**Functional vs Non Functional Requirements**

Requirements analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: *Functional* and *Non-functional requirements*.

**Functional Requirements:** These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

**Non-functional requirements:** These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.  
They basically deal with issues like:

* Portability
* Security
* Maintainability
* Reliability
* Scalability
* Performance
* Reusability
* Flexibility

Following are the differences between Functional and Non Functional Requirements

| **Functional Requirements** | **Non Functional Requirements** |
| --- | --- |
| A functional requirement defines a system or its component. | A non-functional requirement defines the quality attribute of a software system. |
| It specifies “What should the software system do?” | It places constraints on “How should the software system fulfill the functional requirements?” |
| Functional requirement is specified by User. | Non-functional requirement is specified by technical peoples e.g. Architect, Technical leaders and software developers. |
| It is mandatory. | It is not mandatory. |
| It is captured in use case. | It is captured as a quality attribute. |
| Defined at a component level. | Applied to a system as a whole. |
| Helps you verify the functionality of the software. | Helps you to verify the performance of the software. |
| Functional Testing like System, Integration, End to End, API testing, etc are done. | Non-Functional Testing like Performance, Stress, Usability, Security testing, etc are done. |
| Usually easy to define. | Usually more difficult to define. |
| **Example**  **1)** Authentication of user whenever he/she logs into the system. **2)** System shutdown in case of a cyber attack. **3)** A Verification email is sent to user whenever he/she registers for the first time on some software system. | **Example**  **1)** Emails should be sent with a latency of no greater than 12 hours from such an activity. **2)** The processing of each request should be done within 10 seconds **3)** The site should load in 3 seconds when the number of simultaneous users are > 10000 |

The functional requirements of the Internet of Things (IoT) refer to the capabilities and features that IoT systems and devices should possess to achieve their intended objectives. These requirements can vary depending on the specific application and use case, but some common functional requirements of IoT include:

1. **Connectivity**: IoT devices must be able to establish and maintain reliable connections with other devices, networks, or the internet. This may involve wireless technologies such as Wi-Fi, Bluetooth, Zigbee, LoRaWAN, cellular networks, or wired connections.
2. **Data Collection**: IoT devices should be capable of collecting data from various sensors, actuators, and other sources. The data can include environmental parameters, device status, user interactions, and more.
3. **Data Processing and Analytics**: IoT systems should be able to process and analyze the collected data in real-time or near-real-time. This may involve edge computing or cloud-based processing to derive meaningful insights and make data-driven decisions.
4. **Interoperability**: IoT devices and systems should be designed to interoperate with other devices and platforms seamlessly. This allows for easy integration and communication across different IoT ecosystems.
5. **Security**: Security is crucial in IoT to protect data, devices, and networks from unauthorized access, data breaches, and cyber-attacks. This involves encryption, authentication, access control, and secure communication protocols.
6. **Remote Management and Control**: IoT devices should support remote management and control, allowing users or administrators to monitor device status, update firmware, and adjust settings remotely.
7. **Scalability**: IoT systems must be scalable to accommodate a growing number of devices and handle increasing data volume without compromising performance.
8. **Energy Efficiency**: Many IoT devices are battery-powered or have limited power sources. Therefore, they should be designed for energy efficiency to prolong battery life and reduce maintenance.
9. **Localization and Tracking**: Some IoT applications require location awareness and tracking capabilities. Devices may need to determine their location or track the movement of assets or objects.
10. **Real-Time Response**: Certain IoT applications, such as industrial automation or smart grids, demand real-time response to events or emergencies.
11. **User Interface and Interaction**: IoT devices often require user interfaces to interact with users or display relevant information.
12. **Reliability and Availability**: IoT systems should be reliable and available, ensuring continuous operation and minimal downtime.
13. **Standards and Protocols Compliance**: IoT devices and systems should adhere to industry standards and communication protocols to ensure compatibility and interoperability.
14. **Data Privacy and Consent**: IoT systems must handle user data with proper privacy measures and obtain user consent when necessary.

These functional requirements lay the foundation for developing IoT solutions that meet specific use cases and industry needs. They ensure that IoT devices and systems are capable of performing their intended functions effectively and efficiently.

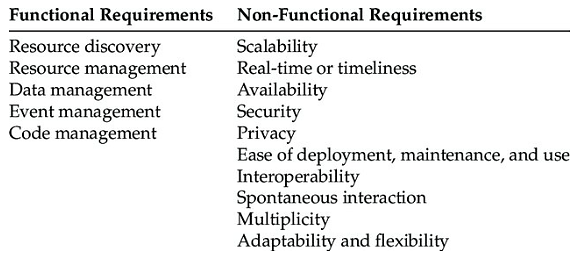


Figure 1. Functional and non-functional requirements for IoT middleware

A comprehensive list of functional requirements can be availed at <https://www.iot-a.eu/public/requirements/copy_of_requirements/>

Questions:

Knowledge Level 1: Remember

Bloom Keyword: Definition

1. What are functional requirements for IoT?
2. What are non-functional requirements for IoT?

Knowledge Level 2: Understand

Bloom Keyword: Explanation

1. Give an example of a functional requirement for IoT.
2. Give an example of a non-functional requirement for IoT.

Knowledge Level 3: Apply

Bloom Keyword: Application

1. Why are functional and non-functional requirements important for IoT?