**Testing Methodologies**

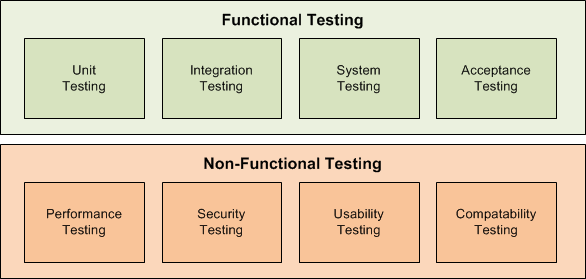
**What are test Methodolgies?**

Testing methodologies are the strategies and approaches used to test a particular product to ensure it is fit for purpose. Testing methodologies usually involve testing that the product works in accordance with its specification, has no undesirable side effects when used in ways outside of its design parameters and worst case will fail-safely (e.g. a nuclear reactor will shut down on failure).

**What are Software Testing Methodologies?**

Software testing methodologies are the different approaches and ways of ensuring that a software application in particular is fully tested. Software testing methodologies encompass everything from unit testing individual modules, integration testing an entire system to specialized forms of testing such as security and performance.

[testing methodology](http://www.inflectra.com/SpiraTest/Highlights.aspx?feature=Test-Case-Management) for making sure that software products/systems being developed have been fully tested to make sure they [meet their specified requirements](http://www.inflectra.com/SpiraTest/Highlights.aspx?feature=Requirements-Management) and can successfully operate in all the anticipated environments with the required usability and security.



**Functional Testing**

The functional testing part of a testing methodology is typically broken down into four components - unit testing, integration testing, system testing and acceptance testing.

**Unit Testing**

The [Unit testing](http://www.inflectra.com/SpiraTest/Integrations.aspx?type=Unit-Test-Frameworks) part of a testing methodology is the testing of individual software modules or components that make up an application or system.

These tests are usually written by the developers of the module and in a test-driven-development methodology (such as Agile, Scrum or XP).

**Integration Testing**

The Integration testing part of a testing methodology is the testing of the different modules/components that have been successfully unit tested when integrated together to perform specific tasks and activities (also known as scenario testing or End to End testing (E2E)). This testing is usually done with a combination of [automated functional tests](http://www.inflectra.com/SpiraTest/Highlights.aspx?feature=Automated-Testing) and manual testing.

**System Testing**

The system testing part of a testing methodology involves testing the entire system for errors and bugs. This test is carried out by interfacing the hardware and software components of the entire system (that have been previously unit tested and integration tested), and then testing it as a whole. This testing is listed under the black-box testing method,

**Acceptance Testing**

The acceptance testing part of a testing methodology is the final phase of functional software testing and involves making sure that all the product/project requirements have been met and that the end-users and customers have tested the system to make sure it operates as expected and meets all their defined requirements:

**User Acceptance Testing**

(UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment.

**Non-Functional Testing**

Testing the application against defined technical qualities. for example: vulnerability, scalability, usability.

## Performance, Load, Stress Testing

performance testing is measuring how a system behaves under an increasing load (both numbers of users and data volumes),

load testing is verifying that the system can operate at the required response times when subjected to its expected load,

and stress testing is finding the failure point(s) in the system when the tested load exceeds that which it can support.

## Security, Vulnerability Testing

Security testing tests the software for confidentiality, integrity, authentication, availability, and non-repudiation. Individual tests are conducted to prevent any unauthorized access to the software code.

**Usability Testing**

The usability testing part of a testing methodology looks at the end-user usability aspect of the software. The ease with which a user can access the product forms the main testing point. Usability testing looks at five aspects of testing, - learnability, efficiency, satisfaction, memorability, and errors.

**Compatibility Testing**

The compatibility part of a testing methodology tests that the product or application is compatible with all the specified operating systems, hardware platforms, web browsers, mobile devices, and other designed third-party programs (e.g. browser plugins). Compatibility tests check that the product works as expected across all the different hardware/software combinations and that all functionality is consistently supported.