***SOFTWARE REQUIREMENTS SPECIFICATION***

**1.0 Introduction**

**This section provides an overview of the entire requirement document. This document describes all data, functional and behavioral requirements for software.**

**1.1 Goals and objectives**

**Overall goals and software objectives are described.**

**1.2 Statement of scope**

**A description of the software is presented. Major inputs, processing functionality and outputs are described without regard to implementation detail.**

**1.3 Software context**

**The software is placed in a business or product line context. Strategic issues relevant to context are discussed. The intent is for the reader to understand the 'big picture'.**

**1.4 Major constraints**

**Any business or product line constraints that will impact the manner in which the software is to be specified, designed, implemented or tested are noted here.**

**2.0 Usage scenario**

**This section provides a usage scenario for the software. It organized information collected during requirements elicitation into use-cases.**

**2.1 User profiles**

**The profiles of all user categories are described here.**

**2.2 Use-cases**

**All use-cases for the software are presented.**

**2.3 Special usage considerations**

**Special requirements associated with the use of the software are presented.**

**3.0 Data Model and Description**

**This section describes information domain for the software**

**3.1 Data Description**

**Data objects that will be managed/manipulated by the software are described in this section.**

**3.1.1 Data objects**

**Data objects and their major attributes are described.**

**3.1.2 Relationships**

**Relationships among data objects are described using CRC cards. No attempt is made to provide detail at this stage.**

**3.1.3 Complete data model**

**An UML Class model for the software is developed**

**3.1.4 Data dictionary**

**A reference to the data dictionary is provided. The dictionary is maintained in electronic form.**

**4.0 Functional Model and Description**

**Description of major software functions along with UML Use Case, sequence, and communication diagrams.**

**4.1 Description for Function n**

**A detailed description of each software function is presented by completing a use case template. Section 4.1 is repeated for each of n functions.**

**4.1.1 Use case name**

**Unique name for function is defined.**

**4.1.2 Actors**

**Entities that produce of consume the information associated with the fucntion.**

**4.1.3 Precondtions**

**A detailed description of the input and output interfaces for the function is presented.**

**4.1.4 Triggers**

**A detailed description of when the function will be utilized by the system.**

**4.1.5 Scenario Description**

**Describe the flow of events needed to accomplish the use case.**

**4.1.6 Post Conditions**

**Any design constraints that will impact the subsystem are noted.**

**4.1.6 Exceptions**

**Describes how to the system should respond to unusual circumstances.**

**4.2 Software Interface Description**

**The software interface(s)to the outside world is(are) described.**

**4.2.1 External machine interfaces**

**Interfaces to other machines (computers or devices) are described.**

**4.2.2 External system interfaces**

**Interfaces to other systems, products or networks are described.**

**4.2.3 Human interface**

**An overview of any human interfaces to be designed for the software is presented.**

**4.3 Use Case Diagrams**

**The control flow for the system is presented with reference to Section 5.0 of this document.**

**4.3 Sequence Diagrams**

**Used to model the class interactions needed for the use cases.**

**4.3 Communication Diagrams**

**Used to model the message passing structure of the system functions.**

**5.0 Behavioral Model and Description**

**A description of the behavior of the software is presented.**

**5.1 Description for software behavior**

**A detailed description of major events and states is presented in this section.**

**5.1.1 Events**

**A listing of events (control, items) that will cause behavioral change within the system is presented.**

**5.1.2 States**

**A listing of states (modes of behavior) that will result as a consequence of events is presented.**

**5.2 State Transition Diagrams**

**Depict the manner in which the software reacts to external events.**

**5.3 Activity Diagram**

**Depict the manner in which the software reacts to internal events.**

**6.0 Restrictions, Limitations, and Constraints**

**Special issues which impact the specification, design, or implementation of the software are noted here.**

**7.0 Validation Criteria**

**The approach to software validation is described.**

**7.1 Classes of tests**

**The types of tests to be conducted are specified, including as much detail as is possible at this stage. Emphasis here is on black- box testing.**

**7.2 Expected software response**

**The expected results from testing are specified.**

**7.3 Performance bounds**

**Special performance requirements are specified.**

**8.0 Appendices**

**Presents information that supplements the Requirements Specification**

**8.1 System traceability matrix**

**A matrix that traces stated software requirements back to the system specification.**

**8.2 Product Strategies**

**If the specification is developed for a product, a description of relevant product strategy is presented here.**

**8.3 Analysis metrics to be used**

**A description of all analysis metrics to be used during the analysis activity is noted here.**

**8.4 Supplementary information (as required)**

**Return to** [APM Documents](http://www.rspa.com/docs/index.html)