.
* If more than one person works on the same file, the changes can be recorded without any conflict.  To highlight the differences, a comparison of the two versions of a file is possible.
* Possible to lock a file so that when it is modified by a user it cannot be accessed by another user, thus forcing serialized change to any given file.
* When a person in the team whats his code to work in isolation, it's also possible by creating branches.
* Maintain an instant audit trail on each and every file: versions, modified date, modifier, and any additional amount of meta-data the system provides for and whichever we choose to implement.

**TEST YOUR UNDERSTANDING**

1. Which of the following options are valid for the relationship between the configuration objects?

**A curved arrow indicates a compositional relation,**

**A double-headed straight arrow indicates an interrelationship**

1. Match the following facts about Version Management

Lock  a file   → **Serialized changes to file.,**

When a member of the team wants his code to work in isolation  → **Create branches,**

If a file is changed and we want to roll back to the previous version  → **Automatic backup**

1. Version Control allows users to lock files so they can only be edited by one person at a time and track changes to files - **True**
2. Which of the below option adhere that the changes to the software are introduced in a controlled manner - **Change control process**
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is responsible for reviewing the items against various specifications for assessing its quality and correctness - **Configuration audit**

**POST QUIZ**

1. John bought a new Laptop with a high-end configuration. To protect his laptop and the applications installed the installed antivirus software. This is an example of **Maintenance**
2. Version Management allows parallel concurrent development. State True or False. - **True**
3. The entry door of the Server room inside the company can be considered equivalent to the concept in **configuration management.**
4. Match the correct option

The process that ensures different versions of the project is managed → **Configuration Management,**

Which authenticates that the change proposed is valid → **Change Control Board,**

The process that ensures that changes made are recorded and controlled → **Change Management,**

The standard document where the requester fills the change in the change management process → **Change Request Form**

1. State true or false. Automated tools are available in the market, for managing change and versioning the software - **TRUE**
2. From the options identify the features that are part of the software configuration management - **Version management, Concurrency control, Synchronisation control**
3. Which of the following describes the change history of an object? - **Evolution graph**
4. \_\_\_\_\_\_\_\_\_ is a committee that makes decisions regarding whether or not proposed changes to a software project can be incorporated. - **Change Control Board**

**HANDS-ON**

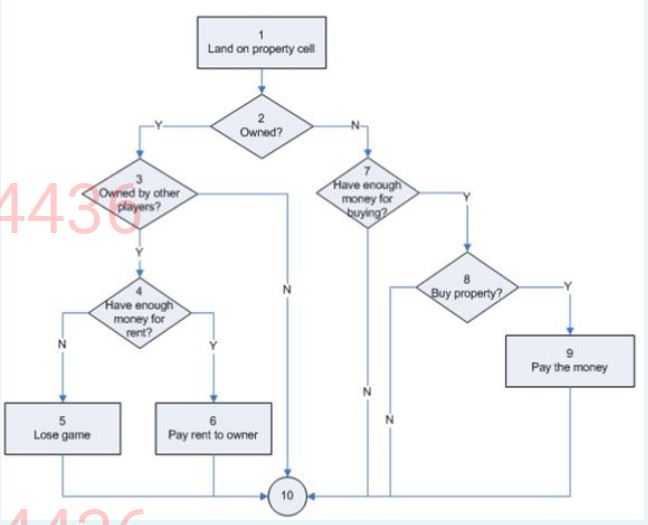
[**Hands-On: Attempt review (tekstac.com)**](https://accenturelearning.tekstac.com/mod/quiz/review.php?attempt=889571&cmid=864)

1. Requirement:

If a player lands on a property owned by other players, he or she needs to pay the rent. If the player does not have enough money, he or she is out of the game. If the property is not owned by any players, and the player has enough money to buy the property, he or she may buy the property at the price associated with the property

Find the mc cab's number for the following flowchart -

**Ans: 6**



1. A retail shop has provided a discount sale on their products. A module for calculating a discount for the total amount of purchase is decided by the following factors

a)      below 5000 then no discount

b)      5000 to 20000 then 5% discount

c)      20001 to 50000 then 10 % discount

d)     above 50000 than 15% discount

Choose from the below appropriate test cases using equivalence partitioning for the above scenario

**Input    Expected Output**

**2000        No Discount**

**7000         5%**

**30000      10%**

**60000      15%**

1. Find the mc cab's number for the below code.

          if code is blank or not in database

                     display "reenter code"

         else

                     if no credit and amount < 500

                                 display "credut not available"

                     else

                                 display "credit passed"

                     end if

         end if

**Ans: 5**

1. A retail shop has provided a discount sale on their products. A module for calculating discount for the total amount of purchase is decided by the following factors

a)      below 5000 then no discount

b)      5000 to 20000 then 5% discount

c)      20001 to 50000 then 10 % discount

d)     above 50000 then 15% discount

Choose from the below appropriate test cases using boundary value analysis for the above scenario

**Input   Expected Output**

**4999 No Discount**

**5000 5%**

**15000   5%**

**20000   5%**

**20001   10%**

**30000   10%**

**50000   10%**

**50001   15%**

1. See the code below

    READ mark;

   IF(mark > 90)

               GRADE = ‘A’;

   ELSE IF(mark > 81 && mark < 90)

               GRADE = ‘B’;

   ELSE IF(mark > 71 && mark < 80)

               GRADE = ‘C’;

   ELSE IF(mark > 61 && mark < 70)

               GRADE = ‘D’;

   ELSE IF(mark < 60)

               GRADE = ‘F’;

   PRINT GRADE;

predict the number of independent paths to be tested.

**Ans: 9**

1. Derive the Cardinality between the student and the School.

A School has many students. The student belongs to a school

Ans: M:1

1. Which of the following statements are true for the below pseudocode

   READ A,B,C

   IF (A>B  AND  A>C)

            PRINT  “A Is Greater”

  IF (B>A  AND  B>C)

            PRINT  “B Is Greater”

   IF (C>A  AND  C>B)

            PRINT  “C Is Greater”

**Condition when all the variables are given the same value is not checked,**

**Code would be more efficient if, if is replaced by else-if**

1. A module is designed for the retail shop to calculate the discount based on the customer type

a)      privileged customer then 3% discount

b)      normal customer then no discount

Design test cases using equivalence partitioning for the above scenario

**Input                      Expected Output**

**Privilege Customer     3%**

**Normal Customer   No Discount**

**Gold Customer       Invalid**

1. For the given code, identify the correct independent paths

Program for Search

{

Binary Search Algorithm

            {

1.     int bottom = 0 ;

2. int top = elemArray.length - 1 ;

    int mid ;

3.     r.found = false ;

4.     r.index = -1 ;

5.     while ( bottom <= top )

    {

6.      mid = (top + bottom) / 2 ;

7.      if (elemArray [mid] == key)

    {

8.     r.index = mid ;

9.      r.found = true ;

10.   return ;

    } // if part

    else

    {

11.        if (elemArray [mid] < key)

12.      bottom = mid + 1 ;

      else

13.  top = mid - 1 ;

    }

    } //while loop

14. } // Binary search

} //Search

Ans:

**1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14**

**1, 2, 3, 4, 5, 6, 7, 11, 12, 5,…**

1. An Employee is Allocated for the Project. Each Employee is described by the empId, Ename, designation, and salary. Each Project is described by projID, ProjName,estimatedCost, and Actual Cost.

For the above scenario identify the correct entities

**Employee, Project**