

Dealing with Nonfunctional Requirements as the Hero, not the Witch

Andre Gous
Requirements Engineer & Founder
Precision Quality Software, Inc., Fallon, NV
andre@pqsw.com



Definitions

[Karl E. Wiegers, *Software Requirements, 2nd Edition*]

- **Functional Requirement (FR)**
A statement of a piece of required functionality or a behavior that a system will exhibit under specific conditions
- **Nonfunctional Requirement (NFR)**
A description of a property or characteristic that a software system must exhibit or a constraint it must respect, other than an observable system behavior



Aspects of NFRs

- Quality Attributes, ”*ility”
 - Availability
 - Performance
 - Portability
 - Scalability
 - Security, Testability, Usability, etc...
- Design Constraints
- External Interfaces

Section 1 of 3

1. Developing a way of thinking about NFRs using a simple non-software example to which everyone can relate
2. Dealing with unreasonable people in the development of NFRs
3. Developing NFRs for software in a reasonable working environment

Audience Participation: Car Door 1

- **Functional Requirement (FR)**
 - List the FRs for a Car Door (Hint: there are four)
- **Nonfunctional Requirement (NFR)**
 - List the quality attribute NFRs



What aspects would make a car door have better or worse quality?

Quality Attributes: Deceptive

- Essential yet appear to be a non-issue
- People who don't appreciate the complexity presume that quality of the product will be
 - High "enough"
 - Free
 - Automatic
- In reality none of the above is the case

Design Constraints

- Where the design is constrained due to project constraints or other reasons
- Typically due to having to adhere to pre-existing architecture, style, etc.
- Could be dictated by culture, economics, re-use of existing components, legal issues, etc.
- Often not explicitly conveyed

Audience Participation: Car Door 2

Constraints

- If there is a design constraint, specify it
- If you don't, the requirements might be implemented in a way you consider ludicrous



What design constraints could be applicable to a car door?

External Interface NFRs

- Describe connections between your system and the outside world
- Examples:
 - Interfacing with a particular device
 - Interfacing with another system
 - Reading or writing files in a particular format
 - Controlling particular pieces of hardware
 - Conforming to formal industry standards

Audience Participation: Car Door 3

External Interfaces

- Talking to the outside world
 - Importing data, exporting data
 - Talking to other systems



What external interfaces could be applicable to a car door?

Section 2 of 3

- Developing an understanding of NFRs using a simple non-software example to which everyone can relate
- Dealing with unreasonable people in the development of NFRs
- Developing NFRs for software in a reasonable working environment

Imagine a Witch-Hunt

- The unreasonable people who were supposed to provide the NFRs didn't do so
- The product is a disaster
- It's so bad that comedians mention it
- The unreasonable people duck the issue
- Senior management want to know how you, the requirements engineer, allowed this disaster to happen

Solution: Official Acceptance Tests

- Identify and list the stakeholders who should help you develop the NFRs
- Get senior management to:
 - Task *them*, not you, with developing a set of acceptance tests on behalf of the organization
 - Agree that if the product passes these tests, it's an official success. No other agenda!
- Ceremoniously celebrate the signing of that tasking agreement. Box 'em in.

Section 3 of 3

- Developing an understanding of NFRs using a simple non-software example to which everyone can relate
- Dealing with unreasonable people in the development of NFRs
- Developing NFRs for software in a reasonable working environment

A Major Challenge: Feasibility

- Finding the point of diminishing returns
- NFRs should be limited due to cost considerations, not lack of imagination & due diligence in developing NFRs
- Early on, prioritize by estimating costs of including a requirement vs. costs of omission
- If it's not in the specification, don't expect it in the product

Think of Achilles's Heel

- Even one vulnerability can cause disaster
- Even if you're not responsible, help the stakeholders build broad coverage
- Imagine you're being interviewed as to why the disaster-related NFR was left out
 - Can you make a credible case?
 - Yes? Document the decision
 - No? Go review the merits of the issue and work the issues and make a good decision

Don't Get Discouraged ...

- Developing high-quality NFRs is hard
- Practice makes you better over time
- Make sure you have the political aspect covered so you work in a safe situation
- Don't get burned. Don't trust anyone, not even nice or friendly-seeming people. Getting burned can terminally discourage you.

Quote Regarding Quality

***“The quality is
remembered long after
the price is forgotten.”
- Gucci***

Credits and Bibliography

- Karl Wieggers, “In Search of Excellent Requirements”
- Karl Wieggers, *Software Requirements, 2nd Edition*, Microsoft Press