Project Fingerpaint

SVVP-1.0

Software Validation and Verification Plan

Authors:
Tessa Belder (0739377)
Lasse Blaauwbroek (0749928)
Thom Castermans (0739808)
Roel van Happen (0751614)
Benjamin van der Hoeven (0758975)
Femke Jansen (0741948)
Hugo Snel (0657700)

Junior Management:
Simon Burg
Areti Paziourou
Luc de Smet

Senior Management:
Mark van den Brand, MF 7.096
Lou Somers, MF 7.145

Technical Advisor:
Ion Barosan, MF 7.082

Customer:
Patrick Anderson, GEM-Z 4.137

Eindhoven - June 23, 2013

Abstract

This is the Software Validation and Verification Plan (SVVP) of the Fingerpaint project, developed in the context of the Software Engineering Project (2IP35). This document contains the procedures for verification and validation and complies with the Software Engineering Standard as specified by the European Space Agency (ESA) [1].

Contents

Document Status Sheet

Document Status Overview

General

Document title:	Software Validation and Verification Plan
Identification:	SVVP-1.0
Author:	Roel van Happen
Document status:	Externally approved

Document History

Version	*Date*	*Author*	*Reason of change*
0.0	07-May-2013	Roel van Happen	Initial version.
0.1	22-May-2013	Roel van Happen	Incorporated feedback from the junior management.
1.0	21-June-2013	-	Externally approved.

Document Change Records Since Previous Issue

General

Date:	21-June-2013
Document title:	Software Validation and Verification Plan
Identification:	SVVP-1.0

Changes

Page	*Paragraph*	*Reason to change*
-	-	Externally approved.

Chapter 1

Introduction

1.1 Purpose

This document describes procedures concerning the testing of the delivered products (product documents and software) of the Fingerpaint project for compliance with the requirements. The requirements that the software has to be verified against can be found in the product documents URD [2], Product Backlog [3], SRD [4], ADD [5] and DDD [6]. 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 Fingerpaint project.

1.2 Scope

The Fingerpaint application is an application designed and developed by Group Fingerpaint for Prof.dr.ir. P.D. Anderson. The first goal of the Fingerpaint application is to provide an intuitive and modern interface for an already existing mixing program. This existing mixing program can calculate how the concentration distribution of a certain mixture changes as the mixer mixes. The second goal of the Fingerpaint application is is to provide this service to devices unable to handle the computational load themselves. To this end, the main computation done by the existing mixing program is done on a server. These two goals are formulated as a set of formal requirements in the URD [2].

1.3 List of definitions

2IP35	The Software Engineering Project
AD	Architectural Design
ADD	Architectural Design Document
ATP	Acceptance Test Plan
Client	Prof.dr.ir. P.D. Anderson
CI	Configuration Item
DD	Detailed Design
DDD	Detailed Design Document

ESA	European Space Agency
PM	Project Manager
QAM	Quality Assurance Manager
SCMP	Software Configuration Management Plan
SEP	Software Engineering Project
SM	Senior Management
SPMP	Software Project Management Plan
SQAP	Software Quality Assurance Plan
SRD	Software Requirements Document
SUM	Software User Manual
SVVR	Software Verification and Validation Report
UR	User Requirements
URD	User Requirements Document

1.4 List of references

[1] ESA, *ESA Software Engineering Standards.* ESA, March 1995.

[2] Group Fingerpaint, “User requirements document,” *SEP*, 2013.

[3] Group Fingerpaint, “Product backlog,” *SEP*, 2013.

[4] Group Fingerpaint, “Software requirements document,” *SEP*, 2013.

[5] Group Fingerpaint, “Architectural design document,” *SEP*, 2013.

[6] Group Fingerpaint, “Detailed design document,” *SEP*, 2013.

[7] Group Fingerpaint, “Acceptance test plan,” *SEP*, 2013.

[8] Group Fingerpaint, “Software user manual,” *SEP*, 2013.

[9] Group Fingerpaint, “Software project management plan,” *SEP*, 2013.

[10] Group Fingerpaint, “Software quality assurance plan,” *SEP*, 2013.

[11] Group Fingerpaint, “Software configuration management plan,” *SEP*, 2013.

[12] Group Fingerpaint, “System test plan,” *SEP*, 2013.

[13] Group Fingerpaint, “Integration test plan,” *SEP*, 2013.

[14] Group Fingerpaint, “Unit test plan,” *SEP*, 2013.

Chapter 2

Verification overview

2.1 Organization

2.1.1 Organization

The QAM checks the verification and validation of the activities of the project. Therefore the QAM attends every internal or external review. If the QAM is not available the vice-QAM will take his place, this means that every time the QAM is mentioned it can also be the vice-QAM. If the QAM runs into problems he reports them to the PM. The PM needs to verify that these problems are resolved. The project uses the following methods of verification and validation:

2.2 Internal reviews

In order to keep the quality of our documents up to standards they will be subject to internal reviews. The team carrying out the internal review of a technical or management document will at least consist of the following persons:

- The QAM. He will make the review document.
- One of the authors of the document.
- At least one other member of the project team, not part of the authoring team.
- The adviser/PM may also be present if necessary.

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

2.3 External reviews

When a document has been internally accepted it should have the desired quality. Having the right amount of quality does not automatically mean that the document conforms to the customers expectations. Therefore an external review is held. The documents which need external reviews are the URD, Product Backlog, SRD, ATP, SUM and ADD.

An external review can only take place after the document has been approved by the adviser. Any documents sent to the adviser will have to be accepted internally first. The external reviews of management documents will be done by the SM. The team carrying out the external review of a technical document will consist of the following people:

- The adviser (if available).
- At least one the author(s) (of the document to be reviewed).
- The QAM. He will make the review document.
- At least one other member of the team.
- When necessary (URD [2], Product Backlog [3], SRD [4], ATP [7] and SUM [8]) also the customer

More details about exteranl reviews can be found in Section 4.1.2.

2.4 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 [1]). The SM is allowed to audit the project to check if the procedures, as described in the management documents SPMP [9], SQAP [10], SCMP [11] 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 a the QAM, 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 QAM 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

2.5 Tests

At all times, tests covering functionality are coded before that functionality is implemented. All tests covering all (impelemented) functionality together comprise the test suite, or regression tests. Before and after each code change, the test suite is run, and should pass before code is changed or committed to version control. This follows and extends on Test-Driven Development.

For each type of testing there is a separate test plan. Note that these test plans are updated iteratively each sprint. The following test plans (plans that outline the approach to testing) can be found as separate documents:

- ATP (Acceptance Test Plan) [7]
- STP (System Test Plan) [12]
- ITP (Integration Test Plan) [13]
- UTP (Unit Test Plan) [14]

The ATP [7] 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 QAM. The produced code and product documents must also be tested to assure that all the requirements are met. This can be found in Section 4.3 and is documented in the appendix C.

2.6 Schedule

The schedules for all phases are given in the SPMP [9].

2.7 Resources

In this project we use the testing framework Selenium. Selenium is a library that automates testing with web browsers. Any tests can be automatically applied to all browsers running the Fingerpaint application. Selenium is described in more detail in the SCMP[11].

2.8 Project responsibilities

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

Member of development team:

- 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.
- Manage the test runs.

Configuration Manager:

- Tag documents that have been committed for review, and the approved versions if changes where needed.

Scrum Master:

- Ensure that the Scrum process is followed.
- Check that the backlog is updated and that stories are clear.

Product Owner:

- Check that the items in the product backlog are user centered rather than technical.

2.9 Tools, techniques and methods

The tools that are used during the project are discussed in the SCMP [11].

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 [10]. Furthermore, it is the task of the SQA team to monitor whether the procedures as defined in the management plans (SPMP [9], SCMP [11], SQAP [10] and SVVP) are followed. This is done during team meetings, reviews and by randomly checking CIs. 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 PMs actions are not satisfactory to the QAM, he can request the senior management to take further action.

3.2 Task iteration policy

Every task performed is to be interally reviewed as described in Chapter 4. Some tasks (see Section 2.3) 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 [9]. In this case, it is the responsibility of the QAM to solve the problem and make sure the task is completed as described in the SPMP. If the QAM is unable to do so, he must report this to the PM. If problems arise concerning the dependencies between tasks 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 manhours.

- An item was forgotten during the execution of a task. Depending on the severity of this item the QAM 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 QAMs opinion, this endangers the completion of the project then the QAM 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 [11].

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 [11].

Chapter 4

Verification activities

4.1 Reviews

Review procedures are held during all phases of the Fingerpaint 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 [9]. 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

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 and 2.3 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 QAM 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 retrieve the highest status. If major defects show up during the review the document needs to be changed to solve these defects. Because this involves great changes it should pass a new internal review before it may be subject to another external review.

4.1.1 Internal reviews

The following table shows the action list for the preparation and execution of the external reviews of documents. The QA is the member of the SQA-team that is present. T is the time of the review meeting.

Nr	Actor	Action	Time
1	QAM/QA	Set a date for the internal review of the document	-
2	Leader	Deliver the paper review version of the document to the reviewers	T - 2 workdays
3	Reviewer	Inspect the document (language errors are underlined)	Before T
4	Reviewer	Discuss all errors other than language errors	T
5	Leader	Write down all necessary changes	T
6	Reviewer	Decide if the document can be approved, provided the stated changes are made	T
7	QA	If the document cannot be approved, an appointment for a new review meeting is made	T
8	Leader	Collect annotated documents	T
9	QAM/QA	See to it that the stated remarks are handled properly by the team delivering the document	After T
10	QAM/QA	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 2.3. 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.

Nr.	Actor	Action	Time
1	QAM	Set a date and place for the external review of the document	After internal acceptance
2	Author	Deliver the paper version of the document to all reviewers	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	QAM	Inspect remarks	Before T
6	Author	Lead the meeting and keep discussions to the point	T
7	QAM	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, QAM	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	QAM	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	QAM	See to it that the remarks are handled properly by the team responsible for the document	After T
13	QAM	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 SQA team, 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 Appendix C.
- Software requirements to component requirements and vice versa, this is documented in Appendix C.
- Component requirements to DD requirements and vice versa, this is documented in Appendix C.
- Integration tests to architectural units and vice versa, this is described in the integration test plans [13]. These tests are performed during the sprints.
- Unit tests to the modules of the detailed design, this is described in the unit test plans [14]. These tests are performed during the sprints.
- Acceptance tests to user requirements and vice versa, this is described in the acceptance test plans [7]. These tests are executed during the sprints.

To support traceability, all requirements are uniquely identified.

Chapter 5

Verification reporting

For the verification and validation of technical CIs (apart from the URD [2]) two parts are added to these CIs:

- A verification report
- A validation report

These reports are presented to and checked by a member of the SQA team. The people performing the test of the CI write the verification report. The people who delivered the CI write the validation report. These are both checked when the CI is reviewed. A validation report is written as a result of the tracing. It contains a traceability table. A verification report is written as a result of a test. It contains the following items:

- Unique reference number of the test plan
- Problems discovered and, if available, solutions to these
- Acceptance or disapproval of the CI. In case of disapproval, accompanied with a short explanation of the reasons of disapproval

For the verification and validation of the entire FINGERPAINT project, progress meetings are held with the SM according to the SPMP [9].

Appendix A

Format of reviews

Internally, we will do reviews of all documents. To keep a history record of these and give an author of a document a good overview of what comments there are on his/her document, we have decided to formalize reviews in a simple format. This will make all reviews consistent and will also make updating a document in response to a review easier.

A review is a list of remarks, where each remark consists of three things:

1. One of the following categories, in which the remark falls:
   - Question;
   - Typo (for "typographical error");
   - Incorrect (content);
   - Missing (content);
   - Structure / layout;
   - Inconsistent;
   - Other.
2. Reference to chapter/section/subsection/paragraph/page to indicate where the subject of the remark is located in the document. For clarity, a (brief) quote may be added here to make finding the part of the text that the remark is about easier.
3. The actual remark. This should be a concise yet complete description of what the problem(s) is (are) according to the reviewer.

Furthermore, every review should contain the name of the reviewer, so that it is clear to who the author can go with questions about the review. Note however that the review should be clear enough, so that questions about the review should not be necessary.

Appendix B

User Requirements 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/Product Backlog 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 (written as stories) should be realistic, that is:

- Clear.
- Verifiable.
  "The product shall be user friendly" is not verifiable.
- Complete.
- Accurate.
  Among other things, the URD/Product Backlog is inaccurate if it requests something that the user does not need, for example a superfluous capacity.
- Feasible.
- Traceable, i.e. every requirement should have an unique identifier.

Appendix C

Sprints phase

C.1 SR phase

C.1.1 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.

C.1.2 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.

C.2 AD phase

C.2.1 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 QAM 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 ITP is an adequate plan for integration testing the software in the DD phase.

C.2.2 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.

C.3 DD phase

C.3.1 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 of the QAM 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 ATP specifies the test design, test procedures and test cases so that all user requirements in the URD can be validated.

Appendix D

Transition phase

D.1 The Transition Review

The outputs of the Transition Phase are formally reviewed in the Transition Review (T-R). Any report by the QAM may also be input for the T-R. Internal reviews are held before the formal external T-R takes place. It ensures that:

- The SUM explains what the software does and instructs the users how to operate the software correctly.
- The STD describes the procedures needed to transfer the product to the client.