



Requirements Specification with the IEEE 830 and IEEE 29148 Standards

Based on slides by Miguel Garzón (2014), Gunter Mussbacher (2009) and
Stéphane Somé (2008)

with material from these standards:

IEEE 830-1998, ISO/IEC 12207, ISE/IEC/IEEE 29148:2011

Table of Contents

- Requirements Specification Document
- IEEE 830 Standard
- Relationship of IEEE 830 and ISO/IEC 12207
- ISO/IEC/IEEE 29148 Standard

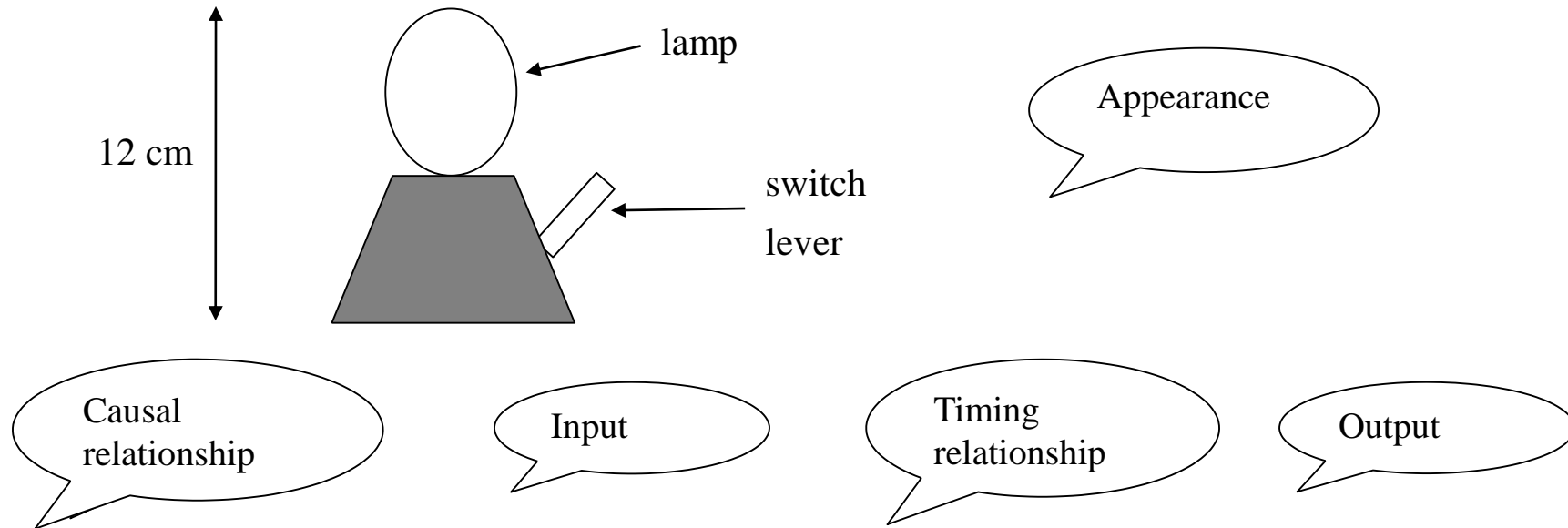
Requirements Specification Document (1)

- Clearly and accurately describes each of the essential requirements (functions, performance, design constraints, and quality attributes) of the system / software and its external interfaces
 - Defines the scope and boundaries of the system / software
- Each requirement must be described in such a way that it is feasible and objectively verifiable by a prescribed method (e.g., by inspection, demonstration, analysis, or test)
- Basis for contractual agreements between contractors or suppliers and customers
- Elaborated from elicitation notes

Requirements Specification Document (2)

- Specifications are intended to a diverse audience
 - Customers and users for validation, contract, ...
 - Systems (requirements) analysts
 - Developers, programmers to implement the system
 - Testers to check that the requirements have been met
 - Project Managers to measure and control the project
- Different levels of detail and formality is needed for each audience
- Different templates for requirements specifications
 - e.g. IEEE 830

Example Specification (1)



- When the switch lever is moved down, then, within 0.1 seconds, the lamp illuminates.
- When the switch lever is moved up, then, within 0.2 seconds, the lamp goes out.

Example Specification (2)

- Extract from the requirements specification
 - R1: The system shall provide illumination of at least 500 candela.
 - R2: The system shall fit within a cube with maximum width of 15cm.
 - R3: The illumination can be switched on and off by a human operator.
 - R4: The system shall respond to operator input within 0.5 seconds.
 - R5: The system shall have a built-in power supply which should be capable of maintaining continuous illumination for at least 4 hours.
 - etc
- Several **alternative** designs could satisfy these requirements

IEEE 830-1998 Standard

- Title of Standard
 - « *IEEE Recommended Practice for **Software** Requirements Specifications* »
- Describes the content and qualities of a good software requirements specification (SRS)
- Presents several sample SRS outlines

IEEE 830-1998 Standard – Objectives

- Help software **customers** to accurately describe what they wish to obtain
- Help software **suppliers** to understand exactly what the customer wants
- Help participants to:
 - Develop a **template** (format and content) for the software requirements specification (SRS) in their own organizations
 - Develop **additional documents** such as SRS quality checklists or an SRS writer's handbook



IEEE 830-1998 Standard – Benefits

- Establish the basis for **agreement** between the customers and the suppliers on what the software product is to do
- Reduce the **development effort**
 - Forced to consider requirements early → reduces later redesign, recoding, retesting
- Provide a basis for realistic **estimates** of costs and schedules
- Provide a basis for **validation** and **verification**
- Facilitate **transfer** of the software product to new users or new machines
- Serve as a basis for **enhancement** requests

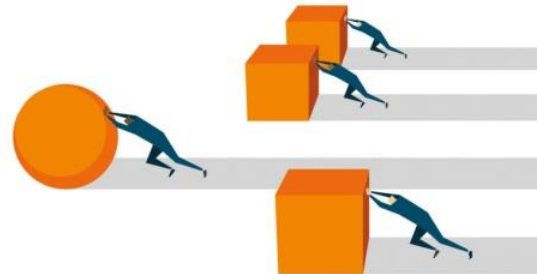


IEEE 830-1998 Standard – Considerations

- Section 4 of IEEE 830 (how to produce a good SRS)
 - Nature (goals) of SRS
 - Functionality, interfaces, performance, qualities, design constraints
 - Environment of the SRS
 - Where does it fit in the overall project hierarchy
 - Characteristics of a good SRS
 - Generalization of the characteristics of good requirements to the document
 - Evolution of the SRS
 - Implies a change management process
 - Prototyping
 - Helps elicit software requirements and reach closure on the SRS
 - Including design and project requirements in the SRS
 - Focus on external behavior and the product, not the design and the production process (describe in a separate document)

IEEE 830-1998 Standard – Considerations

- Nature of the SRS
- Functionality;
- External interfaces;
- Performance;
- Attributes;
- Design constraints imposed on an implementation.



IEEE 830-1998 Standard – Considerations

- Environment of the SRS
 - It is important to consider the part that the SRS plays in the total project plan;
 - The SRS writer(s) should be careful not to go beyond the bounds of that role. This means the SRS:
 - Should correctly define all of the software requirements;
 - Should not describe any design or implementation details;
 - Should not impose additional constraints on the software.

IEEE 830-1998 Standard – Considerations

- Characteristics of a good SRS:

- a) Correct;

- b) Unambiguous;

- Natural language pitfalls;
- Requirements specification languages;
- Representation tools.

- c) Complete (Use of TBDs);

- d) Consistent;

- e) Ranked for importance and/or stability (Degree of stability or necessity);

- f) Verifiable (finite cost-effective process);

- g) Modifiable (structure and style);

- h) Traceable.

- Backward traceability (i.e., to previous stages of development);
- Forward traceability (i.e., to all documents spawned by the SRS).

IEEE 830-1998 Standard – Considerations

- Section 4 of IEEE 830 (how to produce a good SRS)
 - Nature (goals) of SRS
 - Functionality, interfaces, performance, qualities, design constraints
 - Environment of the SRS
 - Where does it fit in the overall project hierarchy
 - Characteristics of a good SRS
 - Generalization of the characteristics of good requirements to the document
 - Evolution of the SRS
 - Implies a change management process
 - Prototyping
 - Helps elicit software requirements and reach closure on the SRS
 - Including design and project requirements in the SRS
 - Focus on external behavior and the product, not the design and the production process (describe in a separate document)

IEEE 830-1998 Standard – Structure of the SRS

- Section 5 of IEEE 830
- Contents of SRS
 - Introduction
 - General description of the software product
 - Specific requirements (detailed)
 - Additional information such as appendixes and index, if necessary

IEEE 830-1998 Standard – Section 1 of SRS

- Title
- Table of Contents
- 1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, Acronyms, and Abbreviations
 - 1.4 References
 - 1.5 Overview
- 2. Overall Description
- 3. Specific Requirements
- Appendices
- Index

•Describe purpose of this SRS
•Describe intended audience

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

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

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

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

IEEE 830-1998 Standard – Section 2 of SRS

- Title
- Table of Contents
- 1. Introduction
- 2. Overall Description
 - 2.1 Product Perspective
 - 2.2 Product Functions
 - 2.3 User Characteristics
 - 2.4 Constraints
 - 2.5 Assumptions and Dependencies
- 3. Specific Requirements
- 4. Appendices
- 5. Index

•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

•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

States assumptions about availability of certain resources that, if not satisfied, will alter system requirements and/or effect the design.

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

IEEE 830-1998 Standard – Section 3 of SRS (1)

- ...
- 1. Introduction
- 2. Overall Description
- 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
- 4. Appendices
- 5. Index

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**

IEEE 830-1998 Standard – Section 3 of SRS (2)

- ...
 - 1. Introduction
 - 2. Overall Description
 - 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
 - 4. Appendices
 - 5. Index
- 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)

IEEE 830-1998 Standard – Templates

- Annex A of IEEE 830
- Section 3 (Specific Requirements) may be organized in many different ways based on
 - Modes
 - User classes
 - Concepts (object/class)
 - Features
 - Stimuli
 - Response
 - Functional hierarchy

Relationship of IEEE 830 and ISO/IEC 12207 (1)

- 12207
 - Common framework for « **Software life cycle processes** »
 - ISO/IEC 12207 = IEEE/EIA 12207
- IEEE 830-1998 and IEEE/EIA 12207.1-1997 both place requirements on documents describing software requirements
- Annex B of IEEE 830 explains the relationship between the two sets of requirements for those who want to produce documents that comply with both standards simultaneously
- Such compliance may be required by customers when requesting proposals or issuing call for tenders

Relationship of IEEE 830 and ISO/IEC 12207 (2)

Table B.2—Coverage of generic description requirements by IEEE Std 830-1998

IEEE/EIA 12207.1-1997 generic content	Corresponding clauses of IEEE Std 830-1998	Additions to requirements of IEEE Std 830-1998
a) Date of issue and status	—	Date of issue and status shall be provided.
b) Scope	5.1.1 Scope	—
c) Issuing organization	—	Issuing organization shall be identified.
d) References	5.1.4 References	—
e) Context	5.1.2 Scope	—
f) Notation for description	4.3 Characteristics of a good SRS	—
g) Body	5. The parts of an SRS	—
h) Summary	5.1.1. Overview	—
i) Glossary	5.1.3 Definitions	—
j) Change history	—	Change history for the SRD shall be provided or referenced.

- Note: Table B.3 is more detailed and shows the correspondence between the two standards at the level of requirements types

ISO/IEC/IEEE 29148:2011

- ISO/IEC/IEEE 29148:2011: Systems and software engineering — Life cycle processes — Requirements engineering
 - <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6146379>
- This International Standard provides a unified treatment of the processes and products involved in engineering requirements throughout the life cycle of systems and software.
- Harmonizes IEEE 830, SWEBOK, and 7 other standards.
- More emphasis on characteristics of good requirements, RE activities and processes, operations (and operation context), and different information items (including their structures) such as specification of requirements for stakeholders, systems and software.
- Complies with ISO/IEC 15288 and ISO/IEC 12207

Stakeholder Requirements Specification Outline

- 1. Introduction**
 - 1.1 Business purpose
 - 1.2 Business scope
 - 1.3 Business overview
 - 1.4 Definitions
 - 1.5 Stakeholders
- 2. References**
- 3. Business management requirements**
 - 3.1 Business environment
 - 3.2 Goal and objective
 - 3.3 Business model
 - 3.4 Information environment
- 4. Business operational requirements**
 - 4.1 Business processes
 - 4.2 Business operational policies and rules
 - 4.3 Business operational constraints
 - 4.4 Business operational modes
 - 4.5 Business operational quality
 - 4.6 Business structure
- 5. User requirements**
- 6. Concept of proposed system**
 - 6.1 Operational concept
 - 6.2 Operational scenario
- 7 Project Constraints**
- 8. Appendix**
 - 8.1 Acronyms and abbreviations

Figure 6 — Example StRS Outline

System Requirements Specification Outline

1. Introduction

- 1.1 System purpose
- 1.2 System scope
- 1.3 System overview
 - 1.3.1 System context
 - 1.3.2 System functions
 - 1.3.3 User characteristics
- 1.4 Definitions

2. References**3. System requirements**

- 3.1 Functional requirements
- 3.2 Usability requirements
- 3.3 Performance requirements
- 3.4 System interface
- 3.5 System operations
- 3.6 System modes and states
- 3.7 Physical characteristics
- 3.8 Environmental conditions
- 3.9 System security
- 3.10 Information management
- 3.11 Policies and regulations
- 3.12 System life cycle sustainment
- 3.13 Packaging, handling, shipping and transportation

4. Verification

(parallel to subsections in Section 3)

5. Appendices

- Assumptions and dependencies
- Acronyms and abbreviations

Figure 7 — Example SyRS Outline

Software Requirements Specification Outline

Verification: This section provides the verification approaches and methods planned to qualify the software. The information items for verification are recommended to be given in a parallel manner with the information items in section 3.

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Product overview

1.3.1 Product perspective

1.3.2 Product functions

1.3.3 User characteristics

1.3.4 Limitations

1.4 Definitions

2. References

3. Specific requirements

3.1 External interfaces

3.2 Functions

3.3 Usability Requirements

3.4 Performance requirements

3.5 Logical database requirements

3.6 Design constraints

3.7 Software system attributes

3.8 Supporting information

4. Verification

(parallel to subsections in Section 3)

5. Appendices

5.1 Assumptions and dependencies

5.2 Acronyms and abbreviations

Figure 8 — Example SRS Outline