SOFTWARECPR®
CRISIS PREVENTION AND RECOVERY, LLC

62304 Amendment 1: Medical device software – Software life cycle processes

SoftwareCPR® Tiered Checklist and Assessment Forms

Prepared by Mary Decareau and Alan Kusinitz with review by Raffaele Caliri.
For training, assessment, or implementation support contact us by leaving a message at www.softwarecpr.com or calling 781-721-2921

1.0 Purpose

This document is intended as a job aide to assessments for conformance to ANSI/AAMI/IEC 62304. It serves as a checklist and provides space to map the internal process to the standard's requirements. The information collected can be used as a mapping of the internal process to 62304 to aide 3rd party conformance assessments.

THIS DOCUMENT INDICATES SUBSTANTATIVE CHANGES TO THE SOFTWARECPR® CHECKLIST FOR AMENDMENT 1 of 62304 in YELLOW HIGHLIGHTS.

2.0 Usage

- **This job aide should only be applied by those who are knowledgeable about 62304 and its proper interpretation** and have an understanding of software engineering and validation principles. Also note that the text is not the full or exact text in the standard.
- **A tiered approach to conformance assessment is incorporated into these forms. One can assess at several levels:**
  - Are all required processes established
  - Are all required tasks and activities performed
  - Are all documentation requirements met
  - Do tasks and deliverables incorporate all required and relevant items (usually by sampling not all in every deliverable)

A group could conform at one or more levels but not be in full conformance. Or a group could completely conform for maintenance or initial development but not both. These forms are intended to highlight the degree of conformance rather then just provide a straight list of items.

The forms provided can be just used as a checklist with notes taken separately for document and procedure references and comments.

DISCLAIMER: These forms should not be used in place of the standard itself and may have unintended omissions or inaccuracies as well as paraphrased verbiage.

SOFTWARECPR®
CRISIS PREVENTION AND RECOVERY, LLC

www.softwarecpr.com
781-721-2921

3.0 Identification and Conclusion

Company/Division/Department/Group:_____

Project/Product:_____

Scope/portion of 62304 Assessed: (Indicate 62304 included or excluded whichever is the shorter list):

_____

_____

Depth of Assessment (Describe which tiers included and the degree of document review and interviewing)

_____

_____

NOTE: If the assessment is focused solely on Legacy software and retrospective methods have been used apply B.4.4 from the standard.

Identify items that are treated as Legacy Software for the purpose of conformance:

_____

_____

Performed by: _____

Analysis and Conclusion: Optional: Normally this would go in a separate report.

State degree of conformance determined using the Tiered method. List:

- Processes Missing:
- Tasks and activities omitted (or summarize)
- Documentation requirements omitted (or summarize)
- Required low level tasks and deliverables omitted (or summarize)

4. High-level Conformance Evaluation

The Procedure/Plan column is to note where the approach or method for the activity is defined. The deliverable/documents column is to note the output of the activity in terms of documents and other deliverables that provide objective evidence that the process and activity was performed. One procedure, plan or document could be referenced multiple times. If all elements of this table are satisfied, one demonstrates conformance with the processes and activities requirements of ANSI/AAMI/IEC 62304. Note that ANSI/AAMI/IEC 62304 also requires specific tasks and these more detailed requirements are not addressed in this table.

The "initially"column indicates whether the initial development was conformant and the "now"column indicates whether the current process is conformant.

Enter NE if the requirement was NOT EVALUATED. Enter NA if it is not applicable. These forms can be just used as a checklist with notes taken separately for document and procedure references and comments.

ANSI/AAMI/IEC 62304	Initially (Y, N, NE)	Now (Y, N, NE)	Procedure, Plan Titles	Deliverables/documents	Comments
4.1 Conformance with 13485 or a national quality management system or a quality management system required by national regulation					State if 13485, 21 CFR 820 or other
4.2Medical Device Risk Management standard ISO 14971					
4.3 Software safety classification					
4.4 Legacy Software					
5 Software development Process					
5.1.1 Software Development plan or plans.					
5.2 Software Requirements Analysis					
5.3 Software Architectural Design ***(no Class A requirements)***					
5.4. Software Detailed Design ***(no Class A requirements)***					
5.5 Software Unit Implementation and Verification					
5.6 Software integration and integration testing ***(no Class A requirements)***					
5.7 SoftwareSystem Testing					
5.8 Software Release					
6 Software Maintenance Process					

6.1. Establish Software Maintenance Plan					
6.2 Problem and modification analysis					
6.3 Modification Implementation					
7 Software Risk Management Process ***(only Section 7.4.1 is required for Class A)***					
7.1 Analysis of software contributing to hazardous situations					
7.2 Risk Control Measures					
7.3 Verification of Risk Control Measures					
7.4 Risk Management of Software Changes					
8 Software Configuration Management Process					
8.1 Configuration Identification					
8.2 Change Control					
8.3 Configuration Status Accounting					
9 Software Problem Resolution Process					
9.1 Prepare Problem Reports					
9.2 Investigate the problem					
9.3 Advise relevant parties					
9.4 Use Change Control process					
9.5 Maintain records					
9.6 Analyse problems for trends					
9.7 Verify software problem resolution					
9.8 Test documentation contents					

Software Safety Classification

The manufacturer shall assign to each software system a safety class according to the Risk of Harm to the patient, operator, or other people resulting from a hazard to which the software system can contribute. This is documented in the Risk Management file.
Class A – Cannot contribute to a hazardous situation or does not result in unacceptable risk after consideration of Risk Control measures external to SW
Class B – Can contribute to a hazardous situation which results in unacceptable risk even after consideration of Risk Control measures external to SW and non-serious injury is possible
Class C –Can contribute to a hazardous situation which results in unacceptable risk even after consideration of Risk Control measures external to SW and Death or serious injury is possible

The manufacturer shall also identify safety classifications of each software item or group of items.

System Class (A,B,C)	Assessor Opinion on Software System Classification

If software items are not all the same then use this as well to **examine a sampling of items classified lower than the system classification**.

Identify Sampled Software Item Classifications and highlight any with questionable Classification	Assessor Rationale

4.4 Legacy Software

For each item of Legacy software (if any) treated under 62304 using Section 4.4 to allow alternative methods to demonstrate conformance identify the following. If there is legacy software but full conformance can be shown using the normal 62304 elements then skip this section.

ANSI/AAMI/IEC 62304 Conformity Requirements	**Y/ N /NE/ NA**	**Procedure, Plan, or Document references** (If level of detail in section 4 is not considered a sufficient mapping)	**Comments**
4.4.2a assess any feedback, including post-production information, on LEGACY SOFTWARE regarding incidents and / or near incidents, both from inside its own organization and / or from users			
b. Risk Managment			
- relationship/role of Legacy SW in device Architecture - validity existing risk measures - hazardous situations - causes - risk control measure for each cause			
4.4.3 Gap Analysis versus 5.2, 5.3, 5.7 and 7 a. validity of available deliverables b. value of creating missing deliverables relative to risks c. deliverables to be created (minimum is software system test records 5.7.5)			

4.4.4 Gap closure activities a. Plan to generate deliverables b. Use of problem resolution process c. Changes to Legacy software in conformance to Clause 6 of 62304			
4.4.5 Rational for continued use of Legacy software			

62304 Processes Detailed Section by Section Checklist

The processes below are required for the safety classifications indicated, unless the manufacturer documents in the Risk Management file a rationale for using a lower classification.

UNLESS NOTED EACH CHECKLIST ITEM APPLIES TO ALL SAFETY CLASSES

For items that are outside the scope of the assessment use **NE** – ***not evaluated*** – and be clear about the scope of the assessment in any summary report or conclusions.

For items that are not relevant use **NA** – *not applicable* – **and document your rationale.**

NOTE: This checklist can be used to evaluate if plans and procedures address all relevant items but for a full assessment results of actual development and maintenance should be evaluated to determine if in practice all conformance was achieved with all items.

5 Software Development Process

5.1 Software development planning

ANSI/AAMI/IEC 62304 Conformity Requirements	**Y/ N /NE/ NA**	**Procedure, Plan, or Document references** (If level of detail in section 4 is not considered a sufficient mapping)	**Comments**
5.1.1 a – e The software development/quality plan(s) addresses:			
a. the processes to be used			
b. the deliverables of the activities and tasks			
c. traceabilty between system requirements, software requirements, software system test and risk control measures.			

d. configuration and change management including SOUP configuration items and software used for development			
e. software problem resolution procedure			
5.1.2 Software development/quality plan(s) get updated			
5.1.3 a. Software development plan(s) references system requirements as inputs			
5.1.3 b. The plan includes or references procedures for coordinating the software development with the system/ device development necessary to satisfy 4.1 (such as system integration, device verification and validation).			
5.1.4 Standards, methods and tools defined in plan(s). Class C.			
5.1.5 software integration and software integration test are included in the plan. Including SOUP. Class B, C			
5.1.6 software verification plan(s) include a) deliverables requiring verification, b) the verification tasks required for each life cycle activity, c) the milestones at which deliverables are verified d) acceptance criteria for verification			
5.1.7 Risk management planning is included in the plan(s) and includes risk management related to SOUP.			

5.1.8 Documentation planning is included in the plan(s) and includes the following for documents to be produced during the software development life cycle: a) Title, name or naming convention b) Purpose c) Deleted in Amendment, note added to refer to Clause 8 for CM of documentation d) Procedures and responsibilities for development, review, approval and modification.			
5.1.9 Plans include CM information including: a. items to be controlled. b. SCM activities and tasks c,d.. organizational responsibilities for CM e. points when the items are to be placed under formal CM f. when the problem resolution process is to be used.			
5.1.10 Supporting development tools, items or settings are included in CM Class B, C			
5.1.11 Plans require software items are placed under formal CM before they are verified. Class B, C			
5.1.12 Plans include a procedure for: a) identifying categories of defects that may be introduced based on the selected programming technology that are relevant to their software system b) documenting evidence that demonstrates that these defects do not contribute to unacceptable risk			

5.2 Software Requirements Analysis

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
5.2.1 software requirements are defined and documented from System Requirements.			
5.2. As appropriate check for the following types of requirements 5.2.2 a. include Functional and capability requirements			
5.2.2 b. Software system inputs and outputs			
5.2.2 c. Interfaces between the software system and other systems.			
5.2.2 d. Alarms, warnings, operator messages			
5.2.2 e. Security			
5.2.2 f. User interface requirements implemented by the software			
5.2.2 g. Data Definition and database requirements			
5.2.2 h. Installation and acceptance reqs at the operation and maintenance site.			
5.2.2 i. reqs for operation and Maintenance			
5.2.2 j. requirements related to IT-network aspects			
5.2.2 k. user maintenance reqs			
5.2.2 l. regulatory reqs such as from performance standards for the device type, regulatory guidance documents for functionality for the device type, …			
5.2.3 risk control measures included as reqs. Class B, C			
5.2.4 device risk analysis re-evaluated and updated based on software reqs.			

5.2.5 System requirements updated based on software reqs			
5.2.6 Verify the software requirements including that: a) system and risk control reqs implemented.			
b) Do not contradict one another			
c) in terms minimizing ambiguity			
d) testable			
e) uniquely identified			
f) are traceable to System requirements			

5.3 Software Architectural Design (No Class A requirements)

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
5.3.1 Documented software architecture including structure and software items. Class B, C			
5.3.2 Documented architecture includes the interfaces between the software items and between software items and external components (HW and SW). Class B, C			
5.3.3 Functional and performance requirements are specified for SOUP items. Class B, C			
5.3.4 System hardware and software necessary for SOUP items are specified. Class B, C			
5.3.5 segregation essential to risk control is specified and states how to ensure it is effective. Class C			
5.3.6 Verify and document the architecture including that it: a) implements system and software and risk control reqs b) supports internal and external interfaces c) supports proper operation of SOUP items Class B, C			

5.4 Software Detailed Design (No Class A requirements)

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
5.4.1 Subdivide software into software units. Class B, C			

5.4.2 design exists for each software unit with enough detail for correct implementation. Class C			
5.4.3 Detailed design exists for the interfaces between the software units and between software units and external components (hw and software) that are detailed enough to implement each unit and its interfaces correctly. Class C			
5.4.4. Verification that the detailed design a) implements the software architecture b) is free from contradiction with the architecture. Class C			

5.5 Software unit implementation ~~and Verification~~

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
5.5.1, Implement units (Class A,B,C)			
5.5.2 -Procedures, methods and strategies exist for verifying each software unit. -Where Verification is done by testing, test procedures evaluated for adequacy. Class B, C			
5.5.3 Acceptance criteria - established for software units prior to integration - Units met acceptance criteria Class B, C			
5.5.4 unit acceptance criteria shall included proper event sequence, data and control flow, planned resource allocation, fault handling, initialization of variables, self diagnosis, memory management, memory overflows and			

boundary conditions. Class C			
5.5.5 Unit test verification has been performed and results documented. Class B, C			

5.6 Software integration and integration testing (No Class A requirements)

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
5.6.1 Software units integrated in accordance with the integration plan. Class B, C			
5.6.2 Verify (not testing usually by review) that units have been integrated into software items and/or the software system and retain records of such verification Class B, C			
5.6.3 software items have been tested in accordance with the integration plan and the results are documented. Class B, C			
5.6.4 integration testing (NOTE: may be combined with system testing) verifies that the software item performs as intended Class B, C			
5.6.5 integration test procedures shall be evaluated for adequacy. Class B, C			
5.6.6 regression testing to identify defects in other units that show up after integration of new units Class B, C			
5.6.7 Integration test records contain: a) the test result including pass/fail determinations and a list of anomalies b) records to permit repeating the test and c) tester identification			

Class B, C			
5.6.8 formal process exists and anomalies found during integration and integration testing are recorded. Class B, C			

5.7 Software System Testing – Amendment 1 changed to require all elements for Class A as well as B and C

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
5.7.1 a)Testing covers all requirements and Tests include input stimuli, expected results, pass/fail criteria and cover all requirements. b) The adequacy of verification strategies and procedures is evaluated Note: it is acceptable to combine integration and system testing in earlier phases.			
5.7.2 Anomalies handled using the formal problem resolution process			
5.7.3 Regression testing after changes and perform any relevant risk management activities			
5.7.4 Evaluated that a) all software requirements have been tested or otherwise verified b) the traceability between software requirements and tests or other verification has been recorded c) test result meet the required pass/fail criteria			
5.7.5 Software system test records contain a) a reference to test case procedures showing required actions and expected results b)the test result (pass/fail and any anomalies) c)the version of the software tested			

d) relevant hardware and software test configurations e) relevant test tools f) date tested g) identify of person responsible for executing the test and recording the results			

5.8 Software Release

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
5.8.1 verification activities are completed and results evaluated before release. Class A, B, C			
5.8.2 Known residual anomalies are documented. Class A, B, C			
5.8.3 Known residual anomalies have been evaluated to ensure they do not pose an unacceptable risk. Class B, C			
5.8.4 Versions of the software that are released are documented. Class A, B, C			
5.8.5 The procedure and environment used to build the release version is documented. Class B, C			
5.8.6 All software development plan (or maintenance plan), activities and tasks are complete along with associated documentation. Class B, C			
5.8.7 The medical device software and configuration items, documentation are archived for a period longer than the life of the medical device software or as specified by relevant regulatory requirements. Class A, B, C			

5.8.8 Procedures ensure that released software can be reliably delivered without change or corruption covering: - replication - media labeling - packaging -protection - storage - delivery Class A, B, C			

6 Maintenance Process

When planning assessments it is recommended to assess both new or original development projects and at least one maintenance release.

6.1 Establish Software Maintenance Plan (all are for all classes)

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
6.1 A software maintenance Plan is established. It includes:			
a) procedures for receiving, documenting, evaluating and tracking feedback after release.			
b) criteria for determing whether feedback is considered to a problem.			
c) use of the software risk management process.			
d) use of the formal problem resolution process. (also in 6.2.2)			
e) use of configuration management process			
f) procedures to evaluate and implement upgrades, bug fixes, patches and obsolescence of SOUP.			

6.2 Problem and Modification Analysis

Section Conformity Requirements	Y/N /NE /NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
6.2.1.1 feedback on released software products are monitored for intended use			
6.2.1.2 Feedback is documented (as problem reports) and evaluated to determine whether a problem exists. Problem reports include actual or potential adverse events or deviations from specifications.			
6.2.1.3 problem reports are evaluated to determine how it affects the safety of software released for intended use and whether a change to the software is needed.			
6.2.2 Problem report process is used to address problems			
6.2.3 Each change request is analyzed for its effect on the organization, intended use of released software and systems with which it interfaces. Class A, B, C			
6.2.4 Modifications to released medical device software are evaluated and approved.			
6.2.5 Changes are communicated to users and regulators as required, including: a) Any problem in released medical device software and the consequences of continued unchanged use. b) The nature of any available changes to released medical device software and how to obtain and install the changes.			

6.3 Modification Implementation

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments

6.3.1 Identified and performed any Clause 5 activities that need to be repeated as a result of the modification.			
6.3.2 Changed software shall be released according to 5.8 Class A, B, C. *Note: Modifications can 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 installation tools.*			

7 Software Risk Management Process (only 7.4.1 applies to Class A software)

7.1 Analysis of software contributing to hazardous situations

Section Conformity Requirements	**Y/N/ NE/ NA**	**Procedure, Plan, or Document references** (If level of detail in section 4 is not considered a sufficient mapping)	**Comments**
7.1.1 Software items that could contribute to a hazardous situation are identified. Class B, C			
7.1.2 Potential causes of hazardous situations have been identified including:. a) Incorrect or incomplete specification of functionality b) Software defects c) Failure or unexpected results from SOUP d) Hardware failures or other software defects that could result in unpredictable software operation (indirect/common causes) e) Reasonably forseeable misuse. Class B, C			
7.1.3 If SOUP failure is a potential cause supplier published anomaly lists were evaluated for relevance. Class B, C			
7.1.4 Potential causes of software items contributing to hazards have been documented. Class B, C			

~~7.1.5 Document Sequences of Events DELETED~~			

7.2 Risk Control measures

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
7.2.1 Risk control measures have been identified for each potential software cause. Class B, C			
7.2.2 Risk control measures implemented in software a) are included in software requirements b) assigned to software items that implement a risk control measure a software class based on the risk that the risk control measure is controlling			

7.3 Verification of Risk Control Measures

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
7.3.1 documented verification for all risk control measures. The risk control measures have been reviewed to determine if it could result in a new hazardous situation. Class B, C			
7.3.2 ~~Document any new Sequences of Events DELETED Not used~~			
7.3.3 Documented traceability from a) hazardous situation to the software item b) software item to specific software cause c) software cause to RCM d) RCM to verification of RCM Class B, C			

7.4 Risk Management of Software Changes

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of detail in section 4 is not considered a sufficient mapping)	Comments
7.4.1 Changes to the software are analyzed to determine whether: a) additional software risk control measures are required. b) additional potential causes are introduced contributing to a hazardous situation Class A, B, C			
7.4.2 software changes, including changes to SOUP are analyzed to determine if the modification could interfere with existing RCMs. Class B, C			
7.4.3 Risk mgmt activities have been performed based on the analysis of the changes. Class B, C			

8 Configuration Management Process

8.1 Configuration Identification (all are for all classes)

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of deail in section 4 is not considered a sufficient mapping)	Comments
8.1.1 Unique identification for configuration items and their versions to be controlled according to the development and configuration planning specified in 5.1			
8.1.2 Each SOUP item is identified by title, manufacturer, and unique SOUP designator/version/patch # etc.			
8.1.3 System configuration documentation includes versions for all items			

8.2 Change Control (all are for all classes)

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of deail in section 4 is not considered a sufficient mapping)	Comments
8.2.1 Configuration items identified to be controlled according to 8.1 are changed only in response to an approved change request. NOTE: Different acceptance processes can be defined for different lifecycle phases. Note if there are.			
8.2.2 Changes are implemented as specified in the change request. Activities that need to be repeated as a result of the change have been performed.			
8.2.3 Changes are verified including repeating any verification that has been invalidated by the change.			
8.2.4 Records exist that show the relationship and dependencies between Change Requests, Problem Reports and Approval of the Change Request.			

8.3 Configuration Status Accounting Tasks (all are for all classes)

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of deail in section 4 is not considered a sufficient mapping)	Comments
8.3 Retrievable records are retained that show the history of the controlled configuration items including system configuration.			

9 Software Problem Resolution Process (all are for all classes)

Section Conformity Requirements	Y/N/ NE/ NA	Procedure, Plan, or Document references (If level of deail in section 4 is not considered a sufficient mapping)	Comments
9.1 Problem reports exist for each problem detected in the software and include a statement of criticality (effect on performance, safety or security) as well as other information			

that may aid in resolution (for example, devices affected, supported accessories affected).			
9.2 Problem are investigated a) to determine the cause, b) evaluate the problem's relevance to safety c) investigation results are documented d) change requests are created for actions needing correct or and rationales for taking no action are documented			
9.3 Relevant parties are advised of the existence of the problem, as appropriate.			
9.4 Change requests are approved observing the requirements of the change control process. **NOTE:** *a special process may exist for emergencies and their appropriateness and overuse checked. If none exists consider if the company is prepared to handle an emergency related to the risk of the device.* 9.5 Records of problem reports and their resolution and verification are kept. The Risk Management file is updated as appropriate.			
9.6 Problem reports are analyzed for trends not just individually			
9.7 Resolutions of problems are verified to determine whether: a) problems are resolved and the problem report closed b) adverse trends have been reversed c) change requests have been implemented in all relevant software items and associated documents d) additional problems have been introduced by the changes.			
9.8 Testing and regression testing documentation following a fix, includes: a. Test results b. Anomalies found			

c. Software version tested d. Relevant hardware and software test configurations e. Relevant test tools f. Date tested g. Identification of the tester.			

END OF CHECKLIST

REMEMBER TO REFER TO THE STANDARD ITSELF AS THIS CHECKLIST IS NOT INTENDED TO BE USED IN ISOLATION FROM THE STANDARD OR KNOWLEDGE AND TRAINING IN PROPER INTERPRETATION OF THE STANDARD.