System Testing

System testing is a level of software testing that focuses on evaluating the entire integrated system to verify that it meets specified requirements and functions correctly as a whole. It aims to assess the system's compliance with functional, performance, security, and other non-functional requirements before it is deployed to production. Here are key aspects of system testing:

1. **Scope**: System testing verifies the behavior and performance of the entire system, including all components, modules, interfaces, and interactions. It tests both functional and non-functional aspects of the system, such as usability, reliability, scalability, and security.
2. **Integration**: System testing is conducted after integration testing, where individual components or modules are combined and tested as a group. It ensures that the integrated system operates as intended and that all components work together seamlessly.
3. **Test Environment**: System testing requires a test environment that closely resembles the production environment in terms of hardware, software, configurations, and data. The test environment should replicate real-world conditions to validate the system's behavior accurately.
4. **Test Cases**: Test cases for system testing are derived from functional and non-functional requirements, user stories, use cases, and system specifications. They cover various scenarios, inputs, outputs, and interactions to ensure comprehensive test coverage.
5. **Functional Testing**: System testing includes functional testing to verify that the system performs all specified functions correctly. Test cases validate individual features, business processes, user interactions, and system workflows.
6. **Non-Functional Testing**: System testing also encompasses non-functional testing to assess performance, usability, reliability, security, and other quality attributes of the system. It includes tests such as performance testing, usability testing, reliability testing, security testing, and compatibility testing.
7. **Regression Testing**: System testing may involve regression testing to ensure that new changes or fixes do not introduce regressions or unintended side effects in existing functionality. Regression test suites are executed to validate that previously tested features still work as expected.
8. **User Acceptance Testing (UAT)**: While system testing focuses on validating the system from a technical perspective, user acceptance testing involves end-users or stakeholders validating the system from a business perspective. UAT typically follows system testing and assesses whether the system meets user requirements and expectations.
9. **Defect Management**: During system testing, defects or issues identified are reported, tracked, and managed using a defect tracking system. Defects are prioritized, assigned, fixed, and retested to ensure they are resolved satisfactorily.
10. **Documentation**: System testing activities are documented to provide traceability, transparency, and accountability. Test plans, test cases, test reports, and other testing artifacts are created to document the testing process and results.