**1. Preparation phase**

* Requirement analysis: Review all system requirements and specifications to understand what needs to be tested.
* Test plan development: Develop a comprehensive test plan that outlines the scope, objectives, resources, schedule, and risk assessment.
* Test case design: Develop detailed test cases that cover all functional and non-functional requirements. This includes both positive and negative test cases.
* Test environment setup: Prepare a test environment that is as close to the production environment as possible. Make sure all necessary hardware, software, network configuration, and other dependencies are in place.

**2. Test execution phase**

* Smoke testing: Perform initial testing to ensure that the basic functions of the application are functioning properly. This helps determine whether the system is stable enough for further testing.
* Functional testing: Verify that the system performs all required functions correctly. This includes:
* Unit testing: Test the correctness of individual components.
* Integration testing: Test the interactions between integrated units or components.
* System testing: Test the entire system to ensure that it meets the specified requirements.
* Non-functional testing: Evaluate the system's performance, security, availability, reliability, etc. This includes:
* Performance testing: Test the system's responsiveness, stability, and scalability under various conditions.
* Security testing: Identify vulnerabilities and ensure that the system is protected from potential threats.
* Usability testing: Evaluate the user-friendliness and intuitiveness of the system.
* Compatibility testing: Ensure that the system can run on different devices, browsers, and operating systems.
* Reliability testing: Verify the system's ability to run without problems under expected conditions.

**3. Defect tracking and management**

* Defect reporting: Record all found defects in the defect tracking system, including details such as severity, steps to reproduce, expected results vs. actual results.
* Defect classification: Prioritize defects based on their severity and impact on the system.
* Defect resolution: Work with the development team to fix defects.
* Defect retesting: Verify that the fix solves the problem and does not introduce new defects.

**4. Regression testing**

* Regression test planning: Identify test cases that need to be re-executed to ensure that recent changes do not affect existing functionality.
* Regression test execution: Execute the identified test cases and verify the stability of the system after the changes.

**5. End of testing**

* Test summary report: Record the test results, including passed and failed test cases, defect summary, and overall quality assessment.
* Test artifact archiving: Store all test artifacts, including test plans, test cases, test scripts, and test results, for future reference.
* Post-test review: Conduct a review meeting with stakeholders to discuss test results, lessons learned, and any necessary improvements.

**6. Continuous Improvement**

* Feedback Integration: Use feedback from the testing phase to improve future testing processes and practices.
* Training and Development: Provide training to the testing team based on identified knowledge or skill gaps.