PXL – IT
42TIW1030 Software Analysis - SRS – IEEE 830

Week 07

Luc Doumen

Nathalie Fuchs

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

Writing the specification

- Three main sections
  - Introduction
  - Overall description
    - Constraints
  - Specific requirements
    - Functional requirements (grouped)
    - Quality requirements

- Therefore ➔

Putting together what we have gathered so far

Document Structure

- Need for structured contents (reference structures)
- Lots of standards and templates available
  - **IEEE 830-1998 (Recommended Practice for Software Requirements Specifications, www.ieee.org)**
  - Other examples
    - IEEE 1233–1998 (Reference structure “System Requirements Specification”)
    - Volere Framework of Atlantic System Guild (www.volere.co.uk)
    - Rational Unified Process (RUP)
    - V-Model XT (Germany)
    - Etc.

Using Standards & Templates

- Can and should be tailored to your project!
- Should at least contain:
  - The goals
  - The functional requirements
  - The non-functional requirements
    - Quality requirements
    - Constraints
  - A glossary
  - A list of abbreviations used

Requirements Document Template

- **Section 1: Introduction**
  - 1.1 The purpose of the product
  - 1.2 Document conventions
  - 1.3 Definitions, acronyms and abbreviations
  - 1.4 Scope of the product (context diagram)
  - 1.5 References

•Describe purpose of this SRS
•Describe intended audience

•Describe the content of the rest of the SRS
•Describe how the SRS is organized

•Define the vocabulary of the SRS (may reference appendix)

•Identify the software product
•Enumerate what the system will and will not do
•Describe user classes and benefits for each

•List all referenced documents including sources (e.g., Use Case Model and Problem Statement; Experts in the field)

Requirements Document Template

- **Section 2: Overall description**
  - 2.1 Product perspective
    - General overview, use cases, interfaces
  - 2.2 Product functions
    - A list of functionalities (incl. quality related) provided
    - Grouping should support a communication point
  - 2.3 User characteristics
  - 2.4 Constraints
  - 2.5 Assumptions and dependencies
  - 2.6 Apportioning of requirements
    - Requirements that may be delayed

•Present the business case and operational concept of the system
•Describe how the proposed system fits into the business context
•Describe external interfaces: system, user, hardware, software, communication
•Describe constraints: memory, operational, site adaptation

Requirements Document Template

- **Section 2: Overall description**
  - 2.1 Product perspective
    - General overview, use cases, interfaces
  - 2.2 Product functions
    - A list of functionalities (incl. quality related) provided
    - Grouping should support a communication point
  - 2.3 User characteristics
  - 2.4 Constraints
  - 2.5 Assumptions and dependencies
  - 2.6 Apportioning of requirements
    - Requirements that may be delayed

•Summarize the major functional capabilities
•Include the Use Case Diagram and supporting narrative (identify actors and use cases)
•Include Data Flow Diagram if appropriate

•Describe and justify technical skills and capabilities of each user class

Requirements Document Template

- **Section 2: Overall description**
  - 2.1 Product perspective
    - General overview, use cases, interfaces
  - 2.2 Product functions
    - A list of functionalities (incl. quality related) provided
    - Grouping should support a communication point
  - 2.3 User characteristics
  - 2.4 Constraints
  - 2.5 Assumptions and dependencies
  - 2.6 Apportioning of requirements
    - Requirements that may be delayed

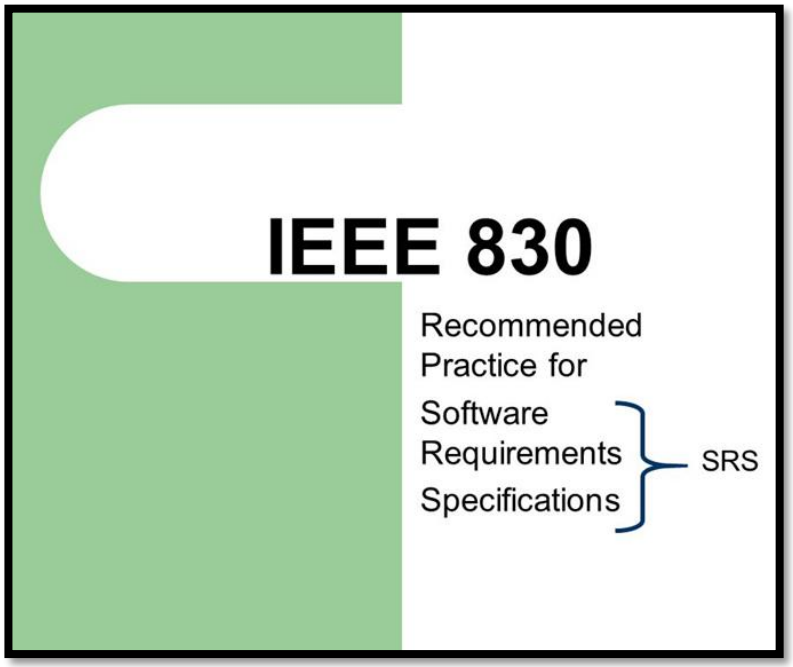
•Describe other constraints that will limit developer's options; e.g., regulatory policies; target platform, database, network software and protocols, development standards requirements

Requirements Document Template

- **Section 3: Specific Requirements**

  3.1 Functional requirements

  3.2 Quality requirements



Specify software requirements in sufficient detail to enable designers to design a system to satisfy those requirements and testers to verify requirements

State requirements that are externally perceivable by users, operators, or externally connected systems

Requirements should include, at a minimum, a description of every input (stimulus) into the system, every output (response) from the system, and all functions performed by the system in response to an input or in support of an output

(a) Requirements should have characteristics of high quality requirements
(b) Requirements should be cross-referenced to their source.
(c) Requirements should be uniquely identifiable
(d) Requirements should be organized to maximize readability

Requirements Document Template

- **Section 3: Specific Requirements**
  - 3.1 External Interfaces
  - 3.2 Functions
  - 3.3 Performance Requirements
  - 3.4 Logical Database Requirements
  - 3.5 Design Constraints
  - 3.6 Software System Quality Attributes
  - 3.7 Object Oriented Models

•Detail all inputs and outputs (complement, not duplicate, information presented in section 2)
•Examples: GUI screens, file formats

•Include detailed specifications of each use case, including collaboration and other diagrams useful for this purpose

•Include:
a) Types of information used
b) Data entities and their relationships

•Should include:
a) Standards compliance
b) Accounting & Auditing procedures

•The main body of requirements organized in a variety of possible ways:
a) Architecture Specification
b) Class Diagram
c) State and Collaboration Diagrams
d) Activity Diagram (concurrent/distributed)

Questions & answers