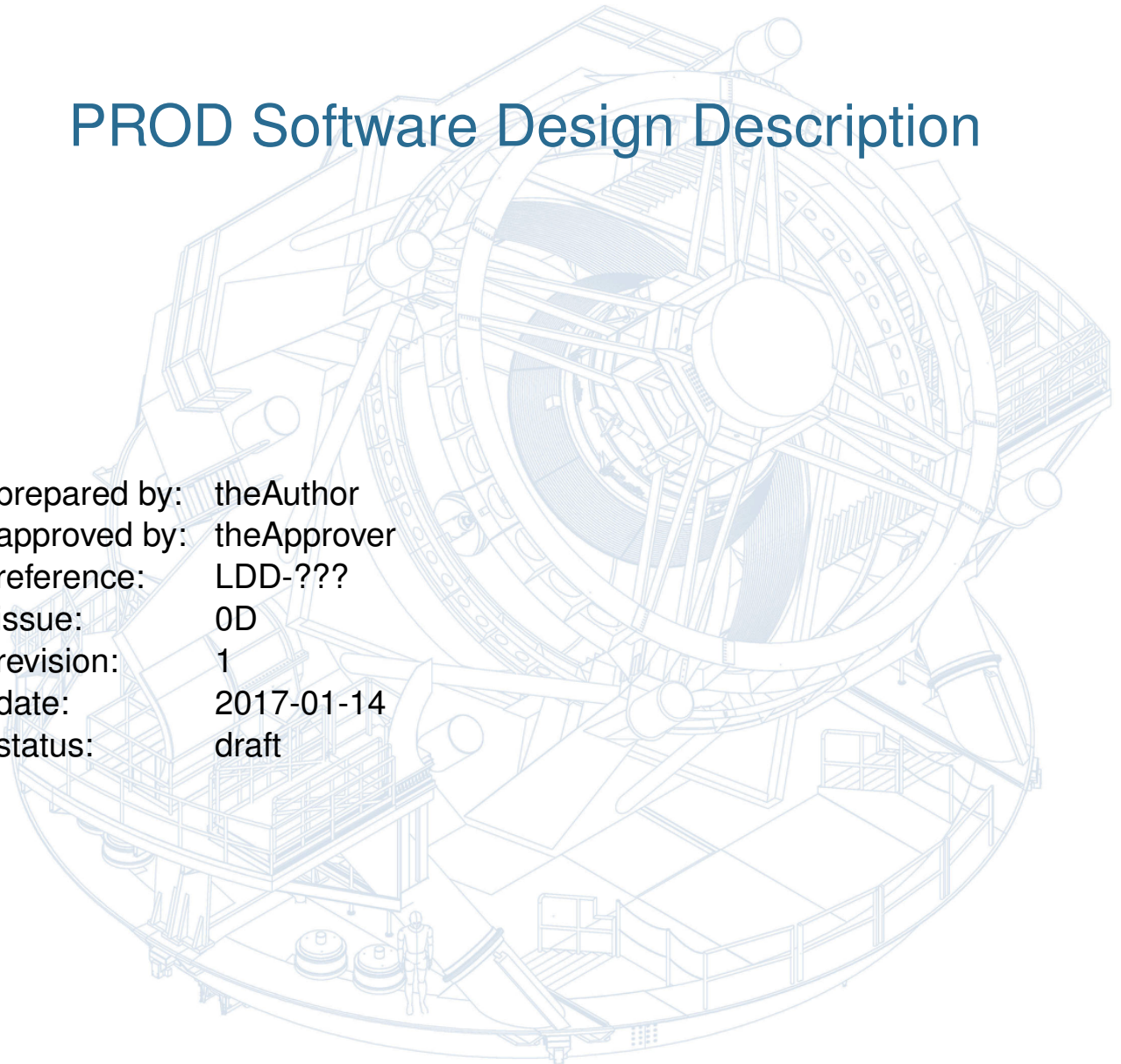




LARGE SYNOPTIC SURVEY TELESCOPE

PROD Software Design Description

prepared by: theAuthor
approved by: theApprover
reference: LDD-???
issue: 0D
revision: 1
date: 2017-01-14
status: draft



Document History

Issue	Revision	Date	Author	Comment
D	1	yyyy-mm-dd	WOM	First draft

Draft

Contents

1	Introduction	5
1.1	Objectives	5
1.2	Scope	5
1.3	Applicable Documents	5
1.4	Reference Documents	5
1.5	Definitions, acronyms, and abbreviations	6
2	Methods and conventions	7
3	Design Overview	7
3.1	Identification	7
3.2	Architecture	7
3.3	Dynamic description	7
4	Software Modules	7
4.1	Software Module N	7
4.1.1	Identification	7
4.1.2	Static Description	8
4.1.3	Dynamic description	8
4.1.4	Interfaces	8
5	Methods(optional)	9
5.1	Methods X	9

5.1.1	Identification	9
5.1.2	Detailed description	9
5.1.3	Interfaces	9
6	Traceability	10

Draft

1 Introduction

All sections of this template should appear in your SDD unless the sections states it may be suppressed. This will give a uniform look and feel to our SDD for the review board.

1.1 Objectives

What is the system supposed to do - a brief description of the function and purpose of the system.

1.2 Scope

To what level is this document applicable - is it just this software system or product or is it across all of DM. This template is applicable to all SDD produced in DPAC to provide a uniform look for such documents.

1.3 Applicable Documents

When applicable documents change a change may be required in this document. The applicable documents are listed here for clarity - the full reference is bellow in Section 1.4. This should be a relatively short list in most cases.

Use citell to cite LiveLink documents such as WOM-001.

TL-001 DPAC Product Assurance Plan
[?] Software Development Plan for DM
[?] Software Requirements Specification for DMWP x product

1.4 Reference Documents

- [1] **[TL-001]**, Levoir, T., Damery, J., Hoar, J., et al., 2012, *DPAC Product Assurance Plan*, GAIA-C1-PL-CNES-TL-001,
URL <http://www.rssd.esa.int/cs/livelink/open/2439085>
- [2] **[WOM-001]**, O'Mullane, W., Lammers, U., 2007, *Work breakdown structures for DPAC*, GAIA-C1-TN-ESAC-WOM-001,
URL <http://www.rssd.esa.int/cs/livelink/open/497865>

1.5 Definitions, acronyms, and abbreviations

The following is a complete list of acronyms used in this document.

Draft

The following table has been generated from the on-line Gaia acronym list:

Acronym	Description
AGIS	Astrometric Global Iterative Solution
AO	Announcement of Opportunity
CCB	Configuration Control Board
CM	Configuration Management
CU	Coordination Unit (in DPAC)
DOC	Department of Commerce (USA)
DPAC	Data Processing and Analysis Consortium
DPC	Data Processing Centre
DU	Detection Unit
ECSS	European Cooperation for Space Standardisation
ESA	European Space Agency
ESAC	European Space Astronomy Centre (VilSpa)
GWP	Gaia Work Package
PA	Product Assurance
QA	Quality Assurance
SCMP	Software Configuration Management Plan
SRS	Software Requirements Specification
SW	Software (also denoted S/W)
TOC	Table of Contents
WBS	Work Breakdown Structure
WP	Work Package

2 Methods and conventions

Use of UML language, design patterns ...

3 Design Overview

3.1 Identification

Identifier: GaiaSpectroscopicArch
Type: Java Archive
Purpose: SRS requirement identifier
Description: This software product aims at
Dependencies: This software product needs the GaiaTools archive to work

3.2 Architecture

General easy to read overview of this software product. Software modules should be introduced here and appear as package or library. From now on, the software modules should be mentioned with their package name or library name.

3.3 Dynamic description

Illustrate the main running sequences involving the software modules of the software product.

4 Software Modules

Here each Software Module defined in the Software Product SRS is described in detail with their algorithms.

4.1 Software Module N

4.1.1 Identification

Identifier: SpectroscopicPkg
Type: Java Package
Purpose: SRS requirement identifier
Description: This software module implements the spectroscopic algorithm

Dependencies: This software module needs the X software module

4.1.2 Static Description

Show the class diagram. Main classes with their main methods should appear in the diagram.

4.1.3 Dynamic description

Main sequence diagrams representing the main activities of the software module should be included. (Scientific algorithms can be subjected to a sequence diagram)

UML sequence diagrams can be used.

Sequence A

Sequence B

Sequence C

4.1.4 Interfaces

Detailed description of the interfaces at 'Module' level introduced in the SRS at DU level.

For each I/O item give at least:

- the name,
- a description and
- the access type: input, output or input/output.
- the data type: long, double, ..
- data ranges.

—Name	Description	Access Type	Data Type	Data ranges —
—				—

5 Methods(optional)

This section is optional. Use this section to describe the main methods if needed.

5.1 Methods X

5.1.1 Identification

Identifier: SpectroscopicMethod
Belong to: SpectroscopicClass
Purpose: SRS requirement identifier
Description: This method implements the main part of the spectroscopic algorithm

5.1.2 Detailed description

This part shows the main steps of the method. A sequence diagram could be provided as an illustration. Scientific concepts could also be described here.

5.1.3 Interfaces

Detailed description of the interfaces at the method level.

For each I/O item give at least:

- the name,
- a description and
- the access type: input, output or input/output.
- the data type: long, double, ..
- data ranges.

—Name	Description	Access Type	Data Type	Data ranges —
—				—

6 Traceability

can be generated with `makerequirementsTrace.rb` from the code

Requirement	Version	Design element implementing requirement	Completion	Comment
S-XXX-NN	1.1	Spectroscopic.Main	I	

The field completion reveals whether the SRS requirement is fully implemented. It can have the following values:

- **I**: the SRS requirement is fully implemented
- **PI**: the SRS requirement is partially implemented
- **NI**: the SRS requirement is not yet implemented

The values **PI** and **NI** are fairly normal in the cyclic development. It is the current status for this version - later versions will have more Is.