



Horus IMSETY

Software Quality Assurance Plan

Version 0.6 14th May 2007

Project team:	Jeroen Keiren	0569081
	Frank Koenders	0575629
	Thijs Nugteren	0574426
	Joeri de Ruiter	0578312
	Stijn Stiefelhagen	0579816
	Carst Tankink	0569954
	Pim Vullers	0575766
	Freek van Walderveen	0566348
Project manager:	Egbert Teeselink	
Senior management:	L. Somers	TU/e (HG 7.83)
	M. v.d. Brand	TU/e (HG 7.44)
Adviser:	R.J. Bril	TU/e (HG 5.09)
Customer:	E. v. Breukelen	ISIS

Computer Science, Eindhoven University of Technology, Eindhoven

Abstract

This is the Software Quality Assurance Plan (SQAP) for the IMSETY project. This project is part of the Software Engineering Project (2IP40) and is one of the assignments at Eindhoven University of Technology. The document complies with the SQAP description from the Software Engineering Standard, as set by the European Space Agency [1]. This document contains the guidelines to maintain the quality of the project, with exception of the procedures for verification and validation, which are described in the Software Validation and Verification Plan [7].

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	1
1.3	List of definitions	2
1.4	List of references	2
2	Management	3
2.1	Organization	3
2.2	Tasks	3
2.3	Responsibilities	3
3	Documentation	4
4	Standards, practices, conventions and metrics	5
4.1	Documentation standards	5
4.2	Design standards	5
4.3	Coding standards	6
4.4	Comment standards	7
4.5	Testing standards	7
4.6	Metrics	7
4.7	Compliance monitoring	7
5	Review	8
6	Test	9
7	Problem reporting and corrective action	10
8	Tools, techniques and methods	12
9	Code control	13
10	Media control	14
11	Supplier control	15
12	Records collection, maintenance and retention	16
13	Training	17
14	Risk management	18

A UR phase	19
A.1 URD	19
A.2 SPMP	19
A.3 SCMP	19
A.4 SQAP	20
A.5 SVVP	20
A.6 GENSO related training	20
B SR phase	21
B.1 SRD	21
B.2 SPMP/SR	21
B.3 SQAP/SR	21
B.4 SVVP/SR	22
C AD phase	23
C.1 ADD	23
C.2 SPMP/AD	23
C.3 SCMP/AD	23
C.4 SQAP/AD	23
C.5 SVVP/AD	24
D DD phase	25
D.1 DDD	25
D.2 ATP	25
D.3 STP	25
D.4 ITP	25
D.5 UTP	25
E TR phase	26
E.1 STD	26
E.2 SUM	26

Document status sheet

Document title	Software Quality Assurance Plan
Document identifier	IMSETY/doc/SQAP
Author(s)	Jeroen Keiren, Frank Koenders
Version	0.6
Document status	Internally approved

Version	Date	Author(s)	Summary
0.1 (Revision 128)	21-02-2007	Jeroen Keiren, Frank Koenders	Ready for internal review
0.2 (Revision 278)	03-03-2007	Jeroen Keiren, Frank Koenders	Fixed problems from internal review
0.3 (Revision 577)	31-03-2007	Jeroen Keiren	Fixed comments from Sylwia Kopczynska
0.4 (Revision 589)	31-03-2007	Jeroen Keiren	Fixed document for AD phase
0.5 (Revision 759)	11-04-2007	Frank Koenders	Fixed problems from internal review
0.6 (Revision 1187)	14-05-2007	Frank Koenders	Written DD appendix

Document change report

Document title	Software Quality Assurance Plan
Document identifier	IMSETY/doc/SQAP
Date of changes	N/A

Section number	Reason for change
D	Ready for internal review.
4.3	Added rule to coding standard.

Chapter 1

Introduction

1.1 Purpose

This document describes the procedures and control methods to obtain the desired quality level
5 of the end products and the process by which these project artefacts are created. This document
serves as a guide for the managers and developers of the IMSETY project. All team members
must read this document and apply the procedures stated in it. The document applies to all
phases of software development as defined in the Software Project Management Plan [5]. Detailed
information about the software quality assurance activities for these phases can be found in the
10 appendices.

1.2 Scope

In the IMSETY project a system has to be designed to support scientific experiments in space.
This system, with user documentation, developer documentation and a demonstration model of
an experiment (in software) have to be developed during this project.

1.3 List of definitions

This section contains a list of terms, definitions and acronyms used in this document.

AD	Architectural Design
ADD	Architectural Design Document
ATP	Acceptance Test Plan
CR	Change Request
DD	Detailed Design
DDD	Detailed Design Document
ESA	European Space Agency
IT	Integration Test Plan
PM	Project Manager
project artefacts	Also referred to as artefacts. All documents and software that have to be delivered for this project, including management documents
QAM	Quality Assurance Manager
SCMP	Software Configuration Management Plan
SPMP	Software Project Management Plan
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SR	Software Requirements
SRD	Software Requirements Document
STD	Software Transfer Document
SUM	Software User Manual
SVVP	Software Verification and Validation Plan
TR	Transfer
UR	User Requirements
URD	User Requirements Document
UTP	Unit Test Plan

1.4 List of references

- [1] European Space Agency. ESA Software Engineering Standards. Technical report, ESA, February 1991. [i](#), [5](#), [8](#), [10](#), [15](#)
- 20 [2] Marshall Cline. C++ FAQ Lite Const correctness. URL <http://www.parashift.com/c++-faq-lite/const-correctness.html>. [7](#)
- [3] Horus. Architectural Design Document, March 2007. [12](#)
- [4] Horus. Software Configuration Management Plan, March 2007. [3](#), [5](#), [10](#), [12](#), [13](#), [14](#), [16](#)
- [5] Horus. Software Project Management Plan, March 2007. [1](#), [3](#), [4](#), [12](#), [18](#), [19](#)
- 25 [6] Horus. Software Quality Assurance Plan, March 2007. [3](#)
- [7] Horus. Software Verification and Validation Plan, March 2007. [i](#), [3](#), [7](#), [8](#), [9](#), [12](#), [19](#), [20](#), [21](#), [23](#), [24](#), [25](#)
- [8] Horus. User Requirements Document, March 2007. [21](#), [25](#)
- [9] Dimitri van Heesch. Doxygen Manual. URL <http://www.stack.nl/~dimitri/doxygen/manual.html>. [7](#)
- 30 [10] Tom Verhoeff. Programmarealisatie 1 codeerstandaard. URL <http://www.win.tue.nl/~wstomv/edu/2ip10/year-2006/coding-standard-v1.4.pdf>. [6](#)

Chapter 2

Management

35 This chapter details the structure of the Software Quality Assurance (SQA) in the project IM-SETY.

2.1 Organization

For a survey of the organizational structure within the project, and the responsibilities of the individual members of the team see the SPMP ([5]). The Quality Assurance Manager (QAM) is responsible for the SQA. In this he is assisted by the vice QAM. When the software quality is endangered the QAM contacts the PM. They will decide whether or not one or more of the following parties have to be informed:

- Customer
- Senior management

45 2.2 Tasks

The main task of the QAM is to check whether the procedures are followed properly and that the standards are handled correctly as defined in the SQAP [6], SVVP [7] and the SCMP [4]. Additionally the QAM inspects whether all group members fulfil their tasks as defined in the SPMP [5] according to the parts of this document applying to their specific tasks. If a problem is detected, the appropriate procedure as defined in chapter 7 will be followed. The QAM has, besides his main task, the following additional tasks:

- Checking the consistency and the coherence between documents.
- Organize internal reviews (initiative lies with the QAM).
- Organize external reviews (initiative lies with the QAM).
- 55 • Attend both internal and external reviews.

Specific tasks arising during the different phases of the project will be added in the corresponding appendices.

2.3 Responsibilities

The main responsibility for the SQA tasks, as described in section 2.2, lies with the QAM. The QAM can delegate tasks to the vice QAM. Every problem found by a team member has to be reported to the QAM. In case the QAM is unavailable for a certain period of time, the vice QAM will assume his tasks.

Chapter 3

Documentation

⁶⁵ The documents to be delivered in the specific phases of the project are listed and outlined in the SPMP ([5]). Document standards are described in chapter 4.

Chapter 4

Standards, practices, conventions and metrics

70 4.1 Documentation standards

Various documents will be produced during this project. The QAM checks that the documents adhere to the ESA Software Engineering Standard guidelines [1], this will be done during checks held by the QAM.

Every document has to be approved by:

- 75 • The author(s)
- The leader of the responsible team
- The QAM

In case these three turn out to be all the same person, the vice QAM has to give his approval as well.

80 The documentation standards the documents need to adhere to are the following:

- All documents must adhere to the ESA Software Engineering Standard guidelines [1].
- All documents must adhere to the house style as described in the SCMP [4].
- All documents must be written in British, academic English.
- Requirements on review and approval as described in chapter 5.
- 85 • Requirements on document identification as described in the SCMP.
- Procedures involving the change of documents (as described in the SCMP).

These standards apply to all artefacts, both written as well as electronic versions. The documents are made available through the project repository, as described in the SCMP.

4.2 Design standards

90 The design standards in the Architectural Design (AD) and Detailed Design (DD) phases will be defined in the Architectural Design Document (ADD), or the Detailed Design Document (DDD) respectively.

4.3 Coding standards

This chapter provides a coding standard for C++ which cares for well-readable and modifiable code. The rules leave a lot of freedom to the programmer. Note that the standard is loosely based on Tom Verhoeff's Coding standard for Delphi [10].

1. Be consistent when the rules do not prescribe certain behaviour.
2. Always use { - } after:
 - `if`
 - 100 • `else`, unless immediately followed by `if`
 - `for`
 - `while`
3. Always indent a multiple of 2 spaces. Never use TABs because TAB distance has not been standardized.
- 105 4. Try to keep the length of lines at most 80 characters.
5. Always write at most one of the following per line:
 - Constant-definition
 - Type-definition
 - Variable-declaration (including explicit type with every variable)
 - 110 • Statement
6. Explain every variable declaration in a comment, unless fully trivial.
7. Always write one empty line before and after the following:
 - function-definitions
 - `for`-, `while`-statements
- 115 8. Never write a space before the following characters, but do put a space after it (unless at the end of a line):
 - `,` `:` `;`
9. Always separate the following with one space of previous and next character:
 - keywords
 - 120 • operators: `+` `-` `*` `/` `=` `<` `>` `<=` `>=` `%` `&&` `||` `&` `|` (etc...)
 - Exceptions to this rule are the pre- and post-increment/decrement (`--i`, `i++`), combinations of operators with `=`, like `+=`, etc., as well as `~`.
10. Always give globals useful names, and only use them after a fierce fight with your fellow programmers. The use of static members is preferred.
- 125 11. Always write defines in capital letters (`#define NAME`).
12. Class names always start with a capital letter (e.g. `MyClass`).
13. Class instances always start with a lower-case letter (e.g. `myObject`)..
14. Function names always start with a lower-case letter (e.g. `myObject->doSomething()`)
15. Use only C++ style casts (`static_cast<Type>()`, `const_cast<Type>()`).

- 130 16. All pointers in a class, that can be newed and deleted, should initially, and after every delete, be set to 0. This is because `delete (void *)0;` is caught and omitted, this eliminates problems with double deletes.
17. Make sure that variables that are not to be changed by functions are passed as `const`. This is also called const-correctness. (See [2] on this subject).
- 135 18. Do not use `using namespace` directives in header files.
19. Document all your code in a style that can be processed by Doxygen (see [9]).

4.4 Comment standards

The comment standards form a part of the Coding standards (section 4.3).

4.5 Testing standards

- 140 The testings standards are described in the SVVP [7].

4.6 Metrics

The quality of software will also be measured by means of metrics. The exact metrics are defined in appendix D, describing the DD phase.

4.7 Compliance monitoring

- 145 The QAM will monitor compliance to the proposed conventions by way of taking samples at random, during which references to other documents are checked. Problems discovered during the review are reported by the QAM.

Chapter 5

Review

150 Standards and procedures for reviews and audits are described in the SVVP [7]. In addition to reviews, random checks will be carried out by the QAM as described below.

Random checks The QAM randomly checks all project and product documents to ensure that all products adhere to the document standards and that all group members do their jobs properly. Management and product documents are tested for adherence to the ESA Software Engineering standards [1], and whether their layout and style adheres to the house style. Furthermore the
155 references to other documents are investigated. Program code is checked for coding standard (section 4.3) adherence. It is also checked that the code in the repository compiles correctly. Random checks are an addition to the reviews. Results of the random checks will only be reported to the author, and will only be documented on QAM's draft version.

¹⁶⁰ Chapter 6

Test

Methods and procedures for testing are detailed in the SVVP [7]. Test results are documented by the QAM and the adherence to the test procedures is reported.

Chapter 7

165 Problem reporting and corrective action

When a problem in approved documentation or code has been detected, it has to be solved. There are several kinds of problems:

- Document problems:
 - 170 – Noncompliance with the ESA Software Engineering Standard guidelines [1].
 - Noncompliance with the house style (SCMP [4]).
 - Noncompliance with other project documents.
 - Incompleteness.
 - Errors.
- 175 • Code problems:
 - Non compliance with coding or commentary standards.
 - Lack of functionality.
 - Wrong functionality.
 - Incompleteness.
 - 180 – Errors (Including compile errors).

The following procedures need to be followed when a problem is detected:

Problem reporting procedure When a problem is detected in an approved version, the person that has discovered the error is responsible for reporting it to the QAM. When a problem has been discovered during a review the QAM is responsible. If the QAM deems it necessary he will report the problem to the PM.
185

Problem solving procedure The procedure of solving a problem consists of several steps:

- The QAM appoints the team leader of the corresponding team to solve the reported error. The team leader is responsible for solving the error.
- 190 • When the team leader is confident that the error is solved, the QAM is notified to check whether the error is solved correctly.
- When the problem cannot be solved, or cannot be solved within a reasonable amount of time, a meeting is set up with the PM, the QAM, and the team leader of the responsible team. During this meeting a decision will be made on how to deal with the problem any further.

195 If the problem to be solved was discovered after internal or external acceptance, the PM (in collaboration with the QAM) first decides whether the problem is important enough to be solved, if so, a Change Request (CR) has to be filled out. This CR has to be approved by:

- In case of previous internal acceptance: the PM, the author(s) of the document and the QAM.
- 200 • In case of previous external acceptance: the PM, the author(s) of the document, the QAM, Senior Management and in case of the URD, SRD and ATP the client.

Requirements change by customer It is also possible that the customer requests a change to the requirements formulated in an externally accepted URD. In this case the customer needs to fill out a CR. After that, the requested change is matched to the URD. If the change conforms 205 to the URD the request is accepted. If the requested change does not conform to the URD, the team decides whether it will accept the changes or not. If such a change is accepted the URD will be changed.

Chapter 8

Tools, techniques and methods

210 The QAM has to make sure that appropriate tools, techniques and methods are used. These are described in SCMP [4], SPMP [5], SVVP [7] and ADD [3]. The QAM checks their use by means of random checks.

With respect to the tool used during this project special interest is paid to:

- Availability of the tools. (Has every group member access to the tools?).
- 215 • Knowledge. The group members working with the tools must have the necessary skills to work with the tools (see also chapter 13).
- The tools must work properly. (Are there errors or malfunctions in tools? Enough capacity?).

Every used tool will be checked at least once before use and once during use. When problems appear the QAM decides together with the PM and librarian whether the problem can be solved, 220 or whether the tool must be replaced by an alternative.

Chapter 9

Code control

It is the librarian's responsibility to assure the correct handling of all artefacts according to the standards described in the SCMP [4]. The following has to be valid:

- 225 • Documents are available to all people who are authorized to access them and to no one else.
- All versions of a document are available.
- Naming conventions are consistently used.

The QAM checks if the procedures and standards as described in the SCMP [4] are handled properly. This is done by reviews and random checks (chapter 5). Problems are reported to the
230 librarian and PM.

Chapter 10

Media control

The QAM checks if the procedures and standards as described in the SCMP [4] are handled properly. This is done in reviews and random checks (chapter 5). Problems are reported to the librarian and PM.

235

Chapter 11

Supplier control

All external software components in the program code, that have an unreliable source, will be tested according to the ESA Software Engineering Standard guidelines [1]. Software components that have reliable sources will undergo some quick tests. These tests will be focused on the parts of this software that are of importance to the project. Whether an external software component is reliable or not is to be decided by the QAM.

Chapter 12

Records collection, maintenance and retention

245

Minutes of meetings and notes of external reviews are added to the project library as described in SCMP [4]. Minutes of meetings are added after the members of the meeting have approved them. Minutes are delivered three workdays after the meeting at the latest. These documents will be kept in digital form throughout the duration of this project. Additionally all progress reports, metrics and review reports will be kept throughout the duration of the project.

250

Chapter 13

Training

During the project there may arise tasks that require special skills. Due to the fact that all group members reached an acceptable level of knowledge in the area of computer science, special training in this area will probably be unnecessary. However, should the need arise for people with specialized knowledge into a certain area for some task, the PM and the QAM will assess the level of knowledge for the task in the group and then they decide whether special action needs to be taken. In that case, detailed information can be found in the appendices of this document.

Chapter 14

260 Risk management

In the SPMP [5], the risks of the project are described. During progress meetings the occurrence of any of the risks described must be discussed and the PM must see to it that the necessary course of action is taken. The QAM will assist in this task.

Appendix A

265 UR phase

In the User Requirements phase of the project (UR), the QAM must see to it that the following documents have been properly reviewed internally before they are submitted for external review.

A.1 URD

Before the internal review the QAM checks whether the URD:

- 270 • Contains a general description of the software that has to be developed.
- Contains requirements on the software to be developed as stated by the customer.
- Contains constraints on the software on the software to be developed.
- Contains a priority list of the requirements.

Furthermore it needs to be checked that every user requirement complies with the requirements
275 specified in SVVP [7].

A.2 SPMP

The QAM must check whether the goals of the project have been clearly described. A life-cycle approach for the project must be defined. The QAM must ensure that the SPMP [5] is realistic by checking:

- 280 • The assumptions made during the planning of the project (by comparing the actual time spent with the reserved time in the planning).
- Restrictions with respect to the planning (e.g. availability of members).
- External problems (e.g. room availability).

A.3 SCMP

285 Before internal review the QAM checks whether the document provides procedures concerning:

- Document identification.
- Document storage.
- Document change control.
- Document status indication.

290 **A.4 SQAP**

Before internal review, the QAM checks whether the SQAP contains:

- Project standards.
- Problem reporting procedures.
- Responsibilities of the project members with respect to quality assurance.

295 **A.5 SVVP**

Before internal review, the QAM checks whether the SVVP contains:

- Reviewing and audits.
- Testing.
- Tracing.

300 During internal reviews (see SVVP [7]) the QAM checks these documents and in case of problems, the author(s) and the team leader are informed. After corrective action has been taken, the QAM checks the document again.

A.6 GENSO related training

305 Because we are required to make use of the GENSO network, we have decided that there are two group members who are doing research into GENSO, and specialize in this area. All GENSO related problems and questions will therefore be redirected to them.

Appendix B

SR phase

In the Software Requirements phase of the project (SR), the QAM must see to it that the following documents have been properly reviewed internally before they are submitted for external review.

During internal reviews (see SVVP [7]) the QAM checks these documents and in case of any problems, the author(s) and the team leader are informed. After corrective action has been taken, the QAM reviews the document again.

B.1 SRD

Before internal reviews the QAM must check whether the SRD:

- Contains requirements on the software to be developed, these requirements must be based on the specific requirements stated in the URD [8].
- Contains constraints on the software to be developed, these constraints must be based on the specific constraints stated in the URD [8].
- Contains a logical model.
- Contains a priority list of the requirements.
- Contains a traceability table (see SVVP [7]).

B.2 SPMP/SR

Before internal reviews the QAM checks whether the SPMP is realistic concerning:

- The assumptions made in the planning.
- Restrictions with respect to the planning (e.g. availability of members).
- External problems (e.g. external software/code).

B.3 SQAP/SR

Before internal reviews the QAM checks whether the SQAP contains:

- The tasks of the QAM during the SR phase.

B.4 SVVP/SR

Before internal reviews the QAM checks whether the SVVP contains:

- The Acceptance Test Plan (could be a document on its own).
- The System Test Plan (could be a document on its own).

335 Appendix C

AD phase

In the Architectural Design phase of the project (AD), the QAM must see to it that the following documents have been properly reviewed internally before they are submitted for external review.

C.1 ADD

340 Before internal review the QAM checks whether the ADD:

- contains an architectural design of the software to be developed, this design must describe a physical model and the interfaces between the different classes. The physical model has to describe the design of the software using implementation terminology. The architectural design also has to contain pre- and postconditions for the methods defined in the physical model.
345
- contains a traceability matrix (see SVVP [7]).

C.2 SPMP/AD

Before internal review the QAM checks whether the SPMP is realistic concerning:

- the assumptions made during the planning.
- 350 • restrictions with respect to the planning (e.g. availability of members).
- external problems (e.g. external software).

C.3 SCMP/AD

Before internal reviews the QAM checks whether the SCMP contains:

- 355 • a description of the tools used in support of version control, code creation, compilation and debugging.

C.4 SQAP/AD

Before internal reviews the QAM checks whether the SQAP contains:

- coding and commentary standards.

C.5 SVVP/AD

³⁶⁰ Before internal reviews the QAM checks whether the SVVP contains:

- an Integration Test Plan (ITP).

During internal reviews (see SVVP [7]) the QAM checks these documents and in case of problems, the author(s) and the team leader are informed. After corrective action has been taken, the QAM reviews the document again.

365 Appendix D

DD phase

In the Detailed Design phase of the project (DD), the QAM must see to it that the following documents have been properly reviewed internally before they are submitted for external review.

D.1 DDD

370 Before internal reviews the QAM checks whether the DDD:

- contains the detailed design of the software to be developed, this design must describe the components and their interfaces to other components.
- contains a detailed design of the software to be developed.
- contains a traceability table (see SVVP [7]).

375 D.2 ATP

Before internal reviews the QAM checks whether the ATP:

- contains all user requirements described in the URD [8].

D.3 STP

Before internal reviews the QAM checks whether the STP:

- 380
- contains sufficient tests to test the system.

D.4 ITP

Before internal reviews the QAM checks whether the ITP:

- contains sufficient tests to the integration of all components.

D.5 UTP

385 Before internal reviews the QAM checks whether the UTP:

- contains sufficient tests to test all units of the system.

Appendix E

TR phase

390 In the Transfer phase of the project (TR), the QAM must see to it that the following documents have been properly reviewed internally before they are submitted for external review.

E.1 STD

Before internal reviews the QAM checks whether the STD:

- contains a list of all deliverables to be transferred.
- contains a procedure to build the software.
- 395 • contains a procedure to install the software.

E.2 SUM

Before internal reviews the QAM checks whether the SUM:

- contains a tutorial on how to use the software.
- contains references to all possible options in the software.