**Software Validation Plan**

**Model: i-DOLPHIN**

**Document No. : Q4-29-015(01) Rev.XX**

This document valid from the date of approval

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

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

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Revision No.** | **Revision history** | **Date** |
| 1 |  |  |
| 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Purpose**

This document is the application specification document of Multi-Purpose Micro Endoscope for Software life cycle processing to IEC 62304: 2006.

This Software life cycle processing and mitigates risk caused by software problems associated with correct use and errors.

1. **Range**

|  |  |  |
| --- | --- | --- |
| **Device** | **Description** | **Range** |
| Monitor | Image shot by device is diagnosed  through Monitor | Minimum  1280\*1024 resolution, 15 inch |
| Microprocessor |  | Minimum 3.3GHz Processor |
| Memory Device |  | Minimum 4GB RAM |
| Operating  System |  | Windows 7, Window 8 |
| Communications | Power supplying to device Data transferring to PC | USB 2.0, USB 3.0 |

1. **Definitions of Terms**

|  |  |
| --- | --- |
| **Terms** | **Definition** |
| Auto Exposure | Automatically adjust exposure value according to distance between camera and object |
| Brightness | Adjust light and darkness |
| Control-Gain | Adjust brightness level (0~3) |
| Control-RGGB | Adjust Red, Green 1, Green 2, Blue value |
| Image Capture | Save the image as .jpeg format |
| Rotate | Rotate displayed image on the screen |
| Rotate-Reverse | Reverse displayed image on the screen |
| Video Recording | Save the video as .avi format |

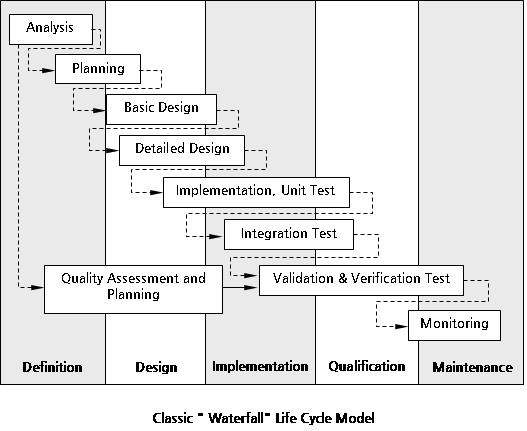
1. **Project Overview**
2. **Software Development Process**
   1. **Project reference**
      1. Risk Management Plan
      2. Risk Management Report
      3. FMEA Software Validation
      4. TRF Document Software Validation
   2. **Standard and Regulatory References**
      1. Regulatory Standards & Guidances
3. FDA Quality System Regulation 21 CFR, Part 820
4. CDRH Guidance: General Principles of Software Validation
   * 1. Industry Standards & Guidances
5. ISO 13485: 2003, Quality management system
6. IEEE std 1012-1986, Standard for Software Verification and Validation
   * + 1. Plans
7. IEEE std 829-1983, Standard for Software Test Documentation
8. EN 60601-1-4[1996]: Programmable electrical medical systems
   1. **Standard and Regulatory References**

To aid the development process, task lists showing specific deliverables, by phase, are used as a form of guidance. A typical list is shown in the table below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase**  **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 |

* 1. **Development Life Cycle Procedure**

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



**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.

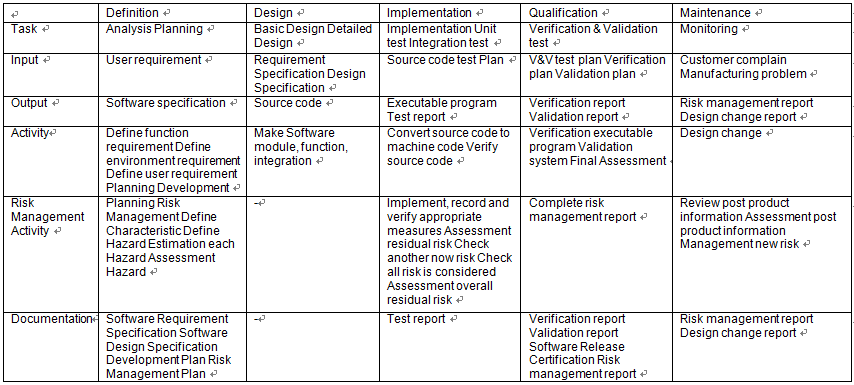
**Design:** This activity considers four functional attributes of the software : data structure, software architecture, functionality, and interface characterization. The design is documented in design documents.

**Implementation:** The implementation phase converts the design into a machine readable form. The software source code is created in accordance with S-denti coding conventions. To verify that the code created is consistent with S-denti 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.

* + 1. Description of Development Life Cycle



1. **Software Documents**

Software Document will be created by Section 5 Software Development Process. Also comes the Project reference in the document, please see 5.1.

IEC 62034: 2006 will be to organize documents by topics presented.

|  |  |
| --- | --- |
| No | Contents |
|
| 1 | Level of Concern |
| 2 | Software description |
| 3 | RISK MANAGEMENT |
| 4 | Software Requirements Specification |
| 5 | Architecture Design Chart |
| 6 | Software Design Specification(SDS) |
| 7 | Traceability |
| 8 | Software development Environment Description |
| 9 | Verification and Validation Documentation |
| 10 | Revision Level History |
| 11 | Unsolved Bugs |
| 12 | Release Version Number |
| 13 | IEC 62304 Checklist : (TRF Report) |
| 14 | Appendix |

[Table1. Contents]

1. **Work Breakdown Structure**
   1. **Software**
      1. Program Language

Editor: MS visual studio

Debugger: MS visual studio

C compiler: MS visual studio Compiler

Assembler: MS visual studio

Linker: MS visual studio

* + 1. Program size

Main UI & sequence:

* 1. **Firmware**
     1. Program Language

Editor : Xilinx

* + 1. Program Language

Main UI & sequence: .Net Framework

1. **Resource Management Plan**
   * 1. Management Members

Jung Hyun Woo, Researcher of R&D Team

* + 1. Hazard Analysis Team: Members, Roles and Responsibilities

Jung Hyun Woo, Researcher of R&D Team

* + 1. V & V Testing Team: Members, Roles and Responsibilities

Oh Jae Hong, Senior researcher of R&D Team

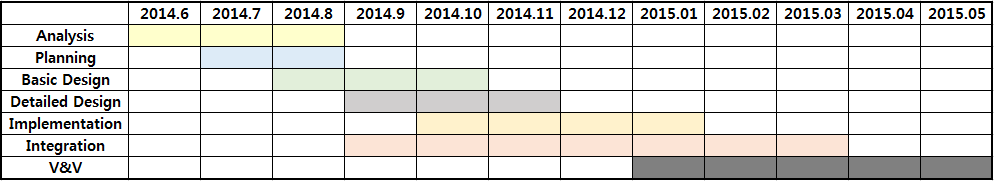
* + 1. Documentation Review: Members, Roles and Responsibilities

Oh Jae Hong, Senior researcher of R&D Team

* + 1. Team Members/Qualifications

General Qualifications/Requirements as described in each individual’s

1. **Project Schedule**



**10 Project Deliverables**

**11 Risk Management Plan**

1. **Project Deliverables**

To ensure the stability of the products through the CE certification. Therefore, this project will be completed.

1. **Risk Management Plan**

Using the Risk Table below will be derived Risk.

Classification chart for the acceptance of risks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Class A** | **Class B** | **Class C** | | |
| **Negligible(1)** | **Minor(2)** | **Serious(3)** | **Critical(4)** | **Catastrophic(5)** |
| **Frequent(5)** |  |  |  |  |  |
| **Probable(4)** |  |  |  |  |  |
| **Occasional(3)** |  |  |  |  |  |
| **Remote(2)** |  |  |  |  |  |
| **Improbable(1)** |  |  |  |  |  |

Risk Level: Severity x Probability

|  |  |
| --- | --- |
|  | Unacceptable risk |
|  | Acceptable risk |

Requirement in relation to the relationship derived through the Risk will create a Software Validation Report.  
  In related information, please see the Software Validation Report that is created later.

1. **Software Verification & Validation Plan**
   1. **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.

* 1. **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 ISO 13485 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.

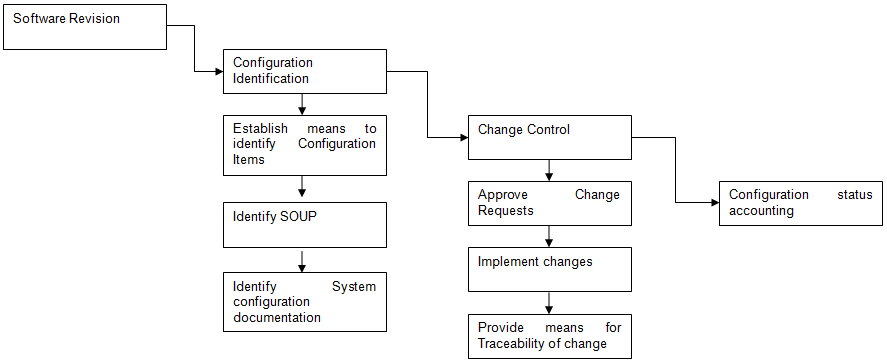
* Algorithm Analysis: Testing to determine that algorithms have been properly implemented according to the requirement and design specifications. Algorithm Analysis is performed using a variety of techniques including comparisons to measured data and hand computations.
* Performance of the product within 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”.

1. **Software Configuration Management Plan**

Software Configuration Management Life Cycle Procedure



1. **Software Quality Plan**

**Software Quality Plan is the section set by the Development Life Cycle Procedure of 5.4.**