|  |  |
| --- | --- |
| **Functional Requirement –** defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. It describes the behavior of the system as it relates to the system’s functionality.  Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.  Functional requirements drive the Application Architecture of a system. | **Non-functional Requirement** - specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. While functional requirements define what a system is supposed to do, non-functional requirements define how a system is supposed to be.  Non-functional requirements are often called "quality attributes" of a system. The non-functional requirement elaborates a performance characteristic of the system. |
| Functional requirements specification documents the operations and activities that a system must be able to perform. It includes:   * Descriptions of data to be entered into the system * Descriptions of operations performed by each screen * Descriptions of work-flows performed by the system * Descriptions of system reports or other outputs * Who can enter the data into the system * How the system meets applicable regulatory requirement   Examples:  **Interface requirements**   * Field 1 accepts numeric data entry. * Field 2 only accepts dates before the current date. * Screen 1 can print on-screen data to the printer.   **Business Requirements**   * Data must be entered before a request can be approved. * Clicking the Approve button moves the request to the Approval Workflow. * All personnel using the system will be trained according to internal SOP AA-101. | Non-functional requirements “quality attributes” is divided into two main categories:  **EXECUTION Quality (**observable during operation)**:**   * Safety * Security * Usability   **EVOLUTION Quality** (embodied in static structure of the system)**:**   * Testability * Maintainability * Extensibility * Scalability   \*these examples are incomplete, complete list is shown below in the last cell. . . |
| The following are a complete list of Non-functional requirements:   * [Accessibility](https://en.wikipedia.org/wiki/Accessibility) * [Auditability](https://en.wikipedia.org/wiki/Auditability) and control * [Availability](https://en.wikipedia.org/wiki/Availability) (see [service level agreement](https://en.wikipedia.org/wiki/Service_level_agreement)) * [Backup](https://en.wikipedia.org/wiki/Backup) * Capacity, current and forecast * [Certification](https://en.wikipedia.org/wiki/Certification) * [Compliance](https://en.wikipedia.org/wiki/Compliance_(regulation)) * [Configuration management](https://en.wikipedia.org/wiki/Configuration_management) * Cost * Dependency on other parties * Deployment * [Documentation](https://en.wikipedia.org/wiki/Documentation) * [Disaster recovery](https://en.wikipedia.org/wiki/Disaster_recovery) * Efficiency (resource consumption for given load) * Effectiveness (resulting performance in relation to effort) * Emotional factors (like fun or absorbing or has "Wow! Factor") * [Environmental protection](https://en.wikipedia.org/wiki/Environmental_protection) * [Escrow](https://en.wikipedia.org/wiki/Source_code_escrow) * Exploitability * [Extensibility](https://en.wikipedia.org/wiki/Extensibility) (adding features, and carry-forward of customizations at next major version upgrade) * Failure management * [Fault tolerance](https://en.wikipedia.org/wiki/Fault_tolerance) (e.g. Operational System Monitoring, Measuring, and Management) * Legal and [licensing](https://en.wikipedia.org/wiki/Software_license_agreement) issues or patent-infringement-avoidability * [Interoperability](https://en.wikipedia.org/wiki/Interoperability) * [Maintainability](https://en.wikipedia.org/wiki/Maintainability) (e.g. Mean Time To Repair - MTTR) * Management * Modifiability * [Network topology](https://en.wikipedia.org/wiki/Network_topology) * [Open source](https://en.wikipedia.org/wiki/Open_source) * [Operability](https://en.wikipedia.org/wiki/Operability) * [Performance](https://en.wikipedia.org/wiki/Computer_performance) / response time ([performance engineering](https://en.wikipedia.org/wiki/Performance_engineering)) * [Platform](https://en.wikipedia.org/wiki/Platform_(computing)) compatibility * [Privacy](https://en.wikipedia.org/wiki/Privacy) (compliance to [privacy laws](https://en.wikipedia.org/wiki/Privacy_law)) * [Portability](https://en.wikipedia.org/wiki/Software_portability) * [Quality](https://en.wikipedia.org/wiki/Quality_(business)) (e.g. faults discovered, faults delivered, fault removal [efficacy](https://en.wikipedia.org/wiki/Efficacy)) * [Readability](https://en.wikipedia.org/wiki/Computer_programming#Readability_of_source_code) * [Reliability](https://en.wikipedia.org/wiki/Reliability_engineering) (e.g. Mean Time Between/To Failures - MTBF/MTTF ) * Reporting * [Resilience](https://en.wikipedia.org/wiki/Resilience_(network)) * Resource constraints (processor speed, memory, disk space, network bandwidth, etc.) * [Response time](https://en.wikipedia.org/wiki/Response_Time#Data_processing) * [Reusability](https://en.wikipedia.org/wiki/Reusability) * [Robustness](https://en.wikipedia.org/wiki/Robustness_(computer_science)) * [Safety](https://en.wikipedia.org/wiki/Safety) or [Factor of safety](https://en.wikipedia.org/wiki/Factor_of_safety) * [Scalability](https://en.wikipedia.org/wiki/Scalability) (horizontal, vertical) * [Security](https://en.wikipedia.org/wiki/Security) (cyber and physical) * Software, tools, standards etc. [Compatibility](https://en.wikipedia.org/wiki/Backward_compatibility) * [Stability](https://en.wikipedia.org/wiki/Stability_Model) * [Supportability](https://en.wikipedia.org/wiki/Serviceability_(computer)) * [Testability](https://en.wikipedia.org/wiki/Software_testability) * [Throughput](https://en.wikipedia.org/wiki/Throughput) * [Transparency](https://en.wikipedia.org/wiki/Transparency_(behavior)) * [Usability](https://en.wikipedia.org/wiki/Usability) (Human Factors) by target user community * [Integrability](https://en.wikipedia.org/wiki/System_integration) ability to integrate components | |