

# **VAIL-Plant**

## Asset Integrity Management System



## **Software Development Process**

**Document Number: VAIL/SDP/2008/008**



Engineering For a Safer World

Public

**Prepared by :** Badar Ibrahim Khan      **Reviewed By:** Shahid Javed Kalair      **Date :** 03/01/2008


**Approved by :** Ijaz Ul Karim Rao      **Revision:** 0      **Page:** 2-of-15

**Revision Control Sheet (RCS)**

<b>Rev. No</b>	<b>Issued Date</b>	<b>Details of Revision</b>
0	03/01/2008	Final Approved Submission

Note :

1. Revision numbers, starting in alphabetic letters indicate “Draft Issues” for Review/comments.
2. Revision numbers , starting from ‘ Zero” in Numeric letters indicate “Issue for implementation and revision number of subsequent revisions will go upwards like 1,2,3 etc., onwards

	<b>VAIL-Plant Software Development Process</b>	
<b>Prepared by :</b> Badar Ibrahim Khan	<b>Reviewed By:</b> Shahid Javed Kalair	<b>Date :</b> 03/01/2008
<b>Approved by :</b> Ijaz Ul Karim Rao	<b>Revision:</b> 0	<b>Page:</b> 3-of-15

### Copyright Notice

Copyright © VELOSI Asset Integrity Ltd. 2008

All rights reserved.

This documentation contains proprietary information of VELOSI. It is provided under the license agreement containing restrictions on use and disclosure. All rights including reproduction by photographic or electronic process and translation into other languages of this material are fully reserved under copyright laws. Reproduction or use of this material in whole or in partial in any manner without permission from Velosi is strictly prohibited.

### Trademarks


VELOSI is a registered trademark.

Other product names that are mentioned in this document may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

<b>Prepared by :</b> Badar Ibrahim Khan	<b>Reviewed By:</b> Shahid Javed Kalair	<b>Date :</b> 03/01/2008
<b>Approved by :</b> Ijaz Ul Karim Rao	<b>Revision:</b> 0	<b>Page:</b> 4-of-15

**Table of Contents**

- 1. Overview .....5**
- 2. Purpose .....5**
- 3. Decomposing the Software Development Process .....5**
  - 3.1. Activities in Software Development .....6
    - 3.1.1. Requirements Engineering .....8
    - 3.1.2. Requirements Review .....8
    - 3.1.3. Requirements Analysis and High level Design .....8
    - 3.1.4. Detailed Design – Application and Data Design .....9
    - 3.1.5. Design Review .....9
    - 3.1.6. Peer Code Review .....10
    - 3.1.7. Commit for Merging .....**Error! Bookmark not defined.**
    - 3.1.8. Code Merging .....11
    - 3.1.9. Testing .....11
    - 3.1.10. Reporting issues and assigning to developers .....12
    - 3.1.11. Tracking to closure after QA team retests application .....**Error! Bookmark not defined.**
    - 3.1.12. Documentation .....11
    - 3.1.13. Deployment and Training .....12
    - 3.1.14. Maintenance .....13
- 4. List of Abbreviations .....14**

	<b>VAIL-Plant Software Development Process</b>	
<b>Prepared by :</b> Badar Ibrahim Khan	<b>Reviewed By:</b> Shahid Javed Kalair	<b>Date :</b> 03/01/2008
<b>Approved by :</b> Ijaz Ul Karim Rao	<b>Revision:</b> 0	<b>Page:</b> 5-of-15

## 1. Overview

The software development process is a prescribed ordering of activities governed by guidelines and structured by templates and tools that produce a product in a consistently repeatable manner.

This software development process describes the activities that constitute the full lifecycle of software development iteration. The end result of this process is a product that is deployed to customer (Internal or external).

The entire software development lifecycle described by this software development process is executed for each release of the product. Each new release accretes function and refinement as this software development process is executed again and again for each iteration. Each iteration builds on the artifacts of the previous iteration.

## 2. Purpose

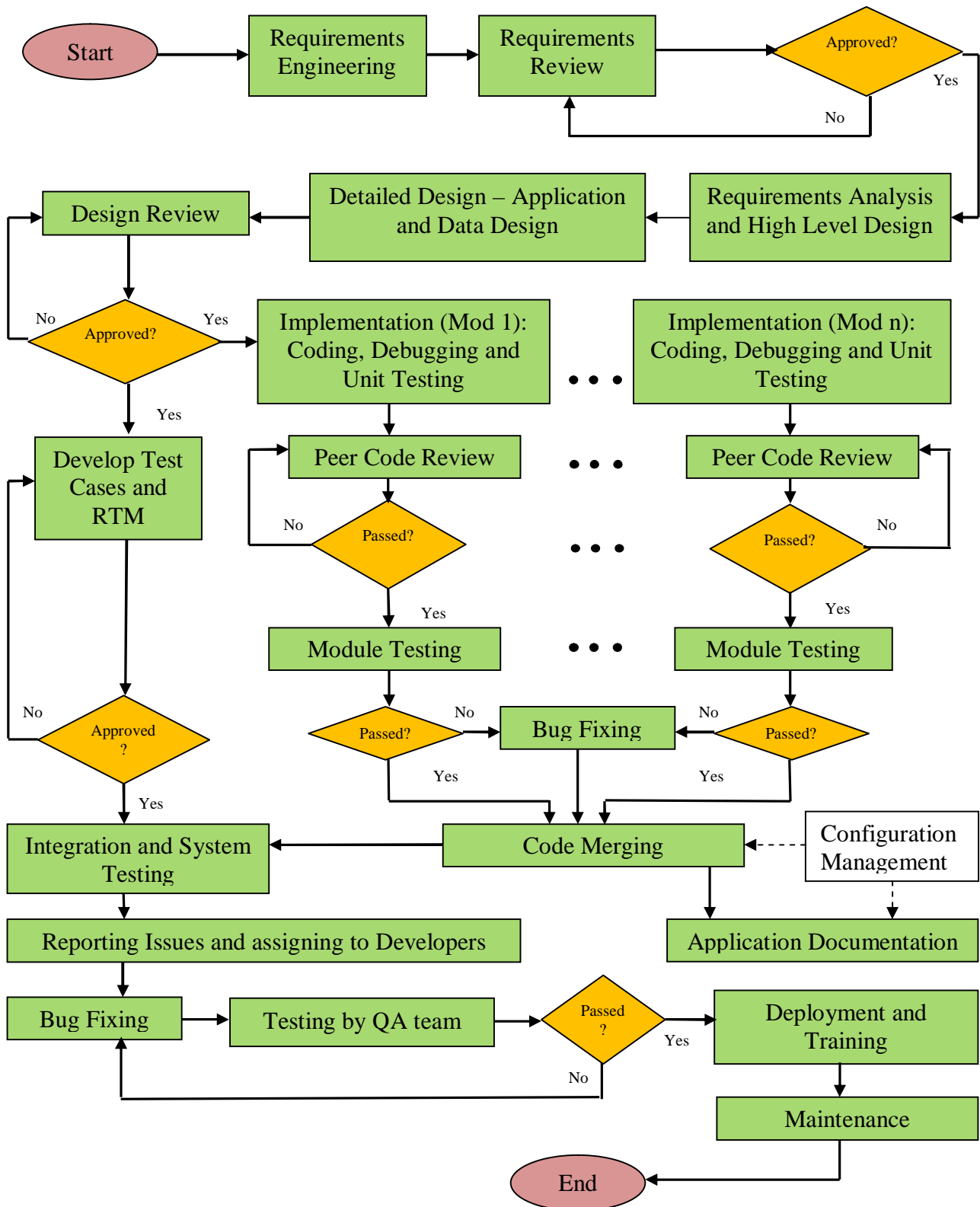
This document elaborates the process followed at Velosi Asset Integrity Limited for the development of software. It describes the actions carried out at each step and the actors involved in the process. It also describes the documents generated at each step of the process.


## 3. Decomposing the Software Development Process

The software development process typically requires the involvement of several different experts who are able to address specific development requirements more precisely. Depending on the size of the application and the actors involved in the development process, building an application may be an intricate undertaking, exposed to a variety of risks that might compromise the success of the final application. In order to control the software development process, it is thus of fundamental importance to understand its constituent activities, its actors, and their interconnections.

### **3.1. Activities in Software Development**

A Combination of Waterfall Model and Incremental model are followed for the software development process. The Flow chart below describes the activities in the Software development Process.



	<b>VAIL-Plant Software Development Process</b>	
<b>Prepared by :</b> Badar Ibrahim Khan	<b>Reviewed By:</b> Shahid Javed Kalair	<b>Date :</b> 03/01/2008
<b>Approved by :</b> Ijaz Ul Karim Rao	<b>Revision:</b> 0	<b>Page:</b> 8-of-15

### 3.1.1. Requirements Engineering

In this phase, the requirements are collected by business analysts and documented as a Software Requirements Specification (SRS) document. UML use case diagrams and activity diagrams can be used as standard representations of usage scenarios.

Requirements engineering aims at understanding a product's needed capabilities and attributes. The analysis concentrates on functional requirements, referring to the functions that the system must be able to support, as well as on nonfunctional requirements, referring mainly to the quality of the offered solution. This implies identifying the general idea behind the system, as well as the stakeholders that require the new solution, the motivations for the production of a new system and the final usage environment. The collected requirements are elaborated with the aim of producing some high-level models of the system that abstract from irrelevant details of the problem domain.


### 3.1.2. Requirements Review

The SRS document is reviewed and any comments are worked upon and submitted for review again. Once approved, the document is used as a blue print for the rest of the project. Approval is sought from the Project Sponsor or the person/client issuing the requirements.

### 3.1.3. Requirements Analysis and High level Design

After a subset of the application's requirements has been understood, the design follows. The design activity aims at specifying a solution, which must meet functional and efficiency requirements, as well as possible constraints derived from the target environment. Requirements previously collected are therefore refined, restricted, and enhanced to satisfy possible technological constraints. At this juncture, system architect prepares the high level design encompassing the architecture of the system.



	<b>VAIL-Plant Software Development Process</b>	
<b>Prepared by :</b> Badar Ibrahim Khan	<b>Reviewed By:</b> Shahid Javed Kalair	<b>Date :</b> 03/01/2008
<b>Approved by :</b> Ijaz Ul Karim Rao	<b>Revision:</b> 0	<b>Page:</b> 9-of-15

#### **3.1.4. Detailed Design – Application and Data Design**

The system designers work at analyzing the requirements in a detailed manner creating use case diagrams to model each requirement. Class diagrams are created and detailed design of the code to be developed is done in this phase. The Data architect simultaneously designs the entity relationship diagram specifying the data elements and their structure. At this phase, clear modules are demarcated for modularizing development. All the above is documented by the system and data designers in the Software Design Document (SDD).


#### **3.1.5. Design Review**

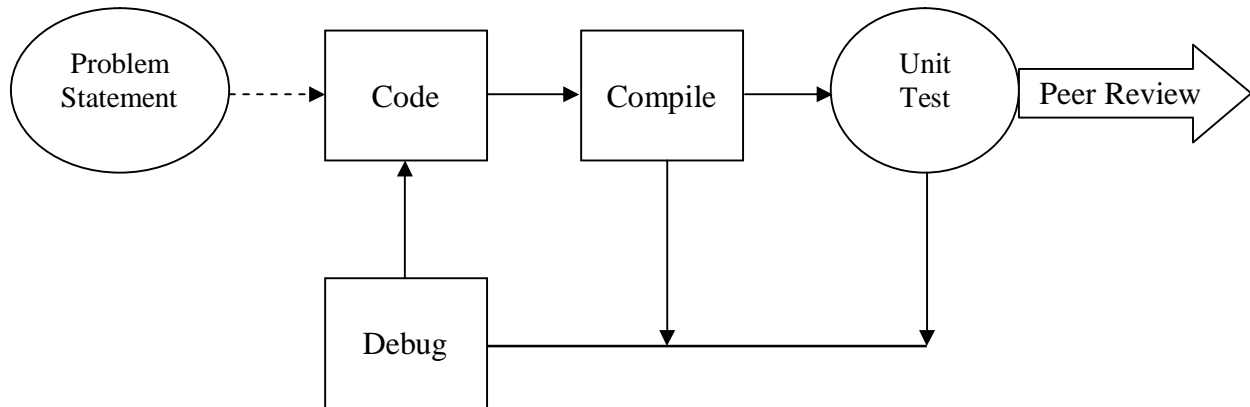
The SDD is reviewed and any comments are worked upon and submitted for review again, until it is approved. The project manager simultaneously assigns modules to individual developers or teams.

#### **3.1.6. Implementation - Coding, Debugging and Unit Testing:**

During implementation, the different design views are transformed into corresponding program code (structured into modules and/or files), database tables, and configuration files. Implementation may require the use of existing code libraries, a variety of different programming languages and communication protocols.

Developers work on the modules assigned to them using the simple iterative process of coding, compiling and debugging any problems. The developer finally tests the solution as part of unit testing. This is shown in the below diagram.

	<b>VAIL-Plant Software Development Process</b>	
	<b>Prepared by :</b> Badar Ibrahim Khan	<b>Reviewed By:</b> Shahid Javed Kalair
<b>Approved by :</b> Ijaz Ul Karim Rao	<b>Revision:</b> 0	<b>Page:</b> 10-of-15




### 3.1.7. Peer Code Review

After unit testing individual modules, the developers issue the code for peer review. In this process a fellow developer or the Development Team Lead reviews the code to confirm that it has been written according to the decided coding standards documented in the document numbered “VAIL/SCS/2008/012”. In case the reviewer issues comments, the developer reworks on the code before resubmitting for peer review.

### 3.1.8. Module Testing

Once the code has passed the Peer Review phase, the developer commits the code into the project branch of the software configuration management tool and informs the Development Team lead about it. The Developer informs the QA Team lead that the module is ready for testing. The QA Team Lead then assigns the module to a team member for testing. Any bugs are logged and communicated to the Development Team Lead who further assigns them to be fixed to the developer/s responsible for the module. Once fixed, the fixed code is once again committed to the Project branch of the software configuration management tool for merging.

	<b>VAIL-Plant Software Development Process</b>	
<b>Prepared by :</b> Badar Ibrahim Khan	<b>Reviewed By:</b> Shahid Javed Kalair	<b>Date :</b> 03/01/2008
<b>Approved by :</b> Ijaz Ul Karim Rao	<b>Revision:</b> 0	<b>Page:</b> 11-of-15

### 3.1.9. Code Merging


This is the responsibility of the Development Team Lead. The Lead merges the code committed into the branches of the project to the main trunk. Then this code is issued to the Quality Assurance Team for performing verification and validation. Configuration Management is the responsibility of the Development Team lead. The Configuration Management Process is detailed in the document titled “VAIL/SCM/2008/003”. In the case of the same file being committed by different developers, the resolution of the conflict through discussion is done by the Development Team Lead. After the code is merged, the application is provided to the Technical Writer to begin developing the documentation such as user manuals, training manuals, installation manual etc as well as to the QA team to begin testing it for quality.

#### 3.1.1. Documentation

Once the application is ready, the technical writer begins work on creating user manuals, installation guides, administrator guides, training material and training manuals. These documents undergo configuration management process to ensure their proper versioning and updates. These documents must undergo the review and approval process and must be modified when any change is implemented in the software.

#### 3.1.2. Integration and System Testing

The Quality Assurance Team members are responsible for verifying that the application functions properly and validate that it meets the requirements. Once the detailed design of the system is finalized, the QA team members design the test cases for each use case scenario. They also design the Requirement Traceability Matrix (RTM) to cross reference and trace every requirement to one or more design elements and test cases. The test cases undergo review and approval process. Once the code is merged by the Development Team Lead, he passes this code

	<b>VAIL-Plant Software Development Process</b>	
<b>Prepared by :</b> Badar Ibrahim Khan	<b>Reviewed By:</b> Shahid Javed Kalair	<b>Date :</b> 03/01/2008
<b>Approved by :</b> Ijaz Ul Karim Rao	<b>Revision:</b> 0	<b>Page:</b> 12-of-15

to the QA team. They use these test cases to test the system and log any issues that they come across.

The most relevant quality concerns addressed by this activity are related to functionality (i.e., the correctness of the application behavior with respect to specified functional requirements), performance (i.e., the throughput and response times of the application in average and peak workload conditions), and usability (i.e., ease of use, communication effectiveness, and adherence to consolidated usage standards). Integration Testing ensures that the modules are working properly after integration and System Testing validates that they confirm to their requirements.

### **3.1.3. Reporting issues, assigning to developers and Bug Fixing**

The QA team leader reports issues to the Development Team Lead who further assigns these issues to individual developers. Developers fix these issues, test at their end and pass the fixes back to the Development Team Lead who merges them back into the main application.

### **3.1.4. Testing by QA team**

Once all fixes have been merged back into the application, the QA again tests the application and closes the issues. In this process, if any issue has not been solved satisfactorily or if other issues have sprung up due to the fix, the QA reports the same to the Development Team Lead and he has them resolved through a developer. Once all issues are fixed, the application is ready for release.

### **3.1.5. Deployment and Training**

In this phase, the application is ready to be rolled out on user machines. It is deployed and end users are trained for its use.

### 3.1.6. Maintenance

Maintaining a deployed and running application involves keeping the application in a healthy state, so as to guarantee high availability and to reduce failures. This includes periodical checks of log files, bug reporting, and the cleaning up of temporary files, as well as the application of bug fixes or security patches, in order to keep the application always up to date. Maintenance also includes implementing changes to the software and coming up with newer versions of the application. This process is detailed in the document titled “VAIL/CMP/2009/002”.

<b>Prepared by :</b> Badar Ibrahim Khan	<b>Reviewed By:</b> Shahid Javed Kalair	<b>Date :</b> 03/01/2008
<b>Approved by :</b> Ijaz Ul Karim Rao	<b>Revision:</b> 0	<b>Page:</b> 14-of-15

#### 4. List of Abbreviations

Abbreviation	Explanation
UML	Unified Modeling Language
QA	Quality Assurance
SRS	Software Requirement Specification
SDD	Software Design Document
RTM	Requirements Traceability Matrix

**Prepared by :** Badar Ibrahim Khan**Reviewed By:** Shahid Javed Kalair**Date :** 03/01/2008**Approved by :** Ijaz Ul Karim Rao**Revision:** 0**Page:** 15-of-15