# Systems and Software Engineering Standards for the Medical Domain

Vera Pantelic

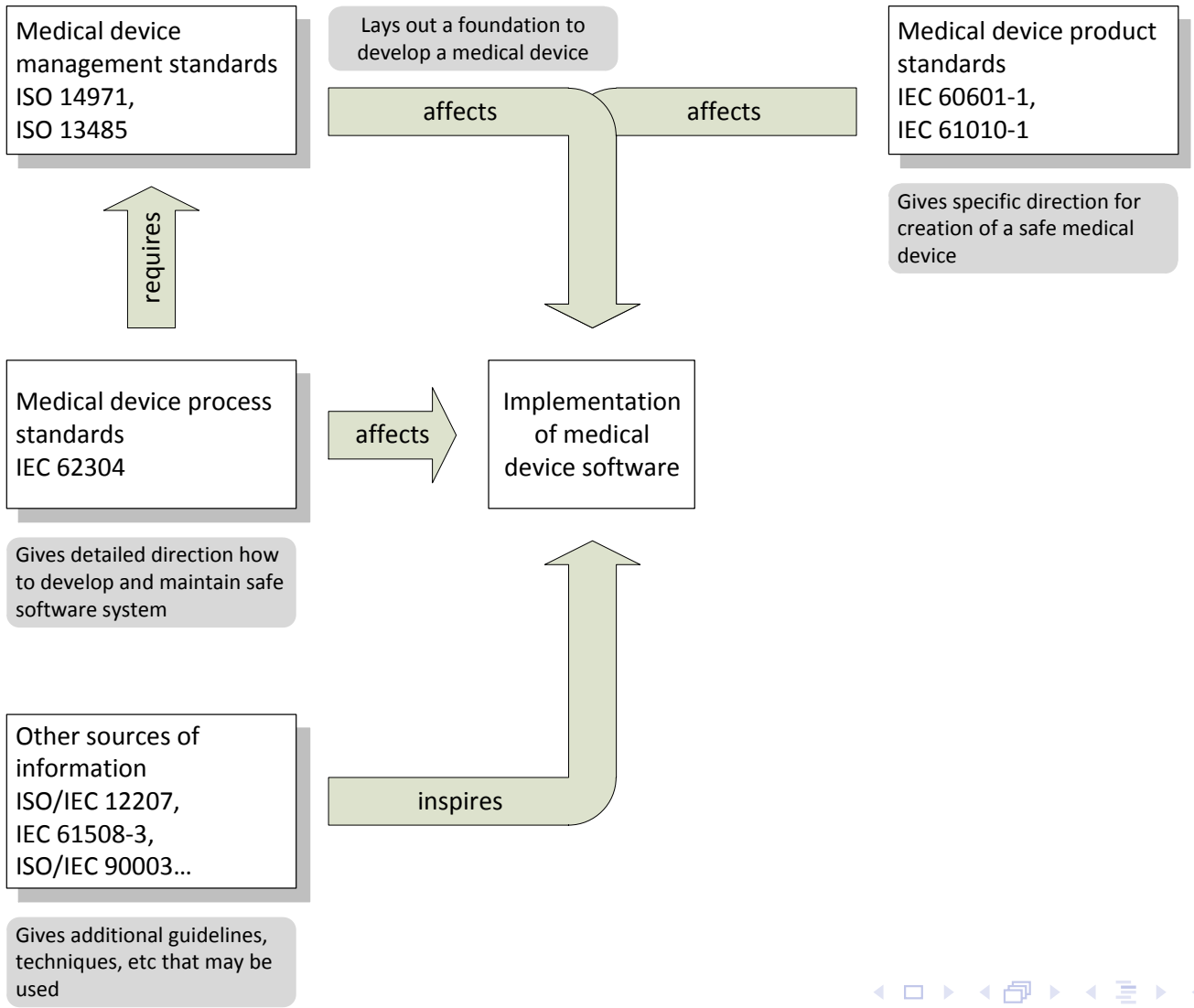
McMaster Centre for Software Certification
Department of Computing and Software
McMaster University

# Outline

# Introduction

- We are mainly interested in the standards important for certification of systems containing software in the medical domain.
- Questions:
  - How are **Safety Integrity Levels (SILs)** defined, derived, and applied?
  - Are **process** and/or **product requirements** defined?
  - Is there an **assurance/safety case** behind a standard?
    - If yes, how? Is the rationale explicit?
  - Do standards address the concept of assurance cases?

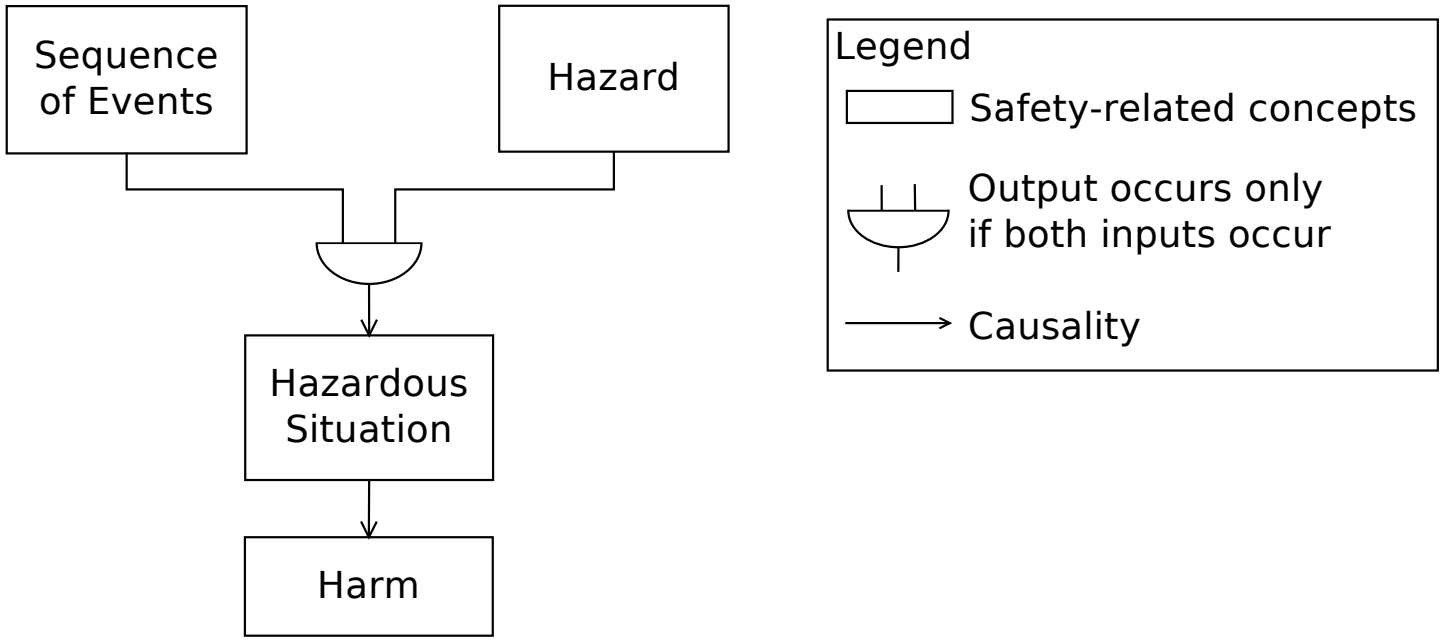
# Standards in Medical Domain (IEC 62304, Annex C.1)



# Vocabulary from ISO/IEC Guide 51:1999

- **harm**
  physical injury, damage, or both to the health of people or damage to property or the environment
- **hazard**
  potential source of harm
- **hazardous situation**
  circumstance in which people, property or the environment are exposed to one or more hazards
- **risk**
  combination of the probability of occurrence of harm and the severity of that harm
- **safety**
  freedom from unacceptable risk

# Estimating Risk




- Probability of occurrence of harm is $P_1 \times P_2$ (often hard to quantify), where $P1$ is the probability of a hazardous situation occurring, and $P2$ is the probability of a hazardous situation leading to a harm.

# We focus on:

- IEC 61508 series (with emphasis on IEC 61508-3:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements),
- IEC 62304:2006 Medical device software – Software life cycle processes.

# IEC 61508: Basics

- It assumes the existence of equipment under control (EUC) and its control system.
- The risks posed by the EUC and its control system should be mitigated until they reach tolerable targets.

# Safety Requirements and SIL

- Safety requirements are specified as
  1. **safety function**: what should be done by an E/E/PE safety related system (or other risk reduction measures) to keep risks at tolerable level, and
  2. **safety integrity**: probability of an E/E/PE safety-related system satisfactorily performing the specified safety functions
- Safety integrity can be specified using a **safety integrity level (SIL)**:
  - one of four possible levels
  - 1 corresponds to the lowest level of integrity, 4 to the highest.

# Application of SILs

- SILs drive the lifecycle processes: higher integrity demands higher rigour.
- For specified lifecycle phases, techniques and measures dependent on a specified SIL are recommended and their effectiveness in satisfying certain properties of artifacts defined as outputs of the lifecycle phases is graded.
- Furthermore, independence of functional safety assessment personnel depends in part on SIL, etc.

# SIL as a Reliability Measure

- Safety integrity levels relate to (dangerous) failure rates:

| Safety integrity level (SIL) | Average probability of a dangerous failure on demand of the safety function ($PFD_{avg}$) |
|---|---|
| 4 | $\geq 10^{-5}$ to $< 10^{-4}$ |
| 3 | $\geq 10^{-4}$ to $< 10^{-3}$ |
| 2 | $\geq 10^{-3}$ to $< 10^{-2}$ |
| 1 | $\geq 10^{-2}$ to $< 10^{-1}$ |

Low demand mode of operation

| Safety integrity level (SIL) | Average frequency of a dangerous failure of the safety function [$h^{-1}$] (PFH) |
|---|---|
| 4 | $\geq 10^{-9}$ to $< 10^{-8}$ |
| 3 | $\geq 10^{-8}$ to $< 10^{-7}$ |
| 2 | $\geq 10^{-7}$ to $< 10^{-6}$ |
| 1 | $\geq 10^{-6}$ to $< 10^{-5}$ |

High demand or continuous mode of operation

- For novel systems and software, the target failure rates are not useful and, even worse, potentially misleading.

# SIL and Confidence

- **Systematic capability:** measure (expressed on a scale of SC 1 to SC 4) of the confidence that the systematic safety integrity of an element meets the requirements of the specified SIL, in respect of the specified element safety function, when the element is applied in accordance with the instructions specified in the compliant item safety manual for the element [IEC 61508-4, Definition 3.5.9]

...

NOTE 3 A Systematic capability of SC N for an element, in respect of the specified element safety function, means that the systematic safety integrity of SIL N has been met when the element is applied in accordance with the instructions specified in the compliant item safety manual for the element.

# Software Safety Integrity Levels and Confidence

- **Software safety integrity level:** systematic capability of a software element.
- “SIL N software” means “software in which confidence is justified (expressed on a scale of 1 to 4) that the (software) element safety function will not fail due to relevant systematic failures.

# Safety Integrity Levels: IEC 62304

- Only severity of harm is used for the estimation of risks.
- There are 3 *safety classes*:
  - *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).
- For each requirement of the standard, it is indicated what classes it applies to.

# Process vs. Product Requirements

- IEC 62304: Only tasks (processes) are required to be documented.
- IEC 61508-3:
  1. Software artifacts are defined as the outputs of lifecycle phases.
  2. Then, techniques and measures are suggested for each artifact dependent on SIL.
  3. Target attributes are defined for each artifact.
  4. Then, for each technique/measure, the standard estimates the effectiveness with which an attribute is achieved.

# Towards Product Requirements: IEC 61508-3

| Safety lifecycle phase | | Objectives | Scope | Require-ments subclause | Inputs (information required) | Outputs (information produced) |
|---|---|---|---|---|---|---|
| Figure 4 box number | Title | | | | | |
| 10.1 | Software safety requirements specification | To specify the requirements for safety-related software in terms of the requirements for software safety functions and the requirements for software systematic capability;<br><br>To specify the requirements for the software safety functions for each E/E/PE safety-related system necessary to implement the required safety functions;<br><br>To specify the requirements for software systematic capability for each E/E/PE safety-related system necessary to achieve the safety integrity level specified for each safety function allocated to that E/E/PE safety-related system | PE system; software system | 7.2.2 | E/E/PE safety requirements specification as developed during allocation (see IEC 61508-1)<br><br>E/E/PE system safety requirements specification (from IEC 61508-2) | software safety requirements specification |

# Techniques/Measures for Software Safety Requirements Specification (SSRS): IEC 61508-3

| Technique/Measure * | | Ref. | SIL 1 | SIL 2 | SIL 3 | SIL 4 |
|---|---|---|---|---|---|---|
| 1a | Semi-formal methods | Table B.7 | R | R | HR | HR |
| 1b | Formal methods | B.2.2, C.2.4 | --- | R | R | HR |
| 2 | Forward traceability between the system safety requirements and the software safety requirements | C.2.11 | R | R | HR | HR |
| 3 | Backward traceability between the safety requirements and the perceived safety needs | C.2.11 | R | R | HR | HR |
| 4 | Computer-aided specification tools to support appropriate techniques/measures above | B.2.4 | R | R | HR | HR |

Table: Software safety requirements specification, Table A.1 from IEC 61508-3 (normative)

# Semi-formal methods for SSRS: IEC 61508-3

| | Technique/Measure * | Ref | SIL 1 | SIL 2 | SIL 3 | SIL 4 |
|---|---|---|---|---|---|---|
| 1 | Logic/function block diagrams | See Note 1 | R | R | HR | HR |
| 2 | Sequence diagrams | see Note 1 | R | R | HR | HR |
| 3 | Data flow diagrams | C.2.2 | R | R | R | R |
| 4a | Finite state machines/state transition diagrams | B.2.3.2 | R | R | HR | HR |
| 4b | Time Petri nets | B.2.3.3 | R | R | HR | HR |
| 5 | Entity-relationship-attribute data models | B.2.4.4 | R | R | R | R |
| 6 | Message sequence charts | C.2.14 | R | R | R | R |
| 7 | Decision/truth tables | C.6.1 | R | R | HR | HR |
| 8 | UML | C.3.12 | R | R | R | R |

Table: Semi-formal methods for SSRS, Table B.7 from IEC 61508-3 (informative)

# Techniques/Measures and Properties of SSRS: IEC 61508-7

| Technique/Measure | | Properties | | | | | |
|---|---|---|---|---|---|---|---|
| | | Completeness with respect to the safety needs to be addressed by software | Correctness with respect to the safety needs to be addressed by software | Freedom from intrinsic specification faults, including freedom from ambiguity | Understandability of safety requirements | Freedom from adverse interference of non-safety functions with the safety needs to be addressed by software | Capability of providing a basis for verification and validation |
| 1a | Semi-formal methods | R1<br>Application-friendly or domain specific specification method and notation used by domain experts | R1<br>Application-friendly or domain specific specification method and notation used by domain experts<br>R2<br>Verification of specification according to coverage criteria | R1<br>Method and notation that helps avoid or detect internal inconsistency, missing behaviour or mathematically inconsistent expressions.<br>R2<br>Verification of specification according to coverage criteria<br>R3<br>Verification of specification based on systematic analysis, and / or systematic avoidance of particular types of intrinsic specification faults | R1<br>Defined notation that restricts opportunity for misunderstanding<br>R2<br>Application of complexity limits in specification | – | R2<br>Defined notation that reduces ambiguity in specification |

Table: Properties for systematic safety integrity - Software safety requirements specification, Table C.1 in IEC 61508-3 (informative)

# Properties of SSRS: IEC 61508-7 (informative)

| | | |
|---|---|---|
| 1.2 | Correctness with respect to the safety needs to be addressed by software | The Software Safety Requirements Specification providing an appropriate answer to the safety needs and constraints assigned to the Software.<br>The objective is to assure that what is specified will really guarantee safety in all the necessary conditions. |
| 1.3 | Freedom from intrinsic specification faults, including freedom from ambiguity | Internal completeness and consistency of the Software Safety Requirements Specification: providing all necessary information for all the functions and situations that can be derived from its statements; expressing no contradicting or inconsistent statements.<br>Contrary to completeness and consistency with respect to safety needs, internal completeness and consistency can be assessed based on the Software Safety Requirements Specification only |

# Implicit Assurance Cases Behind Standards

- IEC 61508-3:
  - The top claim is the software is acceptably safe.
  - The rationale can be found in the use of well-defined and carefully planned safety lifecycle activities, management of functional safety, independent functional assessment, use of competent staff.
  - Outputs of each phase together with their attributes define required evidence.
- IEC 62304:
  - The top claim is that the medical device software is acceptably safe and effective.
  - The rationale is that a set of well-established processes has been followed.
  - Evidence is found in process documentation.

# Assurance Cases in Standards

- Safety cases in ISO-TR 80002:
  - “One could view a safety case as a risk management or residual risk summary with references to more detailed documentation for supporting information and the evidence in the risk management file. It could also include cross references to demonstrate specification and test coverage for all risk control measures.”
- ISO 15026:
  1. ISO 15026-1:2010 Concepts and vocabulary
  2. ISO 15026-2:2011 Assurance case
  3. ISO 15026-3:2011 System integrity levels
  4. ISO 15026-4 Assurance in the life cycle (Under development)

# Conclusions

- Important concepts are still being vaguely and inadequately defined/used in the standards (SILs, product requirements).
- The use of assurance cases has not yet been standardized in the medical domain.
- New standards emerge addressing the increasing complexity and connectivity of devices (e.g., IEC 80001-1:2010).

# Acknowledgments


- Marc Bender, Chris George, Paul Joannou, Zarrin Langari, Mark Lawford, Tom Maibaum, Jackie Wang, Alan Wassyng