

Non-functional Requirements

Katerina Meshkova
skype: Katerina_Meshkova

To measure is to know. If you can not measure it,
you can not improve it

Lord Kelvin (1824 - 1907)

What is a Requirement

A requirement is a statement of one of the following:

1. What a system must do
2. A known limitation or constraint on resources or design
3. How well the system must do what it does



What is NFR (1)

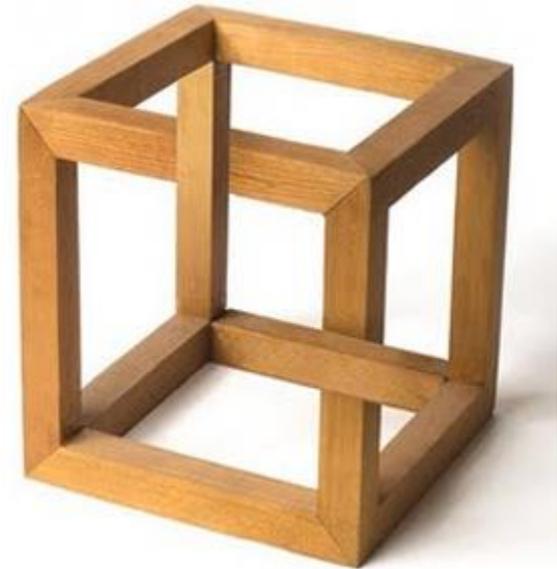
Non-functional requirements define the overall qualities or attributes of the resulting system

Non-functional requirements place restrictions on the product being developed, the development process, and specify external constraints that the product must meet

What is NFR (2)

Examples of NFR include safety, security, usability, reliability and performance requirements

Project management issues (costs, time, schedule) are often considered as non-functional requirements as well



NFR vs FR

Some properties of a system may be expressed either as a functional or non-functional property.

Example(NFR): The system shall ensure that data is protected from unauthorised access.

Example(FR): The system shall include a user authorization procedure where users must identify themselves using a login name and password. Only users who are authorised in this way may access the system data

Bunch of NFR

Accessibility

Audit and control

Availability

Backup

Capacity: current and forecast

Certification

Compatibility compliance

Configuration management

Dependency on other parties

Documentation

Disaster recovery

Efficiency

Effectiveness

Escrow

Extensibility

Failure management

Legal and licensing issues

Interoperability

Maintainability

Modifiability

Open Source

Operability

Patent-infringement-avoidability

Performance/response time

Platform compatibility

Price

Privacy

Portability

Quality

Recovery

Reliability

Resilience

Resource constraints

Response time

Robustness

Scalability

Security

Software, tools, standards

Stability

Safety

Supportability

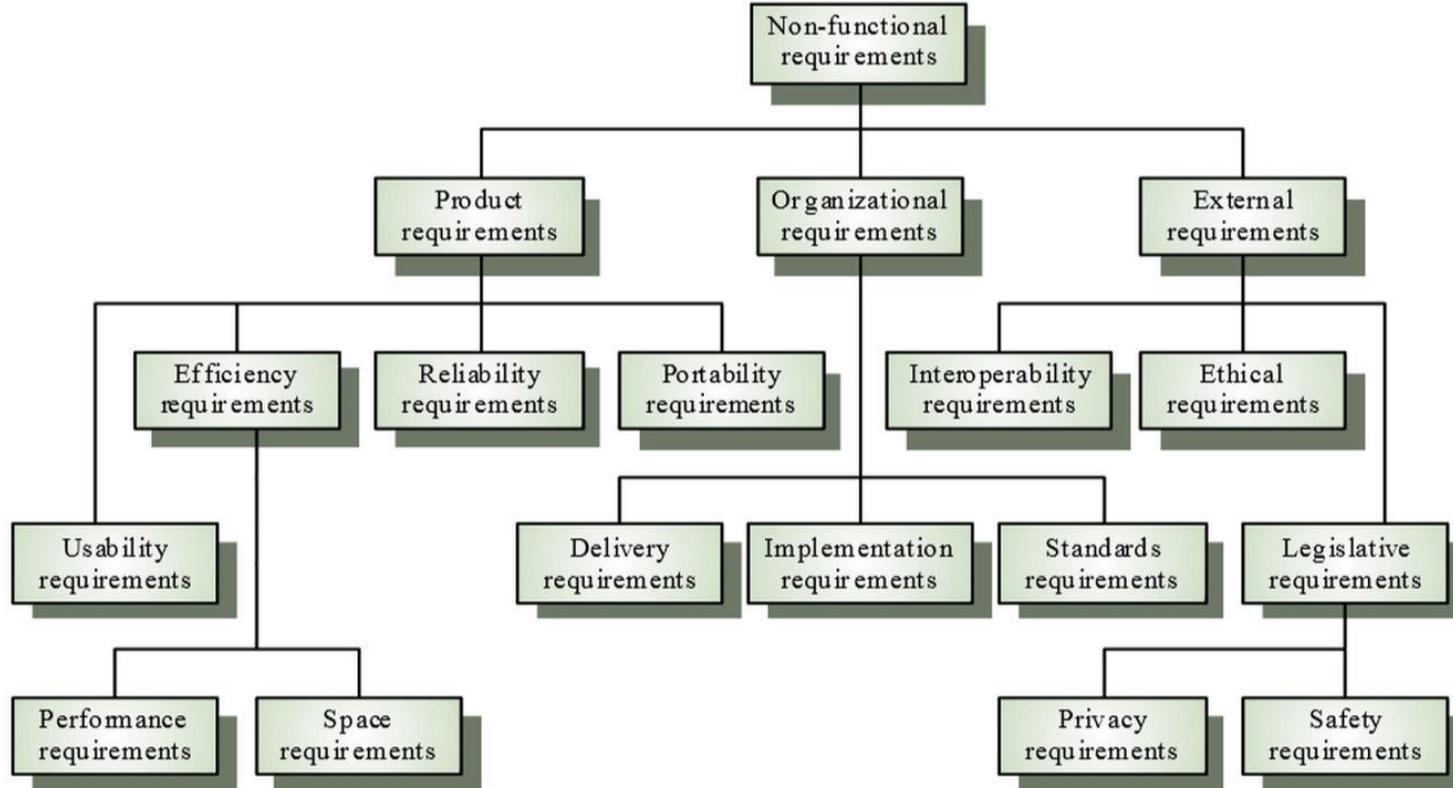
Testability

Usability

ISO-IEEE Std 830-1998

Performance requirements	Documentation requirements
Interface requirements	Security requirements
Operational requirements	Portability requirements
Resource requirements	Maintainability requirements
Verification requirements	Reliability requirements
Acceptance requirements	Quality and Safety requirements

NFRs Classification



NFR Minimal Checklist

- **Performance**
- **Availability**
- **Reliability**
- **Integrity**
- **Usability**
- **Recovery**
- **Compatibility**



Specifying NFR

Non-functional requirements need to be measurable

- Avoid subjective characterization: good, optimal, better, etc.

Values are not just randomly specified

- Must have a rational
- Stakeholder must understand trade-offs
- Important to rank and prioritize



Performance Measures

Usually with probabilities, confidence interval.

Response time

Number of events

processed/denied in
some interval of time

Throughput

Capacity

Usage ratio

Loss of information

Latency

Reliability Measures

Resistance to failure

Ability to perform a
required function under
stated conditions for a
specified period of time

Mean-time to failure

Defect rate

Degree of precision for
computations

Example:

The system defect rate shall
be less than 1 failure per
1000 hours of operation.

Availability Measures

Can be calculated based on Mean-Time to Failure (MTBF) and Mean-Time to Repair (MTTR)

- MTBF : Length of time between failures
- MTTR : Length of time needed to resume operation after a failure
- **Availability** = $MTBF / (MTBF + MTTR)$

Security Measures

There are at least two measures:

1. The ability to resist unauthorized attempts at usage

2. Continue providing service to legitimate users while under denial of service attack (resistance to DDoS attacks)

Maintainability Measures

Ability to make changes quickly and cost effectively

Coupling/cohesion

metrics, number of anti-patterns, cyclomatic complexity

Mean time to fix a defect, mean time to add new functionality

Quality/quantity of documentation

Why bother?

Essentially describe the target quality of the system, so ...

- Missing non-functional requirements (missing quality) impacts every functional requirement
- Affects adoption
 - Careful! Too much could impact time and cost



ISO-IEEE Std 830-1998

An SRS is complete if, and only if:

All significant requirements, whether relating to functionality, performance, design constraints, attributes, or external interfaces. In particular any external requirements imposed by a system specification should be acknowledged and treated.

How document NFR (1)

Depends on

Type of non-functional requirements

At what **level** they apply

Basic Types

- Process
- Data
- Both



How document NFR(2)

Levels

Whole solution

All automated (or all manual) components of the solution

Functional requirement

Whole process

Any level within a process hierarchy

An individual process step

All data

An individual data entity

An individual attribute on an entity

NFRs as user stories

As a customer, I want to be able to run your product on all versions of Windows from Windows 95 on.

As the CTO, I want the system to use our existing orders database rather than create a new one, so that we don't have one more database to maintain.

As a user, I want the site to be available 99.999 percent of the time I try to access it, so that I don't get frustrated and find another site to use.

Final thought

***Writing effective NFRs is
crucial for determining
whether product
performance and quality
goals have been met***



Questions are welcome!

Thank you!