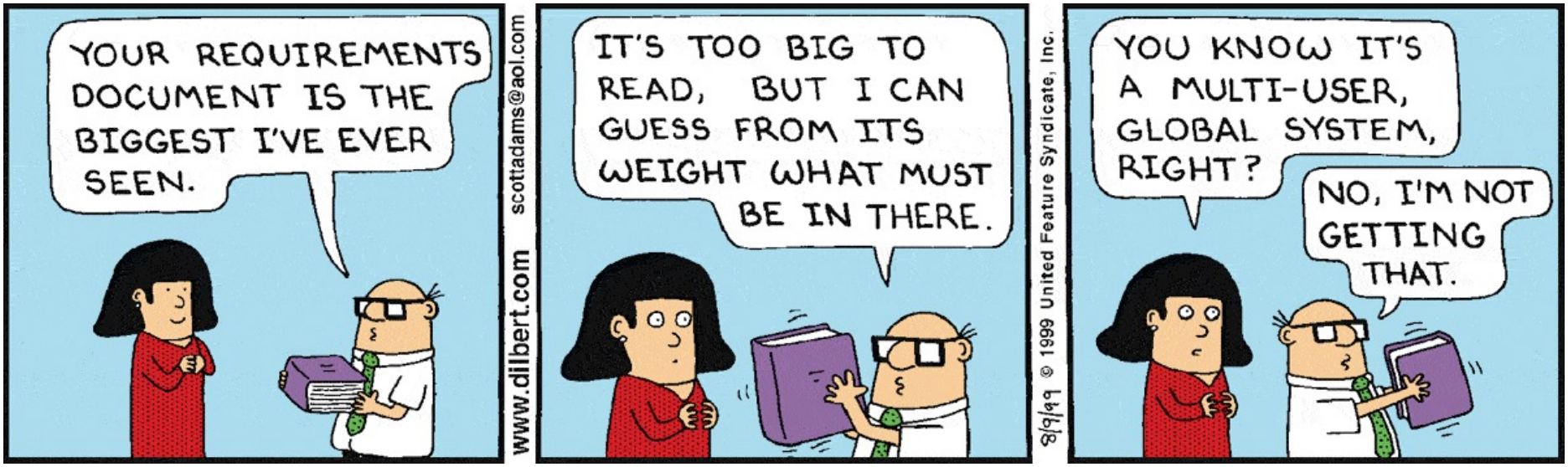
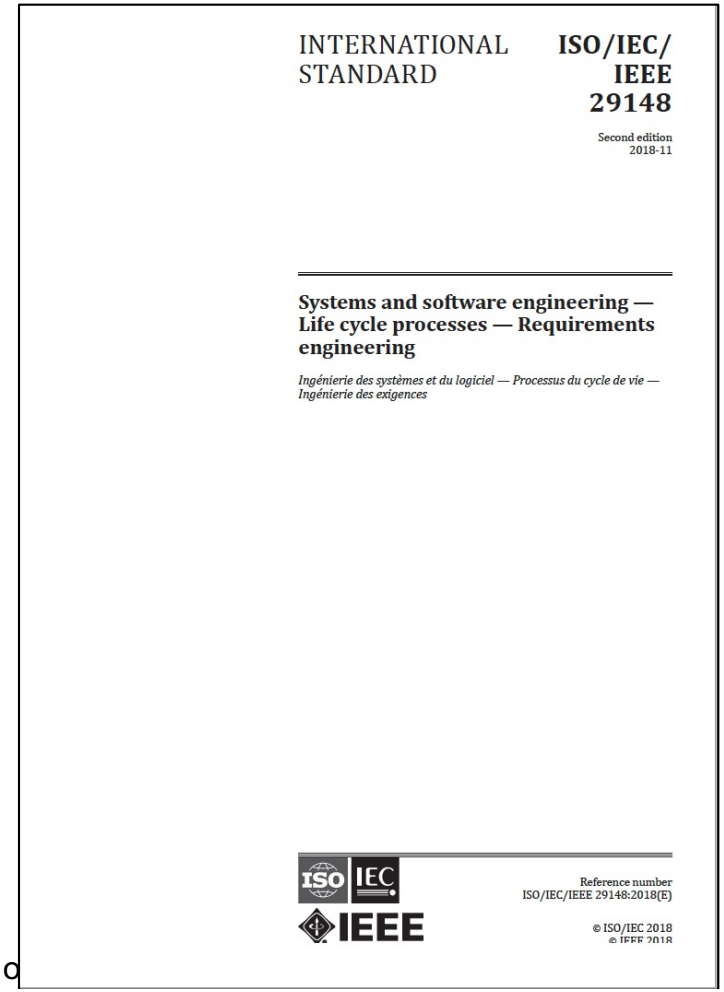
CS445 / ECE451 / CS645 / SE463

Software Requirements Specification & Analysis

Software Requirements Specification (SRS)



Documenting Requirements


INTERNATIONAL STANDARD

ISO/IEC/ IEEE 29148

Second edition 2018-11

Systems and software engineering — Life cycle processes — Requirements engineering

Ingénierie des systèmes et du logiciel — Processus du cycle de vie — Ingénierie des exigences

ISO IEC

IEEE

Reference number ISO/IEC/IEEE 29148:2018(E)

© ISO/IEC 2018

- SRS contents
- SRS structure
- Sample outlines

Organization of an SRS

Table of Contents

Table of Figures

1. Introduction
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 Definitions, acrony… abbreviations

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
   - 3.1 External Interfaces
   - 3.2 Functional Requirements
   - 3.3 Quality Requireme…
   - 3.4 Design Constraints

Appendices

Index

Intro to docoument

Intro to project

Specification details

Example SRSs

From the course's Resources page…

> ### Example SRSs
>
> Here are some example final Software Requirements Specifications (SRSs) from past terms. These aren't exactly like the one that you will produce. These examples were from terms where the whole class worked in teams on SRSs for the same project that had an external customer (so you will see things like minutes from the meetings with the customer). But they will give you an idea of the size of the final project, how to structure the sections, what explanatory text needs to accompany the models in order to produce a coherent document. (As you might guess from the topics of the SRSs, I do some research on automotive software....)
>
> - Example 1 (Active Safety Features)
> - Example 2 (Limited Autonomous Driving Features)
> - Example 3 (Vehicle Stability Control Features)

Section 1. Introduction

1.1 Purpose

- Purpose of the SRS
- Intended audience
- How the SRS is to be used
  - e.g., as a contract between vendor and customer?
- .25-.5 pages

Section 1. Introduction

1.2 Scope

- Name of the software product
- Overview of the product – what it will / will not do
- Benefits, objectives, goals of the product
- Boundaries of the product
- .25-.5 pages

Section 1. Introduction

1.3 Acronyms, Abbreviations, Definitions, Notational Conventions

- Domain-specific terminology
  - Model-related terminology should be in the Glossary
- Naming conventions
- Notational conventions, deviations from standard UML
  - e.g., fonts or colour to denote different model elements

Section 2: Overall description

Provides the background for the system requirements:

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

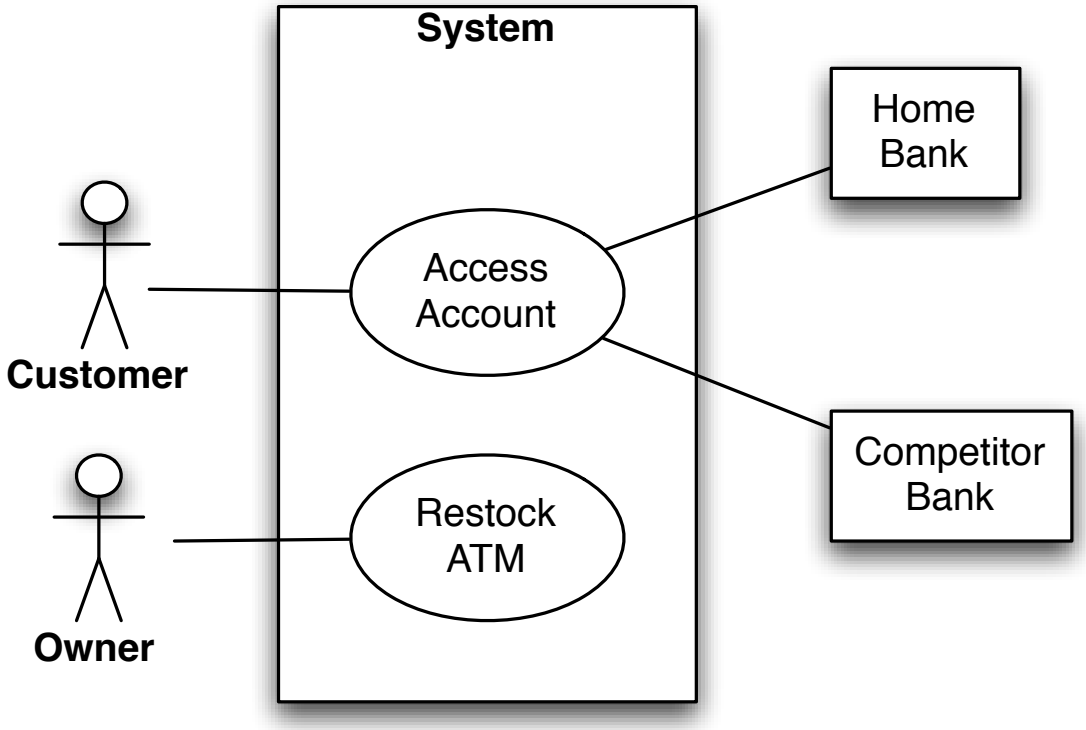
2.4 Limitations

2.5 Assumptions and Dependencies

Describes system's environment, as background to specification of interfaces in Section 3.1

2.1: Product Perspective

Describes the environment/context of the system



2.2: Product Features

An overview of the system's main use cases

- Models and descriptions of use cases
  - Activity diagrams, user stories, use-case descriptions
- Use cases will be specified in detail in Section 3

2.3: User Characteristics

- Assumptions about the user, their background, how much training they will need
  - e.g., different user interfaces for expert vs. novice users
- Only user characteristics that affect the software requirements

2.4: Limitations

Sources of other constraints on requirements

- regulatory policies
- hardware limitations
- parallel operation
- audit functions
- control functions
- criticality of the application
- safety and security considerations
- standards
- laws

2.5: Assumptions / Dependencies

- Assumptions about environmental behaviour
  - fidelity between environment phenomena and sensed input
  - fidelity between interface commands and impact on environment
  - domain assumptions about how the environment behaves
  - assumptions about workload on the system
  - assumptions about working hardware, or about the dependability of adjacent systems
- Conditions could cause the system to fail
- Changes to the environment that could lead to changes in the software requirements

2.5: Assumptions / Dependencies

- Assumptions about environmental behaviour
  - ...lity between environment phenomena and sensed input
  - ... interface commands and impact on environment
  - – domain ... how the environment behaves
  - – assumptions abou... about the dependability of adjacent systems
  - – assumptions about workload on the syst...
- Conditions could cause the system to fail
- Changes to the environment that could lead to changes in the software requirements

External factors that cause requirements to be unsatisfied

Section 3: Specific Requirements

Detailed software requirements and specification

- 3.1 External Interfaces
- 3.2 Functions
- 3.3 Quality Requirements
- 3.4 Design Constraints

3.1: External Interfaces

- Domain Model (interface entities, attributes, relationships)
- Detailed descriptions of all inputs and outputs
  - Name of input (or output)
  - Description of purpose
  - Source of input or destination of output
  - Valid range, accuracy, and/or tolerance
  - Units of measure
  - Timing Relationships to other inputs/outputs
  - Screen formats/organization
  - Window formats/organization
  - Data formats

3.2: Functional Requirements

- State machine model
- Scenarios
- Sequence diagrams
- Functional specifications (i.e., pre/post conditions)
- Constraints

- plus “glue” text that introduces, motivates, and explains each model

Section 3: Specific Requirements

3.3 Quality Requirements

Nonfunctional properties (besides performance) expressed as testable constraints

3.4 Design or Process Constraints

Appendices

- Glossary: The glossary serves as a central place to give a brief description of each term (class, attribute, function, variable).
- References of the sources from which you elicited the documented requirements (textbooks, Web sources, important stakeholders)
- Index

Final SRS Deliverable

Table of Contents (page numbers)

Table of Figures (page numbers)

1. Introduction
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 Definitions, acronyms, conventions
2. Overall description
   - 2.1 Use Case diagram, UC descriptions, Dataflow diagrams
   - 2.2 Activity diagrams, User Stories
   - 2.3 User characteristics, personas
   - 2.4 Limitations
   - 2.5 Assumptions
3. Specific requirements
   - 3.1 Domain model, UI sketches
   - 3.2 Scenarios, State Machines
   - 3.3 Quality Requirements

References

References

ISO/IEC/IEEE, “Systems and Software Engineering – Life Cycle Processes – Requirements Engineering”, International Standard 29148-2018, November 30, 2018.

Section 9.6: “Software Requirements Specification (SRS) content”