***SOFTWARE ANALYSIS 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 Major software functionality**

**List of software functionality matching to the functionality list in the SPMP**

**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**

**Robustness Diagrams - data objects that will be managed/manipulated by the software are described in this section.**

**3.1.1 Entity-Relationship Diagram**

**Each data object in 3.1 must appear as part of the BUILD ERD.**

**3.1.2 Data Flow Diagram**

**Describe flow of data into/out of application – processes match to use-cases**

**3.1.3 Object Relationships**

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

**3.1.4 Complete data model**

**An UML Class model (class diagram) for the software is developed – through attributes and actions (not data typing, method signatures, access)**

**3.1.5 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 Use cases**

**A detailed description of each software function is presented by completing the use case template.**

**Cross reference this document with file name of use case summary document**

**LIST all of the use cases cross-listed with the file names of actual document**

**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.**

**4.2.3.1 User screen interface layouts**

**Be sure to include “exception” screens, if any**

**4.2.3.2 Report layouts**

**Be sure to include “exception” reports, if any**

**4.3 Sequence Diagrams**

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

**4.4 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.**

**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/Test Strategy**

**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)**