SEG3101 (Fall 2010)

Purpose and Structure of Requirements Specifications
(following IEEE 830 Standard)

Gregor v. Bochmann, University of Ottawa

Based on Powerpoint slides by Gunter Mussbacher (2009)
with material from:
IEEE 830-1998 Standard,
Daniel Amyot 2008, Stéphane Somé 2008

Table of Contents

- Purpose and Structure of the Requirements Specification Document

- Two standards about software engineering

  - IEEE 830 (1993, revised 1998) : Software Requirements Specification

  - ISO/IEC 12207 (1995) : Software Live Cylce Processes

    - was slightly revised as IEEE/EIA 12207(1996)
    - Relationship with IEEE 830 (see Annex B)

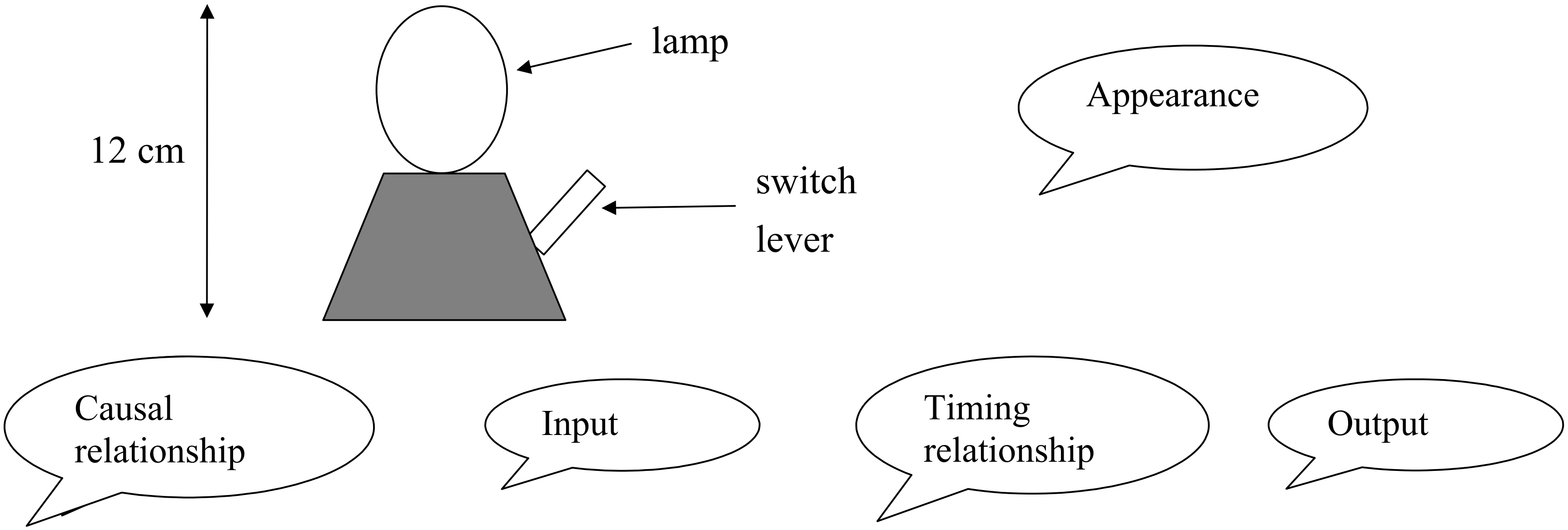
Requirements Specification Document (1)

- 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.

Source: Bray 2004

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

Source: Bray 2004

Purpose of Requirements (Summary)

- 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

- 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 – Considerations

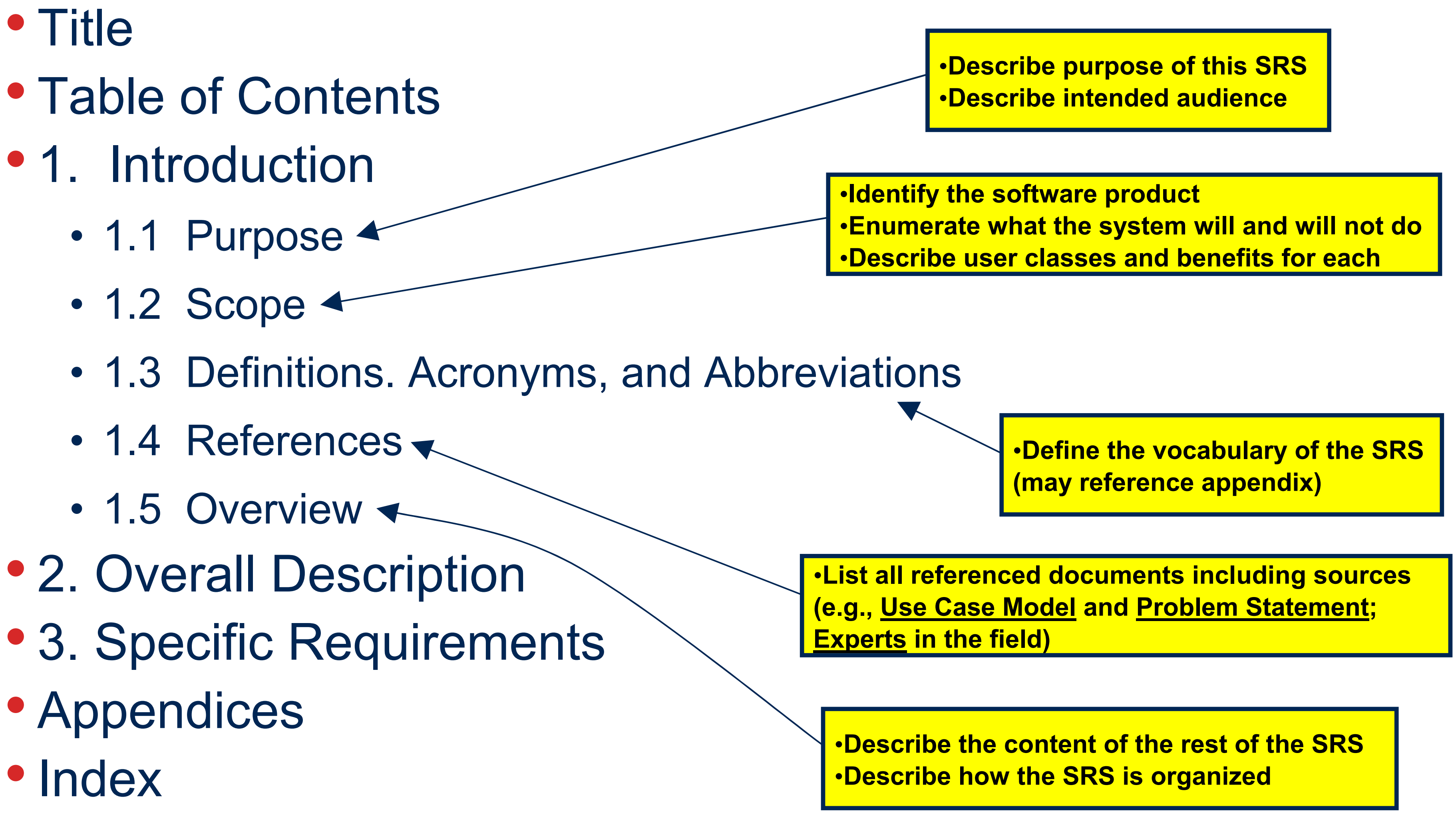
- Section 4 of IEEE 830 (how to produce a good SRS)
  - Nature (goals) of SRS
    - Functionality, interfaces, performance, qualities, design constraints
  - Environment of the SRS
    - Where does it fit in the overall project hierarchy
  - Characteristics of a good SRS
    - Generalization of the characteristics 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 describes a typical document structure

- 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

    •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

  - 2.2 Product Functions

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

  - 2.3 User Characteristics

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

  - 2.4 Constraints

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

  - 2.5 Assumptions and Dependencies

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

- 3. Specific Requirements
- 4. Appendices
- 5. Index

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 Requirem
  - 3.5 Design Constraints
  - 3.6 Software System Quality Att
  - 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
  - Organizations

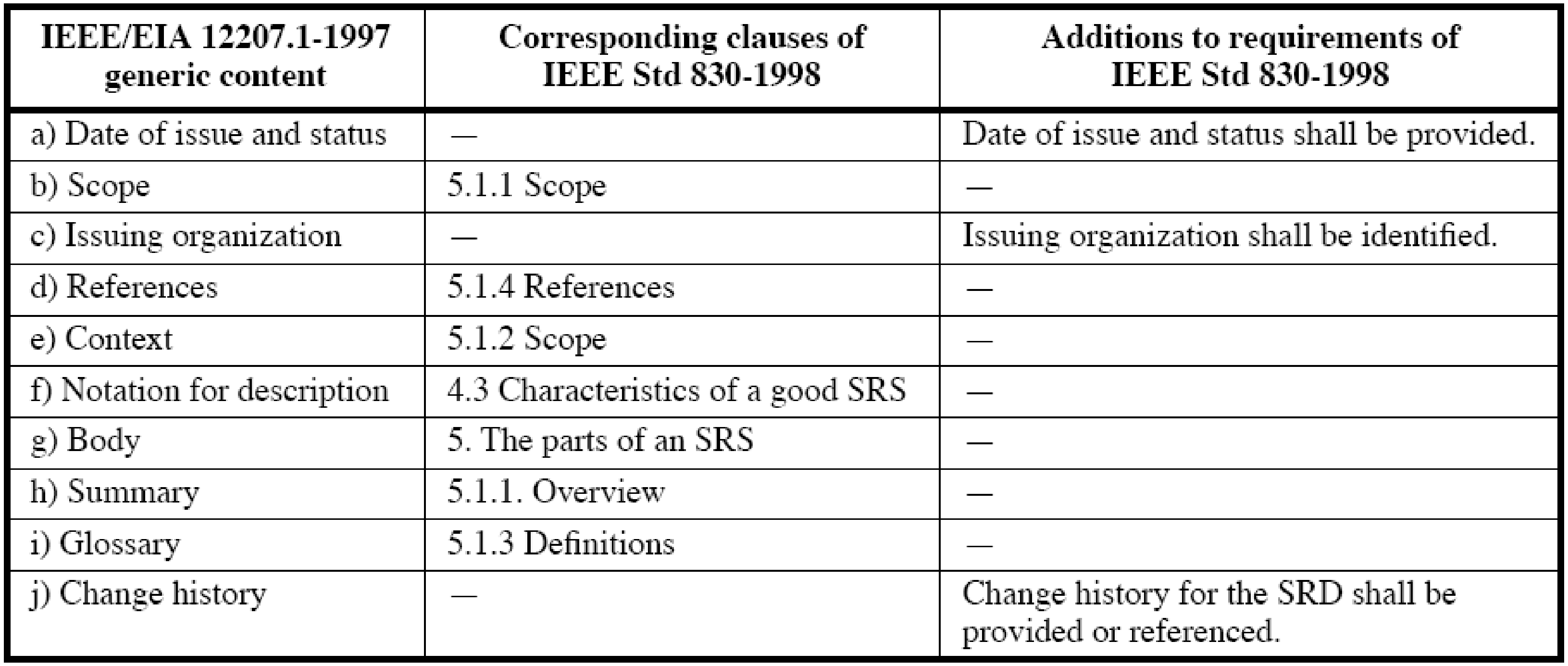
Relationship of IEEE 830 and ISO/IEC 12207 (1)

- ISO/IEC 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 (1)

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.
- Note, however, that the scope of these two standards are quite different.