VERIFICATION AND VALIDATION PLAN

TEMPLATE

DOCUMENT CONVENTIONS

This document is a Verification and Validation Plan template. As such, wording in this document should be supplemented with project-specific information. Therefore, tailor (add, delete, change, or expand) the information provided in this document

Standard conventions are used within this document to direct the reader to specific sections of the text. These sections provide instructions and explanations and require users to substitute their own department-specific information for the generic information provided or to "fill in a blank."

[[text]] Global changes. Items that appear in regular text and are surrounded by double brackets represent changes that can be made globally throughout the document.

Italics Instructions and explanations. Items that appear in italics represent instructions to the user and are not to appear in the completed version of the document.

In some cases where information may already be found in another project document, like the Software Development Plan (SDP), refer to that document rather than duplicate the information here.

Update the header page to reflect the document configuration identifier for the project V&V Plan.

[[PROJECT NAME]]

SOFTWARE VERIFICATION AND VALIDATION PLAN

[[DOCUMENT DATE]]

[[Add your organization name here]]

V&V Plan Approvals:

_____	_____
V&V Manager	**Date**
_____	_____
Project Manager	**Date**
_____	_____
Program Manager	**Date**

Table of Contents

Contents

SECTION 1. OVERVIEW

The purpose of this plan is to define the [[Project Name]] Verification and Validation (V&V) organization, SQA tasks and responsibilities; provide reference documents and guidelines to perform the SQA activities; provide the standards, practices and conventions used in carrying out SQA activities; and provide the tools, techniques, and methodologies to support SQA activities, and SQA reporting.

1.1 SCOPE

This plan establishes the V&V activities performed throughout the life cycle of the [[Project Name]].

V&V processes consist of the Verification process and the Validation process. The Verification process provides objective evidence for whether the products:

- Conform to requirements (e.g., for correctness, completeness, consistency, and accuracy) for all activities during each life cycle process.
- Satisfy the standards, practices, and conventions during life cycle processes.
- Successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities (i.e., builds the product correctly).

The Validation process provides evidence for whether the products:

- Satisfy system requirements allocated to the products at the end of each life cycle activity.
- Solve the right problem (e.g., correctly model physical laws, implement business rules, and use the proper system assumptions).
- Satisfy intended use and user needs in the operational environment (i.e., builds the correct product).

The Verification process and the Validation process are interrelated and complementary processes that use each other's process results to establish better completion criteria and analysis, evaluation, review, inspection, assessment, and test V&V tasks for each life cycle activity.

1.2 V&V OBJECTIVES

The objectives of this plan are to:

- Correlate V&V tasks to project management.
- Identify key review activities
- Ensure V&V is applied at each stage in the SW development process.

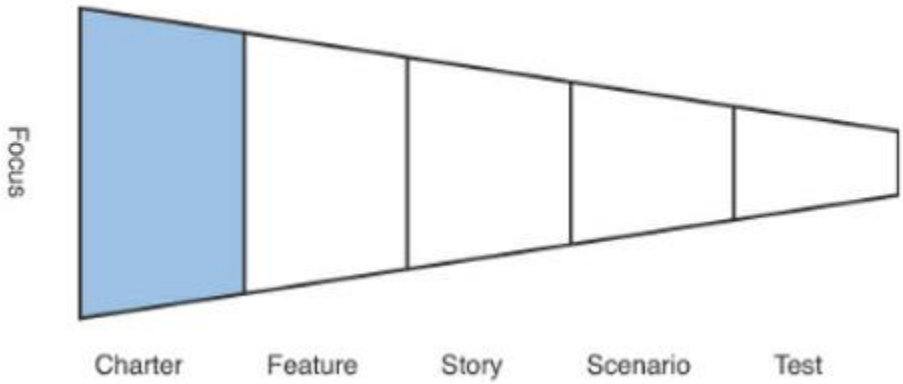
The purpose of the V&V activities are to:

- Discover defects
- Assess whether or not the system is usable in an operational situation.

1.3 SYSTEM OVERVIEW

The [[System Name]] c*omplete the sentence by providing a description of the system and the intended use of the system.*

Software development will be using Acceptance Test Driven Development.



1.3 Project references

1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

1.5 POINTS OF CONTACT

SECTION 2. MANAGEMENT APPROACH

Describe the portion of the software life cycle covered by this V&V Plan, the tasks to be performed with special emphasis on V&V activities, and relationship between these tasks and the planned major checkpoints. The sequence of the tasks should be indicated. Tailor this section to reflect those tasks being verified that relate to the project's current/projected activities.

The scheduling of V&V tasks is driven by the software development schedule. Therefore, a V&V task is performed in relationship to what software development activities are taking place. One or more V&V tasks can be performed concurrently until a task is completed. A task is considered completed when the required report e.g., V&V sign-off, reports, etc. are satisfactorily completed or action items have been closed. The following tasks, requiring coordination and cooperation with the project team, shall be performed by V&V.

As a result of the successful implementation of the V&V Management process:

- The VVP is developed.
- The V&V effort is continuously reviewed.
- Revisions of the VVP are made as necessary based on updated project schedules, development status, or changes in V&V approach.
- V&V results are coordinated with other parties performing life cycle activities.
- V&V task results are reviewed for conformance to task requirements.
- Recommendations are made to program management regarding readiness to proceed to the next stage of the project life cycle, acceptance, and certification.
- Process improvement opportunities in the conduct of V&V are identified.

2.1 RESPONSIBLE TEAMS

The following groups are defined.

Stakeholders:

- Sponsor, procurer, funder
- Customers (manager of organization who will use the system)
- Users
- Requirements engineer
- System architects
- Developers
- Testers
- Documentation writers
- Project managers
- Legal staff
- Manufacturing people
- Sales, marketing, field support, help desk and those who will work with the product and its customers.

Planning team (Plan) – the following roles are needed. One person can play multiple roles.

- Sponsor, procurer, funder
- Customer (manager of organization who will use the system)
- User
- Requirements engineer
- System architect
- Developer
- Tester

Development team (Dev) – the following roles are needed. One person can play multiple roles.

- User
- Requirements engineer
- System architect
- Developer
- Tester

V&V team (V&V) – the following roles are needed. One person can play multiple roles.

- Requirements engineer
- System architect
- Developer
- Tester
- Quality assurance verifier

2.2 MODIFICATION OF ACTIVITES

This document details activities generally helpful for projects. V&V teams can assess what works best in their environment.

2.3 TRAINING

Needed training will be provided by the V&V team.

2.4 DEVELOPMENT/TESTING FRAMEWORK

A testing framework for unit tests and continuous integration is assumed.

SECTION 3. TESTING SCHEDULE

3.1 TESTING SCHEDULE

The master schedule summarizes the various verification and validation tasks and their relationships to the overall project. Describe the project life cycle and project milestones including completion dates. Summarize the V&V tasks and how results provide feedback to the development process to support overall project management functions.

Phase	Item #	V&V Task	When	Precondition	Postcondition	Responsible	Method	Environment	Time
Charter development & presentation									
Charter - check-off	1	Verify objectives are SMART	Each time a charter or mini-charter is developed	Charter developed	Objectives sign-off, ok to present	V&V	Check-off	Dev	1h
	2	Risk analysis	Each time a charter or mini-charter is developed	Charter developed	Risk analysis sign-off	V&V	Check-off	Dev	2h
	3	Assure charter presentation and follow-up occurred	Each time a charter or mini-charter is developed	Presentation, comment period and responses completed	Presentation sign-off	V&V	Check-off	Dev	1h-4h
Features with acceptance criteria developed									
Feature check-off	4	Check-off feature acceptance criteria	For each set of features defined	Features defined, prioritized, reviewed and acceptance criteria developed	Feature acceptance criteria sign-off	V&V	Check-off	Dev	1h
Story with acceptance criteria developed									

Story check-off	5	Check-off story acceptance criteria for sprint	Prior to beginning of sprint	Sprint stories developed with acceptance criteria defined	Story acceptance criteria sign-off	V&V	Check-off	Dev	1h
Sprint - TDD and CI used. Cross story, business, ui, external interface, component tests suggested by development team tester and performed by team, story acceptance tests solidified and performed.									
	6	Sprint tests*	Post sprint	Sprint completed with business value to customers, stories reassigned if necessary.		V&V, plan	Man/auto	Build integration	1w
Stories tested	7	Story tests solidified & performed	After sprint tests	Sprint tests have been completed.	Test sign-off	Plan	Man/auto	Build	1d
	8	Story tests performed by users	After V&V testing of stories	V&V have tested stories.	User story sign-off	V&V	Manual	Test	1d
Features tested	9	Feature tests solidified & performed	After all relevant story tests performed	All story tests for the feature have been completed.	Feature test sign-off		Manual		2 hour of whol team
	10	Story tests performed by users	After V&V testing of stories	V&V have tested features.	User feature sign-off	Users	Manual	Prod	2h, user
Charter tested		Alpha and beta testing	After features tested	Features tested	Anomalies from testing reported	Users	Manual	Prod	2w
	11	Quality attribute and exploratory , regression	Set of features completed	Deployment		Users	Manual	Prod	1w

Sprint tests occur after the sprint is completed by the V&V team (including the tester on the development teams) - Additional business, user interface, external interface, component tests, quality attribute tests and exploratory testing.

For the above activities prepare a schedule for this project. Identify who will perform the activities, date when the activity will be completed, and contingencies should responsible individuals need to be replace.

Deliverable	Responsibility	Due date

3.2 EQUIPMENT REQUIRED

3.3 SOFTWARE REQUIRED

3.4 PERSONNEL REQUIRED

3.5 DELIVERABLES

3.6 TESTING TOOLS

3.6 SITE SUPPLEMENTAL MATERIALS

SECTION 4. TEST REPORTING

4.1 TEMPLATE AND POLICY FOR RECORDING TESTS

Software is to be implemented using test driven development, that is, write the test first and make sure that it fails. Implement the code and check that the test passes.

Continuous integration is used. As code is added to the repository, a fresh build and test is implemented. Failed tests will have:

- Automatic emails to person who committed change where bug is first recognized
- Automatic recording in a bug tracking system
- Commit id
- Call stack

During sprints, accomplish the above in the development/testing framework. Major bugs from this framework are recorded in the test and anomaly tracker.

All user and manual tests performed subsequent to sprint development are to be recorded in the following:

Id	Date	Reference to document	Reference within document	Build	Commit ID	Result	Observer	Comments

4.2 POLICY FOR FAILED TESTS AND ANOMALLIES

When a test has failed, submit an issue/bug in the bug tracker containing at least:

- Test id – from test reporting
- Reference to document, reference within document
- Build number
- Explanation of failure
- Reproduction steps
- Severity (high, medium, low)
- Priority (high, medium, low)
- Reported by
- Assigned to
- Date reported
- Reason
- Status (new, open, active)
- Environment (Windows 10, SQL Server 2007)
- Notes

When tests fail there is a bug review. At that time the planning team decides whether to make updates to charter, feature, story, and scenario. If needed, work is rescheduled.