# Software Quality Assurance Study Guide

**Peer Reviews**

- **Buddy Checks:** informal verification technique in which the product is examined by the author and one other person.
- **Circulation Review:** Can be informal or strict. The product is circulated to each reviewer who reviews it and either attaches comments, questions and recommendations directly on the product or places them in a separate document.
    - Good for long distance review.
- **Technical Review:** Formal team eval to id3entify any discrepancies from specifications and standards.
- **Inspections:** formal verification technique in which products are examined in detail by a group of peers for the explicit purpose o detecting and identifying defects.
    - Process is led by a moderator who is not the author
- **Walkthroughs:** designed to be less formalin which products are examined by a group of peers for the purpose of finding defects, omissions, and contradictions.
    - Normally led by the author.
- **Structured Walkthroughs:** more like an inspection. Its like doing a walkthrough with more inspection requirements. The significant difference is that it is lead by the author.

**Testing**

- Purpose of Testing:
    - Establish confidence that a program or system does what it is supposed to do.
    - Make lack of quality visible
    - Execute a program with the intent of finding errors.
    - Exercise a component to verify that it satisfied a specific requirement.
    - Provide continual assessment of whether the software being produced will meet the needs of the user.
- Key Testing Principles:
    - Complete testing is not possible
    - Testing is creative and difficult
    - Testing must be planned.
    - Testing requires independence.
    - Expected Results:
        - *Define the expected output or result.*
    - Invalid and unexpected input conditions.
    - What is the program expected to do?
    - Probability of more errors.
        - *The probability of the existence of more errors in a section of a program is proportional to the number of errors already found in the section.*
    - Good test case versus successful test case.
        - *A good test case is one that has a high probability of detecting an error that has not been discovered yet. A successful test case is on the detects an error that has not been discovered yet.*

**Types of Testing**

- **Unit Testing:** process of testing the individual components, subsystems, hardware components, such as programmable logic arrays, and software components such as subprograms, subroutines, or procedures.
    - Focuses on white box testing.
- **Integration Testing:** object is to test component interfaces and confirm that the components meet the interface requiremnt5s and that the components can indeed be assembled.
    - Should be used in the system design phase.
- **Systems Testing:** the first time at which the entire system can be tested against the systems requirements specification. Measures and determines what the system capabilities are and ends when the system capabilities have been measured and enough of the problems have been corrected to have confidence that the acceptance is ready to be executed.
- **Acceptance Testing:** purpose is to confirm that a system is ready for operational use and that the confidence built up in system testing is justified.
    - Tests whether the "requirements" have been met.
- **Regression Testing:** test to check that modifications or changes after testing have been retested.

**Test Coverage:** process of finding areas of a program not exercised by a set of test cases, creating additional test cases to increase coverage, and determining a quantitative measure of code coverage that serves an indirect measure of quality.

**Testing Matrices**

- **Test Plan:** a general guide created to define the scope of work that will be applied during testing.
- **Testing Matrix:** a prioritized testing list that includes browsers, clients, devices and platforms.

**Test Automation**

- **UI Automation Testing:**
    - Benefit of UI automation is that scenarios can be executed repeatedly and catch regressions introduced in the application.
    - They can be run on any platform or browser.
    - Usually Very Slow

**Black Box Testing**

- The goal is to be completely unconcerned with the internal behavior and structure of the program. Instead, concentrate on finding circumstances in which the program does not behave according to its specifications.
- Test the usability of the program.

**White Box Testing**

- Examine the internal structure of the program.

**Test Case**

- A test case must consist of two components:

- A description of the input data to the program
- A precise description of the correct output of the program for that set of input data.

**Inspections and Walkthroughs**

- three primary human testing methods are code inspections, walkthroughs, and user (or usability) testing.
- Inspections and walkthroughs are code-oriented methods.

- **Code Inspections:** a set of procedures and error-detection techniques for group code reading.
  - Inspection team usually consists of four people.
    - First of the four plays the role of moderator.
    - They are expected to be a competent programmer, but not the author.
    - Their duties include leading the session, recording all errors found and ensuring that the errors are subsequently corrected.
    - The second team member is the programmer.
    - The remaining team members usually are the program's designer and a test specialist.
  - During the session two activities occur:
    - The programmer narrates, statement by statement, the logic of the program.
    - The program is analyzed with respect to checklists of historically common programming errors.
  - **Common Errors:** an important part of the inspection process is the use of a check list to examine the program for common errors.
    - **Data Reference Errors:** does a referenced variable have a value that is unset or uninitialized?
    - **Data Declaration Errors:** have all variable been explicitly declared?
    - **Computation Errors:** Is it computed properly?
    - **Comparison Errors:** Compare errors.
    - **Control-Flow Errors:** Will every loop eventually terminate?
    - **Interface Errors:** Does the # of parameters equal # of arguments?
    - **Input/output Errors:** if files are explicitly declared, are their attributes correct?

- **Walkthroughs:** a set of procedures and error-detection techniques for group code reading.
  - Very similar to inspection.
    - Same length
    - Someone plays a role similar to that of the moderator.
    - Another plays role of secretary (they record all errors found)
    - Third person is the role of tester.
  - Rather than reading the code, the participants "play computer".
    - The person designated as the tester comes to the meeting armed with a small set of paper test cases. Each test case is mentally executed; the test data are "walked through" the logic of the program.
- **Desk Checking:** third human detection process that is the older way. Can be viewed as a one person inspection or walkthrough.
- **Peer Ratings:** the last human review process is not associated with program testing. (the objective is not to find errors). It is a technique of evaluating anonymous programs in terms of their overall quality, maintainability, extensibility, usability, and clarity.

- The purpose is to provide programmer with self-evaluation.

**Test-Case Design**

- Given constraints on time and cost, the key issue of testing becomes:
  - What subset of all possible test cases has the highest probability of detecting the most errors.
- The least effective is **random-input testing.**

- **White Box Testing**
  - Concerned with the logic (source code) of the program.
  - The ultimate white box test is the execution of every path in the program; but that is not realistic.
  - **Logic Coverage Testing**
    - **Decision Coverage:** you must write enough test cases that each decision has a true and a false outcome at least once.
    - **Condition Coverage:** stronger than decision coverage. Write enough test cases to ensure that each condition in a decision takes on all possible outcomes at least once.
    - **Decision/Condition Coverage:** each condition in a decision takes on all possible outcomes at least once and each decision takes on all possible outcomes at least once.
    - **Multiple-Condition Coverage:** requires that you write sufficient test cases such that all possible combos of condition outcomes in each decision are invoked at least once.

- **Black Box Testing**
  - The goal is to find areas in the program where it does not behave according to its specifications.
  - **Equivalence Partitioning**
    - When testing a program, you are limited to a small subset of all possible inputs.
    -