Quality inside Scia Engineer

CONTENTS

1 Document Perspectives

Subject	QA and QC in Scia Engineer
Purpose	To give an overview of the QA and QC activities (policy, procedures) related to the Scia Engineer package.
Author	CVL
Release	
Document	QA_QC_SE(CVL20150603)
Distribution	
Date	11/01/2011 – updated 03/06/2015

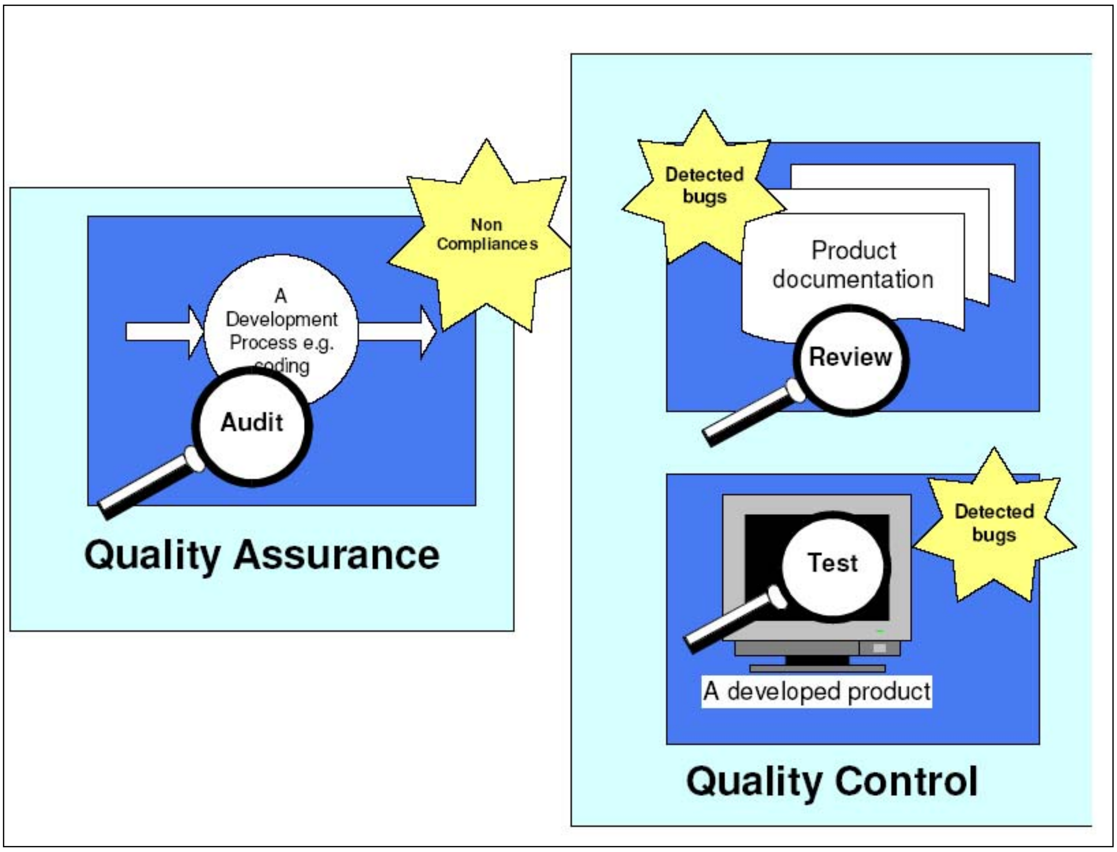
2 Introduction

Quality Control (QC) or ‘Verification and Validation’ (V&V, VER, VAL) is the process of ensuring that software being developed or changed will satisfy functional and other requirements (validation) and each step in the process of building the software yields the right products (verification).
The purpose of Validation (VAL) is to demonstrate that a product fulfils its intended use when placed in its intended environment. The purpose of Verification (VER) is to ensure that selected work products meet their specified requirements. In other words, verification ensures that “you built it right”; whereas, validation ensures that you “you built the right thing”. See also Ref.[1], Chapter 'Verification’ and ‘Validation’.

The two major QC activities are reviews and testing. The review of work products is performed by peers during development of the work products to identity defects for removal. Testing is the operation of the software with real or simulated inputs to demonstrate that a product satisfies its requirements and, if it does not, to identify the specific differences between expected and actual results.

Quality Assurance (QA) is defined as the planned and systematic pattern of all actions necessary to provide adequate confidence that the software has been designed and developed using adequate processes. Quality Assurance evaluates the process and the associated work products. It provides staff and management with objective insights into these processes and work products. See also Ref.[1], Chapter 'Process and Product Quality Assurance'.

The difference between Quality Assurance (QA) and Quality Control (QC) is depicted in the figure below :



In this document, the SCIA policy concerning QC and QA are described (Chapter 3). The used model (Chapter 4) and supporting processes and procedures (Chapter 5) are explained.

3 Policy

3.1 QC Policy

The SCIA policy for the Quality Control is the following :

> “
>
> Product designs are verified in accordance with planned arrangements (design and development project plan). The purpose is to ensure that the design and development outputs have met the design and development input requirements.
>
> “
>
> “
>
> Product designs and development are validated in accordance with planned arrangements (design and development project plan) and that the resulting product is capable of meeting the requirements for specified application or intended use, where known.
>
> “

3.2 QA Policy

The SCIA policy for the Quality Assurance is the following :

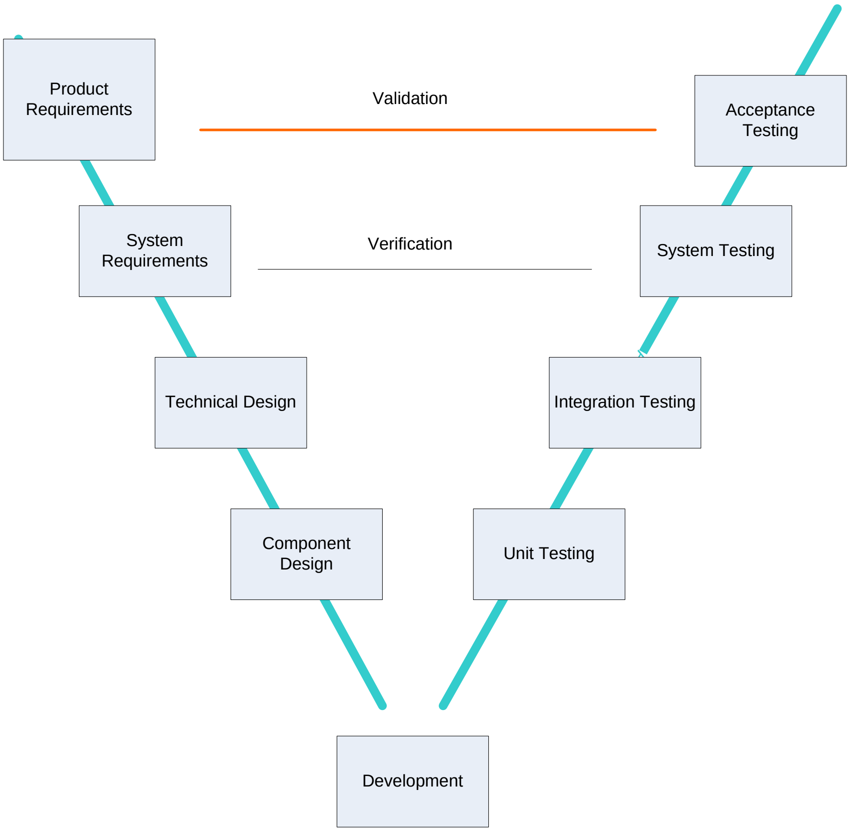
> “
>
> A strategic system of planned and periodical audits is implemented to verify compliance with all aspects of the Quality Management System.
>
> “

4 The V-Model

The applied model for achieving the QC policy is the so-called V-Model. The V-Model, also called the Vee-Model, is a product-development process. The V-Model gets its name from the fact that the process is often mapped out as a flowchart that takes the form of the letter V.

The development process proceeds from the upper left point of the V toward the right, ending at the upper right point. In the left-hand, downward-sloping branch of the V, the business requirements, the application design parameters and the design processes are defined. At the base point of the V, the code is written. In the right-hand, upward-sloping branch of the V, testing and debugging is done. The unit testing is carried out first, followed by bottom-up integration testing. The extreme upper right point of the V represents product release.

The V-model illustrates how QC (testing) activities are integrated into each phase of the software development life cycle.



4.1 Product Requirements

The Product Requirements are representing the high level solution. The Product Requirements are composed by the Expert Team (Development Department) and are approved by the Product Department.

4.2 System Requirements

This represents the list of requirements. It captures the user's wishes from the system. System Use Cases are added to clarify these requirements. This document is composed by the Product Development Engineer. It is reviewed by the Development Manager, and the Software Engineer has to accept the document. The Product Development Engineer describes the Test Cases for the System Testing, related to the System Requirements.

4.3 Technical Design

The Software Engineer creates the technical solution, based on the System Requirements. The Technical Software Architect / Software Engineer Experts approves the Technical Design document.

4.4 Component Design

The Software Engineer breaks the designed system in to smaller units. The unit test design is developed in this stage.

4.5 Unit testing

In the Unit Testing, both module structure and functionality are tested. These Unit Tests are performed on the units and modules while the programmer is coding them. The Software Engineer of a unit owns this unit and is responsible for testing it and finding any bugs contained in it.

4.6 Integration Testing

In the Integration Testing, the separate modules are tested together to find weaknesses and bugs in the system. This way any bottlenecks in the system can be identified and the corresponding module can be redesigned. To evaluate the influence of the new developed modules on the system, automated tests of the existing functionalities are used. Software Engineers and Product Development Engineers are involved.

4.7 System Testing

The System Testing compares the System Requirements against the actual system. Test Cases are derived from the Systems Requirements and use cases. The functionality of the system is tested based on these tests. New automated test files are initiated. The QC Team and the Product Development Engineers are involved.

4.8 Acceptance Testing

The goal of acceptance testing is to verify that the agreed Product Requirements have been achieved. Acceptance testing is the final testing stage before taking the system into full operation. The Product Engineers are involved.

5 Related Processes and Procedures

5.1 Software Product Life Cycle

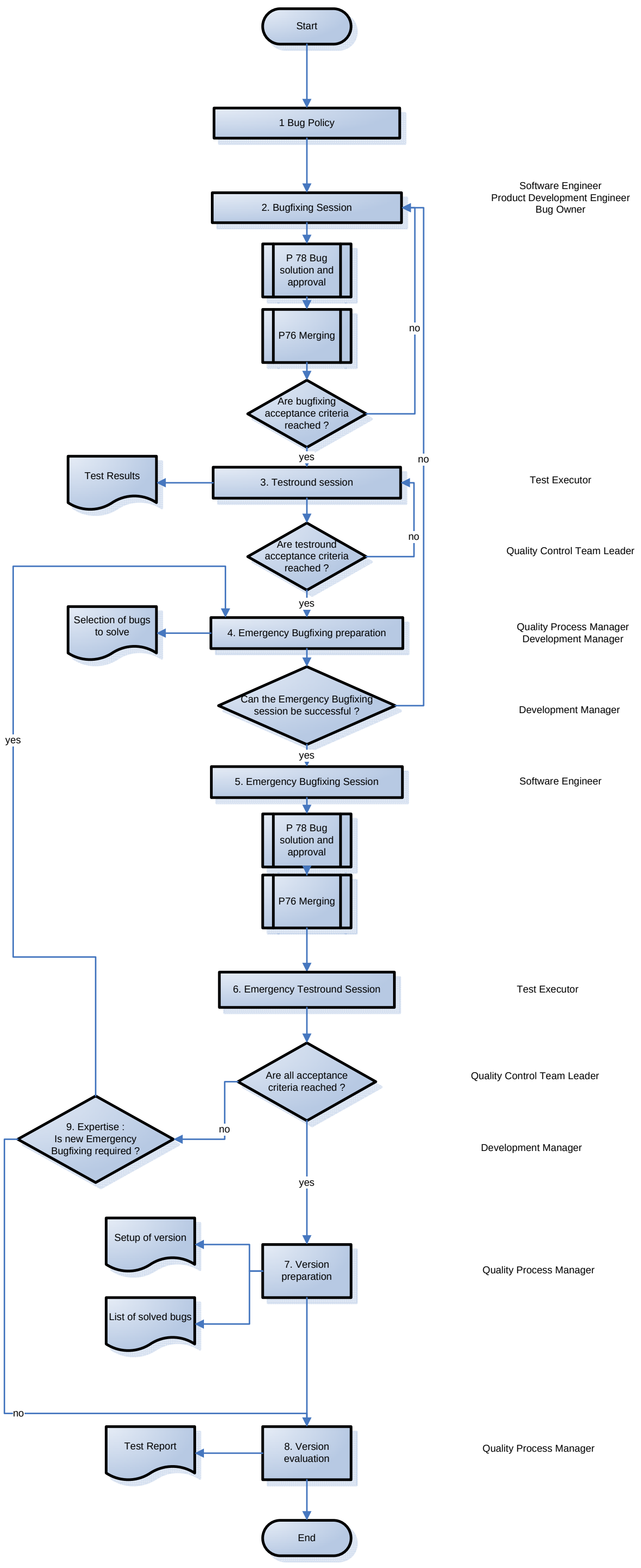
To support the SCIA QC policy and the V-Model, and taking into account the CMMI guidelines (Ref. [1]) , a set of processes and procedures are established and maintained. The total of these processes are representing the Software Product Life Cycle for Scia Engineer.

The development of the standard release is split in three main phases: the product vision and planning, the release planning and implementation, and the external launch of the release. The implementation is executed in several iterations, the so-called Sprints.

Inside this Software Product Life Cycle, the V-Model is applied on two levels: on the version (Release) level and on the iteration (Sprint) level.

5.2 Quality Improvement Procedure

Before a version is released (patch, release), the Quality Improvement Procedure is followed. The Quality Improvement Procedure is a sequence of bug fixing and testing, which ends when the Quality Improvement Acceptance criteria (Chapter 0) are reached.


Start
1 Bug Policy
2. Bugfixing Session
Software Engineer
Product Development Engineer
Bug Owner
P 78 Bug solution and approval
P76 Merging
no
Are bugfixing acceptance criteria reached ?
yes
no
Test Results
3. Testround session
Test Executor
no
Are testround acceptance criteria reached ?
Quality Control Team Leader
yes
Selection of bugs to solve
4. Emergency Bugfixing preparation
Quality Process Manager
Development Manager
Can the Emergency Bugfixing session be successful ?
Development Manager
yes
yes
5. Emergency Bugfixing Session
Software Engineer
P 78 Bug solution and approval
P76 Merging
6. Emergency Testround Session
Test Executor
Are all acceptance criteria reached ?
Quality Control Team Leader
9. Expertise : Is new Emergency Bugfixing required ?
no
Development Manager
yes
Setup of version
7. Version preparation
Quality Process Manager
List of solved bugs
no
Test Report
8. Version evaluation
Quality Process Manager
End

5.3 Quality Improvement Acceptance Criteria

The Quality Improvement procedure (Ref.[5]) has 3 major parts : the Bug Fixing Session, the Test Round Session and the Emergency Bug Fix Session.

5.3.1 Bug Fixing Session

At start of the Bug Fixing Session, the scope of the session is defined according to the Bug Policy and the solving of the bugs is according to set SLA (Service Level Agreements). The SLA for Priority 1 and Priority 2 are monitored.

- The following Service Level Agreements (SLA) will be in place for the External DevTrack (for bugs introduced in 2015) :

PRIORITY	CSupport Priority	SLA
P0	-	Immediate solution and approval – special setup is created
P1	High	Lead Time Q3 < 60 days
P2	Medium	Lead Time Q3 < 90 days
P3	Low	-
NO BUG		Lead Time Q3 < 14 days

For 2015, the Lead Time from 2014 are taken into account, and the SLA are :

PRIORITY	SLA
P0	Immediate solution and approval – special setup is created
P1	Lead Time Q3 < 98 days
P2	Lead Time Q3 < 108 days
P3	-
NO BUG	Lead Time Q3 < 12 days

5.3.2 Emergency Bug fixing Session Completion Criteria

Open Priority 1 bugs are selected for the Emergency Bug fixing Session.
The Emergency Bug Fix Session Completion Criteria are as follows :

- All selected Priority 1 Bugs are solved - – exceptions must be evaluated and clarified
- All automated tests (Poirots and Unit Test) must be OK – exceptions must be evaluated and clarified.

5.3.3 Test Round Session Completion Criteria

The Test Round Session Completion Criteria are as follows :

- The TEC (Test Execution Coverage) must be at least 95%.

5.4 Automated Testing

For each new build, the complete set of automated test cases is executed. The automated test cases are used to test the results of the solver and post processing (steel code, concrete code, …).
For this purpose, a tool called Poirot is developed. It enables to run Scia Engineer in batch mode, export the defined results and compare them to the correct expected status. The tool can check numerical and graphical results.
Currently, there are approximately 7500 files executed each build.

The results of the automated test cases are part of the acceptance criteria – see Chapter **Error! Reference source not found.**.

5.5 QA activities

Quality Assurance (QA) is a planned and systematic means for assuring management that the defined standards, practices, procedures and methods of the products are applied.

The processes are evaluated by means of regular internal audits.

On top of the regular peer reviews, executed inside the Product Department and the Development Department, the documents are evaluated if they are consistent, brief, clear, complete, unique, verifiable and traceable, as described in the relevant check lists, by a member of the Quality Department. The selected work products for evaluation are the Customer Requirements, the System Requirement and the Test Case.

6 References

[1] Chrissis M.B., Konrad M., Shrum S.
CMMI Guidelines for Process Integration and Product Improvement
Addison Wesley, 2003