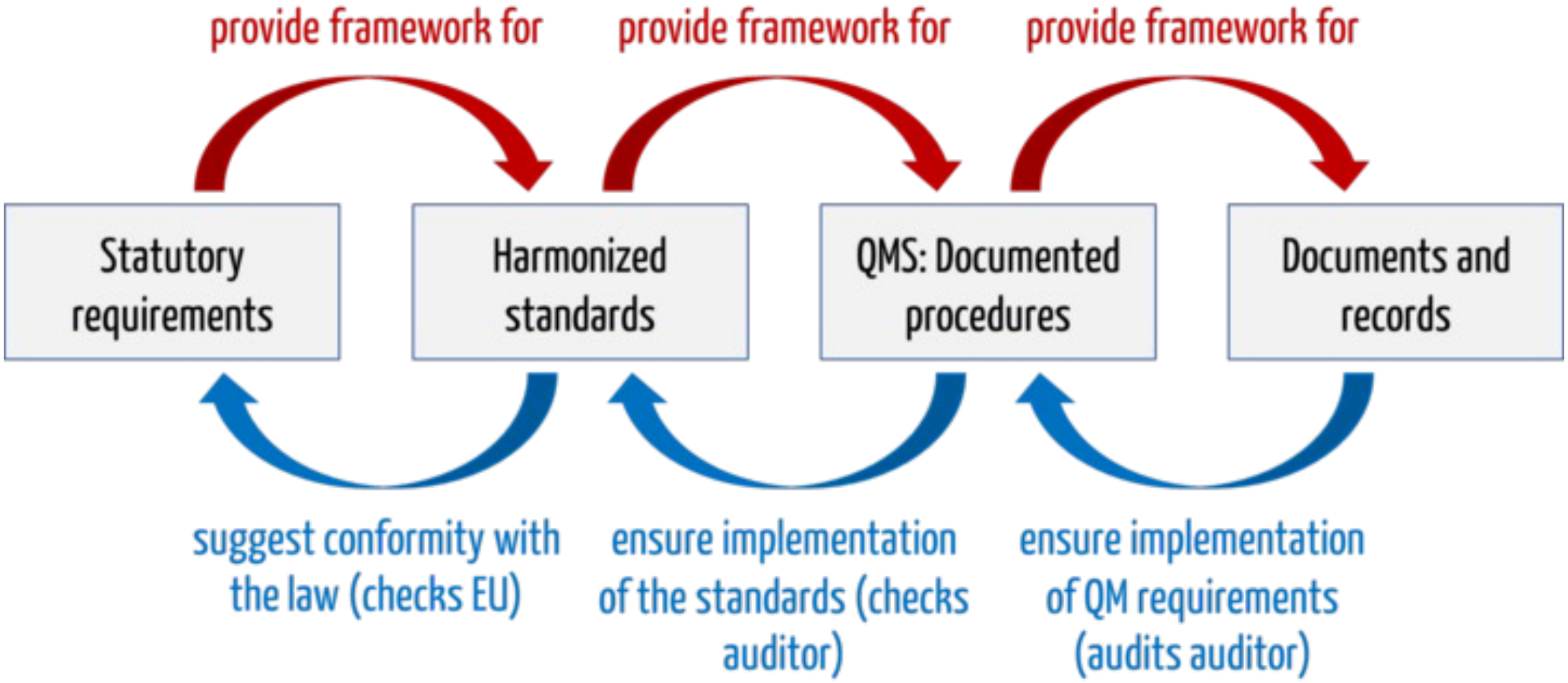
Evidence from Standards

SCSD Lecture

Monday 4 March

Standards and Regulation



Johner Institute

IEC 62304

This standard provides a framework of life cycle PROCESSES with ACTIVITIES and TASKS necessary for the safe design and maintenance of MEDICAL DEVICE SOFTWARE. This standard provides requirements for each life cycle PROCESS. [A1⟩ Each life cycle PROCESS consists of a set of ACTIVITIES, with most ACTIVITIES consisting of a set of TASKS. ⟨A1]

As a basic foundation it is assumed that MEDICAL DEVICE SOFTWARE is developed and maintained within a quality management system (see 4.1) and a RISK MANAGEMENT system (see 4.2). The RISK MANAGEMENT PROCESS is already very well addressed by the International Standard ISO 14971. Therefore IEC 62304 makes use of this advantage simply by a normative reference to ISO 14971. Some minor additional RISK MANAGEMENT requirements are needed for software, especially in the area of identification of contributing software factors related to HAZARDS. These requirements are summarized and captured in Clause 7 as the software RISK MANAGEMENT PROCESS.

Structure of IEC 62304

- Clause 4: General Requirements – QMS, Risk, Safety, Legacy Software
- Clause 5: Software Development Process – see next slides
- Clause 6: Software Maintenance Process – see next slides
- Clause 7: Software Risk Management Process – ISO 14971
- Clause 8: Software Configuration Management Process
- Clause 9: Software Problem Resolution Process
- Appendix A provides a rationale for the clauses of the standard against software safety classes
- Appendix B provides informative guidance on the provisions of the standard
- Appendix C covers the relationship to other standards
- Appendix D covers implementation of the standard

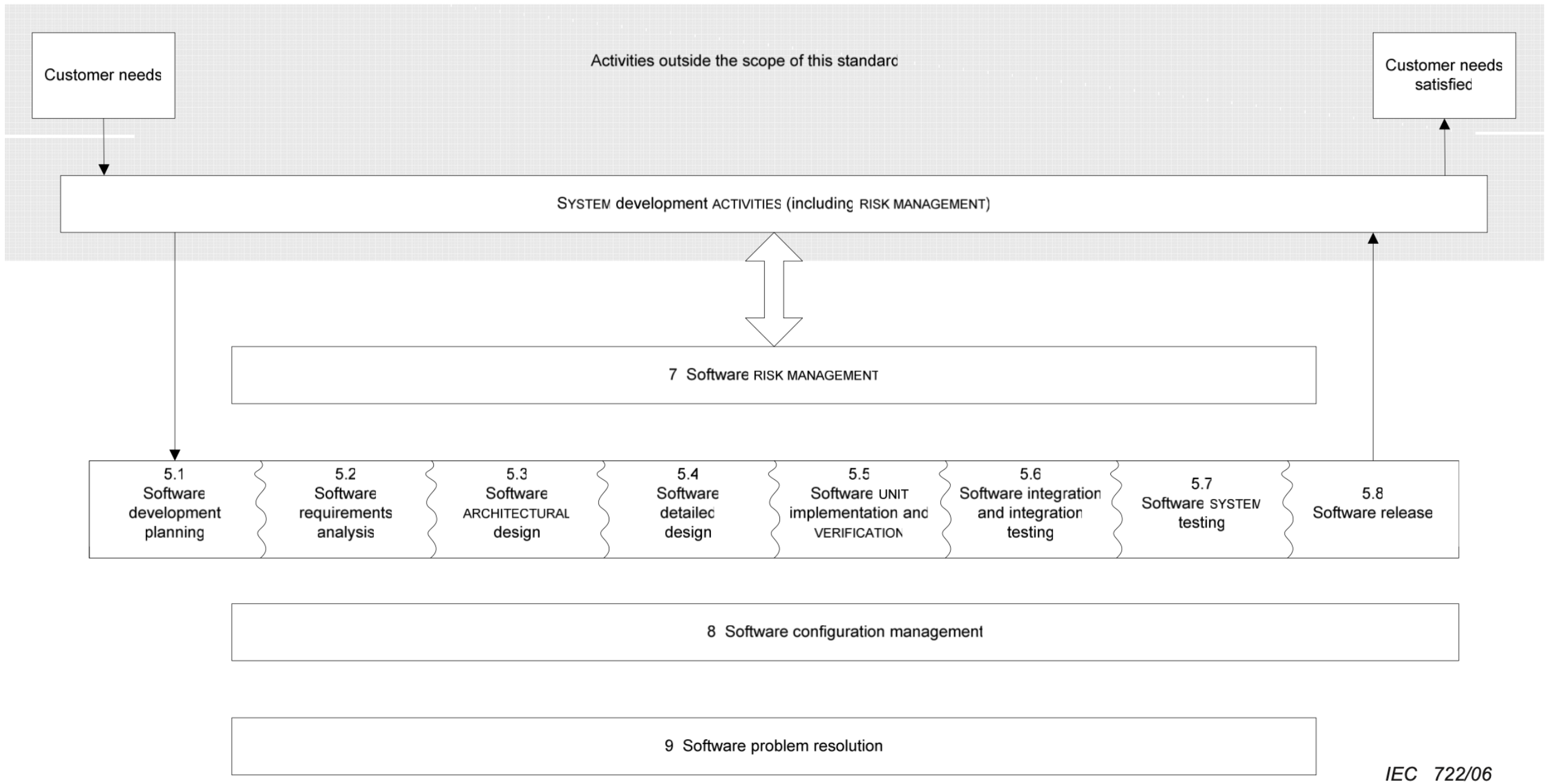


Figure 1 – Overview of software development PROCESSES and ACTIVITIES

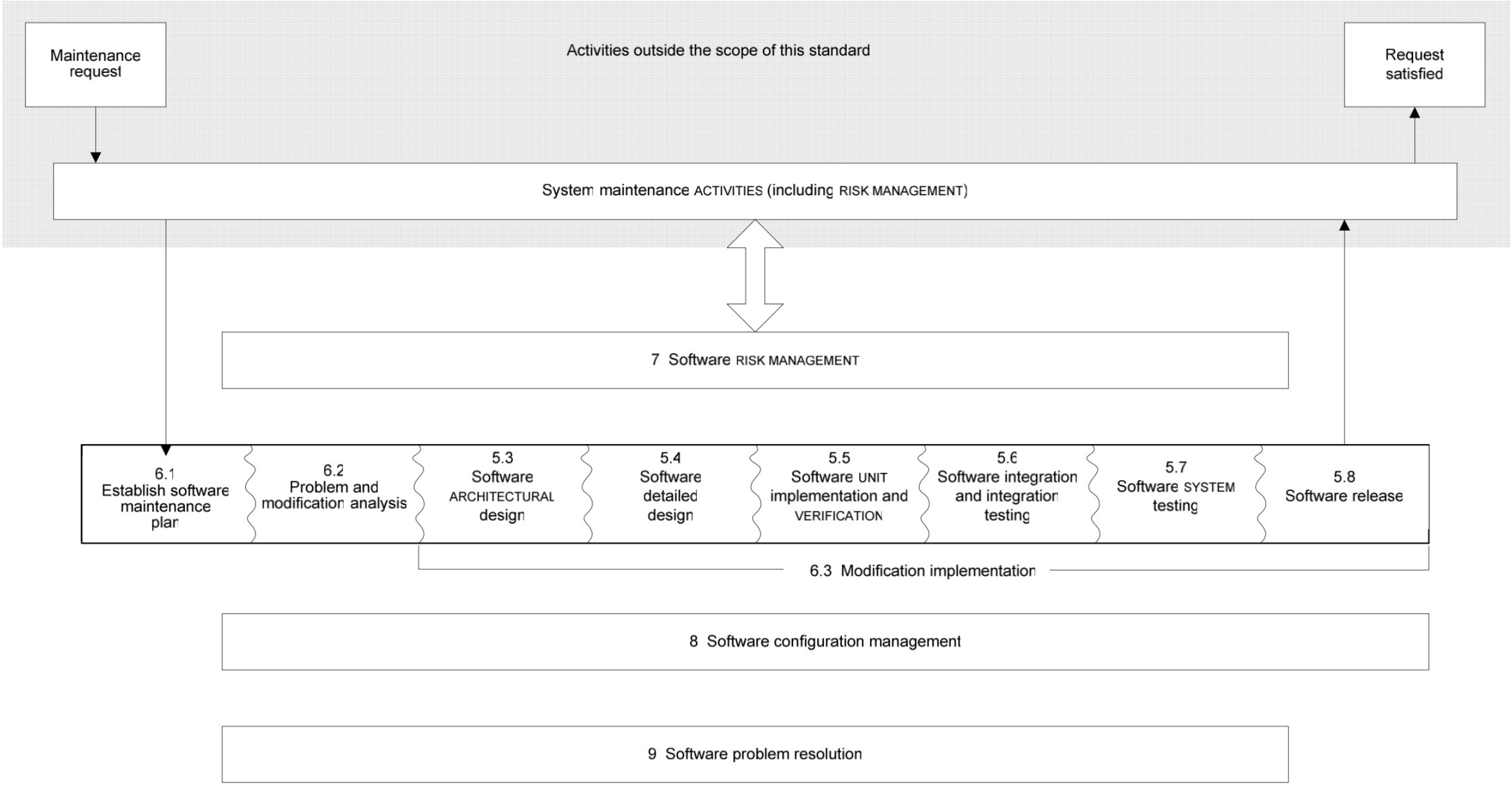


Figure 2 – Overview of software maintenance PROCESSES and ACTIVITIES *IEC 723/06*

Quality Management and Risk Management

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 [7]; 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 [11].

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 RISK of HARM to the patient, operator, or other people resulting from a HAZARDOUS SITUATION to which the SOFTWARE SYSTEM can contribute in a worst-case-scenario as indicated in Figure 3.

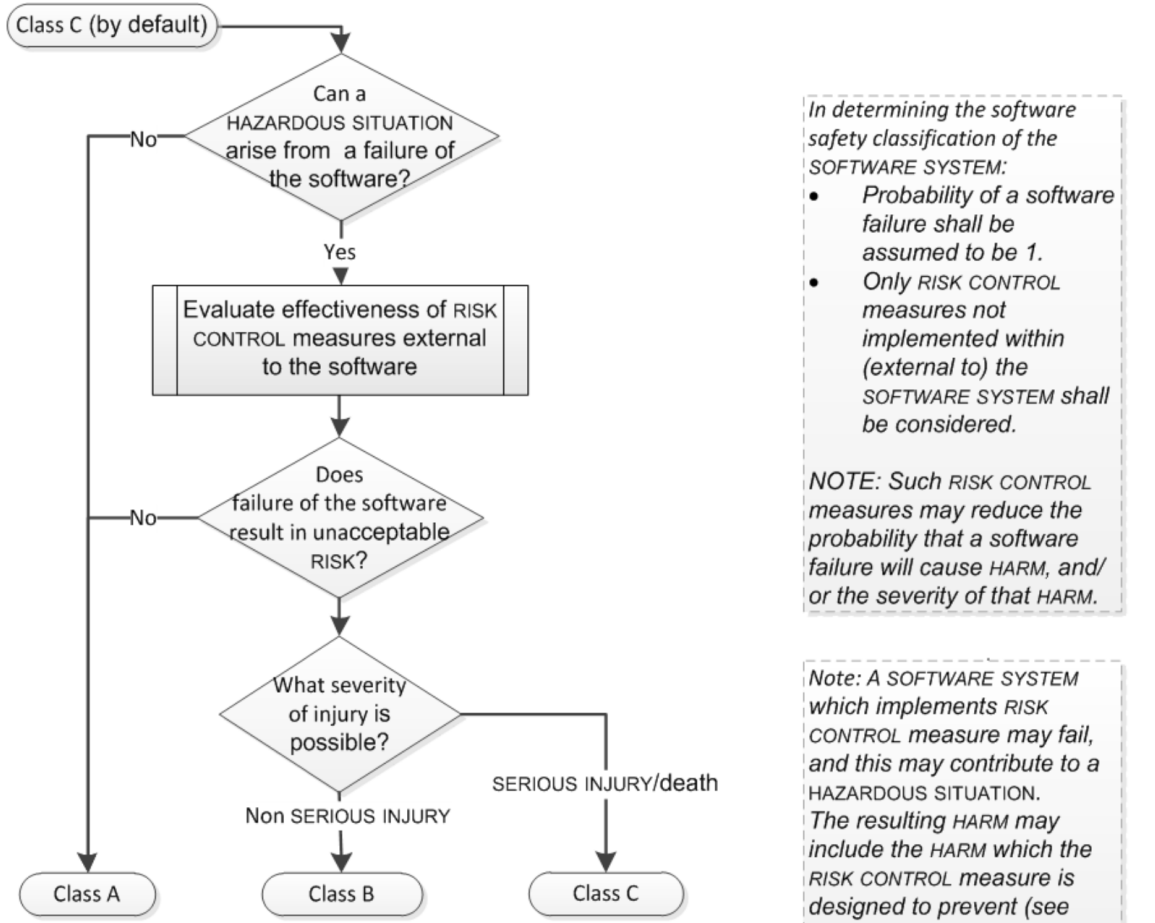


Figure 3 – Assigning software safety classification

Safety Classes

The SOFTWARE SYSTEM is software safety class A if:

- the SOFTWARE SYSTEM cannot contribute to a HAZARDOUS SITUATION; or
- the SOFTWARE SYSTEM can contribute to a HAZARDOUS SITUATION which does not result in unacceptable RISK after consideration of RISK CONTROL measures external to the SOFTWARE SYSTEM.

The SOFTWARE SYSTEM is software safety class B if:

- the SOFTWARE SYSTEM can contribute to a HAZARDOUS SITUATION which results in unacceptable RISK after consideration of RISK CONTROL measures external to the SOFTWARE SYSTEM and the resulting possible HARM is non-SERIOUS INJURY.

The SOFTWARE SYSTEM is software safety class C if:

- the SOFTWARE SYSTEM can contribute to a HAZARDOUS SITUATION which results in unacceptable RISK after consideration of RISK CONTROL measures external to the SOFTWARE SYSTEM and the resulting possible HARM is death or SERIOUS INJURY.

Legacy Systems

- Permits a manufacxturer of the software to avoid the requirements of clauses 5-9.
- By risk managing the legacy software against the standard approach and closing any gaps.

Clause 5 (waterfall stages)

- 5.1: Software Development Planning
- 5.2 Software Requirements Planning
- 5.3 Software Architectural Design
- 5.4 Software Detailed Plan
- 5.5 Software Unit Implementation
- 5.6 Software Integration and Integration Testing
- 5.7 Software System Testing

Realising Clause 5

Table B.1 – Development (model) strategies as defined in ISO/IEC 12207

Development Strategy	Define all requirements first?	Multiple development cycles?	Distribute interim software?
Waterfall (Once-through)	yes	no	no
Incremental (Preplanned product improvement)	yes	yes	maybe
Evolutionary	no	yes	yes

Whichever life cycle is chosen it is necessary to maintain the logical dependencies between PROCESS outputs such as specifications, design documents and software. The waterfall life cycle model achieves this by delaying the start of a PROCESS until the inputs for that PROCESS are complete and approved.

Consistency

Other life cycles, particularly evolutionary life cycles, permit PROCESS outputs to be produced before all the inputs for that PROCESS are available. For example, a new SOFTWARE ITEM can be specified, classified, implemented and VERIFIED before the whole software ARCHITECTURE has been finalised. Such life cycles carry the RISK that a change or development in one PROCESS output will invalidate another PROCESS output. All life cycles therefore use a comprehensive configuration management system to ensure that all PROCESS outputs are brought to a consistent state and the dependencies maintained.

The following principles are important regardless of the software development life cycle used:

- All PROCESS outputs should be maintained in a consistent state; whenever any PROCESS output is created or changed, all related PROCESS outputs should be updated promptly to maintain their consistency with each other and to maintain all dependencies explicitly or implicitly required by this standard;
- all PROCESS outputs should be available when needed as input to further work on the software.
- before any MEDICAL DEVICE SOFTWARE is released, all PROCESS outputs should be consistent with each other and all dependencies between PROCESS outputs explicitly or implicitly required by this standard should be observed.

Subclauses provide detail e.g. 5.2.2

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.

5.2.2 Continued

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.

– system security/malware protection.

f) user interface requirements implemented by software; ⟨A1⟩

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 ⟨A1⟩ IEC 62366-1 [21] among others (e.g., IEC 60601-1-6 [3]) ⟨A1⟩.

5.2.2 Concluded

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 and maintenance site or sites;

i) requirements related to methods of operation and maintenance;

j) requirements related to IT-network aspects;

NOTE 9 Examples include those related to:

– networked alarms, warnings, and operator messages;

– network protocols;

– handling of unavailability of network services. ⟨A1⟩

k) user maintenance requirements; and

l) regulatory requirements.

NOTE 10 The requirements in a) through l) can overlap. ⟨A1⟩

[Class A, B, C]

NOTE 7 All of these requirements might not be available at the beginning of the software development.

NOTE 8 ⟨A1⟩ Among others, ISO/IEC 25010 [12] ⟨A1⟩ provides information on quality characteristics that may be useful in defining software requirements.

Clause 6: Software Maintenance Process

- 6.1 Establish Software Maintenance Plan
- 6.2 Problem and Modification analysis
- 6.3 Modification Implementation

Clause 7: Software Risk Management Process

- 7.1 Analysis of Software Contributing to Hazardous Situations: involves identifying software items and identifying how they might cause hazardous situations (particular mention of SOUP)
- 7.2 Risk Control Measures: all the situations identified in 7.1 should have appropriate risk control measures and they should be added to requirements and development planning.
- 7.3 Verification of Risk Control Measures
- 7.4 Risk Management of Software Changes: ensuring that changes to software are subject to similar measures to the original software.

Verification of Risk Control Measures

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. 〚A1〛 The MANUFACTURER shall review the RISK CONTROL measure and determine if it could result in a new HAZARDOUS SITUATION. 〚A1〛 [Class B, C]

7.3.2 〚A1〛 *Not used* 〚A1〛

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;

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]

Summary

- IEC 62304 is primarily concerned with safety and processes to guarantee safety.
- The emphasis is on the risk of the product in operation.
- Processes are structured to ensure risk is managed effectively
- This is achieved by:
  - prescribing processes that are sensitive to risk classification
  - Requiring effective risk management (ISO 14971)
  - Requiring that quality is effectively managed (ISO 13485)
  - ISO 13485 requires effective document management that organizes the evidence generated by the process