

Non Functional Requirements

Arnav Gupta

December 5, 2024

Contents

1	Requirements Analysis	1
2	Non-Functional Requirements	3

1 Requirements Analysis

Architecture analysis involves requirement engineering.

Requirements come from users and stakeholders who have demands/needs.

A requirement engineer gathers raw requirements, analysis for consistency, feasibility, and completeness, writes a specification, and validates that gathered requirements reflect stakeholder needs.

Considerations during requirement gathering:

- business need vs requirement
- nice-to-have vs must-have
- system goal vs contractual requirement

Types of Requirements

- **Functional Requirements**
 - specify function of system
 - record, compute, transform, transmit
- **Non-Functional Requirements**

- Quality Requirements
 - * Specify how well system performs intended functions
 - * performance, usability, maintenance, reliability, portability
- Managerial Requirements
 - * when to deliver
 - * verification
 - * legal responsibilities if things go wrong
- Context/Environment Requirements
 - * range of conditions in which system should operate

Also data requirements handle system state and IO.

Software Specification: bridge between real-world environment (demands and needs of stakeholders) and software system

System Perspective: block diagram that describes system boundaries, users, and other interfaces

A good requirement spec is:

- correct: requirement reflects need
- complete: necessary requirements included
- unambiguous: all parties agree on meaning
- consistent: all parties match
- ranked for importance and stability: priority and expected changes per requirement
- modifiable: easy to change, maintaining consistency
- verifiable: possible to see whether requirements met
- traceable: to goals/purposes and to design/code
- necessary and feasible

2 Non-Functional Requirements

Often called quality attributes, specify how well system performs functions like speed, ease of use, security, and maintainability.

Non-functional requirements can be mandatory or not mandatory. Become more mandatory as market matures.

NFRs should be clear, concise, and measurable. NFR design requires customers and developers.

NFRs affect many high level subsystems, so must be considered during software architecture design.

Difficult to modify NFR once architecture complete.