Lecture # 3

Software Requirements Engineering

Dr. Shehnila Zardari

Kinds of Software Requirements

- Functional requirements
- Non-functional requirements
- Domain requirements
- Inverse requirements
- Design and implementation constraints

Non-Functional Requirements Discussion

- NFRs are very important to capture the emergent behavior of the system in these there major dimensions
- Product
  - Usability, reliability, portability, efficiency (performance, space)
- Organizational
  - Standards, implementation, delivery
- External
  - Interoperability, ethical, legislative (privacy, safety)

NFRs as Goals

- Non-functional requirements are sometimes written as general goals, which are difficult to verify
- They should be expressed quantitatively using metrics (measures) that can be objectively tested

Example: Goal converted into an NFR

- Goal (unverifiable)
  - The system should be easy to use by experienced controllers and should be organized in such a way that user errors are minimized
- Non-functional requirement (verifiable)
  - Experienced controllers shall be able to use all the system functions after a total of two hours' training. After this training, the average number of errors made by experienced users shall not exceed two per day

Metrics for Non-Functional Requirements (NFRs)

Metrics for NFRs - 1

Property	Measure
Speed	1. Processed transactions/second 2. Response time 3. Screen refresh time

Requirements related to "Speed" can use different measures to quantify the goal

Metrics for NFRs - 2

Property	Measure
Size	1. K bytes 2. Number of function points

Requirements related to “Size” can use different measures to quantify the goal

Metrics for NFRs - 3

Property	Measure
Ease of use	1. Training time 2. Number of help frames

Requirements related to “Ease of use” can use different measures to quantify the goal

Metrics for NFRs - 4

Property	Measure
Reliability	1. Mean time to failure 2. Probability of unavailability 3. Rate of failure occurrence 4. Availability

Requirements related to “Reliability” can use different measures to quantify the goal

Metrics for NFRs - 5

Property	Measure
Robustness	1. Time to restart after failure 2. Percentage of events causing failure 3. Probability of data corruption on failure

Requirements related to “Robustness” can use different measures to quantify the goal

Metrics for NFRs - 6

Property	Measure
Portability	1. Percentage of target-dependent statements 2. Number of target systems

Requirements related to “Portability” can use different measures to quantify the goal

Discussion on Metrics for NFRs

- With the help of these measures the NFRs can be verified quantitatively

- It should also be noted that the cost of quantitatively verifying each NFR may be very high

Kinds of Software Requirements

- Functional requirements
- Non-functional requirements
- Domain requirements
- Inverse requirements
- Design and implementation constraints

Domain Requirements

Domain Requirements - 1

- Requirements that come from the application domain and reflect fundamental characteristics of that application domain
- These can be both the functional or non-functional requirements

Domain Requirements - 2

- These requirements, sometimes, are not explicitly mentioned
- Domain experts find it difficult to convey domain requirements
- Their absence can cause significant dissatisfaction

Domain Requirements - 3

- Domain requirements can impose strict constraints on solutions. This is particularly true for scientific and engineering domains
- Domain-specific terminology can also cause confusion

Domain Requirements - 4

- Example:

In a commission-based sales businesses, there is no concept of negative commission. However, if care is not taken novice developers can be lured into developing systems, which calculate negative commission

Domain Requirements - 5

- Banking domain has its own specific constraints, for example, most banks do not allow over-draw on most accounts, however, most banks allow some accounts to be over-drawn

Kinds of Software Requirements

- Functional requirements
- Non-functional requirements
- Domain requirements
- Inverse requirements
- Design and implementation constraints

Inverse Requirements

Inverse Requirements - 1

- They explain what the system shall **not** do.

  Many people find it convenient to describe their needs in this manner

- These requirements indicate the indecisive nature of customers about certain aspects of a new software product

Inverse Requirements - 2

- Example:

The system shall not use red color in the user interface, whenever it is asking for inputs from the end-user

Kinds of Software Requirements

- Functional requirements
- Non-functional requirements
- Domain requirements
- Inverse requirements
- Design and implementation constraints

Design and Implementation Constraints

Design and Implementation Constraints - 1

- They are development guidelines within which the designer must work
- These requirements can seriously limit design and implementation options
- Can also have impact on human resources

Design and Implementation Constraints Examples

- The system shall be developed using the Microsoft .Net platform
- The system shall be developed using open source tools and shall run on Linux operating system

Summary

- Discussed different kinds of requirements including domain, inverse, and implementation constraints
- Requirements should be explored from different perspectives and categorized differently