

[Institution(s) Name(s)]

[Project Name]

[Project Name Qualification]

User Requirements Document

Issue:	[Document Issue]
Revision:	[Document Revision Number]
Reference:	[Document Reference Number]
Created:	[Document Creation Date]
Last modified:	[Document Last Modification Date]
Prepared By:	[Document Author/Editor]
	[Document Author/Editor]

This document has been prepared using the CERN PSS-05 Templates. The CERN PSS-05 Templates have been prepared by the Information and Programming Technology Group, ECP Division, CERN (The European Laboratory for Particle Physics) and conform to the PSS-05 *Software Engineering Standards* (ISBN 0-13-106568-8) defined by the ESA (European Space Agency) BSSC (Board for Software Standardisation and Control). For more information please contact docsys@ptsun00.cern.ch.

Abstract

Document Abstract (Use Body Abstract tag).

Document Status Sheet

Maintain document history information in the table below. The Document Status Sheet (DSS) provides a history of issues and revisions of the document with a comment indicating the reason for the revision. Note that the Document Title and the Document Reference Number should never change from one revision to another.

Table 1 Document Status Sheet

1. Document Title: [Project Name Qualification] User Requirements Document			
2. Document Reference Number: [Document Reference Number]			
3. Issue	4. Revision	5. Date	6. Reason for change
	0		

Document Change Record

Record all changes between this and previous revision in a Document Change Record (DCR) table. A new DCR per revision is required.

Table 2 Document Change Record (of changes made since issue ...)

Document Change Record			DCR No.	
			Date	
			Originator	
			Approved By	
1. Document Title				
2. Document Reference Number			[Document Reference Number]	
3. Document Issue / Revision Number			[Document Issue]/[Document Revision Number]	
4. Page	5. Paragraph	6. Reason for Change		

Table Of Contents

Abstract	.iii
Document Status Sheet	.iii
Document Change Record	.iv
List of Figures	.vi
List of Tables	vii
1 Introduction	1
1.1 Purpose of the document	.1
1.2 Scope of the software	.1
1.3 Definitions, acronyms and abbreviations	.1
1.3.1 [Definitions]	.1
1.3.2 [Acronyms]	.1
1.3.3 [Abbreviations]	.1
1.4 References	.2
1.5 Overview of the document	.2
2 General Description	3
2.1 Product perspective	.3
2.2 General capabilities	.3
2.3 General constraints	.3
2.4 User characteristics	.3
2.5 Operational environment	.3
2.6 Assumptions and dependencies	.3
3 Specific Requirements	5
3.1 Capability Requirements	.5
3.1.1 [subsection]	.5
3.1.2 [...]	.6
3.2 Constraint requirements	.7
3.2.1 [subsection]	.7
3.2.2 [subsection]	.7
4 List of User Requirements	9
A [Appendix Heading]	11

List of Figures

1 Introduction

Guidelines are provided for each section. Each section may be organised in subsections as necessary.

1.1 Purpose of the document

1.2 Scope of the software

1.3 Definitions, acronyms and abbreviations

Specify special terms, acronyms and abbreviations used in the document. The subsections may be re-organised as necessary. For each definition, acronym or abbreviation use a paragraph pair of either DL1 Term and DL1 Description paragraphs, or, of DL1 Term RIH and DL1 Description paragraphs.

1.3.1 [Definitions]

1.3.2 [Acronyms]

ESA European Space Agency
CERN European Laboratory for Particle Physics

1.3.3 [Abbreviations]

1.4 References

List all external documents referenced in this document.

[1]

1.5 Overview of the document

Provide an overview of the document.

2 General Description

2.1 Product perspective

Describe related external systems and subsystems.

2.2 General capabilities

Describe the main capabilities required and why they are needed.

2.3 General constraints

Describe the main constraints that apply and why they exist.

2.4 User characteristics

Describe who will use the software and when.

2.5 Operational environment

Describe what external systems do and their interfaces with the product.

2.6 Assumptions and dependencies

Describe the assumptions upon which the requirements depend.

3 Specific Requirements

List all specific requirements, with attributes.

User requirements fall into two categories: capabilities needed by users to solve a problem or achieve an objective; constraints placed by users on how the problem is to be solved or the objective achieved.

Each user requirement must include the attributes listed below.

Identifier - each user requirement **shall** include an identifier, to facilitate tracing through subsequent phases.

Need - essential user requirements **shall** be marked as such. Essential user requirements are non-negotiable; others may be less vitally important and subject to negotiation.

Priority - for incremental delivery, each user requirement **shall** include a measure of priority so that the developer can decide the production schedule.

Stability - some user requirements may be known to be stable over the expected life of the software; others may be more dependent on feedback from the SR, AD and DD phases, or may be subject to change during the software life cycle. Such unstable requirements **should** be flagged.

Source - the source of each user requirement **shall** be stated. This may be a reference to an external document (e.g. system requirement document) or the name of the user, or user group, that provided the user requirement.

Clarity - a user requirement is clear if it has one, and only one, interpretation. Clarity implies lack of ambiguity. If a term used in a particular context has multiple meanings, the term **should** be qualified or replaced with a more specific term.

Verifiability - each user requirement **shall** be verifiable. This means that it must be possible to: check that the requirement has been incorporated in the design; prove that the software will implement the requirement; test that the software does implement the requirement.

3.1 Capability Requirements

Capability requirements describe functions and operations needed by users. Quantitative statements that specify performance and accuracy attributes should form part of the specification of a capability.

Space and time dimensions can be useful for organising capability requirements. It is often convenient to describe capability requirements in terms of a sequence of operations.

3.1.1 [subsection]

...

UR [id1] [UR Statement]

Need [How essential is this UR]

Priority [Priority for incremental delivery]

Stability [How subject to change is this UR]

Source [Name of person, group, document, ... from which the UR originates]

Clarity [If more than one interpretation possible, this must be qualified]

Verifiability [Check that UR is incorporated into the design, is implemented in the software, and can be tested]

[Any Other Attribute] [Value of Any Other Attribute]

UR [id2] [UR Statement]

Need [How essential is this UR]

Priority [Priority for incremental delivery]

Stability [How subject to change is this UR]

Source [Name of person, group, document, ... from which the UR originates]

Clarity [If more than one interpretation possible, this must be qualified]

Verifiability [Check that UR is incorporated into the design, is implemented in the software, and can be tested]

[Any Other Attribute] [Value of Any Other Attribute]

...

3.1.2 [...]

...

3.2 Constraint requirements

Constraint requirements place restrictions on how software can be built and operated. For example, definitions of external communications, hardware and software interfaces may already exist, either because the software is a part of a larger system, or because the user requires that certain protocols, standards, computers, operating systems, library or kernel software be used.

Constraints that users may wish to place on the software include the quality attributes of adaptability, availability, portability and security. The user shall describe the consequences of losses of availability, or breaches of security, so that developers can fully appreciate the criticality of each function.

The user may choose to make additional standards applicable; such requirements are additional constraints on the development.

3.2.1 [subsection]

...

UR [id3] [UR Statement]

Need [How essential is this UR]

Priority [Priority for incremental delivery]

Stability [How subject to change is this UR]

Source [Name of person, group, document, ... from which the UR originates]

Clarity [If more than one interpretation possible, this must be qualified]

Verifiability [Check how UR: can be incorporated into the design; implemented in the software; can be tested]

[Any Other Attribute] [Value of Any Other Attribute]

...

3.2.2 [subsection]

A [Appendix Heading]

