

## Tips and common mistakes for the Software Requirement Specification document

Kira Chan, Nick Polanco, Betty H.C. Cheng  
CSE435

Department of Computer Science and Engineering, Michigan State University  
{chanken1, polanco3, chengb}@msu.edu

### General comments:

- 1) If you have a section header without text at the bottom of a page, then create a new page
- 2) Acronyms
  - a. First use: spell out (e.g., Backup Collision Avoidance System (BCAS)), use BCAS in the rest of document
  - b. ``First use'' does not count in title or acronym section (1.3)
- 3) The use of the word section, figure, and table to reference a target is capitalised (e.g., Figure 1 shows..., the data dictionary shown in Table 2..., Finally, Section 7 concludes our work...)
- 4) Be sure to include some text for section headers, briefly describing what is expected

## 2 Overall Description

Section 2 describes the BCAS in detail. First, the document summarizes the product perspective and product functions. Next, the expectations for the users of the system are described. The document then describes the constraints set on the system. Finally, the document explains the assumptions, dependencies, and apportioning of requirements.

### 2.1 Product Perspective

The BCAS is responsible for obstacle detection, collision prevention, and damage mitigation if an accident were to occur for objects residing behind a vehicle, as they may

- 5) Avoid the use of ``this'' if there are multiple targets (ambiguous)
  - a. This section is **ok** (since we are only referring to the current section)
  - b. This subsystem is **not ok** (since there are multiple subsystems that can be confused)
- 6) Avoid ``will''. Instead, simply state:
  - a. Section x overviews...
  - b. Figure y shows a domain diagram for...
- 7) Be sure the figures used are of good resolution and size
  - a. Stretch the diagram vertically
  - b. Use landscape for the page if needed
- 8) Chronological order – When you are describing how your system work, use an order that makes sense.
  - a. Typically, you will walk the reader through a use case of the system (e.g., 1) the user turns on the system by pressing a button. 2) once the system is active, cameras do x. 3) If y condition is met, then z corrective response is triggered)

### **Section 1 Introduction:**

- 1) This section should provide a brief overview and set the stage. Think of it as background lite.
- 2) **Section 1.2 Scope:**
  - a. Should briefly overview the product to be produced, the benefits, or intended use of the system.
  - b. Should also describe who your intended audience is
- 3) The introduction should capture the reader's attention
- 4) After reading the introduction, the reader should have a general idea of what your system wants to do
- 5) **Section 1.3 Acronym:**
  - a. Format this section for readability (e.g., bold the acronym)
- 6) **Section 1.4 Organisation:**
  - a. **This section should state:**
    - i. **The remainder of this document is organised as follows. Section 2 describes background information for the software. Section 3...**

### **Section 2 Overall Description:**

- 1) The goal of this section is to describe your product in full details
- 2) **Section 2.1 Product Perspective:**
  - a. This section should set the stage for your system. It describes the contributions of your system and why it is important
  - b. Think of it as a sales pitch for your system
- 3) **Section 2.2 Product Functions:**
  - a. This section should describe in English text how your system works. Essentially, we are walking the reader through the system and describing how each subsystem/component interact with each other to carry out the given task
  - b. Think of this as a detailed description for your system
- 4) Section 2.3-2.6 should be full English text (not lists) describe the constraints or other assumptions for the system

### **Section 3 Requirements:**

- 1) Each requirement should state ONE requirement. If you can break it down into two requirements, then break it down!
- 2) Organisation: similar to how you sell your product in Section 2.2, your list of requirements should be organised in a manner that makes sense
- 3) If you use subsystems in your project (e.g., Lane Detection System), then these should be an individual requirement, with subrequirements describing the functionality of the subsystem
  - a. The system shall include the Lane Detection System used to detect lane markings
    - i. The LDS shall use camera and other onboard sensors to identify lane markings...
- 4) Be sure the requirements are testable and concise
  - a. If you use values, then given concrete numbers (0.3 seconds vs x seconds)

#### Section 4 Modelling:

- 1) For each type of diagram, you must do the following:
  - a. Clearly state the purpose of the diagram
    - i. Ex: The purpose of the domain model is to...
  - b. Reference the diagram before the figure appears:
    - i. Ex: Figure 2 shows the domain model for the BCAS.
  - c. Describe the notation of the diagram
    - i. Ex: Boxes represent classes. Connection edges denote associations between classes...
  - d. Walk through the diagram in English text
    - i. There are three types of readers. 1) Those who only read the text. 2) Those who do not read the text and only reads the diagram. 3) Those that read both the text and diagram.
    - ii. We would like all three types of reader to interpret the diagram in the same manner
    - iii. We do so by telling the reader how to read the figures in text
- 2) Use case diagrams
  - a. Include a use case diagram
  - b. Include a use case description for EACH use case in the diagram
  - c. Reference each of these use cases as a table first
    - i. Ex: Table 5-10 shows the use case descriptions for the use cases. *Describe each field of the table, then show the tables.*
  - d. **Each requirement in Section 3 should be cross-reference at least once by at least one use case**
- 3) Domain model
  - a. The domain model should show the class and their relationship in your system
  - b. Each class should have methods/functions/operations (pick one and be consistent)
  - c. Data Dictionary describes each class in detail, including their descriptions, attributes, and operations.
- 4) Sequence diagram
  - a. For each scenario provided in your project description provided by your customer (and maybe additional ones your team included), you should include a sequence diagram that describes how the classes interact with each other to successfully re-enact the scenario
  - b. Each box should be a corresponding class in your domain model
  - c. **Each arrow should be a method in the receiving class**
  - d. Once again, describe the sequence of event and what happens during the scenario in text
- 5) State diagrams
  - a. Demonstrates the states for each of your class in domain model

### **Section 5 Prototype**

- 1) Provide a link to the prototype for your project
- 2) Provide some brief information on the prototype, how to access it, and prerequisite to run
- 3) For each scenario demonstrated in the sequence diagrams, you should have a subsection to demonstrate the same behaviour in the prototype
- 4) Walk to reader through your prototype using text and screenshots
  - a. Essentially, imagine if the reader wants to test your prototype but does not have internet access. Describe the behaviour of your system in text.

Please refer to some example SRS documents posted on the CSE435 website to see some examples implemented, particularly for the diagram descriptions.