|  | Information Technology Institute ITI - Intake 38 |
| --- | --- |

Kenovo Blender

Embedded Systems - Software testing and QA Joint Project

Configuration Management (CM) Plan

Version 1.0

04/02/2018

**Document ID: CFG\_MGT\_PLN**

Table of Contents

[1. Introduction 5](#_Toc396111342)

[2. Overview 5](#_Toc396111343)

[2.1 Project Description 5](#_Toc396111344)

[3. Configuration Management Approach 5](#_Toc396111350)

[3.1 Configuration Management Policy 5](#_Toc396111359)

[3.3.1 Naming Convention 5](#_Toc396111360)

[3.3.2 Identifying CIs 8](#_Toc396111360)

[3.2 Establishing Configuration management system 9](#_Toc396111352)

[3.2.1 Selecting configruation management tools 10](#_Toc396111360)

[3.2.2 Folder structure 10](#_Toc396111360)

[3.2.3 Branching Strategy 10](#_Toc396111360)

[3.2.4 Change log 11](#_Toc396111360)

[3.3 Roles & Responsibilities 12](#_Toc396111352)

[4. Configuration Management Administration 11](#_Toc396111354)

[4.1 Baselines 11](#_Toc396111356)

[4.2 Tracking and contolling changes 13](#_Toc396111356)

[Appendix A: Acronyms 14](#_Toc396111366)

[Appendix B: Glossary 15](#_Toc396111367)

[Appendix C: Approvals 16](#_Toc396111369)

[Appendix D: Revision History 17](#_Toc396111371)

List of Tables

[Table 1: Configuration Management Processes 3](#_Toc396111373)

[Table 2: Roles and Responsibilities 3](#_Toc396111374)

[Table 3: Record of Changes 6](#_Toc396111375)

[Table 4: Acronyms 7](#_Toc396111376)

[Table 5: Glossary 8](#_Toc396111377)

[Table 6: Referenced Documents 10](#_Toc396111378)

[Table 7: XLC Template Revision History 13](#_Toc396111379)

# Introduction

This Configuration Management (CM) Plan establishes the technical and administrative direction and surveillance for the management of configuration items (i.e., software, hardware, and documentation) associated with the Kenovo Blender Project that are to be placed under configuration control. This document defines the project’s structure and methods for:

* Identifying, defining, and baselining configuration items (CIs);
* Controlling modifications and releases of CIs;
* Reporting and recording status of CIs and any requested modifications;
* Ensuring completeness, consistency, and correctness of CIs; and
* Controlling storage, handling, and delivery of CIs.

As the project matures, appropriate sections of this plan will require periodic updating.

# Overview

## A direct purpose of this project is to provide a proof of concept for Kenovo™ international as a bidding to acquire a new model of Blenders to satisfy its vast majority of customers worldwide.

# Configuration Management Approach

## Configuration Management Policy

**3.3.1 Naming Convention**

A naming standard will be established for every Configuration item, Baseline, Change request and Approved changes.

Configuration item conventions shall be as follows :

**Document Naming:**

An abbreviation shall be introduced as an identifier and as a prefix to each document name represents key letters in the original wording written in uppercase letter and separated by a space bar.

Example:   
 A project plan will be document as "Project Management Plan" .

**Use case naming:**

A use case shall be named as "**USE\_CASE\_n**" where n is the order of the use case among other use cases.

Example :   
 **USE\_CASE\_1** will represent the first use case.

**Requirement naming:**

A customer requirement shall be named as "**REQ\_n**" where n is the order of the requirement among other customer requirements.

Example :   
 **REQ\_1** will represent the first cusomter requirement.

**Requirements spefications naming:**

An SRS requirement shall be named as "SRS\_n" where n is the order of the specifications among other SRS requirements.

Example:

"SRS\_1" will represent the first SRS requirement "Specifications"

**Specifications naming:**

A design requirement shall be named as "**SPEC\_n**" where n is the order of the specifications among other design specifications.

Example:

"**SPEC\_1**" will represent the first design requirement "Specifications"

**Module naming:**

A design module "Code file" shall be named in a descriptive identifier expressing the functionality or major functionalities of a code file or a module.

Example:   
 "**Dio**" will represent the digital input output driver of a microcontroller.   
 "**Uart\_Config**" will represent a configuration file of a uart peripheral of a microcontroller.

**Testcase naming:**

A Testcase naming will depend on the Testlevel and/or module which belongs to.

It will be expressed in upper case letters followed by Testlevel or module Identifier and a unique numbering.

Example:  
 A high level System testcase shall be named : "**TestCase\_SYS\_n**" , where n represents the numbering.

An integration level testcase shall be named "**TestCase\_INT\_Mod1\_Mod2\_n**", where n represents the numbering.

A unit level testcase shall be named after the module it belongs to "**TestCase\_Dio\_n**", where n represents the numbering.

**Test report naming:**

A Test report naming will depend on the Testlevel and/or module which belongs to.

It will be expressed in upper case letters followed by Testlevel or module.

Example:  
 A high level System test report shall be named : "**Test\_report\_SYS**"

An integration level testreport shall be named "**Test\_report\_INT\_Mod1\_Mod2\_n**".

A unit level test report shall be named after the module it belongs to "**Test\_report\_Dio**.

**Defect naming:**

A defect naming will depend on the artifacts and/or module which belongs to.

It will be expressed in lower case letters followed by document and/or module Identifier and a unique numbering.

Example:  
 A defect found in the SRS shall be named : "**Defect\_SRS\_n**" , where n represents the numbering.

**Software Bug naming:**

A software bug naming will depend on the testcase artifacts and/or module which it was discovered .

It will be expressed in lower case letters followed by module Identifier and a unique numbering.

Example:  
 A bug found in Module\_1 shall be named : "**Defect\_Module\_1\_n**" , where n represents the numbering.

**Other managerial or administrative naming:**

* A change request will be named "**CHG\_REQ\_n**", where n represents the numbering of the change request.
* An approved change request shall be named "**CHG\_n**" where n represents the numbering of the change.
* A document review will be named "**REV\_n**", where n represents the numbering of the review.
* An issue will be named "**ISSUE\_n**", where n represents the numbering of the issue.
* A project Risk shall be named "RIS\_n" where n represents the numbering.
* Baselines convention will be clarified in the Baselines section.
* Configuration items will be noted in the configuration items identification section**.**

**3.3.2 Identifying key configuration items "CIs"**

In this section configuration items, components, and related work products to be placed under configuration management shall be clearly identified and categorized mainly depending on the project phase that the document shall be introduced or based on ownership or on the inter-product dependencies which will be clearly identified and monitored as a core functionality of a typical configuration management system.  
  
 Details and list of all configuration items are clearly identified in the CONFIG\_ITEMS document.

## Configuration Management Policy

**3.2.1 Selecting Configuration management tools**

In this section configuration management tools are identified.

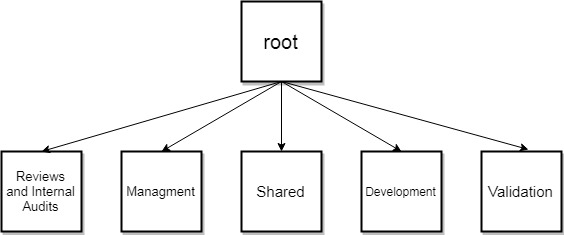
Version control system:

"SmartGit" was selected as a common platform used to store local repositories for each individual with the option to be synchronized with the common remote project repositories.  
   
 "SmartGit" offers all Git's functionalities with simple, interactive GUI with many other features that ensure ease of use and doesn't require any extensive training or special arrangements to be introduced into a project and can be easily integrated with Git online host server.   
  
 Remote Online host server:

"Git" was selected to host the remote common repository and all project files, as it offers all the required version control functionalities and also offers a variety of hosting servers with sufficient storage space and access level to ensure integrity and security of cloudily stored data and project files .

**3.2.2 Project folder structure:**

Based on extensive planning and project expectations a detailed project folder structure was organized as follows :



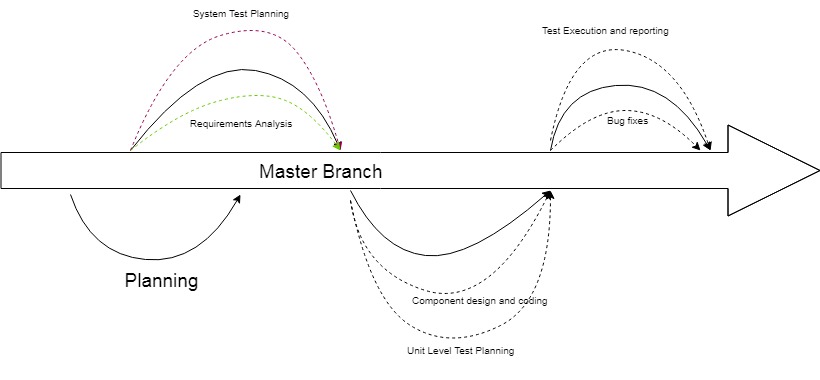
**Folders content**

Table 1: Folder structure

|  |  |
| --- | --- |
| **Should contain** | **Directory** |
| CDD, Code , Any design artifacts,  Unit level test cases and reports | Development |
| Master Test plan ,System level test plan , Test Suites ,Test cases , Test reports | Validation |
| Internal Audits Checklists , Review log | Internal Audit Checklists |
| Project plan, Configuration management Plan, RTM , Issue Log , Schedule ….. | Management |
| SRS , CRS, Manuals , Datasheets,……  Bug report | Shared |

**This section should be updated when creating any new directories or subdirectories to during the project and should request a change that should be logged.**

**3.2.2 Project Branching strategy:**



The Branching strategy will be centered around a "Master Branch" representing the main project flow.

The project shall be divided into four different phases with each has its own branch(s):

* Planning will be having a single branch that will be merged with the center at the end of the phase .
* The second phase will be having two separate branches a "Requirements Analysis" branch for the development team while the testing team will be having "System Test Planning" branch to consolidate its efforts, and shall be merged with the master by the end of this phase.
* The Third phase will be having two branches for "component's design coding" and "Unit Test planning" for development and testing team correspondingly.
* The final phase will be represented by another two separate branches for Tester's Validation testing and developers Bug fixing.

3.2.2

Change Log:

A document that lists all the change requests and their approval, status, progress and affected documentations.   
  
Details and list of all change logs are clearly identified in the "Change log" document.

## Roles & Responsibilities

In this section key personnel responsible for configuration management. their responsibilities for activities such as configuration identification, configuration control, configuration auditing/reporting, etc for software, hardware, and documentation.

Table 2: Roles and Responsibilities Validation team

| Name | Shared | Development | Validation | Planning | Reviews and Audits |
| --- | --- | --- | --- | --- | --- |
| Soha Ahmed | Read/Write | Read | Read/Write | Read/Write | Read/Write |
| Marwan Swailem | Read/Write | Read | Read/Write | Read/Write | Read/Write |
| Esraa Ahmed | Read/Write | Read | Read/Write | Read/Write | Read/Write |
| Toqa Ibrahim | Read/Write | Read | Read/Write | Read/Write | Read/Write |
| Basma Karama | Read/Write | Read | Read/Write | Read/Write | Read/Write |
| Salma Mohammed | Read/Write | Read | Read/Write | Read/Write | Read/Write |
| Abdel Rahman Moussa | Read | Read/Write | Read/Write | Read/Write | Read/Write |
| Mohamed El zeiny | Read | Read/Write | Read/Write | Read/Write | Read/Write |
| Christine Ramsis | Read | Read/Write | Read/Write | Read/Write | Read/Write |
| Khaled Hossam | Read | Read/Write | Read/Write | Read/Write | Read/Write |
| Ahmed Raafat | Read | Read/Write | Read/Write | Read/Write | Read/Write |

Table 3: Roles and Responsibilities Development team

# Configuration Management Administration

## Baselines Baseline are constructed by the assignment of an identifier to a configuration item or a collection of configuration items and associated entities at a specific stage.

In this project Baselines were nominated at 5 stages representing distinctive project project milestones and a set of deliverables.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Baseline No.** | **Description** | **Project Stage** | **Version** | **Related documents** |
| Baseline\_0 | The aim of this baseline is to reserve the initial stage of the project which include planning requirements analysis and test planning | Planning | xx.0 | Customer Requirement Specifications Software Requirement Specification Project Management Plan Project schedule RASIC Risk Matrix Requirements Traceability Matrix Configuration Management Plan Master Test Plan Software Interactive Questionnaire  System level Test Plan |
| Baseline\_1 | The aim of this baseline is to establish a solid Architectural design and test cases | System Design | xx.1 | Customer Requirement Specifications Software Requirement Specification Project Management Plan Project schedule RASIC Risk Matrix Requirements Traceability Matrix Configuration Management Plan Master Test Plan Software Interactive Questionnaire  System level Test Plan System level Test Cases System Architectural Design |
| Baseline\_2 | The aim of this baseline is to establish a reference point to start coding system components. | Component Design | xx.2 | Customer Requirement Specifications Software Requirement Specification Project Management Plan Project schedule RASIC Risk Matrix Requirements Traceability Matrix Configuration Management Plan Master Test Plan Software Interactive Questionnaire  System level Test Plan System Architectural Design Components Design Unit Level Test Plan Unit Level Testcases |
| Baseline\_3 | In this stage the source code will be implemented and unit test cases are run to verify source code quality a | Implementation | xx.3 | Customer Requirement Specifications Software Requirement Specification Project Management Plan Project schedule RASIC Risk Matrix Requirements Traceability Matrix Configuration Management Plan Master Test Plan Software Interactive Questionnaire  System level Test Plan System Architectural Design Components Design Unit Level Test Plan Unit Level Testcases Unit Level Test Report |
| Baseline\_4 | In this stage whole system will be exercised to validate the system functionality with reference to customer needs | Validation | xx.4 | Customer Requirement Specifications Software Requirement Specification Project Management Plan Project schedule RASIC Risk Matrix Requirements Traceability Matrix Configuration Management Plan Master Test Plan Software Interactive Questionnaire  System level Test Plan System Architectural Design Components Design Unit Level Test Plan Unit Level Testcases Unit Level Test Report System Level Test Report |

Table 4: Baselines

## Monitoring and controlling changes

A well established change policy will ensure full documents and data integrity across all different project cycles and phases.

To achieve this, an agreement between team members was reached to institutionalize a predefined steps for change requests and change management and where clarified as follows:

* Authorized access to the configuration management system is granted to ensure access control to all project related documents.
* Store and retrieve configuration items in a configuration management system which is already predefined in the "Configuration management tools" section.
* Share and transfer configuration items between control levels in the configuration management system.
* A change in document will require invoking a "change request" and all details including reasons to change have to be clarified.
* A change request has to be approved by stallholders "Document Authors, Reviewers"
* When approved a change request has to be logged into the "Change Log".
* All details has to provided into the change log and documents self revision history.
* The document name should be changed to include a new version update.

1. Acronyms

Table 5: Acronyms

| Acronym | Literal Translation |
| --- | --- |
| **CI** | Configuration Item |
| **CM** | Configuration Management |
| **PMP** | Project Management Plan |

1. Glossary

Table 6: Glossary

| Term | Definition |
| --- | --- |
| **Baseline** | (1) A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures. (2) A document or a set of such documents formally designated and fixed at a specific time during the life cycle of a configuration item. (3) Any agreement or result designated and fixed at a given time, from which changes require justification and approval.  (IEEE Std. 610-12-1990) A baseline is a configuration identification formally designated and applicable at a specific point in the life cycle of a configuration item. |
| **Build** | An operational version of a system or component that incorporates a specified subset of the capabilities that the final product will provide. (IEEE Std. 610-12-1990) |
| **Configuration** | The functional and physical characteristics of hardware or software as set forth in technical documentation or achieved in a product. (IEEE Std. 610-12-1990) |
| **Configuration Identification** | An element of CM, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation. (IEEE Std. 610-12-1990) |
| Configuration Item (CI) | An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration process. (IEEE Std. 610-12-1990) |
| Configuration Management (CM) | A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements. (IEEE Std. 610-12-1990) |

1. Approvals

The undersigned acknowledge that they have reviewed the Configuration Management Plan and agree with the information presented within this document. Changes to this Configuration Management Planwill be coordinated with, and approved by, the undersigned, or their designated representatives.

**Authors**

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: | Marwan Ahmed Thabet | Date: | 5th of April 2018 |

**Reviewers**

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: | Soha Swailem | Date: | 6th of April  2018 |

Appendix D: Revision History

Table 7: Configuration Management Plan Revision History

| Version  Number | Date | Author/Owner | Description of Change |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |