Chapter 8: Software Testing

1. Program Testing

- Testing aims to show that a program does what it is intended to do and to discover defects before deployment.
- It involves executing a program with artificial data and checking results for errors and anomalies.
- Testing reveals the presence of errors but cannot confirm their absence.
- Part of the broader verification and validation (VV) process, which includes static validation techniques.

2. Goals of Program Testing

- To demonstrate to developers and customers that the software meets its requirements.
- To uncover situations where the software behavior is incorrect, undesirable, or does not conform to its specification.

3. Validation and Defect Testing

- **Validation Testing:** Ensures the system performs correctly with expected test cases.
- **Defect Testing:** Designed to expose defects with deliberately obscure test cases.

4. Verification vs Validation

- **Verification:** "Are we building the product right?" - Software should conform to its specification.
- **Validation:** "Are we building the right product?" - Software should do what the user requires.

5. VV Confidence

- Establish confidence that the system is fit for purpose, depending on its purpose, user expectations, and marketing environment.

6. Inspections and Testing

- **Inspections:** Static process analyzing the system representation to discover problems.
- **Testing:** Dynamic process executing the system with test data to observe behavior.

7. Stages of Testing

- **Development Testing:** Testing during development to discover bugs and defects.
- **Release Testing:** Separate team tests a complete system version before release.
- **User Testing:** Users test the system in their environment.

8. Development Testing

- **Unit Testing:** Testing individual program units or object classes.
- **Component Testing:** Testing integrated units to create composite components.
- **System Testing:** Testing the integrated system as a whole.

9. Automated Testing

- Use test automation frameworks (e.g., JUnit) to write and run tests without manual intervention.

10. Interface Testing

- Detect faults due to interface errors or invalid assumptions.
- Types of interfaces: Parameter, Shared Memory, Procedural, Message Passing.

11. System Testing

- Integrating components and testing interactions.
- Ensures components are compatible and interact correctly.

12. Use-Case Testing

- Use-case scenarios help identify system interactions for testing.

13. Regression Testing

- Ensures changes have not broken previously working code.
- Automated testing simplifies regression testing.

14. Release Testing

- Testing a system release intended for external use.
- Separate team performs validation to ensure system meets requirements.

15. Performance Testing

- Tests the system's emergent properties like performance and reliability.
- Stress testing involves overloading the system to test failure behavior.

16. User Testing

- **Alpha Testing:** Users work with developers at the development site.
- **Beta Testing:** Users experiment with the software and report issues.
- **Acceptance Testing:** Customers decide if the software is ready for deployment.

17. Key Points

- Testing reveals the presence of errors, not their absence.
- Development team handles development testing; a separate team handles release testing.
- Automated tests and scenario testing are essential practices.
- Acceptance testing determines if the software is ready for operational deployment.

Scenarios and Exam Questions

Scenario 1: Weather Station Testing

- **Scenario:** Testing weather station operations like reportWeather, calibrate, test, startup, and shutdown.
- **State Transitions:**
  - Shutdown -¿ Running -¿ Shutdown
  - Configuring -¿ Running -¿ Testing -¿ Transmitting -¿ Running
  - Running -¿ Collecting -¿ Running -¿ Summarizing -¿ Transmitting -¿ Running
- **Example Test Case:**
  - Input: Request for weather report
  - Expected Output: Acknowledgment and return of a summarized report

Scenario 2: Mentcare System

- **Scenario:** Testing the prescription system for allergy warnings.
- **Requirements-Based Test:**
  - **Requirement:** Issuing a warning message if a patient is allergic to prescribed medication.
  - **Test:** Prescribe medication the patient is allergic to and check for the warning message.
  - **Requirement:** Prescriber must provide a reason if ignoring an allergy warning.
  - **Test:** Ignore the warning and ensure the system prompts for a reason.

Exam Questions

1. **What are the main goals of program testing?**
   - To demonstrate the software meets requirements.
   - To uncover situations where the software behavior is incorrect or undesirable.
2. **Describe the difference between validation testing and defect testing.**
   - **Validation Testing:** Ensures the system performs correctly with expected test cases.

- **Defect Testing:** Designed to expose defects with deliberately obscure test cases.

3. **What are the advantages of inspections over testing?**
   - Errors can be discovered without execution.
   - Inspections can be applied to incomplete systems.
   - Inspections can consider broader quality attributes.

4. **Explain the concept of regression testing and its importance.**
   - Regression testing ensures changes do not break previously working code.
   - It is essential for maintaining software quality over updates.