VERIFICATION REPORT
IEC 62304:2006

Medical device software -
Software life cycle processes

Report Reference No	20110915
Compiled by (+ signature)	Markus Weber
Reviewed by (+ signature)	
Approved by (+ signature)	
Date of issue	09/15/2011
Verification laboratory	System Safety, Inc.
Address	5150 Corte Playa Catalina
	San Diego, CA 92124-1558
Verification location	
Applicant	
Address	
Standard	IEC 62304:2006-05
Test Report Form No	03092020
Test procedure	Audit / Review
Procedure deviation	None
Non-standard test method	None
Type of end product	
End product Trademark	
End product Model and/or type reference	
End product Manufacturer	
End product Address	See above
End product Rating(s)	

PEMS/PESS Configuration Information:	No special hardware configuration necessary.
Software Designer (if different than end Product manufacturer).	NA
Address	NA
	NA
Method of Identification of Software:	Revision
Particular Risks Addressed by Software:	As contained in hazard analyses

System Safety Inc.
Safety Livecycle Services

GENERAL INFORMATION

Particulars: verification item vs. verification requirements
No.: N.N.

Possible verification case verdicts

Verification case does not apply to the verification item ---------- : **N**(ot)/**A**(pplicable) or
: **N**(ot)/**A**(ssessed)
Verification item is available ---------- : **N**(oted)
Verification item does meet the requirement ---------- : **P**(ass)
Verification item does meet the requirement under the limited scope of this assessment ------- : **P**(ass) **L**(imited Scope)
Verification item does not meet the requirement ---------- : **F**(ail)
Verification item does not meet the requirement under the limited scope of this assessment -- : **F**(ail) **L**(imited Scope)

Minor non-compliances are noted in regular case and font
Major non-compliances are note in **ALL CAPS** and / or **BOLD**

General remarks

‘’(See enclosure #)" refers to an enclosure appended to this report.
"(See appended table)" refers to a table appended to the report.
Throughout this report a period is used as the decimal separator.
The verification results presented in this report relate only to the item being verified.
This verification report shall not be reproduced except in full without the written approval of the verification laboratory.

SUMMARY OF CONTENTS:

The equipment has been evaluated according to standard IEC 62304:2006-05.

All applicable verifications according to the above-specified standard(s) have been carried out; however the scope was limited to sub-system evaluation.

These verifications fulfil the requirements of standard EN 45001.

Note: As per IEC 62304:2006-05, determination of compliance is by inspection and audit, the attachments should be documents or parts of documents.

Acronyms and Abbreviations:

COTS	Commercial of the shelf software
DFU	Directions for Use
H&RA	Hazard and Risk Analysis
MDD	European Medical Device Directive
PEMS	Programmable Electronic Medical Devices
RMP	Risk Management Plan
SOP	Standard Operating Procedure
SOUP	Software of Unknown Pedigree
V&V	Verification and Validation

Results and Conclusions:

System Safety Inc.
Safety Livecycle Services

Contents

Clause	Requirement	Result- Remark	ABC	Verdict
4.	**General requirements**			
4.1	**Quality management system**			
	The MANUFACTURER of MEDICAL DEVICE SOFTWARE shall demonstrate the ability to provide MEDICAL DEVICE SOFTWARE that consistently meets customer requirements and applicable regulatory requirements *NOTE 1 Demonstration of this ability can be by the use of a quality management system that complies with:* - *ISO 13485]; or* - *a national quality management system standard; or* - *a quality management system required by national regulation.* *NOTE 2 Guidance for applying quality management system requirements to software can be found in ISO / IEC 90003.*			
4.2	**RISK MANAGEMENT**			
	The MANUFACTURER shall apply a RISK MANAGEMENT PROCESS complying with ISO 14971			
4.3	**Software safety classification**			
	a) The MANUFACTURER shall assign to each SOFTWARE SYSTEM a software safety class (A, B, or C) according to the possible effects on the patient, operator, or other people resulting from a HAZARD to which the SOFTWARE SYSTEM can contribute. The software safety classes shall initially be assigned based on severity as follows: - 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 If the 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 (as defined by ISO 14971) by a hardware 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 hardware RISK CONTROL measure, the software safety classification may be reduced from B to A. b) The MANUFACTURER shall assign to each SOFTWARE SYSTEM that contributes to the implementation of a RISK CONTROL measure a software safety class based on the possible effects of the HAZARD that the RISK CONTROL measure is control-			

Clause	Requirement	Result- Remark	ABC	Verdict
	ling. c) The MANUFACTURER shall document the software safety class assigned to each SOFTWARE SYSTEM in the RISK MANAGEMENT FILE. d) When a SOFTWARE SYSTEM is decomposed into SOFTWARE ITEMS, and when a SOFTWARE ITEM is decomposed into further SOFTWARE ITEMS, such SOFTWARE ITEMS shall inherit the software safety classification of the original SOFTWARE ITEM (or SOFTWARE SYSTEM) unless the MANUFACTURER documents a rationale for classification into a different software safety class. Such a rationale shall explain how the new SOFTWARE ITEMS are segregated so that they may be classified separately. e) The MANUFACTURER shall document the software safety class of each SOFTWARE ITEM if that class is different from the class of the SOFTWARE ITEM from which it was created by decomposition. f) For compliance with this standard, 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 MANUFACTURER shall use the PROCESSES and TASKS which are required by the classification of the highest-classified SOFTWARE ITEM in the group unless the MANUFACTURER documents in the RISK MANAGEMENT FILE a rationale for using a lower classification. g) For each SOFTWARE SYSTEM, until a software safety class is assigned, Class C requirements shall apply.			

Clause	Requirement	Result- Remark	ABC	Verdict
5.	**Software development PROCESS - General Requirements**			
5.1	**Software development planning**			
5.1.1	**Software development plan** The MANUFACTURER shall establish a software development plan (or plans) for conducting the ACTIVITIES of the software development PROCESS appropriate to the scope, magnitude, and software safety classifications of the SOFTWARE SYSTEM to. Be developed. The SOFTWARE DEVELOPMENT LIFE CYCLE MODEL shall either be fully defined or be referenced in the plan (or plans). The plan shall address the following: a) the PROCESSES to be used in the development of the SOFTWARE SYSTEM (see Note 4); b) the DELIVERABLES (includes documentation) of the ACTIVITIES and TASKS; c) TRACEABILITY between SYSTEM requirements, software requirements, SOFTWARE SYSTEM d) test, and RISK CONTROL measures implemented in software; e) software configuration and change management, including SOUP CONFIGURATION ITEMS and software used to support development; and f) software problem resolution for handling problems detected in the SOFTWARE PRODUCTS, DELIVERABLES and ACTIVITIES at each stage of the life cycle. *NOTE 1 The SOFTWARE DEVELOPMENT LIFE CYCLE MODEL can identify different elements (PROCESSES, ACTIVITIES,TASKS and DELIVERABLES) for different SOFTWARE ITEMS according to the software safety classification of each SOFTWARE ITEM of the SOFTWARE SYSTEM.* *NOTE 2 These ACTIVITIES and TASKS can overlap or interact and can be performed iteratively or recursively. It is notthe intent to imply that a specific life cycle model should be used.* *NOTE 3 Other PROCESSES are described in this standard separately from the development PROCESS. This does not imply that they must be implemented as separate ACTIVITIES and TASKS. The ACTIVITIES and TASKS of the other PROCESSES can be integrated into the development PROCESS.* *NOTE 4 The software development plan can reference existing PROCESSES or define new ones.* *NOTE 5 The software development plan may be integrated in an overall SYSTEM development plan. NOTE 1 Refer to Annex F for guidance on developing a risk management plan.*			
5.1.2	**Keep software development plan updated** The MANUFACTURER shall update the plan as development proceeds as appropriate			

Clause	Requirement	Result- Remark	A B C	Verdict
5.1.3	**Software development plan reference to SYSTEM design and development** a) As inputs for software development, SYSTEM requirements shall be referenced in the software development plan by the MANUFACTURER. b) The MANUFACTURER shall include or reference in the software development plan procedures for coordinating the software development and the design and development validation necessary to satisfy 4.1.. *NOTE There might not be a difference between SOFTWARE SYSTEM requirements and SYSTEM requirements if the SOFTWARE SYSTEM is a stand-alone SYSTEM (software-only device).*			
5.1.4	**Software development standards, methods and tools planning** The MANUFACTURER shall include or reference in the software development plan: a) standards, b) methods, and c) tools associated with the development of SOFTWARE ITEMS of class C. *[Class C]*			
5.1.5	**Software integration and integration testing planning** The MANUFACTURER shall include or reference in the software development plan, a plan to integrate the SOFTWARE ITEMS (including SOUP) and perform testing during integration. *[Class B,C]* *NOTE It is acceptable to combine integration testing and SOFTWARE SYSTEM testing into a single plan and set of ACTIVITIES.*			
5.1.6	**Software VERIFICATION planning** The MANUFACTURER shall include or reference in the software development plan the following VERIFICATION information: a) DELIVERABLES requiring VERIFICATION; b) the required VERIFICATION TASKS for each life cycle ACTIVITY; c) milestones at which the DELIVERABLES are VERIFIED; and d) d) the acceptance criteria for VERIFICATION of the DELIVERABLES			
5.1.7	**Software RISK MANAGEMENT planning** The MANUFACTURER shall include or reference in the software development plan, a plan to conduct the ACTIVITIES and TASKS of the software RISK MANAGEMENT PROCESS, including the management of RISKS relating to SOUP *NOTE See Clause 7.*			

Clause	Requirement	Result- Remark	ABC	Verdict
5.1.8	**Documentation planning** The MANUFACTURER shall include or reference in the software development plan information about the documents to be produced during the software development life cycle. For each identified document or type of document the following information shall be included or referenced: a) title, name or naming convention; b) purpose; c) intended audience of document; and d) procedures and responsibilities for development, review, approval and modification.			
5.1.9	**Software configuration management planning** The MANUFACTURER shall include or reference software configuration management information in the software development plan. The software configuration management information shall include or reference: a) the classes, types, categories or lists of items to be controlled; b) the software configuration management ACTIVITIES and TASKS; c) the organization(s) responsible for performing software configuration management and ACTIVITIES; d) their relationship with other organizations, such as software development or maintenance; e) when the items are to be placed under configuration control; and f) when the problem resolution PROCESS is to be used.			
5.1.10	**Supporting items to be controlled** The items to be controlled shall include tools, items or settings, used to develop the MEDICAL DEVICE SOFTWARE, which could impact the MEDICAL DEVICE SOFTWARE. *[Class B, C]* *NOTE Examples of such items include compiler/assembler versions, make files, batch files, and specific environment settings.*			
5.1.11	**Software CONFIGURATION ITEM control before VERIFICATION** The MANUFACTURER shall plan to place CONFIGURATION ITEMS under documented configuration management control before they are VERIFIED. *[Class B, C]*			

Clause	Requirement	Result- Remark	ABC	Verdict
5.2	**Software requirements analysis**			
5.2.1	**Define and document software requirements from SYSTEM requirements** For each SOFTWARE SYSTEM of the MEDICAL DEVICE, the MANUFACTURER shall define and document SOFTWARE SYSTEM requirements from the SYSTEM level requirements *NOTE There might not be a difference between SOFTWARE SYSTEM requirements and SYSTEM requirements if the SOFTWARE SYSTEM is a stand-alone SYSTEM (software-only device).*			
5.2.2	**Software requirements content** As appropriate to the MEDICAL DEVICE SOFTWARE, the MANUFACTURER shall include in the software requirements: a) functional and capability requirements; *NOTE 1 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, and* - *need for compatibility with upgrades or multiple SOUP or other device versions.* b) SOFTWARE SYSTEM inputs and outputs; *NOTE 2 Examples include:* - *data* - *characteristics (e.g., numerical, alpha-numeric, format)* - *ranges,* - *limits, and* - *defaults.* c) interfaces between the SOFTWARE SYSTEM and other SYSTEMS; d) software-driven alarms, warnings, and operator messages; e) SECURITY requirements; *NOTE 3 Examples include:* *those related to the compromise of sensitive information,* - *authentication,* - *authorization,* - *audit trail, and* - *communication integrity.* f) usability engineering requirements that are sensitive to human errors and training; *NOTE 4 Examples include those related to:* - *support for manual operations,* - *human-equipment interactions,* - *constraints on personnel, and* - *areas needing concentrated human attention.* *NOTE 5 Information regarding usability engineering requirements can be found in IEC 60601-1-6.* g) data definition and database requirements; *NOTE 6 Examples include:* - *form;* - *fit;* - *function.* h) installation and acceptance requirements of the delivered MEDICAL DEVICE SOFTWARE at the operation			

Clause	Requirement	Result- Remark	ABC	Verdict
	and maintenance site or sites; i) requirements related to methods of operation and maintenance; j) user documentation to be developed; k) user maintenance requirements; l) and regulatory requirements. *NOTE 7 All of these requirements might not be available at the beginning of the software development.* *NOTE 8 ISO/IEC 9126-1 provides information on quality characteristics that may be useful in defining software requirements.*			
5.2.3	**Include RISK CONTROL measures in software requirements** The MANUFACTURER shall include RISK CONTROL measures implemented in software for hardware failures and potential software defects in the requirements as appropriate to the MEDICAL DEVICE SOFTWARE. *NOTE These requirements might not be available at the beginning of the software development and can change as the software is designed and RISK CONTROL measures are further defined*			
5.2.4	**Re-EVALUATE MEDICAL DEVICE RISK ANALYSIS** The MANUFACTURER shall re-EVALUATE the MEDICAL DEVICE RISK ANALYSIS when software requirements are established and update it as appropriate.			
5.2.5	**Update SYSTEM requirements** The MANUFACTURER shall ensure that existing requirements, including SYSTEM requirements, are re-EVALUATED and updated as appropriate a$ a result of the software requirements analysis ACTIVITY.			
5.2.6	**Verify software requirements** The MANUFACTURER shall verify and document that the software requirements: a) implement SYSTEM requirements including those relating to RISK CONTROL; b) do not contradict one another; c) are expressed in terms that avoid ambiguity; d) are stated in terms that permit establishment of test criteria and performance of tests to determine whether the test criteria have been met; e) can be uniquely identified; and f) are traceable to SYSTEM requirements or other source. *NOTE This standard does not require the use of a formal specification language*			

Clause	Requirement	Result- Remark	A	B	C	Verdict
5.3	**Software ARCHITECTURAL design**					
5.3.1	**Transform software requirements into an ARCHITECTURE** The MANUFACTURER shall transform the requirements for the MEDICAL DEVICE SOFTWARE into a documented ARCHITECTURE that describes the software's structure and identifies the SOFTWARE ITEMS. *[Class B, C]*					
5.3.2	**Develop an ARCHITECTURE for the interfaces of SOFTWARE ITEMS** The MANUFACTURER shall develop and document an ARCHITECTURE for the interfaces between the SOFTWARE ITEMS and the components external to the SOFTWARE ITEMS (both software and hardware), and between the SOFTWARE ITEMS. *[Class B, C]*					
5.3.3	**Specify functional and performance requirements of SOUP item** If a SOFTWARE ITEM is identified as SOUP, the MANUFACTURER shall specify functional and performance requirements for the SOUP item that are necessary for its intended use. *[Class B, C]*					
5.3.4	**Specify SYSTEM hardware and software required by SOUP item** If a SOFTWARE ITEM is identified as SOUP, the MANUFACTURER shall specify the SYSTEM hardware and software necessary to support the proper operation of the SOUP item. *[Class B, C]* *NOTE Examples include processor type and speed, memory type and size, SYSTEM software type, communication and display software requirements.*					
5.3.5	**Identify segregation necessary for RISK CONTROL** The MANUFACTURER shall identify the segregation between SOFTWARE ITEMS that is essential to RISK CONTROL, and state how to ensure that the segregation is effective. *[Class C]* *NOTE An example of segregation is to have SOFTWARE ITEMS execute on different processors. The effectiveness of the segregation can be ensured by having no shared resources between the processors.*					

Clause	Requirement	Result- Remark	A	B	C	Verdict
5.3.6	**Verify software ARCHITECTURE** The MANUFACTURER shall verify and document that: a) the ARCHITECTURE of the software implements SYSTEM and software requirements including those relating to RISK CONTROL; b) the software ARCHITECTURE is able to support interfaces between SOFTWARE ITEMS and between SOFTWARE ITEMS and hardware; and c) the MEDICAL DEVICE ARCHITECTURE supports proper operation of any SOUP items. *[Class B, C]*					
5.4	**Software detailed design**					
5.4.1	**Refine SOFTWARE ARCHITECTURE into SOFTWARE UNITS** The MANUFACTURER shall refine the software ARCHITECTURE until it is represented by SOFTWARE UNITS. *[Class B, C]*					
5.4.2	**Develop detailed design for each SOFTWARE UNIT** The MANUFACTURER shall develop and document a detailed design for each SOFTWARE UNIT of the SOFTWARE ITEM. *[Class C]*					
5.4.3	**Develop detailed design for interfaces** The MANUFACTURER shall develop and document a detailed design for any interfaces between the SOFTWARE UNIT and external components (hardware or software), as well as any interfaces between SOFTWARE UNITS. *[Class C]*					
5.4.4	**Verify detailed design** The MANUFACTURER shall verify and document that the software detailed design: a) implements the software ARCHITECTURE; and b) is free from contradiction with the software ARCHITECTURE. *[Class C]*					

Clause	Requirement	Result- Remark	A	B	C	Verdict
5.5	**SOFTWARE UNIT implementation and verification**					
5.5.1	**Implement each SOFTWARE UNIT** The MANUFACTURER shall implement each SOFTWARE UNIT					
5.5.2	**Establish SOFTWARE UNIT VERIFICATION PROCESS** The MANUFACTURER shall establish strategies, methods and procedures for verifying each SOFTWARE UNIT. Where VERIFICATION is done by testing, the test procedures shall be EVALUATED for correctness. *[Class B, C]* *NOTE It is acceptable to combine integration testing and SOFTWARE SYSTEM testing into a single plan and set of ACTIVITIES.*					
5.5.3	**SOFTWARE UNIT acceptance criteria** The MANUFACTURER shall establish acceptance criteria for SOFTWARE UNITS prior to integration into larger SOFTWARE ITEMS as appropriate, and ensure that SOFTWARE UNITS meet acceptance criteria *[Class B, C]* *NOTE 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?*					
5.5.4	**Additional SOFTWARE UNIT acceptance criteria** When present in the design, the MANUFACTURER shall include additional acceptance criteria as appropriate for: a) proper event sequence; b) data and control flow; c) planned resource allocation; d) fault handling (error definition, isolation, and recovery); e) initialization of variables; f) self-diagnostics; g) memory management and memory overflows; and h) boundary conditions. *[Class C]*					
5.5.5	**SOFTWARE UNIT VERIFICATION** The MANUFACTURER shall perform the SOFTWARE UNIT VERIFICATION and document the results. *[Class B, C]*					

Clause	Requirement	Result- Remark	ABC	Verdict
5.6	**SOFTWARE UNIT implementation and verification**			
5.6.1	**Integrate SOFTWARE UNITS** The MANUFACTURER shall integrate the SOFTWARE UNITS in accordance with the integration plan (see 5.1.5). *[Class B, C]*			
5.6.2	**Verify software integration** The MANUFACTURER shall verify and record the following aspects of the software integration in accordance with the integration plan (see 5.1.5): a) the SOFTWARE UNITS have been integrated into SOFTWARE ITEMS and the SOFTWARE SYSTEM; and b) the hardware items, SOFTWARE ITEMS, and support for manual operations (e.g., human equipment interface, on-line help menus, speech recognition, voice control) of the SYSTEM have been integrated into the SYSTEM. *[Class B, C]* *NOTE This VERIFICATION is only that the items have been integrated according to the plan, not that they perform as intended. This VERIFICATION is most likely implemented by some form of inspection.*			
5.6.3	**Test integrated software** The MANUFACTURER shall test the integrated SOFTWARE ITEMS in accordance with the integration plan (see 5.1.5) and document the results. *[Class B, C]*			
5.6.4	**Integration testing content** For software integration testing, the MANUFACTURER shall address whether the integrated SOFTWARE ITEM performs as intended. *[Class B, C]* *NOTE 1 Examples to be considered are:* - *the required functionality of the software;* - *implementation of RISK CONTROL measures;* - *specified timing and other behavior;* - *specified functioning of internal and external interfaces; and* - *testing under abnormal conditions including foreseeable misuse.* *NOTE 2 It is acceptable to combine integration testing and SOFTWARE SYSTEM testing into a single plan and set of ACTIVITIES.*			
5.6.5	**Verify integration test procedures** The MANUFACTURER shall EVALUATE the integration test procedures for correctness. *[Class B, C]*			

Clause	Requirement	Result- Remark	ABC	Verdict
5.6.6	**Conduct regression tests** When software items are integrated, the MANUFACTURER shall conduct REGRESSION TESTING appropriate to demonstrate that defects have not been introduced into previously integrated software. *[Class B, C]*			
5.6.7	**Integration test record contents** The MANUFACTURER shall: a) document the test result (pass/fail and a list of ANOMALIES); b) retain sufficient records to permit the test to be repeated; and c) identify the tester. *[Class B, C]* *NOTE Requirement b) could be implemented by retaining, for example:* *- test case specifications showing required actions and expected results;* *- records of the equipment;* *- records of the test environment (including software tools) used for test.*			
5.6.8	**Use software problem resolution PROCESS** The MANUFACTURER shall enter ANOMALIES found during software integration and integration testing into a software problem resolution PROCESS. *[Class B, C]* *NOTE See Clause 9.*			
5.7	**SOFTWARE SYSTEM testing**			
5.7.1	**Establish tests for software requirements** The MANUFACTURER shall establish and perform a set of tests, expressed as input stimuli, expected outcomes, pass/fail criteria and procedures, for conducting SOFTWARE· SYSTEM testing, such that all software requirements are covered. *[Class B, C]* *NOTE 1 It is acceptable to combine integration testing and SOFTWARE SYSTEM testing into a single plan and set of ACTIVITIES. It is also acceptable to test software requirements in earlier phases.* *NOTE 2 Not only separate tests for each requirement, but also tests of combinations of requirements can be performed, especially if dependencies between requirements exist.*			
	Use software problem resolution PROCESS The MANUFACTURER shall enter ANOMALIES found during software system testing into a software problem resolution PROCESS. *[Class B, C]*			

Clause	Requirement	Result- Remark	A	B	C	Verdict
5.7.2	**Retest after changes** When changes are made during SOFTWARE SYSTEM testing, the MANUFACTURER shall: a) repeat tests, perform modified tests or perform additional tests, as appropriate, to verify the effectiveness of the change in correcting the problem; b) conduct testing appropriate to demonstrate that unintended side effects have not been introduced; and c) perform relevant RISK MANAGEMENT ACTIVITIES as defined in 7.4. *[Class B, C]*					
5.7.3	**Verify SOFTWARE SYSTEM testing** The MANUFACTURER shall verify that: a) the VERIFICATION strategies and the test procedures used are appropriate; b) SOFTWARE SYSTEM test procedures trace to software requirements; c) all software requirements have been tested or otherwise VERIFIED; and d) test results meet the required pass/fail criteria. *[Class B, C]*					
5.7.4	**SOFTWARE SYSTEM test record contents** The MANUFACTURER shall: a) document the test result (pass/fail and a list of ANOMALIES); b) retain sufficient records to permit the test to be repeated; and c) identify the tester. *[Class B, C]* *NOTE Requirement b) 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.*					
5.8	**SOFTWARE release**					
5.8.1	**Ensure software VERIFICATION is complete** The MANUFACTURER shall ensure that software VERIFICATION has been completed and the results EVALUATED before the software is released. *[Class B, C]*					
5.8.2	**Document known residual ANOMALIES** The MANUFACTURER shall document all known residual ANOMALIES. *[Class B, C]*					

Clause	Requirement	Result- Remark	A	B	C	Verdict
5.8.3	**EVALUATE known residual ANOMALIES** The MANUFACTURER shall ensure that all known residual ANOMALIES have been EVALUATED to ensure that they do not contribute to an unacceptable RISK. *[Class B, C]*					
5.8.4	**Document released VERSIONS** The MANUFACTURER shall document the VERSION of the SOFTWARE PRODUCT that is being released					
5.8.5	**Document how released software was created** The MANUFACTURER shall document the procedure and environment used to create the released software. *[Class B, C]*					
5.8.6	**Ensure activities and tasks are complete** The MANUFACTURER shall ensure that all ACTIVITIES and TASKS are complete along with all the associated documentation. *[Class B, C]*					
5.8.7	**Archive software** The MANUFACTURER shall archive: a) the SOFTWARE PRODUCT and CONFIGURATION ITEMS; and b) the documentation for at least a period of time determined as the longer of: the life time of the device as defined by the MANUFACTURER or a time specified by relevant regulatory requirements. *[Class B, C]*					
5.8.8	**Assure repeatability of software release** The MANUFACTURER shall establish procedures to ensure that the released SOFTWARE PRODUCT can be reliably delivered to the point of use without corruption or unauthorized change. These procedures shall address the production and handling of media containing the SOFTWARE PRODUCT including as appropriate: - replication, - media labeling, - packaging, - protection, - storage, and - delivery. *[Class B, C]*					

Clause	Requirement	Result- Remark	ABC	Verdict
6.	**Software maintenance PROCESS**			
6.1	**Establish software maintenance plan**			
6.1.1	The MANUFACTURER shall establish a software maintenance plan (or plans) for conducting the ACTIVITIES and TASKS of the maintenance PROCESS. The plan shall address the following: a) procedures for: - receiving, - documenting, - evaluating, - resolving and - tracking feedback arising after release of the MEDICAL DEVICE SOFTWARE; b) criteria for determining whether feedback is considered to be a problem; c) use of the software RISK MANAGEMENT PROCESS; d) use of the software problem resolution PROCESS for analyzing and resolving problems arising after release of the MEDICAL DEVICE SOFTWARE; e) use of the software configuration management PROCESS (Clause 8) for managing modifications to the existing SYSTEM; and f) procedures to EVALUATE and implement: - upgrades, - bug fixes, - patches and - obsolescence of SOUP.			
6.2	**Problem and modification analysis**			
6.2.1	**Document and EVALUATE feedback**			
6.2.1.1	**Monitor feedback** The MANUFACTURER shall monitor feedback on released SOFTWARE PRODUCT from both inside its own organization and from users			
6.2.1.2	**Document and EVALUATE feedback** Feedback shall be documented and EVALUATED to determine whether a problem exists in a released SOFTWARE PRODUCT. Any such problem shall be recorded as a PROBLEM REPORT (see Clause 9). PROBLEM REPORTS shall include actual or potential adverse events, and deviations from specifications			
6.2.1.3	**Evaluate PROBLEM REPORT'S effects on SAFETY** Each PROBLEM REPORT shall be EVALUATED to determine how it affects the SAFETY of a released SOFTWARE PRODUCT and whether a change to the released SOFTWARE PRODUCT is needed to address the problem.			

Clause	Requirement	Result- Remark	ABC	Verdict
6.2.2	**Use software problem resolution PROCESS** The MANUFACTURER shall use the software problem resolution PROCESS (see Clause 9) to address PROBLEM REPORTS. *NOTE When this ACTIVITY has been done, any change of safety class in the SOFTWARE SYSTEM or its SOFTWARE ITEMS should be known.*			
6.2.3	**Analyze CHANGE REQUESTS** In addition to the analysis required by Clause 9, the MANUFACTURER shall analyze each CHANGE REQUEST for its effect on the organization, released SOFTWARE PRODUCTS, and SYSTEMS with which it interfaces. *[Class B, C]*			
6.2.4	**CHANGE REQUEST approval** The MANUFACTURER shall EVALUATE and approve CHANGE REQUESTS which modify released SOFTWARE PRODUCTS			
6.2.5	**Communicate to users and regulators** The MANUFACTURER shall identify the approved CHANGE REQUESTS that affect released SOFTWARE PRODUCTS. As required by local regulation, the MANUFACTURER shall inform users and regulators about: a) any problem in released SOFTWARE PRODUCTS and the consequences of continued unchanged use; and b) the nature of any available changes to released SOFTWARE PRODUCTS and how to obtain and install the changes			
6.3	**Modification implementation**			
6.3.1	**Use established PROCESS to implement modification** The MANUFACTURER shall use the software development PROCESS (see Clause 5) or an established maintenance PROCESS to implement the modifications. *NOTE For requirements relating to RISK MANAGEMENT of software changes see 7.4.*			
6.3.2	**Re-release modified SOFTWARE SYSTEM** The MANUFACTURER shall release modified SOFTWARE SYSTEMS according to 5.8. Modifications may be released as part of a full re-release of a SOFTWARE SYSTEM or as a modification kit comprising changed SOFTWARE ITEMS and the necessary tools to install the changes as modifications to an existing SOFTWARE SYSTEM			

Clause	Requirement	Result- Remark	A	B	C	Verdict
7.	**Software RISK MANAGEMENT PROCESS**					
7.1	**Analysis of software contributing to hazardous situations**					
7.1.1	**Identify SOFTWARE ITEMS that could contribute to a hazardous situation** The MANUFACTURER shall identify SOFTWARE ITEMS that could contribute to a hazardous situation identified in the MEDICAL DEVICE RISK ANALYSIS ACTIVITY of ISO 14971 (see 4.2). *[Class B, C]* *NOTE The hazardous situation could be the direct result of software failure or the result of the failure of a RISK CONTROL measure that is implemented in software.*					
7.1.2	**Identify potential causes of contribution to a hazardous situation** The MANUFACTURER shall identify potential causes of the SOFTWARE ITEM identified above contributing to a hazardous situation. The MANUFACTURER shall consider potential causes including, as appropriate: a) incorrect or incomplete specification of functionality; b) software defects in the identified SOFTWARE ITEM functionality; c) failure or unexpected results from SOUP; d) hardware failures or other software defects that could result in unpredictable software operation; and e) reasonably foreseeable misuse. *[Class B, C]*					
7.1.3	**EVALUATE published SOUP ANOMALY lists** If failure or unexpected results from SOUP is a potential cause of the SOFTWARE ITEM contributing to a hazardous situation, the MANUFACTURER shall EVALUATE as a minimum any ANOMALY list published by the supplier of the SOUP item relevant to the VERSION of the SOUP item used in the MEDICAL DEVICE to determine if any of the known ANOMALIES result in a sequence of events that could result in a hazardous situation. *[Class B, C]*					
7.1.4	**Document potential causes** The MANUFACTURER shall document in the RISK MANAGEMENT FILE potential causes of the SOFTWARE ITEM contributing to a hazardous situation (see ISO 14971). *[Class B, C]*					

Clause	Requirement	Result- Remark	ABC	Verdict
7.1.5	**Document sequences of events** The MANUFACTURER shall document in the RISK MANAGEMENT FILE sequences of events that could result in a hazardous situation that are identified in 7.1.2. *[Class B, C]*			
7.2	**Analysis of software contributing to hazardous situations**			
7.2.1	**Define RISK CONTROL measures** For each potential cause of the software item contributing to a hazardous situation documented in the risk management file, the manufacturer shall define and document risk control measures. *[Class B, C]* *NOTE The RISK CONTROL measures can be implemented in hardware, software, the working environment or user instruction.*			
7.2.2	**RISK CONTROL measures implemented in software** If a RISK CONTROL measure is implemented as part of the functions of a SOFTWARE ITEM, the MANUFACTURER shall: a) include the RISK CONTROL measure in the software requirements; b) 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 c) develop the SOFTWARE ITEM in accordance with Clause 5. *[Class B, C]* *NOTE This requirement provides additional detail for RISK CONTROL requirements of ISO 14971*			
7.3	**VERIFICATION of RISK CONTROL measures**			
7.3.1	**Verify RISK CONTROL measures** The implementation of each RISK CONTROL measure documented in 7.2 shall be VERIFIED, and this VERIFICATION shall be documented. *[Class B, C]*			
7.3.2	**Document any new sequences of events** If a RISK CONTROL measure is implemented as a SOFTWARE ITEM, the MANUFACTURER shall EVALUATE the RISK CONTROL measure to identify and document in the RISKMANAGEMENT FILE any new sequences of events that could result in a hazardous situation. *[Class B, C]*			
7.3.3	**Document TRACEABILITY** The MANUFACTURER shall document TRACEABILITY of software HAZARDS as appropriate: a) from the hazardous situation to the SOFTWARE ITEM; b) from the SOFTWARE ITEM to the specific software cause;			

Clause	Requirement	Result- Remark	ABC	Verdict
	c) from the software cause to the RISK CONTROL measure; and d) from the RISK CONTROL measure to the VERIFICATION of the RISK CONTROL measure. *[Class B, C]* *NOTE See ISO 14971 - RISK MANAGEMENT report.*			
7.4	**RISK MANAGEMENT of software changes**			
7.4.1	**Analyze changes to MEDICAL DEVICE SOFTWARE with respect to SAFETY** The MANUFACTURER shall analyze changes to the MEDICAL DEVICE SOFTWARE (including SOUP) to determine whether: a) additional potential causes are introduced contributing to a hazardous situation; and a) additional software RISK CONTROL measures are required.			
7.4.2	**Analyze impact of software changes on existing RISK CONTROL measures** The MANUFACTURER shall analyze changes to the software, including changes to SOUP, to determine whether the software modification could interfere with existing RISK CONTROL measures. *[Class B, C]*			
7.4.3	**Perform RISK MANAGEMENT ACTIVITIES based on analyses** The MANUFACTURER shall perform relevant RISK MANAGEMENT ACTIVITIES defined in 7.1, 7.2 and 7.3 based on these analyses. *[Class B, C]*			

Clause	Requirement	Result- Remark	ABC	Verdict
8	**RISK MANAGEMENT of software changes**			
8.1	**Configuration identification**			
8.1.1	**Establish means to identify CONFIGURATION ITEMS** The MANUFACTURER shall establish a scheme 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.			
8.1.2	**Identify SOUP** For each SOUP CONFIGURATION ITEM being used, including standard libraries, the MANUFACTURER shall document: a) the title, b) the MANUFACTURER, and c) the unique SOUP designator of each SOUP CONFIGURATION ITEM being used. *NOTE The unique SOUP designator could be, for example, a VERSION, a release date, a patch number or an upgrade designation.*			
8.1.3	**Identify SYSTEM configuration documentation** The MANUFACTURER shall document the set of CONFIGURATION ITEMS and their VERSIONS that comprise the SOFTWARE SYSTEM configuration			
8.2	**Change control**			
8.2.1	**Approve CHANGE REQUESTS** The MANUFACTURER shall change CONFIGURATION ITEMS only in response to an approved CHANGE REQUEST. *NOTE 1 The decision to approve a CHANGE REQUEST can be integral to the change control PROCESS or part of another PROCESS. This sub clause only requires that approval of a change precede its implementation.* *NOTE 2 Different acceptance PROCESSES can be used for CHANGE REQUESTS at different stages of the life cycle, as stated in plans, see 5.1.1 e) and 6.1 e).*			
8.2.2	**Implement changes** The MANUFACTURER shall implement the change as specified in the CHANGE REQUEST. The MANUFACTURER shall identify and perform any ACTIVITY that needs to be repeated as a result of the change, including changes to the software safety classification of SOFTWARE SYSTEMS and SOFTWARE ITEMS *NOTE This sub clause states how the change should be implemented to achieve adequate change control. It does not imply that the implementation is an integral part of the change control PROCESS. Implementation should use planned PROCESSES, see 5.1.1 e) and 6.1 e).*			
8.2.3	**Verify changes** The MANUFACTURER shall verify the change, including repeating any VERIFICATION that has been invalidated by the change and taking into account 5..7.3 and 9.7.			

Clause	Requirement	Result- Remark	ABC	Verdict
	NOTE This sub clause only requires that changes be VERIFIED. It does not imply that VERIFICATION is an integral part of the change control PROCESS. VERIFICATION should use planned PROCESSES, see 5.1.1 e) and 6.1 e).			
8.2.4	**Provide means for TRACEABILITY of change** The MANUFACTURER shall create an audit trail whereby each: a) CHANGE REQUEST; b) relevant PROBLEM REPORT; and c) approval of the CHANGE REQUEST can be traced			
8.3	**Configuration status accounting**			
	The MANUFACTURER shall retain retrievable records of the history of controlled CONFIGURATION ITEMS including SYSTEM configuration			

Clause	Requirement	Result- Remark	ABC	Verdict
9	**Software problem resolution PROCESS**			
9.1	**Prepare PROBLEM REPORTS**			
	The MANUFACTURER shall prepare a PROBLEM REPORT for each problem detected in a SOFTWARE PRODUCT. PROBLEM REPORTS shall be classified as follows: a) type; *EXAMPLE 1 corrective, preventive, or adaptive to new environment* b) scope; and *EXAMPLE 2 size of change, number of device models affected, supported accessories affected, resources involved, time to change* c) criticality. *EXAMPLE 3 effect on performance, SAFETY, or SECURITY* *NOTE Problems can be discovered before or after release, inside the MANUFACTURER'S organization or outside it.*			
9.2	**Investigate the problem**			
	The MANUFACTURER shall: a) investigate the problem and if possible identify the causes; b) EVALUATE the problem's relevance to SAFETY using the software RISK MANAGEMENT PROCESS (Clause 7); c) document the outcome of the investigation and evaluation; and d) create a CHANGE REQUEST(S) for actions needed to correct the problem, or document the rationale for taking no action. *NOTE A problem does not have to be corrected for the MANUFACTURER to comply with the software problem resolution PROCESS, provided that the problem is not relevant to SAFETY.*			
9.3	**Advise relevant parties**			
	The MANUFACTURER shall advise relevant parties of the existence of the problem, as appropriate. *NOTE Problems can be discovered before or after release, inside the MANUFACTURER'S organization or outside it. The MANUFACTURER determines the relevant parties depending on the situation.*			
9.4	**Use change control process**			
	The MANUFACTURER shall approve and implement all CHANGE REQUESTS, observing the requirements of the change control PROCESS (see 8.2).			
9.5	**Maintain records**			
	The MANUFACTURER shall maintain records of PROBLEM REPORTS and their resolution including their VERIFICATION. The MANUFACTURER shall update the RISK MANAGEMENT FILE as appropriate (see 7.4)			
9.6	**Analyze problems for trends**			
	The MANUFACTURER shall perform analysis to detect trends in PROBLEM REPORTS			

Clause	Requirement	Result- Remark	ABC	Verdict
9.7	**Verify software problem resolution**			
	The MANUFACTURER shall verify resolutions to determine whether: a) problem has been resolved and the PROBLEM REPORT has been closed; b) adverse trends have been reversed; c) CHANGE REQUESTS have been implemented in the appropriate SOFTWARE PRODUCTS and ACTIVITIES; and d) additional problems have been introduced.			
9.8	**Test documentation contents**			
	When testing, retesting or REGRESSION TESTING SOFTWARE ITEMS and SYSTEMS following a change, the MANUFACTURER shall include in the test documentation: a) test results; b) ANOMALIES found; c) the VERSION of software tested; d) relevant hardware and software test configurations; e) relevant test tools; f) date tested; and g) identification of the tester			

System Safety Inc.
Safety Livecycle Services

Mapping of Required Evidence and Client Documents

Standard Clause	Deliverables	Title	Revi-vi-sion	Date
4.1	Software Development SOP			
4.3	Software safety classification			
5.1.1	Software Development Plan			
5.1.4	Software development standards (incl. coding STD)			
5.1.4	Software Methods and Tools Planning			
5.1.5	Integration Test Planning			
5.1.6	Verification Planning			
5.1.6	Software Integration Test Plan			
5.1.6	Software System Test Plan			
5.1.7	Risk Management Plan			
5.1.8	Documentation Plan			
5.1.9	Configuration Management			
5.2.1	Software Requirement Specification			
5.3	Software Architecture Documentation			
5.4	Software Unit Requirement Specification			
5.5	Software Design Specification			
5.5.5	Unit Verification			
5.6	Software Integration Testing Report			
5.7	Software System Testing Report			
5.8.2	Anomalies Documentation			
5.8.3	Residual Risk Evaluation			
5.8.4	Software Release Documentation			
6.1	Software Maintenance Plan			
6.2	Impact Analysis			
6.3	Re-release and Regression Test Reports			
7	Software Risk Management (part of RMF)			