IEC 62304 Action List

for medical devices that rely on one or more software components, parts, or accessories for Basic Safety or Essential Peformance

Preparing Your Device for Market

IEC 62304 is the international standard that defines software development lifecycle requirements for medical device software. The standard was developed from the perspective that product testing alone is insufficient to ensure patient safety when software is involved. The standard requires all aspects of the software development life cycle to be scrutinized.

1 General Requirements

You must have a:

Quality Management System

Risk Management Process in Compliance with ISO 14971

SDLC Process in Compliance with IEC 62304

2 Software Safety Classification

Classification is assigned based on risk 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

The device manufacturer assigns a safety class to each Software System that contributes to the implementation of a Risk Control measure. If a class is not assigned, then Class C requirements apply.

3 Software Development Plan

For all classes, this plan must:

- ❑ Reference system design and development
- ❑ Be updated as development progresses
- ❑ Include Verification Planning
- ❑ Include Risk Management Planning
- ❑ Include Configuration Management Planning

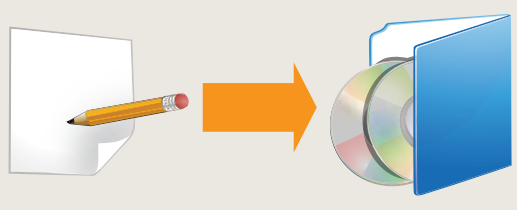
Class B and Class C must also provide:

- ❑ Integration and Integration Testing Planning
- ❑ Supporting items to be controlled
- ❑ Software Configuration Item Control before verification

Class C must also provide:

- ❑ Software Development Standards, Methods and Tools Plan

4 Software Requirements Analysis

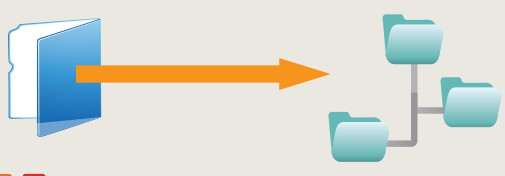


All classes must define and document the software requirements for their system. This includes:

- ❑ Preparing software requirement content
- ❑ Re-evaluating medical device risk analysis
- ❑ Updating system requirements
- ❑ Verifying software requirements

Class B and Class C must also include risk control measures.

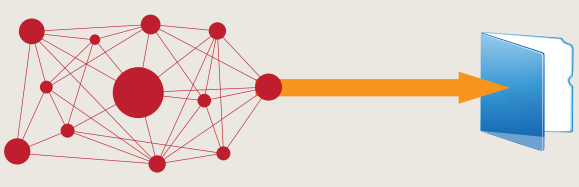
5 Software Architectural Design Requirements



Classes B and C must transform their software requirements and the interfaces of software items into an architecture.

- ❑ Specify functional and performance requirements of SOUP (Software of Unknown Pedigree) items
- ❑ Specify system hardware and software required by SOUP items
- ❑ Identify software segregation necessary for risk control
- ❑ Verify software architecture

6 Software Detailed Design



Classes B and C must refine their Software Architecture into Software Units.

Class C must go into more depth by:

- ❑ Developing Detailed Design for each Software Unit
- ❑ Developing Detailed Design for Interfaces
- ❑ Verifying the Detailed Design

7 Implement Each Software Unit

8 Software Unit Verification Process

Classes B and C must establish a Software Unit Verification Process. Where Verification is done by testing, the test procedures will be evaluated for correctness.

- ❑ Develop Software Unit Acceptance Criteria
- ❑ Complete Software Unit Verification

9 Software Integration & Integration Testing

Classes B and C must:

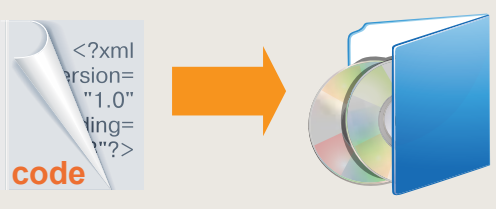
- ❑ Integrate Software Units
- ❑ Verify Software Integration
- ❑ Test Integrated Software
- ❑ Prepare Integration Testing content
- ❑ Verify Integration Test Procedures
- ❑ Conduct Regression Tests
- ❑ Prepare Integration Test Records
- ❑ Use Software Problem Resolution Process

10 Software System Testing

Classes B and C must:

- ❑ Establish tests for Software Requirements
- ❑ Use Software Problem Resolution Process
- ❑ Retest after changes
- ❑ Perform relevant risk management activities
- ❑ Verify Software System Testing
- ❑ Use Software Problem Resolution Process
- ❑ Prepare System Test Record contents

11 Software Release

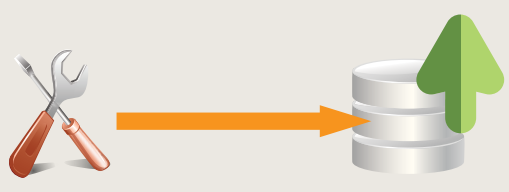


All classes must document the version of the released software.

Additionally, Class B and Class C must:

- ❑ Ensure Software Verification is complete
- ❑ Document known residual anomalies
- ❑ Evaluate known residual anomalies
- ❑ Document how released software was created
- ❑ Ensure activities and tasks are complete
- ❑ Archive software
- ❑ Assure repeatability of Software Release

12 Software Maintenance Process

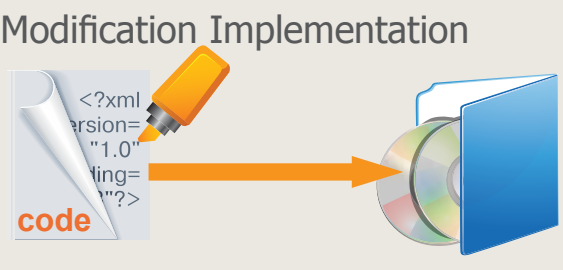


All classes must establish a Software Maintenance Plan. Problem and Modification Analysis is also required. All classes must:

- ❑ Document and evaluate feedback
- ❑ Monitor feedback
- ❑ Evaluate Problem Report's effects on safety
- ❑ Use Software Problem Resolution Process
- ❑ Include change request approval
- ❑ Communicate to users and regulators

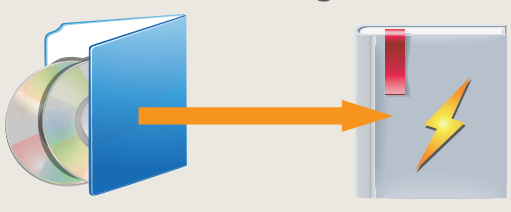
Class B and Class C must also analyze change requests.

13 Modification Implementation



You must use an established process to implement modification. Upon making changes, you are also required to re-release the modified software system.

14 Software Risk Management Process



The Software Risk Management Process is critical for Class B and Class C. These classes must:

- ❑ Conduct an analysis of software that contributes to hazardous situations
- ❑ Identify potential causes of the contribution
- ❑ Evaluate published SOUP anomaly lists
- ❑ Document potential causes
- ❑ Document sequences of events

15 Risk Control Measures

Classes B and C must complete Risk Control Measures, including:

- ❑ Defining said Risk Control Measures
- ❑ Identifying Risk Control Measures implemented in the software
- ❑ Verify Risk Control Measures
- ❑ Document any new sequences of events
- ❑ Document traceability

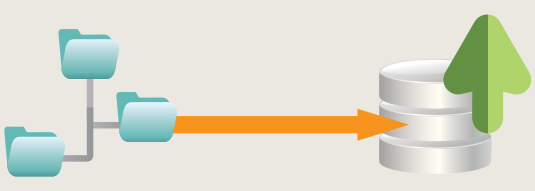
16 Risk Management of Software Changes

All classes must analyze changes to their medical device software with respect to safety.

In addition, classes B and C must:

- ❑ Analyze the impact of software changes on existing risk control measures
- ❑ Perform risk management activities based on these analyses

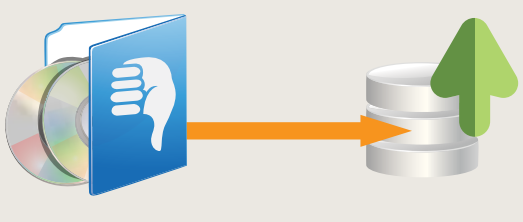
17 Software Configuration Management Process



All classes must implement configuration identification. This includes:

- ❑ Establishing a means to identify configuration items
- ❑ Identifying SOUP
- ❑ Identifying system configuration documentation
- ❑ Configure change controls
- ❑ Approve change requests
- ❑ Implement changes
- ❑ Verify changes
- ❑ Provide means for traceability of change
- ❑ Configure status accounting

18 Software Problem Resolution Process

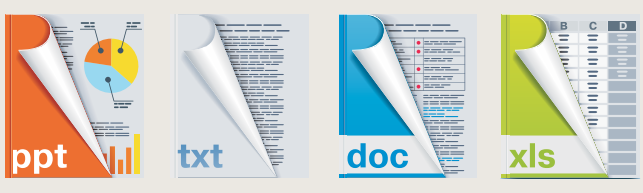


All classes must implement a process for resolving software problems. This includes how you will:

- ❑ Prepare problem reports
- ❑ Investigate the problem
- ❑ Advise relevant parties of the problem
- ❑ Use change control processes
- ❑ Maintain records of the problems
- ❑ Analyze problems for trends
- ❑ Verify Software Problem Resolution
- ❑ Test documentation contents.

19 Submit to Test Lab

When complete, documents should be sent to the test lab in an editable format. The lab will assign your device a pass or fail upon review. If you fail, make necessary changes. If you pass – congratulations!



MethodSense is a life science consulting firm, specializing in helping companies bring medical and technological products to market. They guide medical device, biotech and pharmaceutical companies with quality, regulatory and technology solutions, which enable clients to operate more effectively during the commercialization process and beyond. In addition to their expertise as regulatory consultants, MethodSense has developed a 21 CFR Part 11 compliant software as a service tool, InfoStrength Smart Enterprise Suite, that allows them to guide their clients to success. For more information about MethodSense, please visit www.methodsense.com.

Please note: This IEC 62304 Action List represents a high level summary of the standard's requirements and does not constitute advice for compliance with this or any other regulatory requirement. Manufacturers should consult with their regulatory affairs professional to determine the appropriate compliance strategy for any particular medical device.