PVP 20

Code: 20AM4601D, 20DS4601D

III B.Tech - II Semester – Regular Examinations - APRIL 2025

SOFTWARE TESTING METHODOLOGIES

(Common for AIML & DS)

Duration: 3 hours Max. Marks: 70

Note: 1. This paper contains questions from 5 units of Syllabus. Each unit carries 14 marks and have an internal choice of Questions.
2. All parts of Question must be answered in one place.

BL – Blooms Level CO – Course Outcome

			BL	CO	Max. Marks
		UNIT-I			
1	a)	Discuss the evolution of software testing.	L2	CO1	10 M
	b)	Explain the model for software testing.	L2	CO1	4 M
		OR			
2	a)	Describe the testing principles and their importance.	L2	CO1	7 M
	b)	Explain the phases involved in STLC.	L2	CO1	7 M
		UNIT-II			
3	a)	Apply a procedure to verify High level designs.	L3	CO2	7 M
	b)	Demonstrate the activities required for validation.	L3	CO2	7 M
		OR			
4	a)	Compare and contrast verification and validation.	L3	CO2	7 M

	b)	How would you verify the bugs (bug-Classification) based on SDLC?	L3	CO2	7 M
		UNIT-III			
5	a)	Demonstrate inspection process and benefits of inspection process.	L3	CO3	10 M
	b)	How is scenario-based reading different from checklists? Explain the types of scenario based reading.	L3	CO3	4 M
		OR			
6	a)	Explain the Regression testing, apply Regression testing to produce the quality software.	L3	CO3	7 M
	b)	Compare and contrast Progressive testing and Regressive testing.	L4	CO3	7 M
		UNIT-IV			
7	a)	Explain the various activities performed in the quantitative approach for quality management.	L2	CO3	7 M
	b)	Compare and contrast Quality management and Project management.	L2	CO3	7 M
		OR			
8	a)	Analyze the benefits of investment on quality.	L4	CO3	4 M
	b)	Explain the costs associated with quality.	L2	CO3	3 M
	c)	Compare and contrast quality control, quality assurance, and quality management.	L4	CO3	7 M

		UNIT-V			
9	a)	Explain the process involved in TPI model developed to measure the maturity of a test process.	L3	CO4	8 M
	b)	Classify the different levels of TMM with its purpose.	L2	CO4	6 M
		OR			
10	a)	Analyze the need for Automation in testing.	L4	CO4	4 M
	b)	Why selection of testing activity tools are needed, discuss the categorization of testing tools elaborate?	L4	CO4	10 M

Subject code: 20AM4601D,20DS4601D

PVP20

PRASAD V. POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY
III B.Tech. – II Sem- Regular Examinations APRIL 2025

SOFTWARE TESTING METHODOLOGIES
SCHEME OF EVALUATION
(Common for AIML, DS)

I. Short Scheme

Q.No	Question		CO-Level	Total Marks
1(a)	**Discuss the evolution of Software testing.** Evolution Diagram --- 4M Any 3 Evolution phases --- 6M	10M	CO1-L2	10M
1(b)	**Explain the model for software testing.** Model for software testing Diagram --- 2M Steps Involved in testing --- 2M	4M	CO1-L2	4M
2(a)	**Describe the testing principles and their importance.** Any 5 testing principles	7M	CO1-L2	7M
2(b)	**Explain the phases involved in STLC** Software Testing Life Cycle Diagram --- 3M STLC Phases --- 4M	7M	CO1-L2	7M
3(a)	**Apply a procedure to verify High level designs.** Explanation of Verification of High-Level Design Data Design, Architectural Design, Interface Design	7M	CO2-L3	7M
3(b)	**Demonstrate the activities required for validation.** Explanation of 2 validation activities: 1) Validation Test Plan --- 3M 2) Validation Test Execution --- 4M	7M	CO2-L3	7M
4(a)	**Compare and contrast verification and validation** Any 4 differences between verification and validation	7M	CO2-L3	7M
4(b)	**How would you verify the bugs(bug-classification) based on SDLC?** Explanation to verify the any 4 bugs based on Software Development Life Cycle	7M	CO2-L3	7M
5(a)	**Demonstrate the inspection process and benefits of inspection process** Inspection process Diagram --- 2M Inspection process Phases --- 6M Benefits of inspection process --- 2M	10M	CO3-L3	10M
5(b)	**How to scenario-based reading different from checklist? explain the types of scenario based reading.** Any 2 points difference between scenario-based reading, checklist. --- 2M Any 2 types of scenario based reading techniques --- 2M	4M	CO3-L3	4M
6(a)	**Explain the Regression testing, apply regression testing to produce the quality software.** Definition of Regression testing --- 2M Regression testing to produce the quality software --- 5M	7M	CO3-L3	7M
6(b)	**Compare and Contrast Progressive testing and Regression testing.**	7M	CO3-L3	7M

	Any 4 Differences b/w Progressive vs Regression --- 7M			
7(a)	**Explain the various activities performed in the quantitative approach for quality management.** Explanation of quantitative approach --- 7M	7M	CO3-L2	7M
7(b)	**Compare and Contrast Quality management and Project management** Any 4 Differences b/w QM and PM --- 7M	7M	CO4-L4	7M
8(a)	**Analyze the benefits of investment on quality** Benefits of investment on quality with diagram ---4M	4M	CO4-L4	4M
8(b)	**Explain the costs associated with quality** Types of Quality costs: Prevention costs, Appraisal costs, Failure Costs	3M	CO3-L2	3M
8(c)	**Compare and Contrast quality control, quality assurance and quality management** Any 4 Differences b/w quality control, quality assurance and quality management	7M	CO3-L4	7M
9(a)	**Explain the process involved in TPI model developed to measure the maturity of a test process.** Process involved in Test Process Improvement (TPI) model ---6M Diagram ---2M	8M	CO3-L3	8M
9(b)	**Classify the different levels of TMM with its purpose** Levels of Test Maturity Model and its purpose explanation ---4M TMM levels diagram ---2M	6M	CO2-L2	6M
10(a)	**Analyze the need for Automation in testing.** Mention any 2 points for Automation in testing explanation ---4M	4M	CO4-L4	4M
10(b)	**Why selection of testing activity tools is needed, discuss the categorization of testing tools elaborate?** Selection of testing activity tools explanation ---6M categorization of testing tools ---4M	10M	CO4-L4	10M

PVP20

Subject code: 20AM4601D,20DS4601D

PRASAD V. POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY
III B.Tech. – II Sem- Regular Examinations APRIL 2025

SOFTWARE TESTING METHODOLOGIES
SCHEME OF EVALUATION
(Common for AIML, DS)

I. Long Scheme

UNIT-I

1a) Discuss the evolution of Software testing. **L2 CO1 10M**

- Evolution Diagram --- 4M
- Any 3 Evolution phases --- 6M

Ans)

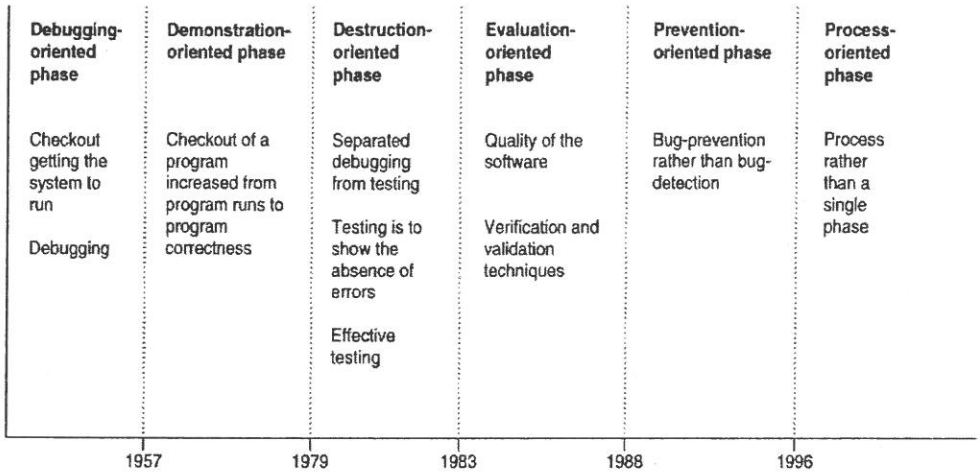

Figure 1.1 Evolution phases of software testing

1b) Explain the model for software testing. **L2 CO1 4M**

- Model for software testing Diagram --- 2M
- Steps Involved in --- 2M

Ans)

Model for Software Testing:

Testing is not an intuitive activity; rather it should be learnt as a process. Therefore, testing should be performed in a planned way. For the planned execution of a testing process, we need to consider every element and every aspect related to software testing.

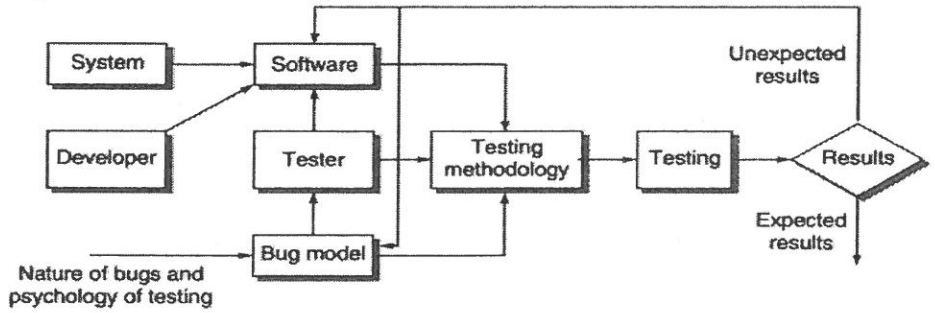

Figure 1.6 Software testing model

The software is basically a part of a system for which it is being developed. Systems consist of hardware and software to make the product run.

1. Software and Software Model
2. Bug Model
3. Testing methodology and Testing

OR

2a) Describe the testing principles and their importance. **L2 CO1 7M**

- Any 5 testing principles ----7M

Ans)

Testing principles can be seen as guidelines for a tester.

1. Effective testing, not exhaustive testing
2. Testing is not a single phase performed in SDLC
3. Destructive approach for constructive testing
4. Early testing is the best policy
5. Probability of existence of an error in a section of a program is proportional to the number of errors already found in that section
6. Testing strategy should start at the smallest module level and expand towards the whole program
7. Testing should also be performed by an independent team
8. Everything must be recorded in software testing Invalid inputs and unexpected behaviour have a high probability of finding an error
9. Testers must participate in specification and design reviews

2b) Explain the phases involved in STLC. **L2 CO1 7M**

- Software testing life cycle Diagram --- 3M
- STLC Phases --- 4M

Ans)

The testing process divided into a well-defined sequence of steps is termed as software testing life cycle (STLC). The major contribution of STLC is to involve the testers at early stages of development. This has a significant benefit in the project schedule and cost. The STLC also helps the management in measuring specific milestones.

STLC consists of the following phases (see Fig. 2.8):

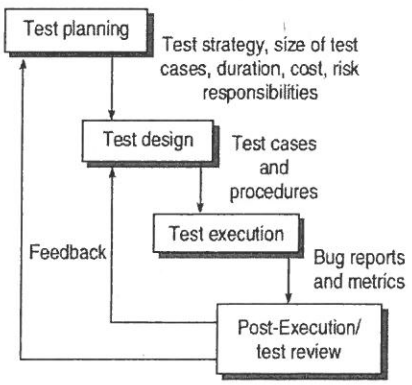

Figure 2.8 Software testing life cycle

UNIT-II

3a) Apply a procedure to verify High level designs. **L3 CO2 7M**

- Explanation of Verification of High-Level Design ---7M

Ans)

Verification of High-Level Design:

All the requirements mentioned in the SRS document are addressed in this phase and work in the direction of designing the solution. The architecture and design is documented in another document called the software design document (SDD).

How To Verify High-Level Design?

High-level design takes the second place in SDLC, wherein there is a high probability of finding bugs. Therefore, high-level design must be verified as a next step in early testing. Unless the design is specified in a formal way, design cannot be verified. So SDD is referred for design verification.

If a bug goes undetected in the high-level design phase, then its cost of fixing increases with every phase. Therefore, verification for high-level design must be done very carefully.

This design is divided in three parts

1. Data Design
2. Architectural Design
3. Interface Design

3b) Demonstrate the activities required for validation. **L3 CO2 7M**

- 1) Validation Test Plan --- 3M
- 2) Validation Test Execution --- 4M

Ans)

Validation Activities:

The validation activities are divided into two ways

1) Validation Test Plan
2) Validation Test Execution

Validation Test Plan:

It starts as the first output of SDLC, i.e. the SRS, is prepared. In every phase, the tester performs two parallel activities—verification at that phase and the corresponding validation test plan.

For preparing a validation test plan, testers must follow the points described below.

- Testers must understand the current SDLC phase.
- Testers must study the relevant documents in the corresponding SDLC phase.

Validation Test Execution:

Test Execution is the process of executing the tests written by the tester to check whether the developed code or functions or modules are providing the expected result as per the client requirement or business requirement. Test Execution comes under one of the phases of the Software Testing Life Cycle (STLC).

OR

4a) Compare and contrast verification and validation L3 CO2 7M

- Any 4 Differences between verification and validation ---7M

Ans)

Differences between Verification and Validation

	Verification	Validation
Definition	Verification refers to the set of activities that ensure software correctly implements the specific function	Validation refers to the set of activities that ensure that the software that has been built is traceable to customer requirements.
Focus	It includes checking documents, designs, codes, and programs.	It includes testing and validating the actual product.
Type of Testing	Verification is the static testing.	Validation is dynamic testing.
Execution	It does *not* include the execution of the code.	It includes the execution of the code.
Methods Used	Methods used in verification are reviews, walkthroughs, inspections and desk-checking.	Methods used in validation are Black Box Testing, White Box Testing and non-functional testing.
Purpose	It checks whether the software conforms to specifications or not.	It checks whether the software meets the requirements and expectations of a customer or not.
Bug	It can find the bugs in the early stage of the development.	It can only find the bugs that could not be found by the verification process.
Goal	The goal of verification is application	The goal of validation is an actual

4b) How would you verify the bugs based on SDLC? L3 CO2 7M

- Bugs can appear in any phase of Software Development Life Cycle, they can be classified based on SDLC phases.

Ans)

- 1.Requirements and Specifications Bugs
- 2.Design Bugs
- 3.Coding Bugs
- 4.Testing Bugs

UNIT-III

5 a) Demonstrate the inspection process and benefits of inspection process L3 CO3 7M

- Inspection process Diagram --- 2M
- Inspection process Phases --- 6M
- Benefits of inspection process --- 2M

Ans)

The stages in the inspections process are:

1. Planning 2. Overview meeting 3. Preparation 4.Inspection meeting 5.Rework 6.Followup.

The Preparation, Inspection meeting and Rework stages might be iterated.

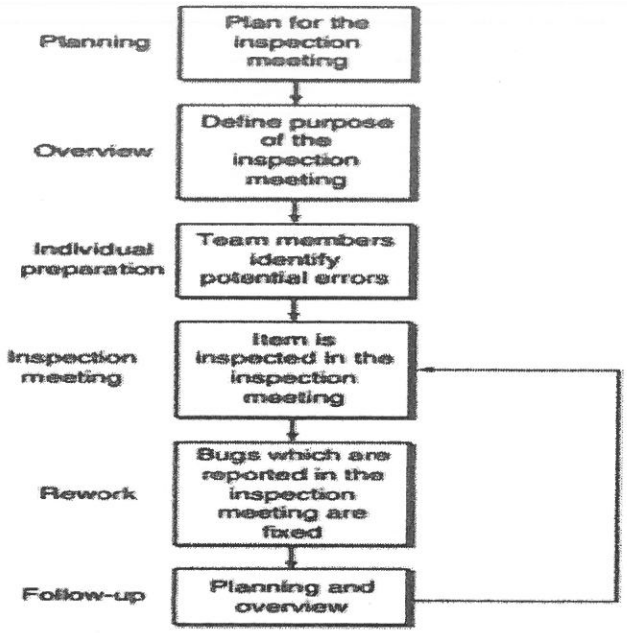

Figure 6.1 Inspection process

5b) How to scenario-based reading different from checklist? explain the types of scenario based reading. **L3 CO3 4M**

- Any 2 points difference between scenario-based reading, checklist. --- 2M
- Any 2 types of scenario based reading techniques --- 2M

Ans)

Scenario-Based Reading (SBR) and Checklist-Based Review are two different techniques used in software inspections and requirements analysis to identify defects. Here's how they differ:

Feature	Scenario-Based Reading (SBR)	Checklist-Based Review
Approach	Guided by realistic usage scenarios (user stories, error conditions, etc.)	Follows a predefined list of questions or checks
Focus	Examines how the system behaves under specific conditions	Ensures compliance with general standards and best practices
Flexibility	Adaptable to different contexts and user perspectives	Rigid, follows a fixed set of criteria
Engagement	Encourages deeper analysis by simulating real-world usage	More mechanical, may miss context-specific issues
Best Used For	Requirements validation, usability, and functional testing	Compliance checks, coding standards, and basic defect detection

scenario based reading techniques

- Perspective-based reading
- Usage-based reading
- Abstraction driven reading
- Task-driven reading
- Function-point based scenarios

OR

6a) Explain the Regression testing, apply regression testing to produce the quality software. **L3 CO3 7M**

- Definition of Regression testing --- 2M
- Regression testing to produce the quality software --- 5M

Ans)

Regression testing can be performed on a new build when there is a significant change in the original functionality. It ensures that the code still works even when the changes are occurring. Regression means Re-test those parts of the application, which are unchanged.

Regression tests are also known as the Verification Method. Test cases are often automated. Test cases are required to execute many times and running the same test case again and again manually, is time-consuming and tedious too.

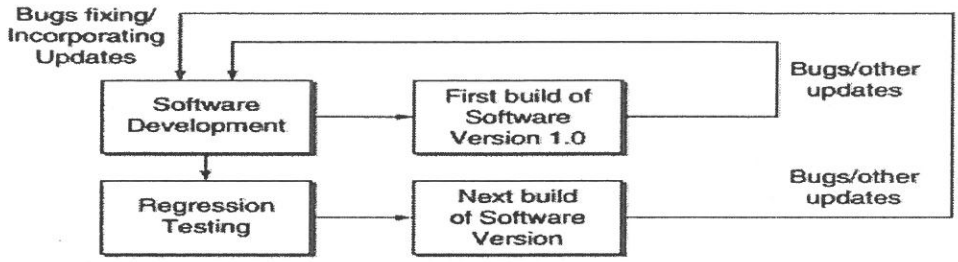

Figure 8.1 Regression testing produces Quality Software

6b) Compare and Contrast Progressive testing and Regression testing. L4 CO3 7M

- Any 4 Differences b/w Progressive vs Regression

Ans)

Aspect	Progressive Testing	Regression Testing
Purpose	Ensures new features or modules work correctly.	Ensures existing functionalities remain unaffected by changes.
Focus	Testing newly developed parts of the system.	Testing previously developed and stable parts of the system.
Trigger Point	Triggered during the development of new features.	Triggered after a code change, such as bug fixes or updates.
Objective	Verify functionality of new features or modules.	Detect unintended side effects of code changes.
Test Cases	Focus on new test cases specific to the new features.	Reuse existing test cases for previously tested features.

UNIT-IV

7a) Explain the various activities performed in the quantitative approach for quality management. L2 CO3 7M

- Quantitative Approach To Qm

Ans)

Quantitative approach that measures every activity in the life cycle, analyses the data and then, gives judgement about the level of quality. If there is a need to increase the level of quality, then auditing is done to check whether all the procedures and standards are being followed. systematic program of activities called the software metrics programme (SMP). We discuss a model for SMP given by Paul Goodman

Paul Goodman Model for SMP:

This model presents a framework for SMP in the lifecycle stages just like SDLC. The idea is to identify and implement the software metrics in the finer detailed intermediate stages. This model comprises of five stages as discussed here

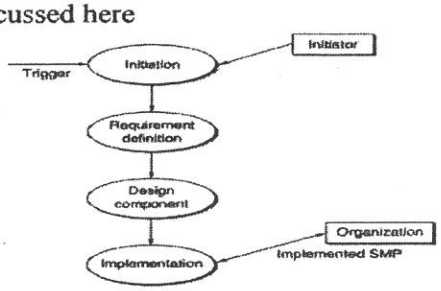

Figure 13.5 Paul Goodman model for SMP

7b) Compare and Contrast Quality management and Project management. L4 CO4 7M

- Any 4 Differences b/w QM and PM ---7M

Ans)

Comparison and Contrast: Quality Management vs. Project Management

Aspect	Quality Management (QM)	Project Management (PM)
Primary Focus	Ensures deliverables meet defined quality standards	Ensures project is completed on time, within scope & budget
Goal	Deliver high-quality products/services	Achieve project objectives efficiently
Key Processes	- Quality Planning - Quality Assurance (QA) - Quality Control (QC) - Continuous Improvement	- Initiation - Planning - Execution - Monitoring & Controlling - Closure
Standards & Frameworks	ISO 9001, Six Sigma, CMMI, TQM (Total Quality Management)	PMBOK (PMI), PRINCE2, Agile (Scrum, Kanban)
Key Metrics	- Defect density - Customer satisfaction - Process capability (Cpk)	- Schedule variance - Cost variance - Risk impact
Responsibility	Quality managers, QA/QC teams	Project managers, team leads, stakeholders
Approach	Preventive (avoids defects) & Corrective (fixes defects)	Predictive (Waterfall) or Adaptive (Agile)
Tools Used	- Control charts - Pareto analysis - Fishbone diagrams	- Gantt charts - Risk matrices - Work breakdown structure (WBS)
When Applied	On-going (throughout product lifecycle)	Temporary (until project completion)

OR

8a) Analyse the benefits of investment on quality L4 CO3 4M

- Benefits of investment on quality with diagram ---4M

Ans)

Despite the costs incurred in prevention and appraisal activities, benefits of these activities greatly outweigh the costs. If we invest in prevention and appraisal activities at an early stage, then failures and failure-costs reduce

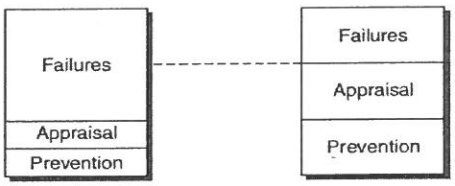

Figure 13.3 Benefits of investment on quality

Thus, by implementing quality evaluation programs, the following benefits are achieved:

- Customer is satisfied, as the end-product is of high quality.
- Productivity increases due to shorter SDLC cycle.
- Failures and failure costs reduce.
- Rework and cost of quality reduce.

8b) Explain the costs associated with quality L2 CO3 3M

- Types of Quality costs: Prevention costs, Appraisal costs, Failure costs

Ans)

Quality of a software is maintained with efforts in both products and processes. But implementation of those quality-related procedures and practices is expensive and incurs cost.

- **Prevention costs** are related with activities that identify the cause of defects and those actions that are taken to prevent them.
- **Appraisal costs** include the costs of evaluating the quality of software products at some level.
- **Failure costs** include the costs to analyse and remove the failures.

8c) Compare and Contrast quality control, quality assurance and quality management

- Any 4 Differences **L4 CO3 7M**

Ans) Comparison Table

Aspect	Quality Control (QC)	Quality Assurance (QA)	Quality Management (QM)
Definition	Detects defects in **finished products**	Prevents defects by improving **processes**	Oversees **both QC and QA** to ensure overall quality
Focus	**Product-oriented** (Outputs)	**Process-oriented** (Systems)	**Strategic** (Organization-wide)
Goal	Identify & fix defects before delivery	Prevent defects through standardized processes	Align quality with business objectives
When Applied	**Reactive** (After production)	**Proactive** (During development)	**Continuous** (Throughout lifecycle)
Key Activities	Testing, inspections, bug fixes	Process audits, documentation, training	Policy creation, metrics analysis, QC+QA integration
Tools/Methods	- Manual/Automated testing - Sampling inspections	- PDCA Cycle - ISO 9001 audits - Six Sigma	- Balanced Scorecard - TQM (Total Quality Management)
Responsibility	QC Engineers, Testers	QA Engineers, Process Analysts	Quality Managers, Leadership

UNIT-V

9a) Explain the process involved in TPI model developed to measure the maturity of a test process. **L3 CO3 8M**

- Process involved in TPI model ---6M
- TPI model diagram ---2M

Ans)

Test Process Improvement (TPI) Model:

The TPI model was developed by Koomen and Pol in 1997 [140,141]. It contains the following parts

- Key process areas
- Maturity levels
- Test maturity matrix
- Checklist
- Improvement suggestions

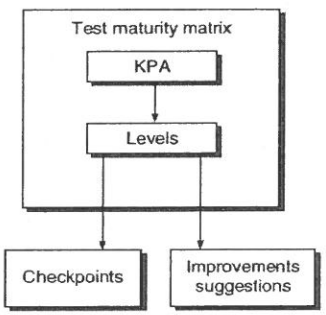

Figure 14.1 TPI model structure

9b) Classify the different levels of TMM with its purpose L2 CO2 6M

- Levels TMM and its purpose explanation ---4M
- TMM levels diagram ---2M

Ans)

TMM Levels

TMM has been deliberately designed similar to CMM. The idea is to ensure the growth in testing maturity is in tune with the growth in software capability maturity. The characteristics of fi ve TMM levels are given below.

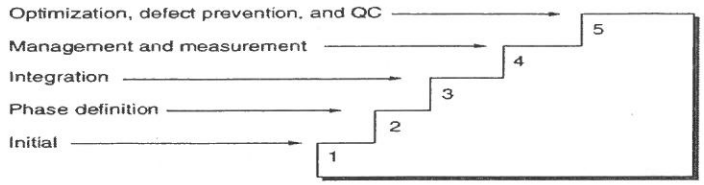

Figure 14.5 Five levels of TMM

1. **Initial level** : The objective of testing at this level is to show only the functioning of the software.
2. **Phase definition** : The primary goal of testing at this maturity level is to show that the software meets its specifications.
3. **Integration:** Test objectives are established with respect to the requirements based on user and client needs and are used for test case plan, design, and success criteria.
4. **Management and measurement** : At this level, testing is recognized as a measured and quantified process.
5. **Optimization, defect-prevention, and quality control**: At this level, the testing process is defined and managed.

OR

10 a) Analyse the need for Automation in testing. L4 CO4 4M

- Mention any 2 points need for Automation in testing explanation ---4M

Ans)

an organization needs to choose a testing tool, the following benefits of automation must be considered.

Reduction of testing effort: execution of test suits through software tools greatly reduces the amount of time required.

Reduces the testers' involvement in executing tests: Automating this process of executing the test suit will relieve the testers to do some other work, thereby increasing the parallelism in testing efforts.

Facilitates regression testing: If we automate the process of regression testing, then testing effort as well as the time taken will reduce as compared to manual testing.

Avoids human mistakes: Manually executing the test cases may incorporate errors in the process or sometimes automation in testing tools will not cause these problems

Reduces overall cost of the software: if testing time increases, cost of the software also increases.

Simulated testing: Load performance testing is an example of testing where the real-life situation needs to be simulated in the test environment. Automated tools, on the other hand, can create millions of concurrent virtual users/data and effectively test the project in the test environment before releasing the product.

Test enablers: Automation reduces the effort required in this case and becomes essential.

10 b) Why selection of testing activity tools are needed, discuss the categorization of testing tools elaborate? L4 CO4 10M

- selection of testing activity tools explanation ---6M
- categorization of testing tools ---4M

Ans)

Testing Activity Tools:

These tools are based on the testing activities or tasks in a particular phase of the SDLC.
Testing activities can be categorized as

1. Reviews and inspections
2. Test planning
3 Test design and development
4. Test execution and evaluation

Tools for review and inspections these tools are for static analysis on many items, some tools are designed to work with specifications

Complexity analysis tools It is important for testers that complexity is analysed so that testing time and resources can be estimated.

Code comprehension These tools help in understanding dependencies, tracing program logic, viewing graphical representations of the program, and identifying the dead code.

Tools for test planning The types of tools required for test planning are:

1. Templates for test plan documentation
2. Test schedule and staffing estimates
3. Complexity analyser

Tools for test design and

Test data generator: It automates the generation of test data based on a user defined format.

Test case generator : It automates the procedure of generating the test cases.

Test execution and evaluation tools

Capture/playback tools: These tools record events (including keystrokes, mouse activity, and display output) at the time of running the system and place the information into a script.

Coverage analysis tools : These tools automate the process of thoroughly testing the software and provide a quantitative measure of the coverage of the system being tested.

- Quantifying the complexity of the design
- Help in specifying parts of the software which are not being covered
- Measure the number of integration tests required to qualify the design
- Help in producing integration tests
- Measuring the number of integration tests that have not been executed
- Measuring multiple levels of test coverage, including branch, condition, decision/condition, multiple conditions, and path coverage

Categorization of Testing Tools:

1.Static Testing Tools
2.Dynamic Testing Tools

These tools are based on the type of execution of test cases, namely static and dynamic, as discussed in software testing techniques:

Static testing tools

For static testing, there are static program analysers which scan the source program and detect possible faults and anomalies. These static tools parse the program text, recognize the various sentences, and detect the following:

- Statements are well-formed.
- Inferences about the control fl ow of the program.
- Compute the set of all possible values for program data.

Dynamic testing tools

- Dynamic testing activities.
- Many a times, systems are difficult to test because several operations are being performed concurrently. In such cases, it is difficult to anticipate conditions and generate representative test cases. Automated test tools enable the test team to capture the state of events during the execution of a program by preserving a snapshot of the conditions. These tools are sometimes called program monitors.