Table of Contents

1. Introduction

1.1 Purpose -- purpose of this document and its intended audience

1.2 Scope -- what product is being described here; what problem is it solving (benefits)

1.3 Definitions, acronyms, and abbreviations -- don't define what TAs and instructors will know

1.4 References -- if you reference a document elsewhere in this spec, list it here, too.

1.5 Overview (high level view of key points in this spec)

2. Overall Description

2.1 Product perspective -- block diagram of components

2.2 Product functions -- summarize in *priority order*

2.3 User characteristics

2.4 Design Constraints

2.5 Assumptions and dependencies

3. Specific Requirements (See chapter 3 in course text)

4. Use Cases – Need thorough coverage of all system responsibilities, from power on (e.g. power on self test, or POST) to power down and everything in between!

**NOTE: Please remember to focus only on your product or system in this document and do not reference your project (i.e. not even “I” or “we” will do X). Also remember to strip out the guideline commentary and examples from this template when completing your draft, including the Table of Contents.**

**Other recommendations:**

* Write in present tense
* State system requirements as “the system must” or “does” instead of “should”

What is a requirement?

An externally visible function or attribute of a system.

Items such as cost, schedule, methods do not appear here.

For further reference on content and formatting your Requirements Specification, see IEEE std 830-1998, especially chapter 5. This IEEE standard was written around software requirements specification (SRS), although the concepts are applicable to embedded systems specification as well.

Also chapter 7 of your course text is strongly recommend reading as you develop your Requirements Specification. Section 7.2.4 Acceptance Testing and section 7.3 are especially pertinent, as Acceptance Testing will directly correlate to your product’s engineering requirements in the future.

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

2.1 Product perspective --

* Block diagram of major components, interconnections, and external interfaces and/or
* How the system operates under various constraints which may include
  + system interfaces -- list each one, identify how sw accomplishes the system requirement and the interface description
  + user interfaces -- *logical* characteristics of each interface between the user and the system; required display layouts; forms; reports; constraints due to user characteristics
  + hardware interfaces -- *logical* characteristics of each interface between the software and the hardware components of the system such as number of ports and their purposes, what devices will be supported for what purpose.
  + software interfaces -- use of other required software products (are you using some open-source tool such as a data base app, math library, operating system?) Available reference material.
  + communications interfaces -- various interfaces to comm such as local network protocols
  + memory -- relevant characteristics or limits on primary and secondary memory (on-board memory? flash drive available?)
  + operations -- normal and special operations user requires such as
    - modes of operation (e.g., novice/expert; interactive/autonomous; etc.)
    - backup/recovery operations
  + site adaptation requirements
    - requirements for data or initialization sequences specific to a given site, mission, or operational mode (safety limits, ...)
    - anything that must be done to adapt the product to a particular installation

2.2 Product functions -- summary in *priority order*

High: Without these functions, we don't have a product.

Medium: Our customer/sponsor would really like to have these.

Low: It will be cool if we have these but our prototype and demo will be fine without them.

(Implement in this order.)

Within each priority category, indicate software role, if any.

Example under "High priority"

*1. "On" switch activates system including red LED. Software: Run self-diagnostics on system and, if ok, display Welcome msg and turn LED to green.*

2.3 User characteristics

In its intended market:

* Expectation of technology comfort level?
* Educational level? Particular field?
* Users are experts in their own domain?
* Users are beginners in their own domain?
* Users might have disabilities?
* Users might not speak English?
* ??

2.4 Design Constraints

This includes anything that limits your options.

Examples:

a) Regulatory policies;

b) Hardware limitations (e.g., signal timing requirements);

c) Interfaces to other applications;

d) Parallel operation;

e) Audit functions;

f) Control functions;

g) Higher-order language requirements;

h) Signal handshake protocols (e.g., XON-XOFF, ACK-NACK);

i) Reliability requirements;

j) Criticality of the application;

k) Safety and security considerations.

2.5 Assumptions and dependencies

For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the Requirements Specification would then have to change accordingly.

3. Specific requirements

Outline your product’s marketing and engineering requirements.

Use table format, examples and explanation from Chapter 3 in your course text, *Design for Electrical and Computer Engineers*, Ford and Coulston.

**Make sure your engineering requirements meet the properties in section 3.2.1.**

(Also see IEEE Std 830-1998 Section 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)

4. Use Cases

Reference WhatUseCaseSectionsMean.pdf and UseCaseFullyDressedExample.pdf, which are excerpts from *Applying UML and Patterns*, 3rd Edition, by Craig Larman.

Use this abbreviated format for your use cases:

**Use Case UC[n]:**

**Primary Actor:**

# Stakeholders/Interests:

# Preconditions:

# Success Guarantee:

**Main Success Scenario (the basic flow):**

**Extensions (alternate flows):**

**Special Requirements:**

**Open Issues:**

Appendixes

Index