|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Author** | **Change description** |
| 1 | 12/04/23 | JF | Initial Release |
| 2 | 23/10/23 | JF | Software design and development was covered in SSI-SOP-10 Design and Development. Moved forms from SOP-10 to SOP-20:  SSI-QF-10F becomes **SSI-QF-20J** Software Release Version Control  SSI-QF-10J becomes **SSI-QF-20K** Software Checklist  SSI-QF-10N becomes **SSI-QF-20L** Software Build Numbers Summary of Changes  SSI-QF-10S becomes **SSI-QF-20M** Firmware Programmable PartRelease  **SSI-QF-10P** Problem Report. This deals with both hardware and software issues and remains in SOP-10 but is referenced in this SOP-20 as many processes relating to software development are fulfilled using this form. |
| 3 | 20/06/24 | JF | Added:  Definition of ‘build’;  SSI-QF-20O Documenting SOUP (or OTS – off the shelf) Software; Reference to IEC 62304:2006 +AMD1:2015, Medical Device – Software Lifecycle Processes;  References to system requirements;  Use of SSI-QF-20L on release of new builds;  Cyber security included in SSI-QF-20G Software Maintenance Plan;  Minor edits for clarity or consistency. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Approved:………GJGD…………. ……..

(CEO)

6Date:…………..…20/06/24…………

# Purpose

This document describes how software is developed and maintained by Stowood Scientific Instruments (Stowood) in order to comply with the BS EN 62304:2006+A1:2015 Medical Device Software – life-cycle processes and fulfil the general safety and performance requirements of the EU MDR 2017/745 (EUMDR), the essential requirements of the MDD 93/42/EEC (MDD), UK MDR 2002 (as amended) (UKMDR) and analogous requirements of SOR/98-282 Canadian Medical Device Regulations and analogous requirements of the US FDA. This procedure is intended to be used in conjunction with **SSI-SOP-10 Design and Development** to ensure the safety and correct performance of a medical device.

The procedure follows sections of BS EN 62304:2006+A1:2015, and is also based on Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices from the FDA.

# Scope

This procedure applies to the development of medical device software when software is itself a medical device or when software is an embedded or integral part of a medical device.

This process is intended to meet the requirements for software Class A or B per BS EN 62304:2006+A1:2015 and FDA ‘level of concern’ of Moderate. If the software is determined to be of a higher level, then the development process will be reviewed to ensure that all required steps are in place.

**N.B. Compliance of LEGACY SOFTWARE such as Visi-Download is not covered by this process. Please refer to Clause 4.4 of BS EN 62304:2006+A1:2015. Procedures for the release of new builds of Visi-Download and problem reporting and resolution are covered herein.**

# Responsibilities

## CEO/Project Lead:

* + 1. Leadership and monitoring of the overall project progress.
    2. Ensuring team members are trained to relevant SOPs.
    3. Ensuring completion of the required activities per this SOP.
    4. Leading the design review process and ensuring that design review action items are generated, documented and completed.
    5. Defining the system requirements

## QARCM

* + 1. The Quality Assurance and Regulatory Compliance Manager will ensure that procedures are followed and that regulatory compliance is achieved; in particular that **SSI-QF-20B Software Development Plan** is followed. **Visi-Download has already been developed so there is no Software Development Plan for it.**

## Software Manager:

* + 1. Defining the software requirements to meet the system requirements
    2. Providing input on risks, hazards and harms associated with the software.
    3. Aligning software planning activities with risk management activities, system and software engineering activities and hardware engineering activities.
    4. Tracking and monitoring software risk, during development and after release.
    5. Defining software engineering life cycle, phases and iterations.
    6. Verifying that the software design reflects the system requirements and is adequately captured in the software requirements
    7. Ensuring that risk control measures are incorporated.
    8. Reviewing software changes software development/maintenance plan and project milestones for identification and planning of software quality assurance and verification and validation (V&V) activities. Track and monitor software functional configuration.
    9. Developing test protocols that verify the requirements.

## Support Engineer

3.4.1 Reports and records on **SSI-QF-10P** **Problem Report** any problems or observations arising in the course of his role as first-line support to users of Stowood products including software

3.4.2 Participates in weekly review meetings to elaborate on problems / observations reported, identify their causes, evaluate their seriousness, consider risk and generate solutions.

## Independent Reviewer:

3.5.1 For the development of any new software Stowood will involve an individual with suitable qualifications and experience who has not been involved with the software development to join its design review meetings to provide an independent review of the process. The independent reviewer will usually be an independent consultant but may be someone from within SSI who meets these criteria.

# References

* 1. SSI-SOP-10, Design and Development
  2. SSI-SOP-13, Risk Management
  3. SSI-QF-20A Software Safety Classification
  4. SSI-QF-20B Software Development Plan
  5. SSI-QF-20C Software Requirements Traceability
  6. SSI-QF-20D Software Architecture Design
  7. SSI-QF-20E Software Test Protocol
  8. SSI-QF-20F Software Test Report
  9. SSI-QF-20G Software Maintenance Plan
  10. SSI-QF-20H Software Maintenance Report
  11. SSI-QF-20I Software Development Summary Report
  12. SSI-QF-20J Software Release Version Control
  13. SSI-QF-20K Software Checklist
  14. SSI-QF-20L Software Build Numbers Summary of Changes
  15. SSI-QF-20M Firmware Programmable Part Release
  16. SSI-QF-20N Software or Firmware code
  17. SSI-QF-20O Documenting SOUP (or OTS – off the shelf) Software
  18. SSI-QF-20P Software Requirements Specification
  19. SSI-QF-20Q Software Functional Specification
  20. SSI-QF-20R Software User Requirements
  21. SSI-QF-20S Software Analysis Verification
  22. SSI-QF-10C Design Review
  23. SSI-QF-10D Traceability Matrix
  24. SSI-QF-10E Design Transfer Checklist
  25. SSI-QF-10G Design Change Record
  26. SSI-QF-10P Problem Report
  27. SSI-QF-13E Design Failure Mode and Effects Analysis (DFMEA)
  28. VISI-006 Risk Assessment and Implementation Control

### 4.23 IEC 62304:2006+AMD1:2015, Medical Device – Software Lifecycle Processes

# Definitions for Software Purposes

* 1. Acceptable – in relation to a risk, one which has been determined to be acceptable by application of procedures in **SSI-SOP-13 Risk Control**
  2. Activity - a set of one or more interrelated or interacting tasks
  3. Anomaly - any condition that deviates from the expected based on requirements, specifications, design documents, standards, etc. or from someone’s perception or experience. Anomalies may be found during, but not limited to, the review, test, analysis, compilation, or use of software products or their documentation.
  4. Architecture - organizational structure of a software system or component.
  5. Asset – physical or digital item that has value
  6. Authorisation – permission granted to a person to access software, data or processes (‘privileges’), or the act of granting the same
  7. Availability – property of being accessible or usable on demand by authorised persons
  8. Build – a new release of a software package identified by a Build No. in the format yymmdd that has passed all approval processes
  9. Confidential – describes information that may not be made available or disclosed except to authorised legal or proper persons or processes
  10. Combined Vulnerability Disclosure (CVD) – process by which interested parties work together to reduce the risks from disclosure of vulnerabilities
  11. Cybersecurity – the protection to an acceptable degree of software and other electronic data and systems from unauthorised access to prevent theft, improper use, lockout, disclosure, disruption, modification, deletion or destruction. Cybersecurity must be maintained during the software lifecycle.
  12. Configuration Item - entity that can be uniquely identified at a given reference point.
  13. Deliverable - required result or output (includes documentation) of an activity or task.
  14. Evaluation - a systematic determination of the extent to which an entity meets its specified criteria.
  15. IT Security - protection of computer systems from adverse effects including disruption or misdirection of the services they provide.
  16. Integrity - property whereby data has not been altered in an unauthorized manner since it was created, transmitted or stored
  17. Operation Security - Protection against the intentional, unauthorised modification or corruption of procedures or workflows
  18. Medical device software - software system that has been developed for incorporation into a medical device or that is intended for use as a medical device in its own right.
  19. Problem report - a record of actual or potential behaviour of a software product that a user or other interested person believes to be unsafe, inappropriate for the intended use, contrary to specification, capable of improvement, or problematic in some way. A manufacturer can reject a problem report as a misunderstanding, error or insignificant event. A problem report can relate to a released software product or to a software product that is still under development.
  20. Process – a set of interrelated or interacting activities that transform inputs into outputs.
  21. Regression testing - the testing required to determine that a change to a system component has not adversely affected functionality, reliability or performance and has not introduced new defects.
  22. Security - protection of information and data so that unauthorised people or systems cannot read, steal, modify, deny authorised access to, or destroy them while allowing access to authorised persons or systems.
  23. Serious injury – injury or illness that directly or indirectly:
      + Is life threatening,
      + Results in permanent impairment of a body function or permanent damage to a body structure, or
      + Necessitates medical or surgical intervention to prevent permanent impairment of a body function or permanent damage to a body structure

NOTE: Permanent impairment means an irreversible non-trivial impairment or damage to a body structure or function.

* 1. Software Development Lifecycle Model - conceptual structure spanning the life of the software from definition of its requirements to its release for manufacturing, which:
     + Identifies the system requirements and the process, activities and tasks involved in development of a software product to meet the system requirements,
     + Describes the sequence of and dependency between activities and tasks, and
     + Identifies the milestones at which the completeness of specified deliverables is verified.
  2. Software Item - any identifiable part of a computer program.
  3. NOTE: Three terms identify the software composition. The top level is the software system. The lowest level that is not or cannot be further decomposed is the software unit. All levels of composition, including the top and bottom levels, can be called software items. A software system, then, is composed of one or more software items, and each software item is composed of one or more software units or decomposable software items. The responsibility is left to the manufacturer to provide the definition and granularity of the software items and software units.
  4. Software product – set of computer programs, procedures and associated documentation and data, if any.
  5. Software system - integrated collection of software items organized to accomplish a specific function or set of functions.
  6. Software unit - software item that is not subdivided into other items.
  7. Note: Software units can be used for the purpose of software configuration management or testing.
  8. SOUP (Software of Unknown Provenance) - software item that is already developed and generally available and that has not been developed for the purpose of being incorporated into the medical device (also known as “off-the-shelf software”) or software previously developed for which adequate records of the development processes are not available.
  9. System - integrated composite consisting of one or more of the processes, hardware, software, facilities, and people, that provides a capability to satisfy a stated need or objective.
  10. Task - a single piece of work that needs to be done.
  11. Threat – potential for a breach of security.
  12. Traceability - degree to which a relationship can be established between inputs and outputs in the development process.
  13. Note: Validation – Validation is not covered in BS EN 62304:2006+A1:2015 but validation must occur before software is released.
  14. Verification - confirmation through provision of objective evidence that specified requirements have been fulfilled.
  15. Version – Modified version of a software product resulting in a new version that has passed all approval processes. Stowood refers to versions as ‘builds’ see definition above.
  16. Vulnerability – a weakness in security

# PROCEDURE

## Quality Management System

* + 1. All products shall be developed within the BS EN ISO 13485 quality management system operated by Stowood.

## Risk Management

* + 1. Development shall follow the risk management process set out in **SSI-SOP-13 Risk Management.**

## Software Safety Classification

* + 1. Under IEC 62304 a software system is assigned a safety class A, B or C according to the possible effects resulting from a hazard to which the software system contributes or may contribute. The classification is arrived at and recorded using **SSI-QF-2OA** **Software Safety Classification.**
    2. The software safety classes are initially be assigned based on severity:
       - Class A: No injury or damage to health is possible
       - Class B: Non-serious injury is possible
       - Class C: Death or serious injury is possible
    3. If a hazard could arise from a failure of the software system to behave as specified, the probability of such failure shall be assumed to be 100 percent.
       - * If the risk of death or serious injury arising from a software failure is subsequently reduced to an acceptable level by a risk control measure, either by reducing the consequences of the failure or by reducing the probability of death or serious injury arising from that failure, the software safety classification may be reduced from C to B;
         * and if the risk of non-serious injury arising from a software failure is similarly reduced to an acceptable level by a risk control measure, the software safety classification may be reduced from B to A.
    4. Each software system that contributes to the implementation of a risk control measure is assigned a software safety class. The software safety class assigned to each software system is documented in the risk management file/risk management plan (Refer to **SSI-SOP-13 Risk Management**). For Visi-Download refer to **Visi-006 Risk Assessment and Control Implementation**.
    5. When a software system is decomposed into software items, and when a software item is decomposed into further software items, such software items inherits the software safety classification of the original software item (or software system) unless a rationale for classification into a different software safety class is arrived at and recorded using  **SSI-QF-20A** **Software Safety Classification document.** Such a rationale shall explain how the new software items are segregated so that they may be classified separately.
    6. Wherever a process is required for software items of a specific classification and the process is necessarily applied to a group of software items, the processes and tasks which are required by the classification of the highest-classified software item in the group shall be used unless a rationale is documented for using a lower classification.
    7. For each software system, until a software safety class is assigned, Class C requirements shall apply.
    8. The FDA classifies the safety of software as a *Major, Moderate* or *Minor* ‘Level of Concern’ based on the severity of injury that a device could permit or inflict. This is prior to any risk control measures. (See Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices). Tables 1 and 2 of the guidance document give some specific circumstances for the classification.
       - *Major*: a failure or latent flaw could directly result in death or serious injury to the patient or operator. The level of concern is also *Major* if a failure or latent flaw could indirectly result in death or serious injury of the patient or operator through incorrect or delayed information or through the action of a care provider.

This regulation defines serious injury as an injury or illness that:

1. is life threatening;
2. results in permanent impairment of a body function or permanent damage to a body structure; or
3. necessitates medical or surgical intervention to preclude permanent impairment of a body function or permanent damage to a body structure.
   * + - *Moderate*: a failure or latent design flaw could directly result in a minor injury to the patient or operator. The level of concern is also *Moderate* if a failure or latent flaw could indirectly result in a minor injury to the patient or operator through incorrect or delayed information or through the action of a care provider.

A minor injury is “any injury that does not meet the definition of a serious injury”.

* + - * *Minor*: failures or latent design flaws are unlikely to cause any injury to the patient or operator.
      1. If the device is intended to be marketed in the USA, then the Level of Concern and how it was determined shall be recorded in **SSI-QF-20A Software Safety Classification.**

## Software Development Process

### Software Development Planning

* + - 1. A software development plan shall be created for each project using the **SSI-QF-20B** **Software Development Plan Template** (Summary required for FDA).
      2. The plan shall address the following:
         * The processes to be used in the development of the software system
         * the deliverables (including documentation) of the activities and tasks;
         * traceability between system requirements, software requirements, software system test, and risk control measures implemented in software;
         * software configuration and change management, including SOUP (software of unknown provenance) configuration items and software used to support development. SOUP shall be documented using **SSI-QF-20O** **Documenting SOUP (or OTS – off the shelf) Software**; and
         * software problem resolution for handling problems detected in the software products, deliverables and activities at each stage of the life cycle.
      3. The software development plan shall reference:
         * the system requirements used as input to software development
         * the procedures for coordination of development with the system development (e.g. system integration, verification and validation).
      4. The software development plan shall include plans to integrate software items (including SOUP) and perform testing during integration. (Not required if Software Level of Concern is *Minor*).
      5. The software development plan shall include information on deliverables requiring verification, the verification tasks for each lifecycle activity, the milestones at which deliverables are verified and the acceptance criteria for verification of deliverables.
      6. The software development plan shall include reference to how tasks and activities of the risk management process (including SOUP risks) will be carried out.
      7. All documentation produced shall follow **SSI-SOP-08 Document Control.**
      8. Software code files shall be named such that the function is easily identifiable. Code versions shall be controlled using a revision control system.
      9. SSI identifies software versions by build number. Each new version is given a build number, which follows the format ‘yymmdd’, e.g. 231026 for a new build no. released on 26/10/2023. The build number for software shall be recorded on **SSI-QF-20J Software Release Version Control**
      10. The software development plan shall include or make reference to:
      * The system requirements;
      * The classes, types or categories of software to be used;
      * The software configuration management activities and tasks;
      * the staff members responsible for performing software configuration management and activities;
      * their relationship with other staff members responsible for software development or maintenance. These may be the same people;
      * when the items are to be placed under configuration control; and
      * when the problem resolution process is to be used.
      1. Supporting Items to be controlled (Class B)

The plan should also include activities to control supporting items that shall include tools, items or settings, used to develop the medical device software, which could affect the medical device software.

* + - 1. Configuration Item before Verification (Class B)

Configuration items shall be placed under documented configuration management control before they are verified.

* + - 1. Software Integration and Integration Testing Planning (Class B)

Include or reference a plan to integrate the software items (including SOUP) and perform testing during integration. It is acceptable to combine integration testing and software system testing into a single plan and set of activities.

### Software Requirements Analysis

* + - 1. A software requirements specification shall be produced from, and with reference to, the system requirements. Software requirements shall be retraceable to the system requirements. This shall be documented using **SSI-QF-20C Software Requirements Traceability Matrix.**
      2. The risk analysis shall be re-analyzed when software requirements are established and updated as appropriate.
      3. As appropriate, the software requirements specification should include:
      * Functional and capability requirements. Examples include:
        + performance (e.g., purpose of software, timing requirements),
        + physical characteristics (e.g., code language, platform, operating system),
        + computing environment (e.g., hardware, memory size, processing unit, time zone, network infrastructure) under which the software is to perform,
        + need for compatibility with upgrades or multiple SOUP or other device versions.
      * Software system inputs and outputs. Examples include data characteristics (e.g., numerical, alpha-numeric, format), ranges, limits, and defaults)
      * Interfaces between the software system and other systems
      * Software-driven alarms, warnings, and operator messages
      * IT and Operation Security requirements to ensure device security, patient safety, confidentiality and integrity, including:

**Secure Communications** (establishing data interfaces (wired, Wi-Fi, Ethernet, Bluetooth, USB etc.), secure communication protocols such as HTTPS or TLS so that data transmitted is protected from eavesdropping or tampering;

**Data Protection** (identification of data types such as patient data e.g. demographic data, diagnoses, examination data e.g. laboratory values, definition of measures to protect sensitive patient data or safety-related data, such as encryption and access controls; all data transmissions should also be secured to prevent interception or tampering);

**Device Integrity** (measures such as secure boot, code signing, tamper-resistant hardware, audit logging function in the system-level architecture);

**Authentication and authorisation** (protection against unauthorized access e.g. username/password, biometrics, hardware keys; authorisation controls so that users only have access to the data and functions that they need to perform their job);

**Physical Access** (definition of controls such as physical locks and seals)

**Reliability and availability** (design features that will allow the device to detect, resist, and recover from cybersecurity attacks in order to maintain its essential performance).

* + - * Usability engineering requirements that are sensitive to human errors and training (Examples include those related to support for manual operations, human-equipment interactions, constraints on personnel, and areas needing concentrated human attention)
      * Data definition and database requirements (Examples include form, fit, function)
      * Installation and acceptance requirements of the medical device software when delivered to the enduser
      * Requirements related to methods of operation and maintenance
      * User documentation to be developed
      * User maintenance requirements, and
      * Regulatory requirements.
      1. The document shall be updated as new requirements become apparent.
      2. Once the software requirements specification has been established, the risk analysis shall be re-evaluated to check that requirements have not introduced new risks.
      3. Software requirements specification verification shall be performed to verify that the following are met:
         * The system requirements including those related to risk control have been implemented
         * The software specifications items are not contradictory
         * The software specification items are not ambiguous
         * Software specification items are stated in a way that can be tested
         * Items are uniquely identified
      4. Software specification items are traceable to the system requirements. The verification shall be conducted as part of a formal design review detailed in section 6.5.2.9 below.
      5. Risk Control Measures (Class B)

Include risk control measures implemented in software for hardware failures and potential software defects in the requirements as appropriate to the medical device software.

* + - 1. A design review shall be conducted at this stage (**SSI-QF-10C** **Formal Design Review**) to ensure that requirements’ development with appropriate risk management activities, has reached an adequate level of completion. Minimum required attendees for this review include the Project Lead, Software Manager and an Independent Reviewer.

### Software Architectural Design

* + - 1. The architectural design of the software shall be documented using **SSI-QF-20D Software Architectural Design.**
      2. Software Architecture (Class B)

The **Software Requirement Specification** documented in **SSI-QF-20C** shall be transformed into a Software Architecture describing the software structure and identifying software. This should be done in the form of high level diagrams, which should help to identify separate software items and software functions. (Not required if software is of FDA Minor Level of Concern).

Within the software architecture document, the interfaces between different software items and between software items and external components (including hardware) shall be documented. (Not required if software is of FDA Minor Level of Concern).

* + - 1. SOUP (Class B)

The functional and performance requirements for any SOUP items should be defined. (See also FDA OTS Software Guidance).

Any hardware or software requirements necessary to support proper operation of any SOUP items shall be identified and documented (for e.g. processor type and speed, memory type and size, system software type, communication and display software requirements). (See also FDA OTS Software Guidance).

* + - 1. Software Bill of Materials (SBOM)

See IMDRF/CYBER WG/N73DRAFT:2022, Principles and Practices for Software Bill of Materials for Medical Device Cybersecurity, and NTIA Software Transparency Healthcare POC - How-To Guide for SBOM Generation.

* + - * 1. As part of the software architecture, an SBOM shall be developed to track and manage the software components used in products and systems as follows:

Create an SBOM that lists all of the components used in the software, including both SOUP and OTS (Off-the-Shelf) components and any custom code developed in-house.

The SBOM should include for each component where it resides, name, version, supplier and level of support provided by supplier, end of support date, and any relevant security information/known vulnerabilities.

Ensure that the SBOM is kept up-to-date and record any changes made to the software throughout its lifecycle.

Use the SBOM to identify and address any vulnerabilities or security issues in the software, and to provide transparency to customers and regulators about the software's components and security.

* + - 1. Software Architecture Verification (Class B)
      2. It shall be verified that the following are met:
         * The architecture of the software implements the system requirements and **SSI-QF-20C Software Requirements Traceability** including risk control
         * It is able to support the interfaces between different software items and between software items and external components (including hardware)
         * It supports proper operation of any SOUP items
      3. The above should be carried out as part of the software architecture sign off.
      4. A formal design review should be used for the formal verification of the software architecture **SSI-QF-10C Formal Design Review.**

### Software Detailed Design

* + - 1. The Software Architecture shall be refined and software items subdivided (if appropriate) to be represented by software units. (Not required if software is of FDA Minor Level of Concern).
      2. FDA requires that Software Design Specification documentation be created (Not required if software is of FDA Minor Level of Concern).
      3. “The Software Design Specification (SDS) describes the implementation of the requirements for the Software Device. In terms of the relationship between the SRS and the SDS, the SRS describes what the Software Device will do and the SDS describes how the requirements in the SRS are implemented. The information presented in the SDS should be sufficient to ensure that the work performed by the software engineers who created the Software Device was clear and unambiguous, with minimal ad hoc design decisions. The SDS may contain references to other documents, such as detailed software specifications”. (FDA Guidance for Premarket Submission).
      4. A detailed design for any interfaces between the software unit and external components (hardware or software), as well as any interfaces between software units shall be captured as part of the SDS. The SDS can be captured on **SSI-QF-20D Software Architecture Design, SSI-QF-20C Software Requirements Specification a**nd **SSI-QF-10D Design Traceability Matrix** or a separate document.
      5. Traceability between SRS and SDS shall be documented in **SSI-QF-20C**

### Software Unit Implementation and Verification

* + - 1. Each software unit shall be implemented using the tools set out in the Configuration Management section of **SSI-QF-20B Software Development Plan.**
      2. Code review may be carried out as a way to verify unit implementation.
      3. Prior to beginning preparation for verification, a Design and Implementation review(s) **SSI-QF-10C Formal Design Review** shall be conducted to verify the architecture and ensure that the design has been appropriately documented and code developed in accordance with the requirements. The minimum required attendees for these reviews are the Software Engineer and an Independent Reviewer.
      4. Software Unit Verification Process (Class B)

Strategies, methods and procedures for verifying each software unit shall be established.

Where verification is done by testing, the test procedures shall be evaluated for correctness.

FDA requirements for Moderate Level of Concern require a summary list of validation and verification activities and the results of these activities at the unit level. (Not required if software is of FDA Minor Level of Concern). This may include details of formal and informal testing at a unit level.

* + - 1. Acceptance Criteria (Class B)

Acceptance criteria for software units shall be established prior to integration into larger software items as appropriate.

It shall also be ensured that software units meet acceptance criteria.

Examples of acceptance criteria are:

– does the software code implement requirements including risk control measures?

– is the software code free from contradiction with the interfaces documented in the detailed design of the software unit

– does the software code conform to programming procedures or coding standards?

### Software Integration and Integration Testing (Class B)

* + - 1. Software units shall be integrated in accordance with the integration plan (as described in the **Software Development Plan SSI-QF-20B)**.
      2. The following shall be verified and recorded:
         * The software units have been integrated into software items and the software system, and
         * The hardware items, software items, and support for manual operations (e.g., human equipment interface, on-line help menus) of the system have been integrated into the system.
      3. This verification may be conducted as part of the software release process.
      4. The integrated software items shall be tested per the integration plan and results documented.
      5. Integrated software may be tested informally at this stage, but formal testing shall be carried out at a system level. (Note 1 of BS EN 62304:2006+A1:2015 Section 5.1.5 allows the combination of integration testing and software system testing into a single plan and set of activities.)
      6. When software items are integrated, regression testing shall be conducted, as appropriate to demonstrate that defects have not been introduced into previously integrated software.
      7. Test Record Requirements

Test result documentation (pass/fail and a list of anomalies);

Retention of sufficient records to permit the test to be repeated. This could be implemented by retaining, for e.g.:

- test case specifications showing required actions and expected results;

- records of the equipment;

- records of the test environment (including software tools) used for test.

Identification of the tester.

Any anomalies found during software integration and integration testing shall be entered into a software problem resolution process.

* + - 1. Software test protocol and their results shall be documented using **SSI-QF-20E Software Test Protocol** and reports shall be documented using **SSI-QF-20F Software Test Report.**
      2. FDA requirements for Moderate level of concern require a summary list of validation and verification activities and the results of these activities at the integration and system level. (Not required if software is of FDA Minor Level of Concern).

### Software System Testing (Class B)

* + - 1. Software system testing shall be conducted and suitable tests established with input stimuli and expected outcomes, and with pass/fail criteria to ensure that all of the requirements of the system requirements and **SSI-QF-20C Software Requirement Traceability** are met. The tests should also ensure that integrated software items (including SOUP) are performing correctly.
      2. Verification shall be conducted to ensure that:
         * The verification strategies and the test procedures used are appropriate;
         * software system test procedures trace to software requirements;
         * all software requirements have been tested or otherwise verified; and
         * test results meet the required pass/fail criteria.
      3. The verification shall be conducted as part of the software release process.
      4. System Test Record Contents:
         * Test result (pass/fail and a list of anomalies);
         * Sufficient records to permit the test to be repeated. This could be implemented by retaining, for example:
         * test case specifications showing required actions and expected results;
         * records of the equipment; and
         * records of the test environment (including software tools) used for test.
         * Identify the tester.
      5. Software test protocols and their results shall be documented using **SSI-QF-20E Software Test Protocol** and reports shall be documented using **SSI-QF-20F Software Test Report.**
      6. If any changes are made to any software units as a result of the testing process, additional testing shall be carried out as appropriate to test and verify effectiveness of the change in correcting the problem and that unintended side effects have not been introduced. Relevant risk management activities shall be carried out.

### Security Testing

* + - 1. Security verification testing shall be conducted, as required, to ensure that the code is free of significant known vulnerabilities and that security controls have been effectively implemented. This includes taking into consideration where and by whom the device will be used.
      2. Specific security testing methods that can be used are as follows:
         * Periodic security testing such as static code analysis, dynamic analysis, robustness testing, vulnerability scanning, or software composition analysis to identify known vulnerabilities or software weaknesses.
         * Technical security analyses such as penetration testing to identify unknown vulnerabilities through fuzz testing or checks for alternative entry points.
         * Vulnerability assessment to analyze the effect of vulnerabilities and identify countermeasures for remediation and mitigation.
      3. Security Test Record Contents:
         1. Test methods used to test the security of the device, including any tools, techniques, or procedures used.
         2. Results of the testing, including any vulnerabilities discovered, as well as the severity and effect of those vulnerabilities.
         3. Conclusions and an overall assessment of the security of the device, based on the results of the testing. Include any recommendations for improving the security of the device.
      4. References to standards and internal SOPs used in the testing process.

### Software Release

* + - 1. Before a new build is released verification must have been completed and the results evaluated through a review involving an Independent Reviewer using **SSI-QF-10C** **Formal Design Review**. This review shall also:
         * Document all known residual anomalies.
         * Ensure that all known residual anomalies have been evaluated to ensure that they do not contribute to an unacceptable risk.
         * Ensure that all activities and tasks are complete along with all the associated documentation.
      2. The conclusions of the software release process shall be documented using the Software Summary Report (**SSI-QF-20I Software Summary Report**). **This is for FDA only.**
      3. New Build numbers and release dates shall be recorded in **SSI-QF-20J Software Release Version Control** to which **SSI-QF-20L Summary of Changes** recording of the main changes in each new build will be attached.
      4. Software code and supporting documents are stored on Stowood’s share drive. . These shall continue to be stored for at least the life of the product unless regulation requires longer storage.
      5. Procedures shall be established to ensure that the software product can be reliably delivered to the point of use without corruption or unauthorized change. These procedures address the production and handling of media containing the software product including as appropriate replication, media labelling, packaging, protection, storage, and delivery.
      6. A software description document shall be created including a summary overview of the features and software operating environment. (FDA Guidance). **For Visi-Download this is fulfilled by the Visi-Download Product Information Sheet.**

## Software Maintenance Process

### Software Maintenance Plan

* + - 1. A software maintenance plan (shall be documented for conducting the activities and tasks of the maintenance process using **SSI-QF-20G Software Maintenance Plan**. The plan shall address the following:
         * Procedures for receiving, documenting, evaluating, resolving and tracking feedback arising after release of the medical device software
         * criteria for determining whether feedback is considered to be a problem
         * use of the software risk management process
         * use of the software problem resolution process for analysing and resolving problems arising after release of the medical device software
         * use of the software configuration management process for managing modifications to the existing system; and
         * procedures to evaluate and implement upgrades, bug fixes, patches and obsolescence of SOUP.
         * All of the above are captured on **SSI-QF-10P Problem Report.**
      2. The results of the maintenance activities shall be captured once annually on **SSI-QF-20H Software Maintenance Report.**

### Cybersecurity Management Plan

* + - 1. A cybersecurity management plan (which may be part of **SSI-QF-20G Software Maintenance Plan**) shall be put in place, as required, to address the following key elements:
         * Total product life cycle (TPLC) Vigilance: monitoring and identifying newly discovered cybersecurity vulnerabilities, assessing their threat, and appropriately responding to them.
         * Vulnerability disclosure: process for gathering information from vulnerability finders, developing mitigation and remediation strategies, and disclosing the existence of vulnerabilities and mitigation or remediation approaches to stakeholders
         * Updates, remediation and recovery: software update and remediation actions in response to identified vulnerabilities as well as restoring the device to its normal operating condition following a cybersecurity incident.
         * Information sharing: participation in Information Sharing Analysis Organizations (ISAOs) or Information Sharing and Analysis Centers (ISACs) that promote the communication and sharing of updated information about security threats and vulnerabilities.

### Problem and modification analysis

* + - 1. Feedback about software may come from many sources, including patients and clinicians.
      2. Feedback shall be documented and evaluated using **SSI-QF-10P Problem Report** to determine whether a problem exists in a released software product. Any such problem shall be recorded on **SSI-QF-10P**, classified, and processed using that form. Opportunities for functional improvements are not non-conformities.
      3. Each problem shall be evaluated to determine if it affects the safety of a released software and whether a change to the released software is needed to address the problem, the problem should be monitored, changes should be considered for the next build number or an immediate change is required to resolve a serious issue.
      4. The software problem resolution process shall be used to address non-conformances related to software. When this activity has been done, any change of safety class in the software system or its software items should be considered.
      5. Each proposed change shall be analysed for its effect on the organisation, released software, and systems with which it interfaces using **SSI-QF-10G Design Change Record.**
      6. Changes which modify released software products shall be evaluated and approved. Refer to **SSI-SOP-10** **Design and Development** for further details on this process.
      7. Changes shall be verified by testing to ensure that they have been properly integrated and have not introduced any additional problems.
      8. Testing of changes to software shall be carried out and recorded in the same way as testing for software development.
      9. As required by local regulation, users and regulators may need to be informed about:
         * any problem in released software products and the consequences of continued unchanged use; and
         * the nature of any available changes to released software products and how to obtain and install the changes.

### Modification Implementation

* + - 1. The software development process shall be used to implement the required modifications.
      2. Modified software shall be released in accordance with software release processes.

## Software Risk Management Process

The software risk assessment as documented below shall be documented using **SSI-QF-13E** **Design Failure Mode and Effects Analysis.**

* + 1. **Analysis of software contributing to hazardous situations [Class B]**
       1. Software items that could contribute to a hazardous situation shall be identified in the risk management file. (The hazardous situation could be the direct result of software failure or due to failure of a risk control measure implemented in software).
       2. Potential causes of the software item contributing to a hazardous situation shall be identified. Potential causes that should be considered are:
          - incorrect or incomplete specification of functionality;
          - software defects in the identified software item functionality;
          - failure or unexpected results from SOUP;
          - hardware failures or other software defects that could result in unpredictable software operation; and
          - reasonably foreseeable misuse.
       3. If failure or unexpected results from SOUP is a potential cause of the software item contributing to a hazardous situation, the anomaly list published by the supplier of the SOUP item relevant to the version of the SOUP item used in the medical device shall be evaluated to determine if any of the known anomalies result in a sequence of events that could result in a hazardous situation. (FDA OTS Software Guidance).

### Risk Control Measures

* + - 1. For each potential cause of the software item contributing to a hazardous situation documented in the risk management file, define and document risk control measures. The risk control measures can be implemented in hardware, software, the working environment or user instruction.
      2. If a risk control measure is implemented as part of the functions of a software item:
         * include the risk control measure in the software requirements;
         * assign a software safety class to the software item based on the possible effects of the hazard that the risk control measure is controlling; and
         * develop the software item in accordance with this procedure

### Verification of Risk Control Measures (Class B)

* + - 1. The implementation of each risk control measure shall be verified. Risk control measures should be reviewed to determine that it could not result in a new hazardous situation.

### Risk Management of software changes

* + - 1. Any changes to the medical device software (including SOUP) shall be analysed to determine if any additional causes that could contribute to a hazardous situation are introduced, and whether any additional risk control measures are required.

## **Software Configuration Management Process**

“The software configuration management process is a process of applying administrative and technical procedures throughout the software life cycle to identify and define software items, including documentation, in a system; controlling modifications and releases of the items; and documenting and reporting the status of the items and change requests. Software configuration management is necessary to recreate a software item, to identify its constituent parts, and to provide a history of the changes that have been made to it.”

(BS EN 62304:2006+A1:2015 Annex B, Section B.8)

The intention of this process is to ensure that released products are not modified without authority, and that versions of the product can be recreated.

### Configuration identification

* + - 1. A scheme shall be established for the unique identification of configuration items and their versions to be controlled for the project. This scheme shall include other software products or entities such as SOUP and documentation.
      2. For peripheral interface controller (PIC) microcontroller based software, configuration management shall be carried out on the software at a hex level (i.e. at the level of a single item to be programmed into the PIC). These items shall include a unique identification such as a release version or build number that can either be displayed on the products display, or read by using an external connection.
      3. For each SOUP configuration item being used, including standard libraries. Using **SSI-QF-20O** **Documenting SOUP (or OTS – off the shelf) Software,** the following shall be documented:
         * the title,
         * the manufacturer, and
         * the unique SOUP designator of each SOUP configuration item being used. The unique SOUP designator could be, for example, a version, a release date, a patch number or an upgrade designation.
      4. The set of configuration items and their versions that comprise the software system configuration shall also be documented.

### Change Control

* + - 1. Configuration items shall be changed after an **SSI-QF-10G Design Change Record** has been approved. Refer to **SSI-SOP-10** for details on change control process.
      2. Where changes are implemented, activities that need to be repeated shall be identified and performed. This includes carrying out any additional activities which result from a change in the software safety class of the software system or software items.
      3. Verification of the change shall be carried out, including repeating any verification which has been invalidated by the change. **SSI-QF-10G Design Change Record** shall be used for recording software changes.

## Software Problem Resolution Process

* + 1. Any problem detected in the released software will be recorded on **SSI-QF-10P Problem Report** and processed accordingly**.** A CAPA will not be raised unless the problem entails a significant increase in risk or a new risk.
    2. Problems shall be investigated by following **SSI-QF-10P** to identify the cause and apply risk management, if required. The outcome and evaluation of the investigation, actions taken, changes made, and their rationale are recorded on **SSI-QF-10P.**
    3. Where appropriate, relevant parties shall be advised of the problem.
    4. The change control process of **SSI-SOP-10** shall be followed.
    5. Records of problem reports and their resolution shall be maintained on **SSI-QF-10P**. The Risk Management File shall be updated as appropriate.
    6. Problems Reported on **SSI-QF-10P** shall be reviewed and analysed as part of the product post market surveillance.
    7. Resolutions shall be verified to determine whether:
       - the problem has been resolved;
       - any trends have been noted and, if adverse, reversed;
       - change requests have been implemented where required;
       - any new or additional risks have been introduced and
       - any additional problems have been introduced.
    8. These steps are recorded on **SSI-QF-10P** When testing, retesting or regression testing software items and systems following a change, include the following in the test documentation:
       - test results;
       - anomalies found;
       - the build number of software tested;
       - relevant hardware and software test configurations;
       - relevant test tools;
       - date tested; and
       - identification of the tester.

# Firmware

**7.1** Firmware used in SSI-manufactured devices will be subject to proportionate similar processes as for software set out in this **SOP-20**. On each new firmware release **SSI-QF-20M Firmware Programmable Part Release** will be completed setting out the changes made and their rationale.

# Records

* 1. All records such as meeting minutes, nonconformance reports, change requests and test documentation shall be maintained in accordance with **SSI-SOP-09 Records Control**. All documents shall be maintained in accordance with **SSI-SOP-8 Document Control**.

# Authorisation to train others and to modify SOP.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Who is authorised to train others (add to training record) | | | Authorised to modify SWI (add to training record) | | |
| **Who is authorised to train** | **Authorised by** | **Date authorised** | **Who is authorised to modify** | **Authorised by** | **Date authorised** |
| GJGD | GJGD | 1/1/23 | GJGD | GJGD | 1/1/23 |
| JF | GJGD | 1/1/23 | JF | GJGD | 1/1/23 |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Initial training | | Training on document. Mark date of training under the issue number of document you are trained on. | | | | | | |
| **Major change? If yes add to training log** | | **Y** | **Y** |  |  |  |  |  |
| **Trainee** | **Trainer** | **1** | **2** |  |  |  |  |  |
| Gwilym Davies | GJGD | 7/7/23 | 20/06/24 |  |  |  |  |  |
| Paul Gladding | GJGD | 7/7/23 | 20/06/24 |  |  |  |  |  |
| Jerome Oswald-Jones | GJGD | 7/7/23 | left |  |  |  |  |  |