**Software UNIT implementation**

**and verification**

**Model: i-DOLPHIN**

**Document No. :**

This document valid from the date of approval

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Title | Name | Date | Signature |
| Prepared by |  |  |  |  |
| Reviewed by |  |  |  |  |
| Approved by |  |  |  |  |

|  |
| --- |
| **META BIOMED CO., LTD.** |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision | Description | Author |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1 Software development Environment Description 4](#_Toc442981557)

[1.1 Development Life Cycle 4](#_Toc442981558)

[**1.1.1** Development Life Cycle Procedure 1](#_Toc442981559)

[**1.1.2** Description of Development Life Cycle 2](#_Toc442981560)

[1.2 Software Maintenance Life Cycle 3](#_Toc442981561)

[**1.2.1** Software Maintenance Life Cycle Procedure 3](#_Toc442981562)

[**1.2.2** Description of Software Maintenance Life Cycle Procedure 4](#_Toc442981563)

[1.3 Software Configuration Management Life Cycle 4](#_Toc442981564)

[**1.3.1** Software Configuration Management Life Cycle Procedure 4](#_Toc442981565)

[**1.3.2** Description of Software Configuration Management 5](#_Toc442981566)

[2 Verification and Validation Documentation 6](#_Toc442981567)

[2.1 Abbreviations and Symbols 6](#_Toc442981568)

[2.2 Definitions 6](#_Toc442981569)

[2.3 Reference 7](#_Toc442981570)

[**2.3.1** Regulatory Standards & Guidances 7](#_Toc442981571)

[**2.3.2** Industry Standards and Guidances 7](#_Toc442981572)

[**2.3.3** Internal Standards & Guidances 7](#_Toc442981573)

[2.4 Roles and Responsibilities 8](#_Toc442981574)

[**2.4.1** Management Members 8](#_Toc442981575)

[**2.4.2** Hazard Analysis Team : Members, Roles and Responsibilities 8](#_Toc442981576)

[**2.4.3** V & V Testing Team : Members, Roles and Responsibilities 8](#_Toc442981577)

[**2.4.4** Documentation Review : Members, Roles and Responsibilities 8](#_Toc442981578)

[**2.4.5** Team Members/Qualifications 8](#_Toc442981579)

[2.5 V.V & T Item & Methods 9](#_Toc442981580)

[**2.5.1** Static Analysis 9](#_Toc442981581)

[2.6 Documentation Handling 9](#_Toc442981582)

# Software development Environment Description

## Development Life Cycle

To aid the development process, task lists showing specific deliverables, by phase, are used as a form of guidance. Atypical list is

shown in the table below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase**  **Se** →  ↓**Group** | **Definition** | **Design** | **Implementation** | **Qualification** | **Maintenance** |
| System  Architecture | *Requirement*  *Specifications* | *Architecture Specification*  *System Design Specification*  *Design Reviews* |  | Test Support | Support |
| Software |  | *SW Requirements and Spec.*  *Software Design and Spec.*  *Design Reviews* | *Completed Software*  *Build Environment*  *Code Reviews & Inspections*  *Test Requirements Specification*  *Updated Software Design Specs* | Test Support | Support |
| Design  Assurance | Preliminary Schedule  Independent V & V | Source Code Analysis  Problem Analysis  Hazard Analysis  Independent V & V | Additional test cases  Automated tests  Independent V & V | Execution of Tests  V & V Documentation  V & V Report  Internal Audit Report  Product & Test Certification | Support |
| Technical  Publications | Documentation  Support Plan | Documentation Design | Completed user and service  documentation | Documentation reviews | Support |

Items in italics indicate a document or an activity which produces a document.

**Group Responsibilities by Development Phase**

### Development Life Cycle Procedure

Software development process at ABC. Follows a classic waterfall methodology, as illustrated in the following chart

Analysis

Planning

Basic Design

Detailed Design

Implementation, Unit Test

Integration Test

Validation & Verification Test

Monitoring

**Definition**

**Design**

**Implementation**

**Qualification**

**Maintenance**

Quality Assessment and Planning

**Classic " Waterfall" Life Cycle Model**

**Analysis.** The activity consists of establishing requirements for as much as possible before the design phase. The requirements are documented in the software-specific requirements and design notes. When enough of the requirements are gathered and analyzed for this iteration of the prototype, the activities shift into the design phase.

**Planning.** This activity constructs the software development plan. This plan outlines the tasks, responsibilities, resources, and other items pertinent to the specific development project.

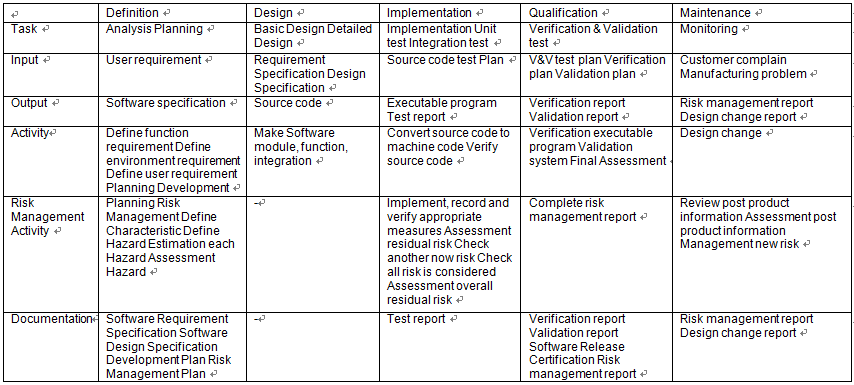
**Design.** This activity considers four functional attributes of the software: data structure, software architecture, functionality, and interface characterization. The design is documented in design documents.

**Implementation.** The implementation phase converts the design into a machine readable form. The software source code is created in accordance with SCUS's coding conventions. To verify that the code created is consistent with META BIOMED Co., Ltd.. standards, design reviews/inspections are held.

**Qualification.** The qualification phase verifies that the design and code implemented meet the requirements. Testing ensures that defined input will produce actual results that agree with required results.

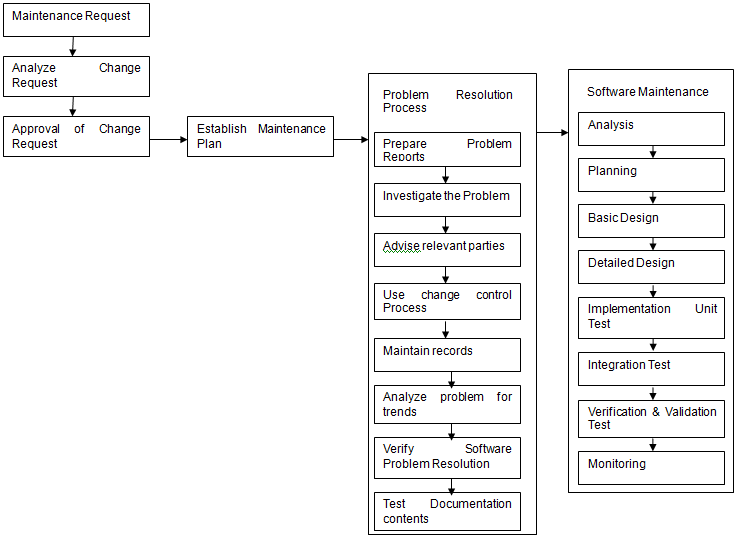
**Maintenance.** Once the software enters a released state, the only software changes allowed are to fix identified defects in the code or to support approved enhancements. All identified system software defects are recorded on System Discrepancy Reports(SDR) forms or submitted electronically to the Modification Request(Ultrasound) database.

### Description of Development Life Cycle

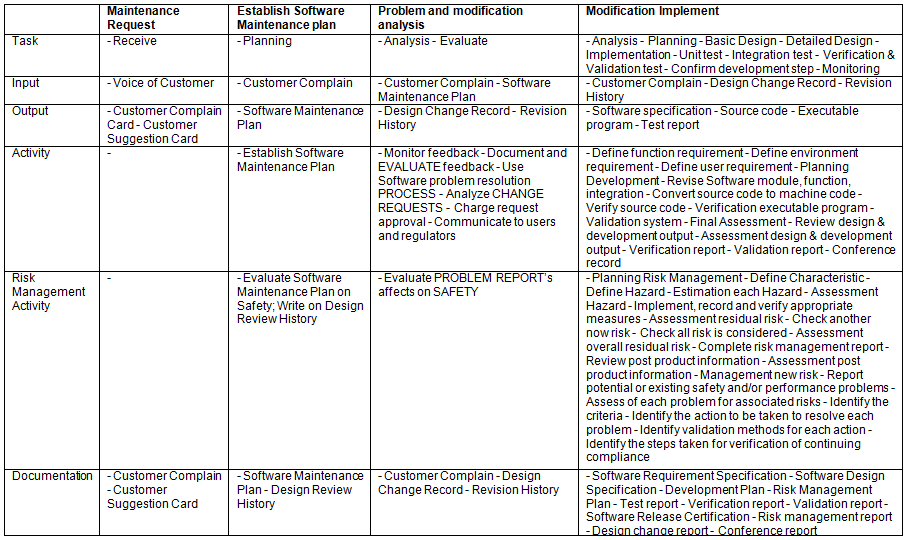


## Software Maintenance Life Cycle

### Software Maintenance Life Cycle Procedure

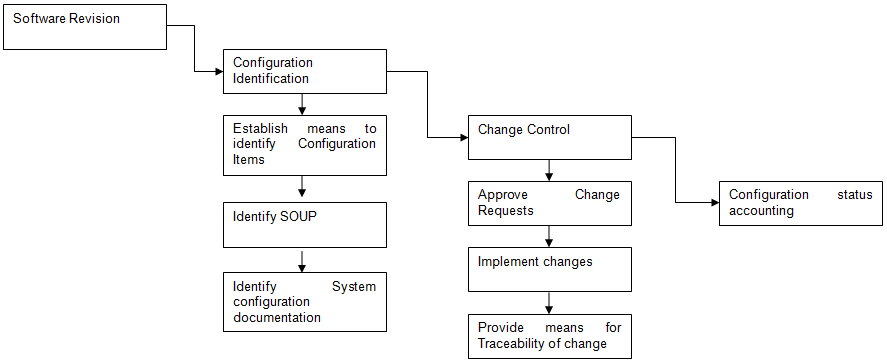


### Description of Software Maintenance Life Cycle Procedure

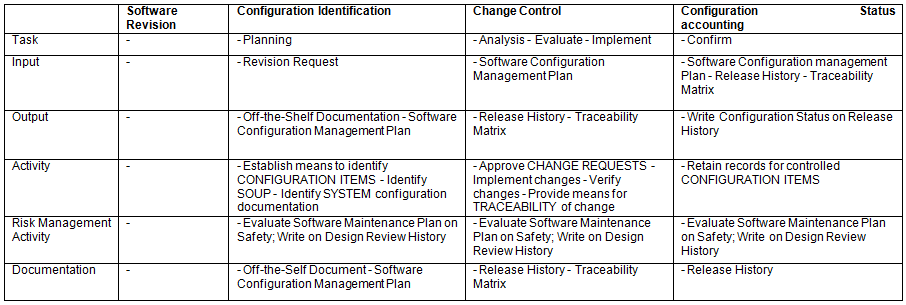


## Software Configuration Management Life Cycle

### Software Configuration Management Life Cycle Procedure



### Description of Software Configuration Management



# Verification and Validation Documentation

## Abbreviations and Symbols

ISO: International Organization for Standardization

SOP: Standard Operating Procedure,

QM: Abbreviation for Quality Manual Document

QR: Abbreviation for Quality Record

QWI: Abbreviation for Quality Work Instruction

SRS: Abbreviation for Software Requirements Specification

SVVP: Abbreviation for Software Verification and Validation Plan.

SVVR: Abbreviation for Software Verification and Validation Report.

## Definitions

In addition to an ordinary English-language meaning, each term listed in this section has a specific meaning applicable to the scope of this document.

Some of the terms are also defined in the ISO13485 standard

* Acceptance Testing: Testing to determine if the software correctly implements hardware and software requirements in an operational environment.

Acceptance testing also challenges the adequacy of user documentation.

* Algorithm Analysis: Testing to determine that algorithms have been properly implemented according to the requirement and design specifications. Algorithm Analysis is performed using a variety of techniques including comparisons to measured data and hand computations.
* Black Box Testing: See the definition for “Functional Testing”.
* Clear Box Testing: See the definition for “Structural Testing”.
* Functional Testing: Also known as “Black Box Testing”. Functional Testing consists of executing a variety of functional tests to assess the performance of the product within its integrated system. Functional Testing methods focus on the functional requirements of the software as defined by the SRS, and include but are not limited to performance tests, interface tests, use case based tests, and design maturity tests.

Functional Testing approaches are used in Integration, System, Beta and Acceptance testing.

* Performance Tests: Functional tests intended to validate the System’s performance against prescribed industry standards and other performance requirements as specified in the SRS. Performance Tests are included in integration testing.
* System Testing: The process of testing an integrated hardware and software system in a production environment to verify that the system meets its specified requirements.
* Unit Testing: Testing conducted to verify the implementation of the design for a single element of software and/or hardware, or a collection of software and/or hardware elements. Employs the static and dynamic testing methodologies defined as “Structural Testing”.
* Structural Testing: It replaced to Section 5.1

## Reference

### Regulatory Standards & Guidances

1. FDA Quality System Regulation 21 CFR, Part 820
2. CDRH Guidance : General Principles of Software Validation

### Industry Standards and Guidances

* + 1. ISO 13485 : 2003, Quality management System
    2. IEEE Std 1012-1986, Standard for Software Verification and Validation Plans
    3. IEEE Std 829-1983, Standard for Software Test Documentation

4) EN 60601-1-4 [1996] : Programmable electrical medical systems

### Internal Standards & Guidances

Design & development procedure

## Roles and Responsibilities

### Management Members

Jung Hyun Woo, Researcher of R&D Team

### Hazard Analysis Team : Members, Roles and Responsibilities

Jung Hyun Woo, Researcher of R&D Team

### V & V Testing Team : Members, Roles and Responsibilities

Oh Jae Hong, Senior researcher of R&D Team

### Documentation Review : Members, Roles and Responsibilities

Oh Jae Hong, Senior researcher of R&D Team

### Team Members/Qualifications

General Qualifications/Requirements as described in each individual’s training file.

## V.V & T Item & Methods

### Static Analysis

a. Code Review: performed by S/W engineer. (Appendix 1)

b. Module Test: performed by S/W engineer. (Appendix 2)

c. Integration Test: performed by S/W engineer. (Appendix 3)

## Documentation Handling

Documentation will be handled in accordance with META BIOMED Co., Ltd. standards outlined in QM Document Control.