|  |
| --- |
| GENUS INNOVATION LIMITED |
| Design Implementation and Module Integration Procedure |
|  |
|  |
| **Genus** |
|  |

|  |
| --- |
| This procedure covers the Implementation, Verification and Integration of the modules to be delivered for the Validation. |

Contents

[Overview 2](#_Toc447274717)

[Objective 2](#_Toc447274718)

[Scope 2](#_Toc447274719)

[Inputs 2](#_Toc447274720)

[Entry Criteria/Triggers 2](#_Toc447274721)

[Tasks 2](#_Toc447274722)

[Verification 5](#_Toc447274723)

[Guidelines 6](#_Toc447274724)

[Standards 6](#_Toc447274725)

[Testing 6](#_Toc447274726)

[Module Testing 6](#_Toc447274727)

[Integration Testing 7](#_Toc447274728)

[Applicable Measurements 7](#_Toc447274729)

[Exit Criteria/Outputs 7](#_Toc447274730)

# Overview

This procedure covers the Implementation, Verification and Integration of the modules to be delivered for the Validation.

# Objective

* To ensure successful integration of the modules/work products.

# Scope

This procedure applies to Development Projects.

# Inputs

* Design documents.
* Updated Requirement Traceability Table.
* Integration sequence, Integration Procedure and Integration Environment.
* Integration test cases.

# Entry Criteria/Triggers

* Approved and Baselined Design Data.
* The practitioners have undergone QMS trainings with focus on performing their processes.

# Tasks

| Sr.No | Task | Owner/Role |
| --- | --- | --- |
|  | **Design Implementation** |  |
|  | Study Design documents. | Design Team/s |
|  | Ensure availability of required environment for development. | Design Team/s |
|  | Develop code as per the Firmware Design Document, Develop Hardware/PCBs Layouts as per the Hardware Design Document and Schematics and Develop Mechanical Drawings (2D/3D) as per the Mechanical Design Document. The guidelines for firmware development are “Firmware Development Guidelines” (GDLN\_FRWSTD). | Design Team/s |
|  | **Review and Verification** |  |
|  | Perform self review of implemented modules. | Design Team/s |
|  | Identify Peer review team. | Project Manager |
|  | Conduct Peer Review and log defects in “Review and Testing Defect Log”.  Use appropriate checklists such as “Code Review Checklist” (CHKL\_CODRVW), “Hardware Review Checklist” (CHKL\_HDWRVW), and “Mechanical Review Checklist” (CHKL\_MCHRVW). | Peer Review Team |
|  | Close identified defects. | Design Team/s |
|  | Re-review and Baseline the Design Documents and Implemented Modules. | Design Team/s |
|  | Get the PCBs fabricated and assembled. Get the prototype samples of Mechanical modules. Procure off the shelf electrical parts. | Design Team/s |
|  | Update “Requirement Traceability Table” (TMPL\_REQTRT) for traceability between requirements and Implementation.  This might generally be done in the form of a Team Meeting, involving all relevant technical team members. | Design Team/s |
|  | **Develop Module Test Cases** |  |
|  | Develop module wise Test Cases using templates “Firmware Test Cases” (TMPL\_FWTCAS), “Hardware Test Cases” (TMPL\_HWTCAS), “Mechanical Test Cases” (TMPL\_MCTCAS).   * Define the test conditions. * Define the intended environment and the tools required. | Design Team/s |
|  | Conduct Peer Review and log the defects in “Peer Review and Defects Log”. | Peer Review Team |
|  | Close identified defects. | Design Team/s |
|  | **Conduct Module Tests** |  |
|  | Perform module testing as per the developed Module Test Cases.  Refer “[Testing Guidelines](#_Testing)”. | Testing Team |
|  | Verify Interfaces (output files, output file formats, Protocols, Connectors, Mounting holes, Studs, Slots). This is done to verify that the modules to be integrated are fit for integration. | Testing Team |
|  | Log the defects in “Review and Testing Defect Log” , if required | Testing Team |
|  | Close defects. The closure of all unit testing defects is an implicit go-ahead for Integration. | Design Team/s |
|  | Analyze the defects. Identify and document the root causes that are supposed to be causing the defects. The analysis can be performed in a meeting with the technical staff. | Functional Head/s |
|  | **Module Level Integration** |  |
|  | Integrate modules with reference to the integration sequence and integration procedures. Refer “Inerface Control and Integration Design Document” (TMPL\_INTDSN). | Design Team/s |
|  | Verify integration at each integration step, Document defects in “Review and Testing Defect Log”. | Peer Review Team |
|  | Close defects. | Design Team/s |
|  | Approve and baseline integrated product. | Functional Head/s |
|  | Make available integrated product for Integration Testing. | Design Team/s |
|  | **Conduct Integration Testing** |  |
|  | Perform integration testing as per the Integration Test Cases.  Refer “[Testing Guidelines](#_Testing)”. | Testing Team |
|  | Document defects in “Review and Testing Defect Log”. | Testing Team |
|  | Close defects. | Design Team/s |
|  | Make available integrated product for System Level Testing and intimate Validation Manager. | Project Manager |
|  | Analyze the defects. Identify and document the root causes that are supposed to be causing the defects. The analysis can be performed in a meeting with the technical staff. | Functional Head/s |
|  | **Product Support Documentation** |  |
|  | Update Product support documentation and User Manual. | Design Team/s |

|  |  |  |
| --- | --- | --- |
|  | Library Creation |  |
|  | Identify and extract the functions/Components that can potentially be library functions/components. Modify the functions/components so as to make them fit for inclusion in the respective libraries. | Design Teamt/ Project Manager |
|  | Document the extracted functions/components in the “Library Function/Component Specifications” of Master Library Documents | Design Team/ Project Manager |
|  | Make the extracted functions/components a part of the identified library using “Firmware Master Library” Function/Component Specifications (MSTL\_FRWLIB), “Hardware Master Library” Function/Component Specifications (MSTL\_HDWLIB) and “Mechanical Master Library” Function/Component Specifications (MSTL\_MCHLIB) and get them approved from the respective Functional Head/s. | Design Team/ Project Manager |
|  | Baseline the approved library and “Library Function Specifications” Documents. | Project Manager |

\* Improvements/Suggestions are solicited on “Process Improvement Proposals Database”.  
\*For details on the Roles and Responsibilities of the practitioners, Refer "Roles and Responsibility" document in the QMS.

# Verification

* Peer review of Source code, PCB layouts, Mechanical 2D/3D Drawings by Peer Review Team
* Interface verification by Testing Team
* Peer review of Modules Test Cases by Peer Review Team
* Peer review of Integration Test Cases by Peer Review Team
* Validation and Approval of the Library
* Review of the process and its work products by PPQA members.
* Review of the process and its work products by Senior Management.

# Guidelines

Refer "Configuration Management Procedure" (PRCD\_CONFIG) for Access Rights, location of work products, naming convention and types of controls.

## Standards

Refer “Firmware Development Guidelines” (GDLN\_FRWSTD) for Firmware development.

## Testing

Testing is the process of executing the functionality in a controlled manner, in order to answer the question “Does the product behave as specified?”

Levels of testing include:

* Module Testing
* Integration Testing
* System Level Testing

Following list provides some rules to follow as an aid to effective testing:

* Always test against a specification. If tests are not developed from a specification, then it is not testing. Hence Testing is totally reliant upon adequate specifications.
* Document the testing process: specify tests and record test results.
* Test hierarchically against each level of specification. Finding more errors earlier will ultimately reduce cost.
* Always test positively: that the product does what it should, but also negatively: that it doesn’t do what it shouldn’t.

### Module Testing

A module is the smallest testable part of a product. Module testing is performed to ensure that individual units of implementation are fit for use and detailed design for the unit has been correctly implemented. Module testing is intended to detect many bugs at a stage of product development where they can be corrected economically. Given modules which have been tested, the integration process is greatly simplified. Developers will be able to concentrate upon the interactions between modules and the overall functionality without being flooded by lots of little bugs within the units.

### Integration Testing

In integration testing, progressively larger groups of tested modules are integrated and tested until the product works as a whole. Integration Testing is to identify defects involving the interaction of integrated components. Before performing Integration Testing, it is important that all the components have been successfully unit tested. Some of the Integration Testing approaches are:

* Top-Down Approach

Low-level modules first tested and gradually worked up toward the high-level modules. A small number of components are combined into a simple system and tested. After ensuring that they work together, more components are added and tested with the already verified subsystem.

* Bottom-Up approach

Major components are integrated gradually, and then system is tested with the lower level components of the system. Once it is verified that the existing components work together within the existing framework, additional components can be added and tested.

The bottom-up approach consists of tests that gradually increase in scope, while the top-down approach consists of tests that are gradually refined as new components are added.

# Applicable Measurements

* Number of Defects identified in Review
* Number of Defects identified in Module Testing
* Number of Defects identified in Integration Testing

# Exit Criteria/Outputs

* Verified and Baselined implemented design
* Integrated Product
* Approved and Baselined Library and Library Function Specifications