Functional versus Non-Functional Requirements and Testing

One of the fundamental objectives of any project is to collect both the functional and non-functional requirements. These need to be kept in balance and harmony, and most importantly not compromised as the project progresses, as happened with the HMS Titanic's construction project.

Thomas Andrews was the one of the *principal architects* of the three White Star super liners, and accountable for turning the vision into reality. He was also *the project leader/manager*. Involved from the outset of the Titanic project he took responsibility for the requirements, both functional and non-functional.

Functional Requirements

The official definition for a functional requirement specifies what the system should do:
"A requirement specifies a function that a system or component must be able to perform."

Functional requirements specify specific behavior or functions, for example:
"Display the heart rate, blood pressure and temperature of a patient connected to the patient monitor."

Typical functional requirements are:

- Business Rules
- Transaction corrections, adjustments, cancellations
- Administrative functions
- Authentication
- Authorization –functions user is delegated to perform
- Audit Tracking
- External Interfaces
- Certification Requirements

- Reporting Requirements
- Historical Data
- Legal or Regulatory Requirements

Non-Functional Requirements

The official definition for a non-functional requirement specifies how the system should behave:
"A non-functional requirement is a statement of how a system must behave, it is a constraint upon the systems behavior."

Non-functional requirements specify all the remaining requirements not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviors, for example: *"Display of the patient's vital signs must respond to a change in the patient's status within 2 seconds."*

Typical non-functional requirements are:

- Performance - Response Time, Throughput, Utilization, Static Volumetric
- Scalability
- Capacity
- Availability
- Reliability
- Recoverability
- Maintainability
- Serviceability
- Security
- Regulatory
- Manageability
- Environmental
- Data Integrity

- Usability
- Interoperability

Non-functional requirements specify the system's 'quality characteristics' or 'quality attributes'. Potentially many different stakeholders have an interest in getting the non-functional requirements right. This is because for many large systems the people buying the system are completely different from those who are going to use it (customers and users).

Requirements Traceability Matrix

This matrix is used to trace project life cycle activities and work products to the project (or business) requirements. The matrix establishes a thread that traces all requirements from identification through to implementation, that is business requirements to system requirements and if necessary component requirements. Traceability is Bi-directional and so is used to manage change and provides the basis for test planning. Traceability tracks the allocation of each requirement to the following:

- Design documents and build components
- Customer acceptance criteria
- Test cases and results (pass/fail) for chosen test type
- Defines impact of change to any elements, requirements, or tests

Functional Testing

Functional testing is concerned with the functional requirements and covers how well the system executes its functions. These include user commands, data manipulation, searches and business processes, user screens, and integrations.

Non-Functional Testing

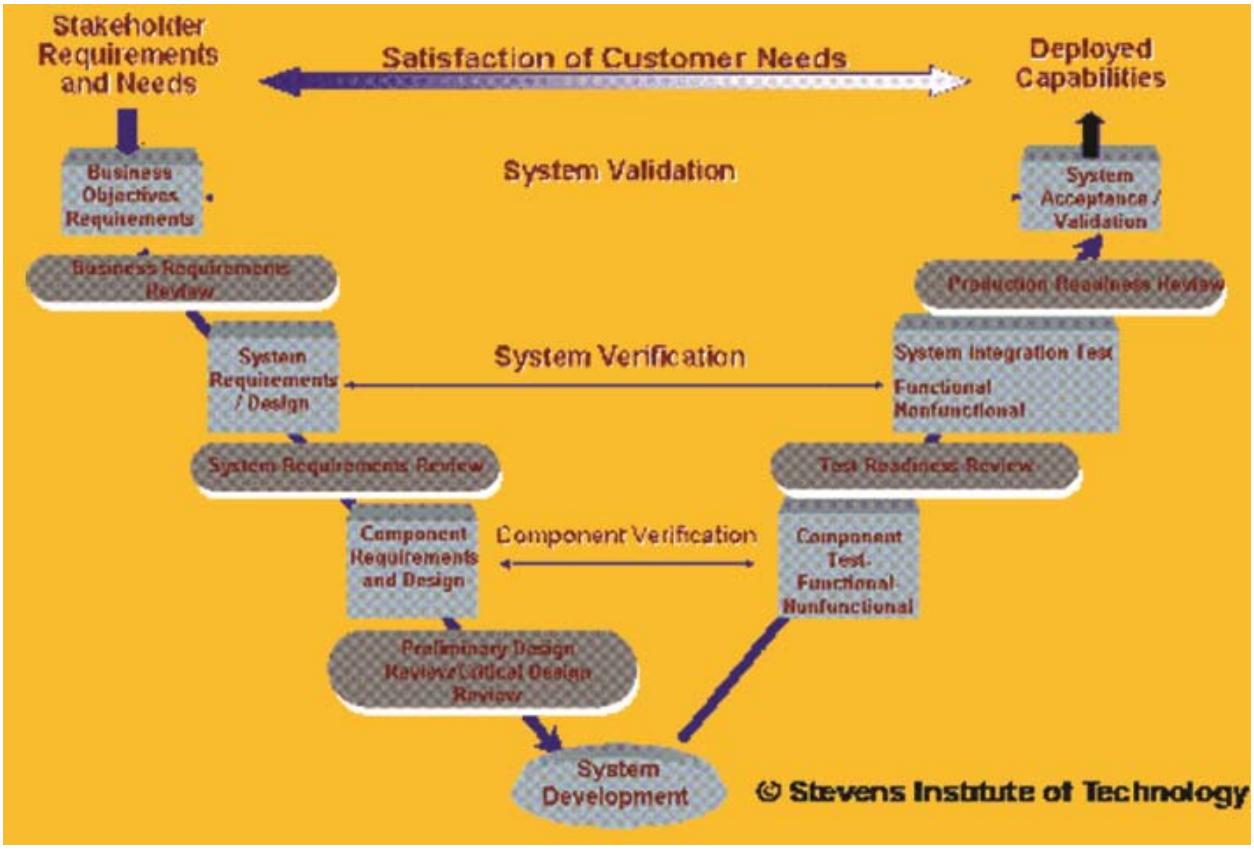
Non-functional testing is concerned with the non-functional requirements and is designed to evaluate the readiness of a system according to several criteria not covered by functional testing. It enables the measurement and comparison of the testing of non-functional attributes of software systems.

- Performance Testing
- Security Testing
- Usability Testing

- Dependability Testing
  - Reliability, Maintainability, Availability, Recoverability
- Miscellaneous Testing
  - Interoperability, Compatibility, Portability, Configuration, Installability

"V" Model

Testing does not have to occur once the 'code' has been delivered. It can start early with analyzing the requirements and creating test criteria of 'What' you need to test. The process for doing this is called the “V” model. It decomposes requirements and testing. Because it allows testing and coding as a parallel activity which enables the changes to occur more dynamic.



The cost of catching and correcting errors early in the cycle is generally under 20% of the cost post implementation.