Software Requirements Specification Template

Author: Uwe van Heesch
Version: 1 (21/09/16)

1 Introduction

1.1 Overall Description

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. If a separate description of the product scope is available (e.g. in the PvA), refer to it rather than duplicating its contents here.>

1.2 User Classes and Characteristics

<Identify the various user classes (actors) that you anticipate will use this product. User classes may be differentiated based on the subset of product functions used. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. >

1.3 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions,

and any other software components or applications with which it must peacefully coexist.>

1.4 Design and Implementation Constraints
<Describe any items or issues that will limit the options available to the developers. These might include: hardware (e.g. specific mobile platforms), specific technologies, tools, and databases to be used; interfaces to other applications; programming language required; or communications protocols>

1.5 Product Functions
<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. In most cases, this section will primarily contain a use case diagram and brief use case descriptions >

2 Domain Model
<Provide a diagram showing important real-situation conceptual classes in the application domain. Do NOT include software classes. Describe each of the conceptual classes in a glossary.>

3 Use-case Descriptions

<In this section, each use-case is described in detail, optionally accompanied by a system sequence diagram (SSD) and operation contracts. Make sure that the use case descriptions are consistent with the domain model and the use case diagram from Section 1.5>

3.1 Use case 1
<Don't really say "Use case 1." State the use-case name instead.>

3.1.1 Fully-dressed use case description
<Provide a fully-dressed use-case description in the format you know from the OOAD course>

3.1.2 System Sequence Diagram (optional)
<In case the use-case entails complex scenarios, you may decide to create a system sequence diagram showing events generated by external actors, the order of events and inter-system events. All systems are treated as a black box>

3.1.3 Operation Contracts (optional)
<If the use case contains complex manipulations of domain objects, you may decide to specify operation contracts for all system operations included in the use case/ SSD.>

3.2 Use case 2 (and so on)
…

4 Other functional requirements (optional)
<Use this section to describe functional requirements that cannot be expressed in the shape of use cases, for instance because they do not concern goal-oriented interactions of an actor with the system.>

Code	Description
FR1	The system shall maintain an audit trail.
FR2	..

5 Non-functional Requirements
<Describe non-functional requirements in this section. Please refer to existing software quality models like ISO_IEC_IEEE_25010 or FURPS+.>

5.1 Performance Efficiency

Code	Description
NFR1	Responses to all user-initiated actions in the web-interface need to be rendered in less than 1 second.
NFR2	..

5.2 Security

Code	Description
NFR3	Personal user information needs to remain confidential to all parties other than system administrators.
…	

5.3 Reliability (and so on)

6 User interface sketches (optional)
<Provide low-fidelity user interface sketches. Map the sketches to use cases and other requirements if applicable.>