Software Requirements Engineering (SE2001)

Dr. Syed Muazzam Ali Shah

HEC Approved Ph.D. Supervisor

(Assistant Professor)

Department of Software Engineering

NUCES-FAST Karachi

Contact#: (021) 111-128-128 Ext. 130

Website: sites.google.com/view/muazzam-kazmi/home

Official page: https://khi.nu.edu.pk/personnel/dr-syed-muazzam-ali-shah-2/

Google Scholar: https://scholar.google.com.tw/citations?hl=en&user=0vcfR-IAAAAJ

Non Functional Requirements 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

Non Functional Requirements as Goals

- **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)

- Speed/Performance
- Size
- Ease of Use
- Reliability
- Robustness
- Portability

Metrics for NFRs

Property	Measure
Speed/Performance	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

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

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

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

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

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.

Domain Requirements

- 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

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

Domain Requirements

- 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: 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 Examples

- 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.

Inverse Requirements

- 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: Example

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

Design and implementation constrains

- 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 constrains

- 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 constrains : Example

- 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.

THANK YOU

Dr. Syed Muazzam Ali Shah
muazzam.ali@nu.edu.pk