

Software Validation and Verification Plan

Eindhoven, November 13, 2009

SVVP-2.0.1499



Where innovation starts

Project Manager:

Wilco Belgraver Thissen, 0514143

Quality Assurance Manager:

Jelle Hellings, 0592127

Senior management:

Mark van den Brand, HG 5.59

Lou Somers, HG 5.36

Advisor:

Erik Luit, HG 7.12

Customer:

Natalia Sidorova, HG 7.84

Project team:

Roy Berkeveld, 0608170

Gijs Direks, 0611093

Michael van Duijkeren, 0535368

Neal van den Eertwegh, 0610024

Dion Jansen, 0590077

Koen Kivits, 0608715

Sander Leemans, 0608896

Kevin van der Pol, 0620300

Nick van der Veeke, 0587266

Computer Science, TU/e

Abstract

This document is the Software Validation and Verification Plan (SVVP) of GROUP QIS. This project is part of the Software Engineering Project (2IP35) and is one of the assignments at Eindhoven University of Technology. The document complies with the SVVP from the Software Engineering Standard, as set by the European Space Agency [1].

This document describes how the quality will be assured during the project. Among other things, this document establishes procedure and policy for maintaining quality and outlines quality requirements for other documents.

Contents

1	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.3	List of definitions	6
1.4	List of references	6
2	Verification overview	8
2.1	Organization	8
2.1.1	Organization	8
2.2	Reviews	8
2.2.1	Internal review	8
2.2.2	External reviews	9
2.3	Audits	9
2.4	Tests	10
2.5	Schedule	10
2.6	Resources	11
2.7	Project responsibilities	11
2.8	Tools, techniques and methods	11
3	Administrative procedures	12
3.1	Anomaly reporting and resolution	12
3.2	Task iteration policy	12
3.3	Deviation policy	13
3.4	Control procedures	13
3.5	Standards	13
4	Verification activities	14
4.1	Reviews	14
4.1.1	Internal reviews	15
4.1.2	External reviews	15
4.2	Formal proofs	16
4.3	Tracing	16
5	Verification reporting	18
A	Format of reviews	19

A.1	General review information	19
A.2	Problems	19
A.2.1	Legend	19
A.3	Background information	20
A.3.1	Major discussions	20
A.3.2	Decisions	20
A.4	Metrics	20
B	UR phase	21
B.1	The User Requirements Review	21
B.2	Requirements for User Requirements	21
C	SR phase	22
C.1	The Software Requirements Review	22
C.2	Requirements for Software Requirements	22
D	AD phase	23
D.1	The Architectural Design Review	23
D.2	Design Quality	23
E	DD phase	25
E.1	The Detailed Design and Production Review	25

Document Status Sheet

Document status overview

General

Document title: Software Validation and Verification Plan
Identification: SVVP-2.0.1499
Authors: Kevin van der Pol
Document status: Final

Document history

<i>Version</i>	<i>Date</i>	<i>Author</i>	<i>Reason of change</i>
0.1	28-09-2009	Kevin van der Pol	Internal review
0.2	28-09-2009	Kevin van der Pol	Internal review
1.0.677	Approved
2.0.964	21-10-2009	Jelle Hellings	External review with advisor

Document Change Records since previous issue

General

Date: 28-09-2009
Document title: Software Validation and Verification Plan
Identification: SVVP-2.0.1499

Changes

<i>Page</i>	<i>Paragraph</i>	<i>Reason to change</i>
8	2.2 Reviews	Unclear who needs to review which documents
15, 16	4.1 Reviews	Limited flexibility
18	5 Verification reporting	Removed, not in scope of this document
17	4.3 Tracing	Reference too many
18	5 Verification reporting	Added back in to keep in line with ESA standard
9	2.2.2 External reviews	Comment from senior management

Chapter 1

Introduction

1.1 Purpose

This document describes procedures concerning the testing of the delivered products (product documents and software) of the GROUP QIS project for compliance with the requirements. The requirements that the software has to be verified against can be found in the product documents URD[13], SRD[9], ADD[3] and DDD[4]. The modules to be verified and validated are defined in the AD phase. The goal of verifying and validating is to check whether the software product to be delivered conforms to the requirements of the client and to ensure a minimal number of errors in the software. This project document is written for managers and developers of the GROUP QIS project.

1.2 Scope

QIS is an application designed and developed by GROUP QIS for the Departement of Mathematics and Computer Science at the Eindhoven University of Technology. The purpose of the application is to support workload division within the department and to support various parties regarding the management of working hours and tasks, holidays, courses and employees.

1.3 List of definitions

2IP35	The Software Engineering Course
AD	Architectural Design phase
ADD	Architectural Design Document
AT	Acceptance Test
ATP	Acceptance Test Plan
Client	Department of Mathematics and Computer Science of Eindhoven University of Technology
CI	Configuration Item
CM	Configuration Manager
DD	Detailed Design phase
DDD	Detailed Design Document
ESA	European Space Agency
TU/e	Eindhoven University of Technology
PM	Project Manager
QM	Quality Manager
SCMP	Software Configuration Management Plan
SEP	Software Engineering Project
SM	Senior Management
SPMP	Software Project Management Plan
SQAP	Software Quality Assurance Plan
SR	Software Requirements definition phase
SRD	Software Requirements Document
STD	Software Transfer Document
SUM	Software User Manual
SVVP	Software Verification and Validation Plan
SVVR	Software Verification and Validation Report
TR	Transfer phase
UR	User Requirements definition phase
URD	User Requirements Document
VPM	Vice Project Manager

1.4 List of references

- [1] ESA Board for Software Standardization and Control (BSSC). European space agency software engineering standards. February 1991. (ESA PSS-05-0 Issue 2).
- [2] GROUP QIS. Unit test plan. Technical report, Eindhoven University of Technology, Computer Science, ? 2007.
- [3] GROUP QIS. Architectural design document. Technical report, Eindhoven University of Technology, Computer Science, ? 2009.
- [4] GROUP QIS. Detailed design document. Technical report, Eindhoven University of Technology, Computer Science and Engineering, ? 2009.
- [5] GROUP QIS. Integration test plan. Technical report, Eindhoven University of Technology, Computer Science, ? 2009.

- [6] GROUP QIS. Software configuration management plan. Technical report, Eindhoven University of Technology, Computer Science, September 2009.
- [7] GROUP QIS. Software project management plan. Technical report, Eindhoven University of Technology, Computer Science and Engineering, sep 2009.
- [8] GROUP QIS. Software quality assurance plan. Technical report, Eindhoven University of Technology, Computer Science, September 2009.
- [9] GROUP QIS. Software requirements document. Technical report, Eindhoven University of Technology, Computer Science, ? 2009.
- [10] GROUP QIS. Software user manual. Technical report, Eindhoven University of Technology, Computer Science, 2009.
- [11] GROUP QIS. Svvp - acceptance test plan. Technical report, Eindhoven University of Technology, Computer Science, ? 2009.
- [12] GROUP QIS. System test plan. Technical report, Eindhoven University of Technology, Computer Science, ? 2009.
- [13] GROUP QIS. User requirements document. Technical report, Eindhoven University of Technology, Computer Science, September 2009.

Chapter 2

Verification overview

2.1 Organization

2.1.1 Organization

The QM checks the verification and validation of the activities of the project. Therefore the QM attends every internal or external review. If the QM is not available the vice-QM will take his place, this means that every time the QM is mentioned it can also be the vice-QM. If the QM discovers problems which threaten the general progress of the project, i.e. problems meeting deadlines, it is his responsibility to inform the PM about these problems.

2.2 Reviews

The following types of review are used:

2.2.1 Internal review

The simple internal review is used for documents which will not reach the client directly.

Documents

The simple internal review is used for all documents.

Reviewers

The following persons must review the document:

- The QM.
- One of the authors of the document.

- Any other project member, if they wish to.
- The advisor, for product documents.

More details about internal reviews can be found in section 4.1.1.

2.2.2 External reviews

The external review is used to see if the document matches client expectations. The external review can only be done after the document has been accepted in an internal review.

Documents

The external review is used for the URD[13], SRD[9], ATP[11], SUM[10].

Reviewers

The following persons must review the document:

- A representative of the project team, preferably one of the authors of the document.
- The advisor.
- The client.
- Any other project member, if they wish to.
- Anyone the client wants.

More details about external reviews can be found in section 4.1.2.

2.3 Audits

Audits are reviews that assess compliance with software requirements, specifications, baselines, standards, procedures, instructions, codes and licensing requirements. Physical audits check that all items identified as being part of the configuration are present in the product baseline. A functional audit checks that unit tests, integration tests and system tests have been carried out and records their success or failure. Functional and physical audits can be performed before the release of the software (ESA Software Engineering Standard Section[1]). The SM is allowed to audit the project to check if the procedures, as described in the management documents SPMP[7], SQAP[8], SCMP[6] and this document are followed. Audits are not routine checks, but the SM can request them. The following rules apply to all audits:

- Only the SM can request audits.
- Audit requests must be directed to the PM.
- In the audit request the following information must be included:

- Names of the auditors (at least two persons)
- Possible dates of the audit
- Purpose of the audit
- Items that will be checked
- The audit is attended by the QM, the PM and possibly others as indicated by SM.
- The results of the audit are reported by a group member in a written report to the PM and the QM within one week of the audit. This report must contain the following information:
 - Date of the audit
 - Participants in the audit
 - Checked items
 - Conclusion
 - Recommendations

2.4 Tests

The testing is done following the ESA Life cycle verification approach [1]. The following test plans (plans that outline the approach to testing) can be found as separate documents:

- ATP (Acceptance Test Plan) [11]
- STP (System Test Plan) [12]
- ITP (Integration Test Plan) [5]
- UTP (Unit Test Plan) [2]

The ATP [11] has to be approved by the client, as it will define the terms on which the final product will be accepted. The results of the tests are presented to the PM and the QM. The produced code and product documents must also be tested to assure that all the requirements are met. Requirements can be found in section 4.3 and in the appendices C, D and E.

If any special software testing tools are used for testing the software, these will be mentioned in the SCMP[6].

2.5 Schedule

The schedules for all phases are given in the SPMP[7].

2.6 Resources

In this project no special tools or people are used for verifying documents.

2.7 Project responsibilities

Some roles as defined in the SPMP have responsibilities related to verification and validation. These responsibilities are:

Project Manager:

- The PM is responsible that the work is adequately inspected.

Quality Assurance Manager:

- Assuring that the requirements of the documents are adhered to.
- Assuring the documents conform to the specified layout and contain the proper information.
- Lead the review sessions.
- Oversee the acceptance tests.

Configuration Manager:

- Copy documents to Master library after finalization.

2.8 Tools, techniques and methods

The tools that are used during the project are discussed in the SCMP[6].

Chapter 3

Administrative procedures

3.1 Anomaly reporting and resolution

Everything that is not up to standards it should be up to or does not conform to requirements it should conform to, is an anomaly. Procedures for anomaly resolution can be found in the SQAP[8]. Furthermore, it is the task of the QM to monitor whether the procedures as defined in the management plans (SPMP[7], SCMP[6], SQAP and SVVP) are followed. This is done during team meetings, reviews and by randomly checking CI's. Findings are reported to the PM. It then is the responsibility of the PM to enforce compliance with defined procedures. If the results of the PM's actions are not satisfactory to the QM then he can request the senior management to take further action.

3.2 Task iteration policy

Every task performed is to be internally reviewed as described in chapter 4. Some tasks (as described in section 2.2.2) need an external review. If, during a review, problems are discovered concerning the correct conclusion of the task, a decision is made concerning the iteration of the task. Guidelines are provided for the following cases:

- The team responsible was unable to complete their task, most probably because of one of the risks as described in section 3.3 of the SPMP. In this case, it is the responsibility of the QM to solve the problem and make sure the task is completed as described in the SPMP. If the QM is unable to do so, he must report this to the PM. If problems arise concerning the dependencies between tasks then these are to be reported to the PM.
- A structural error was found in the execution of the task, for example the output of a piece of code that does not comply with the requirements. In this case, the team that is responsible performs the task again. If necessary the PM schedules extra man-hours.
- An item was forgotten during the execution of a task. Depending on the severity of this item the QM will decide whether a redo of the task is needed, or only the forgotten item needs to be fixed. This case will most probably occur in processing review remarks.

3.3 Deviation policy

During the project, the procedures described in the management documents are followed. However, if in the QM's opinion, this endangers the completion of the project then the QM can decide to deviate from these procedures. If the decision is made to deviate from the procedures described in the management documents, the PM must be informed of such a deviation.

3.4 Control procedures

Procedures assuring that configuration items are not accidentally or deliberately changed are described in the SCMP[6].

3.5 Standards

Before both internal and external reviews, the authors certify that the document is according to ESA Software Engineering standard [1], and that the document complies with the standard layout as detailed in SCMP.

Chapter 4

Verification activities

4.1 Reviews

Review procedures are held during all phases of the GROUP QIS project. Configuration items are reviewed in the phase they are delivered; an overview of which item is delivered in which phase can be found in the SPMP [7]. All project and product documents have one of the following statuses:

- Draft (initial status)
- Internally approved with proposed changes
- Internally approved
- Conditionally approved
- (Externally) approved

This status can be found in the documents themselves. Note that approved technical documents are not modified (unless the completion of the project is endangered). With respect to the approved management documents only appendices for every phase are added during the project. The appendices are approved during review meetings.

For a document to become (internally) approved it has to be reviewed. Here internal and external reviews of technical and management documents (section 2.2.1 and 2.2.2 respectively) are distinguished.

As noted above each document starts with the draft status. Once there has been a internal review it either becomes Internally approved or needs some changes. When there are only minor changes needed the document will be internally approved when the QM has confirmed that the changes have been made. If it concerns major changes a new internal review will be needed.

Internally approved documents can be scheduled for external review. During this review the document can reach the highest status of externally approved if there are no defects. If there are only minor defects the document may be conditionally approved, these defects need to be solved to achieve the highest status. If major defects show up during the review the document needs to be changed to solve these defects. Because this involves considerable changes it should pass a new internal review before it may be subject to another external review.

After review, the document may need to be tagged. This is the responsibility of the authors of the document. See the SCMP [6] for information about tagging documents.

4.1.1 Internal reviews

In the following table, T is the time of the review meeting. If possible, the reviewers may do their review earlier than this table suggests, but no later. So if a document is handed in for review on day X, it may also be reviewed on day X+1. In any case it must be reviewed no later than X+2.

Nr	Actor	Action	Time
1	QM	Set a date for the internal review of the document	-
2	Author	Deliver review version of document to reviewers (hard copy or soft copy)	T - 2 workdays
3	Reviewer	Inspect the document	Before T
4	Reviewer	Communicate all errors to the authors, mark major errors	T
5	Reviewer	Decide if the document can be approved, provided the stated changes are made	T
6	QM	If the document cannot be approved, make an appointment for a new review meeting	T
7	Author	Collect annotated documents	T
8	Author	Make review document	After T
9	Author	If the document cannot be approved, process proposed changes to the document	After T
10	QM	See to it that the stated remarks are handled properly by the team delivering the document	After T
11	QM	Grant the document the status internally accepted if all requested changes are made	After T

4.1.2 External reviews

For the organization of external reviews see section 2.2.2. The following table shows the action list for the preparation and execution of the external reviews of documents. T is the time of the review meeting. This procedure is only for the external review of product documents. The metrics of the external review will be sent to the SM. The official format for reviews is described in appendix A.

If possible, the reviewers may do their review earlier than this table suggests, but no later.

Nr.	Actor	Action	Time
1	QM	Set a date and place for the external review of the document	After internal acceptance
2	Author	Deliver review version of document to reviewers (hard copy or soft copy)	T - 5 workdays
3	Reviewer	Inspect the document and write down all errors explicitly	Before T
4	Reviewer	Deliver remarks to the moderator	Before T
5	QM	Inspect remarks	Before T
6	Author	Lead the meeting and keep discussions to the point	T
7	Author	Document everything that is discussed during the review	T
8	Reviewer	Discuss all comments that need explanation or discussion	T
9	Author	Collect the remarks on the documents	After T
10	Reviewer, QM	Decide the status of the document at the end of the meeting. There are three possible outcomes: the document is rejected and a new appointment is made the document is accepted and the status Approved is granted	After T
11	Author	Make minutes of the review, and hand these together with the remarks of the reviewers to the Senior Management. Also make sure they will go to the configuration management system	After T

Only when the document is rejected do actions 12 and 13 apply.

12	QM	See to it that the remarks are handled properly by the team responsible for the document	After T
13	QM	Grant the document the status Approved if all reviewers inform that their remarks are handled properly, eventually after another review if the remarks included major changes	After T

4.2 Formal proofs

Formal proof will be given where considered necessary by the QM, or when asked by the person(s) responsible for a certain product.

4.3 Tracing

During the project the relation between the input and the output of a phase must be checked several times. A traceability table as result of the final trace is included in the output document

of the phase. In this table the CI is traced to the input of the phase. During the software life cycle it is necessary to trace:

- User requirements to software requirements and vice versa, this is documented in the SVVP/SR (Appendix C).
- Software requirements to component requirements and vice versa, this is documented in the SVVP/AD (Appendix D).
- Component requirements to DD requirements and vice versa, this is documented in the SVVP/DD (Appendix E).
- Integration tests to architectural units and vice versa, this is described in the integration test plans [5]. These tests are performed during the DD-phase.
- Unit tests to the modules of the detailed design, this is described in the unit test plans[2]. These tests are performed during the DD-phase.
- System tests to software requirements and vice versa, this is described in the system test plans[12]. These plans are executed during the DD-phase.
- Acceptance tests to user requirements and vice versa, this is described in the acceptance test plans[11]. These tests are executed during the TR-phase.

To support traceability, all requirements are uniquely identified.

Chapter 5

Verification reporting

Review reports are written for internal and external reviews (see sections 2.2.1 and 2.2.2). The format of these reports is described in Appendix A.

Appendix A

Format of reviews

The review document should be in a separate document in the same folder as the document under review. The filename should be “<number of external reviews>.<number of internal reviews>.<revision number>”.

A.1 General review information

Project name	GROUP QIS
Document name	
Version	
Type of review	
Date	
Start time	
End time	
QM	
Author(s)	
Non-author participants	
Review outcome	

A.2 Problems

Nr.	Ref.	Type	Remark	Severity	Action
-----	------	------	--------	----------	--------

A.2.1 Legend

Number: Sequence number of remark starting at 1.

Reference: Reference to page/section and line number(s).

Type:

M Missing

- X Extra/Superfluous
- A Ambiguous
- I Inconsistent
- U Unclear, improvement desirable
- S Not conforming to standards
- R Risk-prone
- F Factually incorrect
- N Not implementable
- E Editorial

Severity: Major or minor, optional.

Action: Decision about future action:

- No action
- Local change
- Global change
- Investigate

A.3 Background information

A.3.1 Major discussions

Abstract of the most important major review discussions.

A.3.2 Decisions

List of decisions made during the review session.

A.4 Metrics

Number of problems identified:

Major	
Minor	
Total	

Appendix B

UR phase

B.1 The User Requirements Review

The outputs of the User Requirements Definition Phase are formally internally and externally reviewed in the User Requirements Review (UR-R). It ensures that the URD states the user requirements clearly and completely and a general description of the processes to be supported (the environment) is present. The SPMP, SCMP, SVVP and SQAP are only internally reviewed.

B.2 Requirements for User Requirements

User requirements should be realistic, that is:

- Clear.
- Verifiable.
“The product shall be user friendly” is not verifiable.
- Complete.
- Accurate.
Among other things, the URD is inaccurate if it requests something that the user does not need, for example a superfluous capacity.
- Feasible.

In the URD each requirement must ...

- ... **have a unique identifier**. Traceability in other phases depends on this identifier.
- ... **have a priority**. Essential requirements have to be met for the software to be acceptable.
- ... **be marked if unstable**. Unstable requirements may depend on feedback from later phases.

Appendix C

SR phase

C.1 The Software Requirements Review

The outputs of the Software Requirements Definition Phase are formally reviewed in the Software Requirements Review (SR-R). Internal reviews are held before the formal external SR-R takes place. It ensures that:

- The SRD states the software requirements clearly, completely and in sufficient detail to start the design phase.
- The STP [12] is an adequate plan for system testing the software in the DD phase.

C.2 Requirements for Software Requirements

In the SRD each requirement must:

- have a unique identifier.
Traceability in other phases depends on this identifier.
- be marked essential or not.
Essential requirements have to be met for the software to be acceptable.
- have a priority if the transfer will proceed in phases.
- be marked if unstable.
Unstable requirements may depend on feedback from other phases.
- have references that trace it back to the URD.
A software requirement is caused by one or more user requirements.
- be verifiable.

Besides the requirements the SRD must contain a traceability matrix containing trace from the software requirements to the user requirements and vice-versa. This matrix should be complete, meaning all requirements should be traceable.

Appendix D

AD phase

D.1 The Architectural Design Review

The outputs of the Architectural Design phase are formally reviewed in the Architectural Design Review (AD-R). Any report by the QM may also be input for the AD-R. Internal reviews are held before the formal external AD-R takes place. It ensures that:

- The ADD describes the optimal solution to the problem stated in the SRD.
- The ADD describes the architectural design clearly, completely and in sufficient detail to start the detailed design phase.
- The ADD is a good design for QIS.
- The ITP is an adequate plan for integration testing the software in the DD phase.

D.2 Design Quality

A good design is:

- Adaptable: it is easy to modify and maintain.
- Efficient: it makes a minimal use of available resources.
- Understandable: it is not only clear for the developers but also for outsiders.
- Modular: the components are simple and independent from each other:
 - A change in one component has minimal impact on other components.
 - A small change in requirements does not lead to system wide changes.
 - The effects of an error condition are isolated to its source component.
 - A component is understandable as a stand-alone unit, without reference to others.

Good components obey the information hiding principle: software design decisions are encapsulated so that the interface reveals as little as possible about its inner workings. For example, a component should hide how its data is stored: it can be in memory (in an array, list, tree, ...) or in a temporary file.

Some rules to choose components:

- Minimize coupling between components:
 - Minimize the number of items that are passed between components;
 - Pass only the data that are needed (data coupling);
 - Do not pass a structure of which only a small part is being used (stamp coupling);
 - Avoid the use of control flags (control coupling);
 - Do not use global data (common coupling).
- Maximize cohesion inside a component: put elements into a component that are related to each other; they contribute for example to the same task.
- Restrict fan-out: restrict the number of child components.
- Maximize fan-in: reuse components as often as possible.
- Use factoring: avoid duplication of functionality. Cut the common functionality from the components and put it into a reusable component.
- Separate logical and physical functionality: top-level components must be separated from physical aspects (the data they deal with); the level of abstraction of a component must be according to its place in the hierarchy.

Appendix E

DD phase

E.1 The Detailed Design and Production Review

The outputs of the Detailed Design and production Phase are formally reviewed in the Detailed Design and production Review (DD-R). Any report by the QM may also be input for the DD-R. Internal reviews are held before the formal external DD-R takes place. It ensures that:

- The DDD describes the detailed design clearly, completely and in sufficient detail to allow development and maintenance by software engineers not involved in the project.
- Modules have been coded according to the DDD.
- Modules have been verified according to the unit test specifications in the UTP.
- Major components have been integrated according to the ADD.
- Major components have been verified according to the integration test specifications in the ITP.
- The software have been verified against the SRD according to the system test specifications in the STP.
- The ATP specifies the test design, test procedures and test cases so that all user requirements in the URD can be validated.
- The SUM explains what the software does and instructs the users how to operate the software correctly.