**1.0 Software Testing Fundamentals**

**1.1 Overview of Software Testing**

* **Definition and Importance:** Software testing is the process of evaluating a software application to find defects or bugs. It is crucial in the Software Development Life Cycle (SDLC) to ensure quality and reliability of the software.
* **Goals and Principles:**
  + Goals include finding defects, ensuring the software meets requirements, and verifying functionality.
  + Principles involve early testing, exhaustive testing is impossible, defects clustering, and the pesticide paradox.

**1.2 Levels of Software Testing**

* **Unit Testing:** Tests individual units or components of the software.
* **Integration Testing:** Tests how units interact and function together.
* **System Testing:** Tests the complete system as a whole.
* **Acceptance Testing:** Validates if the system meets the user requirements.

**1.3 Software Testing Techniques**

* **Static vs. Dynamic Testing:**
  + **Static Testing:** Reviews and inspections without executing code (e.g., walkthroughs, inspections).
  + **Dynamic Testing:** Executes code to validate functional behavior (e.g., unit testing, system testing).
* **White-box, Black-box, and Grey-box Testing:**
  + **White-box Testing:** Tests internal structures or workings of an application (code-based).
  + **Black-box Testing:** Tests functionality without knowledge of internal structures or workings.
  + **Grey-box Testing:** Combination of white-box and black-box testing; partial knowledge of internal workings.

**1.4 Types of Testing**

* **Functional Testing:**
  + **Regression Testing:** Ensures new changes don't adversely affect existing functionality.
  + **Smoke Testing:** Checks basic functionalities to ensure the application is stable.
  + **Sanity Testing:** Quick testing to ensure the stability of a particular functionality.
  + **User Acceptance Testing (UAT):** Validates if the system meets user requirements.
* **Non-Functional Testing:**
  + **Performance Testing:** Evaluates speed, responsiveness, and stability under various load conditions.
  + **Load Testing:** Tests system behavior under specific load conditions.
  + **Stress Testing:** Tests system behavior beyond normal operational capacity.
  + **Usability Testing:** Tests ease of use for end-users.
  + **Security Testing:** Tests vulnerabilities and risks associated with the system.

**1.5 Test Planning**

* **Creating Test Strategies and Test Plans:** Defines scope, objectives, resources, schedule, and risks of testing.
* **Resource and Environment Planning:** Identifies and allocates necessary resources and test environments.

**1.6 Test Design**

* **Writing Test Cases and Test Scripts:** Documents test scenarios, conditions, and expected results.
* **Designing Tests Based on Requirements and Specifications:** Ensures alignment with functional and non-functional requirements.

**1.7 Test Execution**

* **Executing Test Cases and Recording Results:** Implements tests, records outcomes, and compares actual vs. expected results.
* **Managing Test Cycles:** Iterative process of planning, executing, and evaluating tests.

**1.8 Defect Management**

* **Defect Lifecycle Management:** Processes for detecting, documenting, prioritizing, fixing, and verifying defects.
* **Reporting and Tracking Defects Using Tools:** Utilizes defect tracking systems to manage and monitor defects throughout their lifecycle.