**Software Validation Plan**

**Model: SmartSONO MS-09**

**Document No. : (문서 번호 작성 해야 함)**

This document valid from the date of approval

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Title | Name | Date | Signature |
| Prepared by | Researcher | H.W Jung | 2015.03.02 |  |
| Reviewed by | Director | J.H. Oh | 2015.03.02 |  |
| Approved by | CTO | G.Y. Kim | 2015.03.02 |  |

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

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision | Description | Author |
| 2014.03.02 | 0 | The initial creation | H.W Jung |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 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 use errors.

# Range

|  |  |  |
| --- | --- | --- |
| Device | Description | Range |
| Display monitor | Display device supporting min.  Resolution 1280\*1024. | -Monitor: 15 inch |
| CPU | MS-09 is used for efficient image processing. | -Support SCUS |
| RAM | Minimum memory size for efficient image reconstruction is required. | -1GB or above |
| Mother board | - PCI slot for plugging SICC card  - RS232 communication port to connect OCB and RGU | -VGA slot: 2 or above |

# Definition of Terms

|  |  |
| --- | --- |
| Terms | Definition |
| Tx | According to the probe options to meet user purpose, Tx cycle and signal processing must be changed to match. |
| Rx | In the process of received signal, determine the gain according to the noise and Amplitude by H/W setting. |
| Range gate | As a function of PW (Pulse Wave), it is used when the acquisition and processing of the information about a particular input field. It is used mainly to obtain information about blood flow. |
| Frequency | Every probe has its own Frequency to use. By using this Frequency, it can be obtain information of the desired value. |
| MI | By calculating Mechanical index, it indicates the size of the mechanical impact on the tissue of ultrasound. If it is out of specification, it can increase reliability of equipment by generating a warning. |
| TI | By calculating the Thermal index, it indicates the size of the temperature effects on body tissue of ultrasound. If it is out of specifications, it can increase reliability of the equipment by generating a warning. |
| Dynamic Range | TGC and overall screen brightness can be adjusted through the Dynamic Range internally. In general, the deep section comes to relatively dark due to a weak signal. It is used to complement of it. |
|  |  |

# Project Overview

타사 제품에 비해 저가형으로 널리 보급하기 위하여 이 사업을 시작 하였다. 하지만 저가형이라고 하지만, 기본적인 기능이 있고 사용자에게 거부감 없는 제품을 만들 것이다. 따라서 타사의 제품과 경쟁할 수 있도록 할 것이다.

이 제작을 위한 System의 모델 명은 SmartSONO MS-09로 지정 할 것이다.

Generally lower cost compared to other products, and started this business.  
However, as a low cost, but its basic functions and may be reluctant products to users.  
So, we will allow you to compete with third-party products.  
The system is producing the model name will be designated as SmartSONO MS-09.

# Software Development Process

## Project reference

### Risk Management plan

### Risk Management report

### FMEA SW validation

### TRF Document SW Vaildation

## Standard and Regulatory References

### Requlatory 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

## Standard and Regulatory References

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 |

## 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.

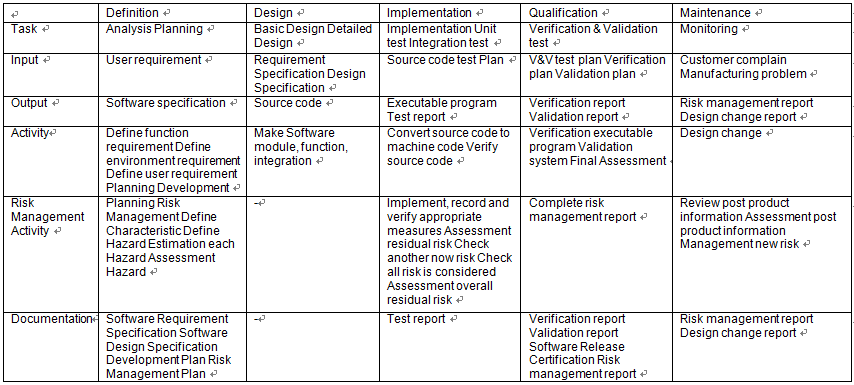
**Design.** This activity considers four functional attributes of the software: data structure, software architecture, functionality, and interface characterization. The design is documented in design documents.

**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 Documents

Software Document는 5절의 Software Development Process에 의해 작성이 될 것이다. 또한 여기서 나오는 문서는 5.1의 Project reference에 참조 바란다.

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

IEC 62034:2006에 제시된 항목에 의하여 문서를 구성 할 것이다.

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]

# Work Breakdown Structure

# 7.1 Software

### Program Languae

Editor : MS visual studio

Debugger : MS visual studio Compiler

C compiler : MS visual studio

Assembler : MS visual studio

Linker : MS visual studio

### Program size Main UI & sequence: about 100MB

# 7.2 Firmware

### 7.2.1 Program Languae

Editor : Visual DSP

C compiler : MS visual studio

Assembler : MS visual studio

Linker : MS visual studio

### 7.2.2 Program Languae

### Main UI & sequence: about 10MB

# Resource Management Plan

# 8.1 Role and Responsibility

8.1.1 Management Members

Jung Hyun Woo, Researcher of R&D Team

8.1.2 Hazard Analysis Team : Members, Roles and Responsibilities

Jung Hyun Woo, Researcher of R&D Team

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

Oh Jae Hong, Senior researcher of R&D Team

8.1.4 Documentation Review : Members, Roles and Responsibilities

Oh Jae Hong, Senior researcher of R&D Team

8.1.5 Team Members/Qualifications

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

# 8.2 Project Term Organization



**8.3 Human Resources**

Software Engineer: H.W Jung

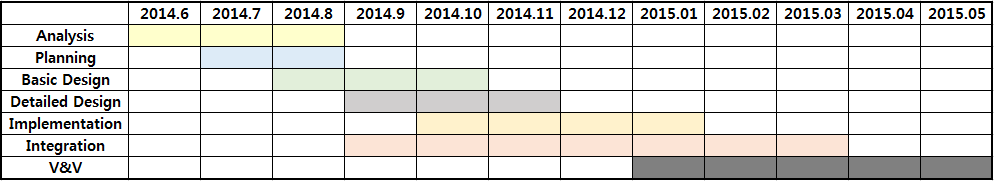
Hardware Engineer: J.H Oh

Firmware Engineer: J.W Kim

Design Engineer: Y.H Lee

Safety Engineer: H.W Jung

# Project Schedule



## Dfadf

# Project Deliverables

## CE 인증을 통하여 제품의 안정성을 보장 한다. 따라서 이 프로젝트를 완료 할 것이다.

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

# Risk Management Plan

아래의 Risk Table을 이용하여 Risk를 도출 할 것이다.

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 |

## 도출 된 Risk를 통하여 Requirement와 관계를 연관하여 Software Validation Report를 작성 할 것이다.

이에 관한 내용은 추후 작성되는 Software Validation Report를 참조 바란다.

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.

# Software Verification & Validation Plan

## 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.

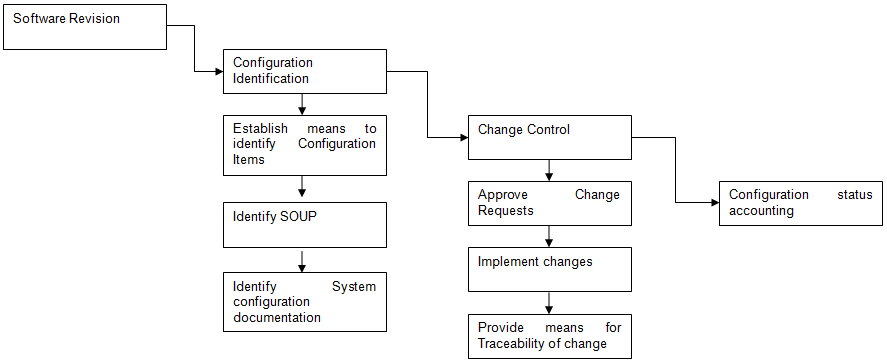
* Algorithm Analysis: Testing to determine that algorithms have been properly implemented according to the requirement and design specifications. Algorithm Analysis is performed using a variety of techniques including comparisons to measured data and hand computations.
* Performance of the product within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

# Software Configuration Management Plan

Software Configuration Management Life Cycle Procedure



# Software Quality Plan

5.4절의 Development Life Cycle Procedure에 의해 Software Quality Plan이 설정이 된다.

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