



Salesmen (Group2 SE)

---

## Software Quality Assurance Plan

---

*author(s):* Patrick Provinciael



**Abstract:** This document details the Software Quality Assurance Plan for the Salesmen Project. This project is done in function of group 2 of the course Software Engineering at the Vrije Universiteit Brussel.

*14-04-2010*

# History of versions

Version	Date	Author(s)	Changes
v0.0.5	18-11-2009	P. Provinciael	First Draft
v0.0.7	19-11-2009	P. Provinciael	Minor Additions
v0.0.8	21-11-2009	P. Provinciael	Remarks by Nick De Cooman
v0.1.1	09-12-2009	P. Provinciael	Addition of QAed repository, review of entire document
v0.2.0	21-12-2009	P. Provinciael	Made the document consistent with the other documents
v0.2.9	10-04-2010	P. Provinciael	Updated: Standards and Conventions, Meeting and Emailing Procedures, Testing
v0.3.0	14-04-2010	P. Provinciael	Updated Bibliography

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Scope . . . . .	3
1.2	Definitions and acronyms . . . . .	3
<b>2</b>	<b>Management</b>	<b>4</b>
2.1	Overview . . . . .	4
2.2	QA Members . . . . .	4
2.3	Tasks . . . . .	4
2.4	Responsibilities . . . . .	5
<b>3</b>	<b>Documentation</b>	<b>6</b>
3.1	Overview . . . . .	6
3.2	Minimal requested documents . . . . .	6
<b>4</b>	<b>Standards, Conventions, Practices and Metrics</b>	<b>7</b>
4.1	Overview . . . . .	7
4.2	Standards and Conventions . . . . .	7
4.3	Documentation Standard . . . . .	7
4.4	Work Practices . . . . .	7
4.5	Meeting Procedures . . . . .	8
4.6	Emailing Procedures . . . . .	8
4.7	Metrics . . . . .	8
4.8	Maximum Amount of Errors . . . . .	9
<b>5</b>	<b>Reviews and audits</b>	<b>10</b>
5.1	Overview . . . . .	10
<b>6</b>	<b>Testing</b>	<b>11</b>
6.1	Unit-testing . . . . .	11
6.2	Integration and system tests . . . . .	11
6.3	Lay-out Design Control . . . . .	11
6.4	Software Test Document . . . . .	11
<b>7</b>	<b>Problem Reporting and Corrective Action</b>	<b>12</b>
7.1	Code Problems . . . . .	12
7.2	Team Interaction Problems . . . . .	12
7.3	Documentation Problems . . . . .	12

---

<b>8 Tools, Technologies and Methodologies</b>	<b>13</b>
8.1 Sourcecode control . . . . .	13
8.1.1 Quality assured code . . . . .	13
8.2 Media control . . . . .	13
<b>9 Riskmanagement</b>	<b>14</b>
<b>References</b>	<b>14</b>

# Chapter 1

## Introduction

### 1.1 Scope

This document details the Software Quality Assurance Plan for the Salesmen Project. This project is done in function of group 2 of the course Software Engineering at the Vrije Universiteit Brussel.

The purpose of this document is to define the various control mechanisms to assure the quality of the delivered project is high enough. It will also explain the various conventions used to assure the quality.

The Quality Assurance team is not meant to create the quality. It's sole purpose is to assure that at the end of the project, all members have followed the decided standards and mechanisms. This will be a continuous job. This way, early errors will have little impact on the rest of the project.

### 1.2 Definitions and acronyms

- QAM: Quality Assurance Manager
- SQAP: Software Quality Assurance Plan
- SCMP: Software Configuration Management Plan
- SPMP: Software Project Management Plan
- SDD: Software Design Document
- SRS: Software Requirements Specification
- STD: Software Test Document
- KLOC: Kilo lines of code

# Chapter 2

## Management

### 2.1 Overview

The quality assurance team will consist of two members. The quality assurance manager (QAM) and the assistant QAM. These persons will be responsible of assuring that all members follow the guidelines as set by this document. Each individual of the project is responsible for maintaining these guidelines. The QAM and in some cases also the Assistant QAM will proofread every document and code, before it is actually released to the public. Their comments are binding and can only be overruled by the project manager or by a majority of the project members in a, by the project manager approved, voting. The comments of the assistant QAM can also be overruled by the comments of the QAM.

### 2.2 QA Members

The members of the Quality Assurance team can always be found in the SPMP[1].

### 2.3 Tasks

The QAM is responsible for the following tasks:

- quality control of official documents.
- quality control of the minutes.
- quality control of the code. He may also add additional comments to the code.
- quality control of the tests for the code.
- quality control of the workload, to ensure no redundant tasks will be done.

These tasks will be detailed further in this document.

The Assistant QAM is responsible for the following tasks:

- quality control when requested by the QAM.
- quality control of documents and code written by the QAM.
- taking over the QAM tasks when QAM is absent.

The assistant QAM is allowed to do tasks of the QAM, without request of the QAM. However, the results of these actions can be overruled by the QAM.

When a project member does not agree with a comment of the assistant QAM, he can appeal to the QAM. When a project members does not agree with a comment of the QAM, he can appeal to the Project Manager.

## **2.4 Responsibilities**

It is the responsibility of the QAM to assure that the project members follow the guidelines as set by this document. If any person fails to do so, the QAM is allowed to directly confront the member with this.

It is the responsibility of the Project Manger to assure that the QA team follows the guidelines as set by this document.

It is the responsibility of every individual member to uphold the QA Requirements as set by this document. Individual members are also responsible for repairing the errors pointed out by the QA team.

## Chapter 3

# Documentation

### 3.1 Overview

This chapter will detail the control mechanisms to assure the quality of the documentation of the project.

### 3.2 Minimal requested documents

The following documents will be produced:

- SQAP: Software Quality Assurance Plan
- SCMP: Software Configuration Management Plan
- SPMP: Software Project Management Plan
- SRS: Software Requirements Specifications
- SDD: Software Design Document
- STD: Software Test Document
- Java broncode documentation.



## Chapter 4

# Standards, Conventions, Practices and Metrics

### 4.1 Overview

This section will detail the used standards and conventions. It will also detail certain practices that each individual member must follow in order to retain a high standard. There will also be sections on mailing procedures, as well as meeting procedures.

A section regarding the metrics used in the project can be found in this section as well.

### 4.2 Standards and Conventions

There are standards and conventions on java code and xhtml code. These conventions and standards are prone to change. Any programmer has the right to request a change. The current locations of standards and conventions are as follows:

**Java** <http://code.google.com/p/salesmen/wiki/JavaCodingConvention>

**xhtml** <http://code.google.com/p/salesmen/wiki/XhtmlCodingConvention>

### 4.3 Documentation Standard

All documents must be written in LaTeX. The source code of the latest revision of a document must be available at all times on the SVN server of the project. Documentation of code must exist in a separate document, as well as commented within the code itself.

All non-code documentation must follow a given template. This template will be decided by the QAM. This is done to assure the consistence of the documents.

The current used template is the softproj template. This template was created by Patrick Provinciael and can be found on the SVN-server of the project.

### 4.4 Work Practices

Every individual member of the project must complete the following control tasks regarding his own work, before allowing the QA team to control the quality.

- He must follow the quality requirements during the work, not after.

- He must strictly follow the deadlines.
- He must allow access for all members to his work through the SVN-Server at all times.
- He must proof read his own work at least once.

By following these tasks, a member will decrease the amount of time spent on repairing his own work.

## 4.5 Meeting Procedures

- Meetings will be run by the Project Manager. He will mail the agenda the day before the meeting to all members. The project Manager is also responsible of deciding the meeting location, date and time. He must make sure that this information is mailed to the attendees well before the actual meeting begins.
- Members unable to come must notify the Project Manager three hours before the actual meeting. They must provide a valid reason for doing so.
- Members may raise additional issues on a meeting after the agenda has been finished. These issues will then be handled if there is time left, or put on the agenda of next meeting. The minutes will detail these additional issues.
- If an emergency meeting must be arranged, the Project Manager must notify members at least twenty-four hours before the meeting. Members have twelve hours to reply on the notification. Members who have not verified their attendance after this time must be called on the phone by the Project Manager.
- A minutes document will be written during every meeting. This document must be available on the official Salesmen website within three days after the meeting.

## 4.6 Emailing Procedures

All emails must be sent to the google groups [10] of the project. They should contain the target person(s) in the subject line. There will be no private correspondence. This to assure that subjects are handled by the right persons. Whenever a subject is sent to the wrong person, the right person can still reply. This is impossible with private communication. The sole exception on this rule is personal communication between the manager and another member regarding behavior of the member.

## 4.7 Metrics

The following metrics will be used to control the amount of work spent by the members on the project.

- The time spent on working for the project.
- The amount of lines produced by each individual member, expressed in KLOC.

## 4.8 Maximum Amount of Errors

This section will detail the maximum amount of errors that this project may have to qualify for good quality.

- Requirement errors : No more than one error in every twenty requirements.
- Design errors : No more than one error for every 5 charts
- Code errors : No more than ten errors for every KLOC, noticed by the customer.

In addition to this, the code must contain a good error handling system. Whenever an error appears in the final product, a special tracking mechanism must show when exactly the error appeared. This must be noted in a special log file.

## Chapter 5

# Reviews and audits

### 5.1 Overview

The purpose of the reviews and audits is to ensure that the project members focus on the quality of the application. Every document will be inspected to assure the quality. This way we can assure that these documents are complete and consistent.

There must be a review with the release of every document. There must also be a review of a previous and coming cycle at the start of each new implementation cycle.

## Chapter 6

# Testing

### 6.1 Unit-testing

A unit-testing framework will be written. This framework will be written by the QAM and reviewed by the assistant QAM. The framework will be made with the sole purpose of black-box testing. This project will not handle white-box testing.

Every individual member is responsible for writing his own tests. These tests will be reviewed by the QA team to assure that it is testing every possible input. The QA team has the right to add or delete tests within the separate test files.

### 6.2 Integration and system tests

These tests will be primarily done by the QAM. Project members are encouraged to perform their own tests.

These tests will be started as soon as the first stable version of the software is released. Afterwards these tests will be done every time a new stable version is released.

### 6.3 Lay-out Design Control

The Project Manager will ensure that the final lay-out of the final product is good. He will look at the color-scheme, user-friendliness and the overall lay-out.

The QAM ensures that the final lay-out is without bugs and errors. He will also look at the usability.

All members are urged to test the lay-out and to ask their close relatives and friends to test the lay-out.

### 6.4 Software Test Document

The Software Test Document (STD)[6] will be written by the QAM and reviewed for QA by the assistant QAM. It will detail all aspects of testing. As well as requirements on how testfiles must be constructed.

## Chapter 7

# Problem Reporting and Corrective Action

### 7.1 Code Problems

Code errors will be reported through the issue system of the Google Code website[9]. They will be tracked by the implementation manager. He will also assign the right persons to solve a problem.

### 7.2 Team Interaction Problems

If a person has a problem with another member(s), he should contact the Project Manager. He must make a detailed report of the problems, including his own errors in the case. The Project Manager will then contact everyone else involved to get their side of the story, without notifying them of what every person has already told him. Afterwards the Project Manager will attempt to verify the complete story with every person involved and attempt to find a suitable solution.

When no other persons are involved, the Project Manager must attempt to find out the problem of this one person. He must then find a suitable solution for his problem.

Only in extreme cases can the Project Manager exclude someone from the project. Extreme cases are as follows, but not limited to:

- Physically assaulting another member.
- Intentionally not telling the truth.
- Refusing to do tasks without a good reason.

### 7.3 Documentation Problems

Minor errors in a document may be solved by any member directly in the source code. Minor errors are either grammatical or spelling-related.

Major errors must be reported to both the QAM and the person responsible for the document. The person responsible for the document must make sure to solve this error within twenty-four hours. The QAM must check if this errors is solved after this time. If not, he must remind the person responsible of his duties.

## Chapter 8

# Tools, Technologies and Methodologies

### 8.1 Sourcecode control

The QAM will assure that all code follows the chosen standards and conventions. The Configuration manager will make sure that there is always a working copy on the SVN server.

#### 8.1.1 Quality assured code

To make a clear distinction between assured code and non-assured code, the QAM will add a copy of the assured code to a separate branch of the repository. Each time code is assured for quality, this process must be repeated.

To ensure that a stable version of the code is quality assured, all stable versions must be made with the quality assured branch. This ensures that every piece of code is quality assured by the QAM. This will result in quality assured software.

### 8.2 Media control

The Configuration Manager will assure that the following tasks are completed by individual members:

- Every Project member must have a personal copy of the Salesmen repository on his hard drive.
- The Configuration Manager will make periodical additional backups on a separate hard drive. These should not be used for working on the project. Other members are encouraged to make periodical backups as well.
- Members must make sure that every file they are working on has a working copy on the SVN repository.

## Chapter 9

# Riskmanagement

Whenever a project member discovers a potential risk, he must notify the Project Manager as soon as possible. The Project Manager must add this risk to the agenda. He must also consider several solutions to the problem.

Risk management is discussed in more detail in the SPMP[1].



# Appendix A

## References

- [1] Software Project Management Plan, Nick De Cooman, [http://wilma.vub.ac.be/~se2\\_0910/docs/spmp/index.html](http://wilma.vub.ac.be/~se2_0910/docs/spmp/index.html), October 26, 2009.
- [2] Software Configuration Management Plan, Jorne Laton & Sina Khakbaz Heshmati, [http://wilma.vub.ac.be/~se2\\_0910/docs/scmp/index.html](http://wilma.vub.ac.be/~se2_0910/docs/scmp/index.html), October 8, 2009.
- [3] Software Design Document, Bart Maes, [http://wilma.vub.ac.be/~se2\\_0910/docs/sdd/index.html](http://wilma.vub.ac.be/~se2_0910/docs/sdd/index.html), November 30, 2009.
- [4] Software Requirements Specification, Wouter Van Rossem, [http://wilma.vub.ac.be/~se2\\_0910/docs/srs/index.html](http://wilma.vub.ac.be/~se2_0910/docs/srs/index.html), November 12, 2009.
- [5] Software Quality Assurance Plan, Patrick Provinciael, [http://wilma.vub.ac.be/~se2\\_0910/docs/sqap/index.html](http://wilma.vub.ac.be/~se2_0910/docs/sqap/index.html) November 22, 2009.
- [6] Software Testing Document, Patrick Provinciael, [http://wilma.vub.ac.be/~se2\\_0910/docs/std/index.html](http://wilma.vub.ac.be/~se2_0910/docs/std/index.html) December 06, 2009.
- [7] TestNG Unit Testing Framework, <http://testng.org/>
- [8] Salesmen Project Website, Salesmen, [http://wilma.vub.ac.be/~se2\\_0910/](http://wilma.vub.ac.be/~se2_0910/), November, 2009.
- [9] Code Repository, Salesmen, <http://code.google.com/p/salesmen>, November, 2009.
- [10] Salesmen Google Groups Communication Channel, Salesmen, <http://groups.google.com/group/salesmen>, November, 2009
- [11] Slides Software Engineering, Dirk Vermeir, [http://tinf2.vub.ac.be/~dvermeir/courses/software\\_engineering/slides.pdf](http://tinf2.vub.ac.be/~dvermeir/courses/software_engineering/slides.pdf), October 1, 2009.