SOFTWARE ENGINEERING & PROJECT MANAGEMENT (SEPM)

MODULE 5 – Software Testing & Maintenance

Based on Mumbai University PYQs (2023–2025)

This document contains detailed descriptive answers for ALL Module 5 questions from the uploaded SEPM question bank.

Covered Questions:

1. Explain Software Testing in detail.
2. Explain White Box Testing and Black Box Testing.
3. Explain Testing Strategies.
4. Explain Unit Testing, Integration Testing, Validation Testing, and System Testing.
5. Explain Verification and Validation.
6. Explain Alpha Testing and Beta Testing.
7. Explain Debugging Approaches.
8. Explain Software Maintenance.
9. Types of Software Maintenance.
10. Explain Reengineering and Reverse Engineering.
11. Explain Smoke Testing and Regression Testing.
12. Explain Basis Path Testing.
13. Explain Test Case Design.

Q1. Explain Software Testing in Detail.

Introduction

Software Testing is one of the most important activities in Software Engineering.

No software is completely free from errors. Testing helps developers identify defects and ensure software quality.

Software Testing improves:

- Reliability
- Performance

- Security
- Quality
- Customer satisfaction

Definition of Software Testing

Software Testing is the process of executing software with the intention of finding defects and verifying that the software satisfies specified requirements.

Objectives of Software Testing

1. Identify defects.
2. Improve software quality.
3. Verify software functionality.
4. Ensure customer satisfaction.
5. Improve reliability.
6. Reduce maintenance cost.

Need for Software Testing

Testing is required because:

1. Software systems are complex.
2. Errors may cause financial loss.
3. Security vulnerabilities may exist.
4. Failure may affect business operations.
5. Users expect reliable software.

Software Testing Process

```
Test Planning → Test Case Design → Test Execution → Defect Reporting → Retesting
```

Characteristics of Good Testing

1. Detects maximum defects.

2. Improves reliability.
3. Reduces development risk.
4. Saves maintenance cost.
5. Ensures requirement compliance.

Levels of Testing

1. Unit Testing
2. Integration Testing
3. Validation Testing
4. System Testing

Testing Principles

1. Testing Shows Presence of Defects

Testing cannot prove absence of defects.

2. Early Testing

Testing should start early.

3. Exhaustive Testing is Impossible

All possible test cases cannot be tested.

4. Defect Clustering

Most defects occur in small number of modules.

5. Pesticide Paradox

Repeated tests may stop finding new defects.

Advantages of Software Testing

1. Improved software quality.
2. Better reliability.
3. Increased customer trust.

4. Reduced failures.
5. Better security.

Disadvantages of Software Testing

1. Time-consuming.
2. Increases project cost.
3. Cannot guarantee error-free software.

Conclusion

Software Testing is essential for developing reliable and high-quality software systems.

Q2. Explain White Box Testing and Black Box Testing.

Introduction

Testing techniques are classified into:

1. White Box Testing
2. Black Box Testing

Both techniques are important for identifying software defects.

White Box Testing

White Box Testing is a testing technique where internal program structure is visible to the tester.

It is also called:

- Structural Testing
- Glass Box Testing
- Logic Driven Testing

Features of White Box Testing

1. Internal logic is tested.
2. Source code knowledge is required.
3. Performed by developers.
4. Path coverage is important.

White Box Testing Techniques

1. Basis Path Testing

Tests independent execution paths.

2. Loop Testing

Tests loops in the program.

3. Condition Testing

Tests logical conditions.

Advantages of White Box Testing

1. Detects logical errors.
2. Optimizes code.
3. Improves security.
4. Measures code coverage.

Disadvantages of White Box Testing

1. Requires programming knowledge.
2. Time-consuming.
3. Difficult for large systems.

Black Box Testing

Black Box Testing is a testing technique where internal program structure is hidden.

The tester focuses only on functionality.

Features of Black Box Testing

1. Tests functionality.
2. No knowledge of source code required.
3. Performed by testers.
4. Based on requirements.

Black Box Testing Techniques

1. Equivalence Partitioning

2. Boundary Value Analysis

3. Cause Effect Graphing

4. Decision Table Testing

Advantages of Black Box Testing

1. User-oriented testing.
2. No programming knowledge required.
3. Detects missing functions.
4. Suitable for large systems.

Disadvantages of Black Box Testing

1. Limited path coverage.
2. Difficult to identify hidden errors.
3. Large number of test cases required.

Difference Between White Box and Black Box Testing

White Box Testing	Black Box Testing
Internal structure visible	Internal structure hidden
Logic-based testing	Functional testing
Requires programming knowledge	No programming knowledge required
Performed by developers	Performed by testers
Tests code structure	Tests system functionality

Conclusion

White Box and Black Box Testing complement each other and help improve overall software quality.

Q3. Explain Testing Strategies.

Introduction

Testing Strategy defines the systematic approach used for software testing.

A proper testing strategy ensures:

- Better quality
- Reduced defects
- Improved reliability

Levels of Testing

1. Unit Testing
2. Integration Testing
3. Validation Testing
4. System Testing

---

1. Unit Testing

Individual modules are tested independently.

Objectives

- Detect coding errors
- Verify module functionality

Advantages

1. Early defect detection.
2. Easier debugging.
3. Improved module quality.

---

2. Integration Testing

Combined modules are tested together.

Objectives

- Detect interface errors
- Verify module interaction

Types of Integration Testing

a) Top-Down Integration

Testing starts from top modules.

b) Bottom-Up Integration

Testing starts from lower modules.

c) Sandwich Integration

Combination of top-down and bottom-up.

Advantages

1. Detects interface defects.
2. Improves module interaction.

---

3. Validation Testing

Checks whether software satisfies customer requirements.

Objectives

- Requirement verification
- Customer satisfaction

Advantages

1. Ensures correct functionality.
2. Improves user satisfaction.

---

4. System Testing

Entire system is tested as a whole.

Types of System Testing

1. Recovery Testing
2. Security Testing
3. Stress Testing
4. Performance Testing

Advantages

1. Complete system verification.
2. Improved reliability.

Diagram of Testing Strategy

```
Unit Testing
      ↓
Integration Testing
      ↓
Validation Testing
      ↓
System Testing
```

Conclusion

Testing strategies ensure systematic software testing and improve software quality significantly.

Q4. Explain Verification and Validation.

Introduction

Verification and Validation are important quality assurance activities.

They ensure that:

- Software is developed correctly.
- Correct software is developed.

Verification

Verification checks whether software is developed according to specifications.

It answers:

“Are we building the product right?”

Verification Activities

1. Reviews
2. Walkthroughs
3. Inspections
4. Static Analysis

Advantages of Verification

1. Early defect detection.
2. Reduced development cost.
3. Improved quality.

Validation

Validation checks whether software satisfies customer requirements.

It answers:

“Are we building the right product?”

Validation Activities

1. Functional Testing
2. System Testing
3. Acceptance Testing

Advantages of Validation

1. Improved customer satisfaction.

2. Requirement verification.
3. Better usability.

Difference Between Verification and Validation

Verification	Validation
Checks specifications	Checks customer requirements
Static process	Dynamic process
Performed before execution	Performed after execution
Detects design defects	Detects functionality defects

Conclusion

Verification and Validation together ensure high-quality software development.

Q5. Explain Alpha Testing and Beta Testing.

Alpha Testing

Alpha Testing is performed at the developer’s site before software release.

Features

1. Performed by developers.
2. Conducted in controlled environment.
3. Detects defects before release.

Advantages

1. Early defect detection.
2. Improved software quality.

Beta Testing

Beta Testing is performed by actual users at customer site.

Features

1. Performed by end users.
2. Conducted in real environment.
3. Collects customer feedback.

Advantages

1. Real-world testing.
2. Improved customer satisfaction.

Difference Between Alpha and Beta Testing

Alpha Testing	Beta Testing
Performed by developers	Performed by users
Controlled environment	Real environment
Before Beta Testing	After Alpha Testing
Internal testing	External testing

Conclusion

Alpha and Beta Testing improve software reliability before final deployment.

Q6. Explain Debugging Approaches.

Introduction

Debugging is the process of identifying and correcting software defects.

Objectives of Debugging

1. Find root cause of defect.
2. Correct software errors.
3. Improve reliability.

Debugging Approaches

1. Brute Force Method

Uses print statements and memory dumps.

---

2. Backtracking

Starts from error point and traces backward.

---

3. Cause Elimination Method

Possible causes are eliminated one by one.

Advantages of Debugging

1. Improved software quality.
2. Reduced defects.
3. Better reliability.

Conclusion

Debugging is an essential activity for correcting software defects.

Q7. Explain Software Maintenance.

Introduction

Software Maintenance modifies software after deployment.

Maintenance is necessary because:

- Requirements change
- Errors are discovered
- Technology evolves

Definition

Software Maintenance is the process of modifying software after delivery to correct faults, improve performance, or adapt to changing environment.

Objectives of Software Maintenance

1. Correct defects.
2. Improve performance.
3. Add new features.
4. Improve maintainability.

Types of Software Maintenance

1. Corrective Maintenance

Fixes software defects.

Example

Bug fixing.

---

2. Adaptive Maintenance

Modifies software for new environment.

Example

Operating system updates.

---

3. Perfective Maintenance

Improves performance and usability.

Example

Adding new features.

---

4. Preventive Maintenance

Prevents future problems.

Example

Code restructuring.

Maintenance Process

```
Problem Identification → Analysis → Modification → Testing → Release
```

Advantages of Maintenance

1. Increased software life.
2. Improved performance.
3. Better customer satisfaction.

Disadvantages of Maintenance

1. Expensive.
2. Time-consuming.
3. Difficult for legacy systems.

Conclusion

Software Maintenance is essential for keeping software reliable and useful after deployment.

Q8. Explain Smoke Testing and Regression Testing.

Smoke Testing

Smoke Testing verifies whether major functionalities are working.

Objectives

1. Detect major failures.
2. Verify software stability.

Advantages

1. Early defect detection.
2. Saves testing effort.

Regression Testing

Regression Testing ensures that modifications do not affect existing functionality.

Objectives

1. Detect side effects.
2. Verify old features.

Advantages

1. Improved reliability.
2. Prevents new defects.

Difference Between Smoke and Regression Testing

Smoke Testing	Regression Testing
Checks major functionality	Checks modified functionality
Performed after build	Performed after changes
Basic testing	Detailed testing

Conclusion

Smoke and Regression Testing improve software stability and reliability.

Q9. Explain Basis Path Testing.

Introduction

Basis Path Testing is a White Box Testing technique.

It uses:

- Control Flow Graph
- Cyclomatic Complexity

for designing test cases.

Steps in Basis Path Testing

1. Draw CFG.
2. Calculate Cyclomatic Complexity.
3. Identify independent paths.
4. Design test cases.

Cyclomatic Complexity Formula

```
V(G) = E - N + 2
```

Where:

- E = Edges
- N = Nodes

Advantages

1. Complete path coverage.
2. Improved testing efficiency.
3. Better defect detection.

Conclusion

Basis Path Testing improves software reliability through path-based testing.

END OF MODULE 5