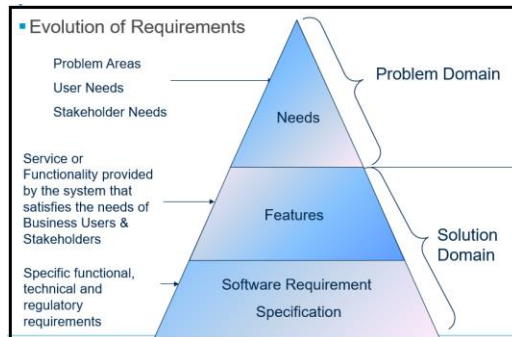


Evolution and Types of Requirements

Evolution of Requirements



Evolution involves the translation of stakeholder requests into a set of system features. These in turn are detailed into specifications for functional and nonfunctional requirements. Detailed specifications are translated into test procedures, design, and user documentation.

Who provides the Requirements?

The “Stakeholders” are individuals who affect or are affected by the proposed software product.

Different categories of stakeholders:

1. Customers
2. End Users
3. Development Team
 - i. Business Analyst
 - ii. Designers
 - iii. Testers

Requirement Categories:

Requirements are typically placed into two categories.

A.Functional: Functional Requirements, which say what the system should do or how it should work

B.Non-Functional: Non-Functional Requirements, which say what constraints there are on the system and its development. They represent quality attributes of the system.

Types of Requirements:

1. **Functional Requirement:** Defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs.”
2. **User Requirements:** These are Existing system functioning, Basic System operations and work flow, business rules etc.
3. **User Interfaces:** The user interface provides means of: Input, allowing the users to manipulate a system Output, allowing the system to produce the effects of the users' manipulation.
4. **Interface (or External API) Requirements:** These are Requirements of interfaces to external systems.
5. **Constraints:** These are Architectural, design, or implementation related restrictions, limitations, controls, checks or decisions – to be treated as requirements.
6. **Data (Informational) Requirements:** These Requirements that specify some mandatory property of a data type or value. Attributes of the objects- Data to create the object.

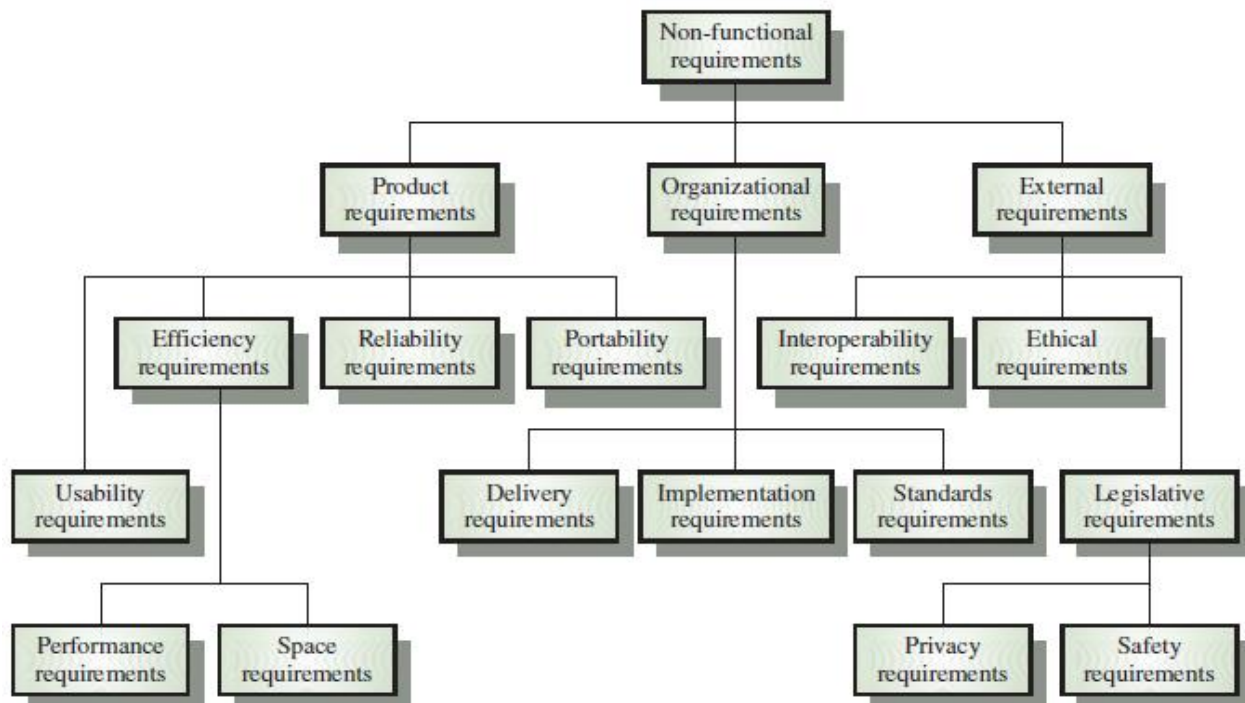
Non Functional Requirements:



Though functional requirement is core part of software application but we should not overlook Nonfunctional requirements.

Functionality	The degree to which the software satisfies stated needs as indicated by the following sub attributes: suitability, accuracy, interoperability, compliance, and security.
Usability	The degree to which the software is easy to use as indicated by the following aspects: understand ability, learnability, and operability.
Reliability	The amount of time that the software is available for use as indicated by the following aspects: maturity, fault tolerance, recoverability.
Performance	It is concerned with characteristics such as throughput, response time, recovery time, start-up time, and shutdown time.
Supportability	It is concerned with characteristics such as testability, adaptability, maintainability, compatibility, configurability, install ability, scalability, and localizability.

Other Non-Functional Requirements: FURPS“+”



What is a “Good” Software Requirement?

A user requirement is good if it is:

1. **Correct:** A set of requirements is correct if and only if every requirement stated therein represents something required of the system to be built.
2. **Complete:** No requirements or necessary information should be missing.
3. **Unambiguous:** Each requirement statement should have only one interpretation, and each requirement should be specified in a coherent, easy-to-understand manner.
4. **Consistent:** A consistent requirement does not conflict with other requirements in the requirement specification
5. **Verifiable:** A verifiable requirement is stated in such a way that it can be tested by inspection, analysis or demonstration
6. **Attainable:** The requirement should be doable within existing constraints such as time, money, and available resources.
7. **Traceable:** A traceable requirement has a unique identity or number. Each requirement should be traceable back to its source
8. **Manageable & Organized:** While documenting requirements sentences should be short, avoid paragraphs, define terms in glossary and avoid redundancies.



Think of these characteristics as a series of filters. A good requirement will pass through all eight filters.