Requirements Specification
Document Outline

1. Introduction

The document begins with an introductory description of the desired system. Except as noted below, the system is described in present tense, third person, active voice. The purpose of using present tense is to render this document historically relevant throughout the lifetime of the system. That is, the requirements specification is a permanent part of the system documentation, before the system is implemented, during implementation, and afterward. In subsections 1.4 and 1.5 only, the system is referred to in future tense in order to distinguish between the state of operations before and after system installation.

The purpose of third person is to avoid inconsistent references to actors involved in the requirements, particularly in the functional requirements scenarios of Section 2. The user is referred to as "the user", or some other third-person identifier. The software system referred to as "the system", or some other third-person identifier. In this way, it is precisely clear what actions the human user performs and what action the software system performs. As appropriate, users can be defined as multiple context-specific persona, such as

a. *"the user"* -- a general end user

b. *"the administrator"* -- someone who administers the software system, often with special privileges

c. if appropriate, the name of some specific individual who uses the software,

The purpose of active voice is to reinforce the clarity of the actors portrayed in the scenarios. For example, we say "When the user selects X ..." rather than "After X has been selected ...".

1.1. Problem Statement

The problem statement is a succinct presentation of the specific problem(s) to be solved and the needs to be met by the system.

1.2. System Personnel

The description of system personnel defines all of the people involved with the system and its development. Personnel categories include some or all of the following:

a. system end users

b. paying customers

c. project managers

d. domain experts

e. system analysts

f. system developers

Membership among these groups may overlap.

1.3. Operational Setting

The operational setting is the environment in which the system is to be used. The discussion includes how operations are conducted in the environment before and after the system is installed. The following questions are addressed in the discussion of the operational setting:

a. What computer-based support is in use prior to installation of the new system?

b. Does the new system need to interface with existing system(s) or are the existing system(s) to be replaced entirely?

1.4. Impact Analysis

The impact analysis assesses the impacts of the system within its operational setting. Both positive impacts (e.g., increased productivity, higher product sales) as well as negative impacts (e.g., job displacement, potential negative legal impacts) are addressed.

1.5. Related Systems

Related systems are those computer systems in use elsewhere that provide functionality similar or related to the functionality of the system being specified here. Both commercially available and public domain systems are considered and the following issues are addressed:

a. What is good about the related system, i.e., what features it has that should be included in the system being proposed here.

b. What is bad about the related system, i.e., what features should not be included in the proposed system, or what features should be included but done in a different or better way.

c. What is missing, i.e., what new features should be included in the proposed system that are not found in the related system

d. Any other noteworthy features.

Following the evaluations of specific related systems, a feature comparison matrix summarizes the features of all the related systems, as well as the proposed system.

1.5.1. Related System A

Evaluation of a related system.

. . .

1.5.n. Related System N

Ibid.

1.5.o. Feature Comparison Matrix

See separate handout for details.

1.6. Feasibility Analysis (Not Required in CSC 308)

The feasibility analysis addresses issues related to the user community and, if appropriate, the commercial market for which the system is targeted. Some or all of the following techniques can be used in the feasibility analysis:

a. user/customer surveys

b. customer demographic analysis

c. cost/benefit/risk analysis

d. prototype usage studies, where potential customers use a system prototype to test reactions

1.7. Document Conventions (Optional in CSC 308)

This subsection is an overview of any noteworthy conventions used in the requirements specification document. If any of the conventions are documented in detail, the details can be included in appendices or cited in appropriate external documents.

2. Functional Requirements

Functional requirements define the specific functions that the software system performs, along with the data operated on by the functions. The functional requirements are presented in scenarios that depict an operational system from the perspective of its end users. Included are one or more examples of all system features and an enumeration of all the specific requirements associated with these features.

2.1. User Interface Overview

The user interface overview presents the initial user screen with a brief description of the top-level functionality. For example, if the system uses a command menu as the top-level interface, each of the menus is exposed and each menu item is briefly described.

2.2. *System-Specific Requirements*

Specific functional requirements for the system are presented in user-oriented scenarios. Scenarios are organized into subsections to an appropriate depth (e.g., 2.2.1, 2.2.3.4.5).

. . .

2.n. *System-Specific Requirements*

Ibid.

3. Non-Functional Requirements

Non-functional requirements address aspects of the system other than the specific functions it performs. These aspects include system performance, costs, and such general system characteristics as reliability, security, and portability. The non-functional requirements also address aspects of the system development process and operational personnel.

3.1. System-Related Non-Functional Requirements

Non-functional system requirements include some or all of the following:

a. performance
   i. time
   ii. space

b. operational environment
   i. hardware platform
   ii. software platform
   iii. external software interoperability

c. standards conformance

d. general characteristics

  - i. reliability
  - ii. robustness
  - iii. accuracy of data
  - iv. correctness
  - v. security
  - vi. privacy
  - vii. safety
  - viii. portability
  - ix. modifiability and extensibility
  - x. simplicity versus power

3.2. Process-Related Non-Functional Requirements

Non-functional process requirements include some or all of the following:

- a. development time
- b. development cost
- c. software life cycle constraints
- d. system delivery
  - i. extent of deliverables
  - ii. deliverable formats
- e. installation
  - i. developer access to installed environment
  - ii. phase-in procedures to replace existing system
- f. standards conformance
- g. reporting
- h. marketing
  - i. pricing
  - ii. target customer base
- i. contractual requirements and other legal issues

3.3. Personnel-Related Non-Functional Requirements

Non-functional personnel requirements include some or all of the following:

- a. for developers:
  - i. credentials
  - ii. applicable licensing, certification
- b. for users:
  - i. skill levels
  - ii. special accessibility needs
  - iii. training

4. Developer Overview

The developer overview is an executive summary oriented towards the system design and implementation team. Included are high-level descriptions of the major objects and operations specified in Section 5 below, specifically:

a. an executive summary of the major objects and operations defined in the formal specification;
b. data and operation dictionaries;
c. high-level diagrams, such as UML and dataflow.

5. Formal Specifications

The formal specification defines all objects and operations of the system, including formal specification of all quantifiable requirements.

6. Requirements Specification Rationale

The rationale discusses important development decisions that were made and why. The discussion addresses development alternatives that were considered and why they were rejected.

Appendix A: User Interview Transcripts

Raw and edited transcripts of interviews conducted with clients.

Appendix B: Help Content

Specific content and wording for online help documentation.

Appendix C: ...

. . .

Additional appendices are provided as necessary, such as detailed data formats, glossary of domain-specific technical terms, details of environment-specific or platform-specific requirements.