
ERAS-test Documentation

Release 0

Ezio

May 01, 2013

CONTENTS

1	SWRS Template	3
1.1	Change Record	3
1.2	Introduction	3
1.3	General Description	4
1.4	Functional Requirements	4
1.5	Interface Requirements	5
1.6	Performance Requirements	6
1.7	Development and Test Factors	6
1.8	Use-Case Models	6
1.9	Notes	6
1.10	Appendix A	6
2	Glossary	7
3	Indices and tables	9

Contents:

SWRS TEMPLATE

1.1 Change Record

- Issue #3: add some feature.
- Issue #2: fix some other bugs.
- Issue #1: fix bug in software.
- Add first version of the software.

1.2 Introduction

1.2.1 Purpose

Describes the purpose of the document, and the intended audience.

1.2.2 Scope

Describes the scope of this requirements specification.

1.2.3 Applicable Documents

1.2.4 Reference Documents

1.2.5 Glossary

See the *Glossary* page for a list of terms, like *IMS* or *ERAS*.

1.2.6 Overview

Provides a brief overview of the package defined as a result of the requirements elicitation process.

1.3 General Description

1.3.1 Problem Statement

This section describes the essential problem(s) currently confronted by the user community. In other words, this section should discuss what purpose this software package fulfills.

1.3.2 Functional Description

Describes the general functionality of the software, which will be discussed in more detail below.

1.3.3 Environment

Describes the environment in which this software will function.

1.3.4 User objectives

User1

Describe all the users and there expectations for this package

1.3.5 Constraints

Describe any constraints that are placed on this software.

1.4 Functional Requirements

This section lists the functional requirements in ranked order. Functional requirements describe the possible effects of a software system, in other words, what the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe how the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

1.4.1 Requirement

Description

Criticality

- High | Normal | Low

Dependency

Indicate if this requirement is dependant on another.

1.5 Interface Requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include library routines, token streams, shared memory, data streams, and so forth.

1.5.1 User Interfaces

Describes how this product interfaces with the user.

GUI (Graphical User Interface)

Describes the graphical user interface if present. This section should include a set of screen dumps or mockups to illustrate user interface features. If the system is menu-driven, a description of all menus and their components should be provided.

CLI (Command Line Interface)

Describes the command-line interface if present. For each command, a description of all arguments and example values and invocations should be provided.

API (Application Programming Interface)

Describes the application programming interface, if present. For each public interface function, the name, arguments, return values, examples of invocation, and interactions with other functions should be provided. If this package is a library, the functions that the library provides should be described here together with the parameters.

Diagnostics

Describes how to obtain debugging information or other diagnostic data.

1.5.2 Hardware Interfaces

A high level description (from a software point of view) of the hardware interface if one exists. This section can refer to an ICD (Interface Control Document) that will contain the detail description of this interface.

1.5.3 Software Interfaces

A high level description (from a software point of view) of the software interface if one exists. This section can refer to an ICD (Interface Control Document) that will contain the detail description of this interface.

1.5.4 Communication Interfaces

Describe any communication interfaces that will be required.

1.6 Performance Requirements

Specifies speed and memory requirements.

1.7 Development and Test Factors

1.7.1 Standards Compliance

Mention to what standards this software must adhere to.

1.7.2 Hardware Limitations

Describe any hardware limitations if any exist.

1.7.3 Software validation and verification

Give a detail requirements plan for the how the software will be tested and verified.

1.7.4 Planning

Describe the planning of the whole process mentioning major milestones and deliverables at these milestones.

1.8 Use-Case Models

If UML Use-Case notation is used in capturing the requirements, these models can be inserted and described in this section. Also providing references in paragraphs 5, 6 and 7 where applicable.

1.9 Notes

1.10 Appendix A

GLOSSARY

ERAS European Mars Analog Station

IMS Italian Mars Society

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*