CS403 SOFTWARE ENGINEERING (RGPV)

UNIT–4 : SOFTWARE ANALYSIS AND TESTING ⭐⭐⭐⭐⭐

> 🎯 **Most Important Unit for Exam**
>
> Agar tumhare paas time kam hai to sabse pehle Unit-4 padho.
>
> RGPV me Testing se har saal direct questions aate hain.
>
> Is unit se **15–25 marks** tak aa sakte hain.

UNIT OVERVIEW

Why This Unit is Important?

Software develop karne ke baad usme errors (bugs) ko detect karna zaruri hota hai.

Testing ensure karti hai ki software correctly kaam kare.

Weightage

⭐ Approx 20–25 Marks

Most Repeated Questions

🔥 Black Box Testing vs White Box Testing

🔥 Testing Levels

🔥 Unit Testing

🔥 Integration Testing

🔥 System Testing

🔥 Static vs Dynamic Analysis

🔥 Software Testing Process

🔥 Test Case Design

SOFTWARE TESTING

Simple Explanation

Software ko execute karke ya analyze karke errors dhoondhne ki process ko Testing kehte hain.

Definition

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

Objectives

✅ Find defects

✅ Improve quality

✅ Verify requirements

✅ Ensure reliability

SOFTWARE ANALYSIS

Analysis ka purpose software ki quality evaluate karna hai.

---

1. STATIC ANALYSIS ⭐⭐⭐⭐⭐

Frequently Asked

Definition

Software ko execute kiye bina analyze karna.

Simple Meaning

Code run nahi hota.

Sirf code ko read aur review kiya jata hai.

Examples

- Code Review
- Code Inspection
- Walkthrough
- Documentation Review

Advantages

✅ Early error detection

✅ Low cost

✅ Improves code quality

Disadvantages

❌ Runtime errors detect nahi kar sakta

Diagram

```
Source Code
      ↓
Review
      ↓
Error Detection
```

---

2. DYNAMIC ANALYSIS ⭐⭐⭐⭐⭐

Frequently Asked

Definition

Software ko execute karke analyze karna.

Simple Meaning

Program run hota hai aur uska behavior observe kiya jata hai.

Examples

- Unit Testing
- Integration Testing

- System Testing

Advantages

✅ Runtime errors detect

✅ Actual behavior check

Disadvantages

❌ More time consuming

Diagram

```
Program Execution
        ↓
Observe Output
        ↓
Detect Errors
```

STATIC vs DYNAMIC ANALYSIS

⭐⭐⭐⭐⭐

Static Analysis	**Dynamic Analysis**
No Execution	Program Executes
Finds coding errors	Finds runtime errors
Faster	Slower
Lower Cost	Higher Cost

Code Review	Testing

🔥 Expected 7 Marks Question

---

3. CODE INSPECTION ⭐⭐⭐⭐

Definition

Formal review technique for finding defects in source code.

Steps

```
Planning
    ↓
Review
    ↓
Error Detection
    ↓
Correction
```

Advantages

- Detects bugs early
- Improves quality

Disadvantages

- Time consuming

SOFTWARE TESTING FUNDAMENTALS

⭐⭐⭐⭐⭐

Important Terms

Error

Human mistake.

Example:

Wrong formula written.

Fault (Bug)

Error present in code.

Failure

System gives wrong result.

Relation

```
Error
  ↓
Fault
  ↓
Failure
```

TESTING PROCESS ⭐⭐⭐⭐⭐

Very Important

Steps

```
Requirement Analysis
          ↓
Test Planning
          ↓
Test Case Design
          ↓
Test Execution
          ↓
Defect Reporting
          ↓
Test Closure
```

Explanation

Requirement Analysis

Requirements study karte hain.

Test Planning

Testing strategy decide karte hain.

Test Case Design

Test cases banate hain.

Execution

Software run karte hain.

Reporting

Bugs report karte hain.

Closure

Testing complete.

TESTING LEVELS ⭐⭐⭐⭐⭐

Most Important

Diagram

```
Unit Testing
      ↓
Integration Testing
      ↓
System Testing
      ↓
Acceptance Testing
```

---

1. UNIT TESTING ⭐⭐⭐⭐⭐

Definition

Testing of individual module.

Example

Login Module Testing

Advantages

✅ Easy debugging

✅ Early bug detection

---

2. INTEGRATION TESTING ⭐⭐⭐⭐⭐

Definition

Testing interaction between modules.

Example

Login Module + Database Module

Types

Top Down

Parent first

Bottom Up

Child first

---

3. SYSTEM TESTING ⭐⭐⭐⭐⭐

Definition

Testing complete software system.

Example

Complete ATM System Testing

Advantages

- Verifies entire system

---

4. ACCEPTANCE TESTING ⭐⭐⭐⭐

Definition

Customer validates software.

Types

Alpha Testing

Developer's site.

Beta Testing

Customer site.

TEST CRITERIA ⭐⭐⭐⭐

Definition

Conditions used to determine whether testing is sufficient.

Examples

- Statement Coverage
- Branch Coverage
- Path Coverage

TEST CASE DESIGN ⭐⭐⭐⭐⭐

Most Expected

Definition

Process of creating test cases.

Test Case Format

Test Case ID	**TC01**
Input	Username, Password
Expected Output	Login Success
Actual Output	Login Success
Result	Pass

Example

Input:

Username = Admin

Password = 1234

Expected:

Login Successful

TEST ORACLE ⭐⭐⭐⭐

Frequently Asked

Definition

Mechanism used to determine expected output.

Example

Calculator

2 + 2 = 4

Expected output = Oracle

Types

- Human Oracle
- Document Oracle
- Existing System Oracle

TEST TECHNIQUES ⭐⭐⭐⭐⭐

Very Important

---

1. BLACK BOX TESTING ⭐⭐⭐⭐⭐

Most Important

Definition

Testing without knowledge of internal code.

Focus

Input → Output

Example

ATM

Input:

PIN

Output:

Access Granted

Tester doesn't know internal code.

Techniques

Equivalence Partitioning

Boundary Value Analysis

Decision Table

Advantages

✅ User perspective

✅ No coding knowledge needed

Disadvantages

❌ Internal logic not tested

---

2. WHITE BOX TESTING ⭐⭐⭐⭐⭐

Most Important

Definition

Testing based on internal program structure.

Focus

Code Logic

Example

Checking loops and conditions.

Techniques

Statement Coverage

Branch Coverage

Path Coverage

Advantages

✅ Thorough testing

✅ Finds hidden errors

Disadvantages

❌ Programming knowledge required

BLACK BOX vs WHITE BOX TESTING

⭐⭐⭐⭐⭐

Black Box Testing	White Box Testing
No code knowledge	Code knowledge required
Tests functionality	Tests internal structure
User view	Developer view
Input & Output	Logic & Paths
Easy	Complex
Functional Testing	Structural Testing

🔥 Most Repeated RGPV Question

TESTING TOOLS ⭐⭐⭐

Examples:

- Selenium
- JUnit
- TestNG
- LoadRunner
- JMeter

Selenium

Web application testing.

JUnit

Java Unit Testing.

JMeter

Performance Testing.

OBJECT ORIENTED ANALYSIS & DESIGN (OOAD) ⭐⭐⭐⭐

Frequently Asked

Definition

Software development approach based on objects and classes.

Basic Concepts

Object

Real-world entity.

Example:

Student

Class

Blueprint of object.

Example:

Student Class

Encapsulation

Data hiding.

Inheritance

Reuse properties.

Polymorphism

Many forms.

OOAD vs STRUCTURED APPROACH

⭐⭐⭐⭐

OOAD	Structured
Object Based	Function Based
Reusable	Less Reusable
Easy Maintenance	Difficult Maintenance
Uses Classes	Uses Functions
Modern Approach	Traditional Approach

UNIT–4 IMPORTANT QUESTIONS

2 Marks

1. Define Software Testing.
2. What is Static Analysis?
3. What is Dynamic Analysis?
4. Define Test Case.
5. What is Test Oracle?

---

5 Marks

1. Explain Unit Testing.
2. Explain Integration Testing.
3. Explain System Testing.
4. Explain Test Case Design.
5. Explain Testing Process.

---

7 Marks

1. Black Box vs White Box Testing.
2. Static vs Dynamic Analysis.
3. Explain Testing Levels.
4. Explain Test Oracle.
5. Explain OOAD.

---

10 Marks

1. Explain Software Testing Process.
2. Explain Black Box Testing with techniques.
3. Explain White Box Testing with techniques.
4. Explain Testing Levels in detail.
5. Compare OOAD and Structured Software Engineering.

PYQ ANALYSIS & 2026 PREDICTION

Topic	Trend	Probability
Black Box Testing	Frequently Asked	Very High
White Box Testing	Frequently Asked	Very High
Testing Levels	Frequently Asked	Very High
Static vs Dynamic Analysis	Frequently Asked	Very High
Test Case Design	Frequently Asked	High
Test Oracle	Sometimes Asked	High
OOAD	Sometimes Asked	High

LAST NIGHT REVISION SHEET

✅ Static Analysis = No Execution

✅ Dynamic Analysis = Execution

✅ Error → Fault → Failure

✅ Testing Levels = Unit → Integration → System → Acceptance

✅ Black Box = Input/Output

✅ White Box = Code Logic

✅ Alpha = Developer Site

✅ Beta = Customer Site

✅ Selenium = Web Testing

✅ JUnit = Java Testing

✅ OOAD = Object-Based Design