SW-Documentation

Seminar

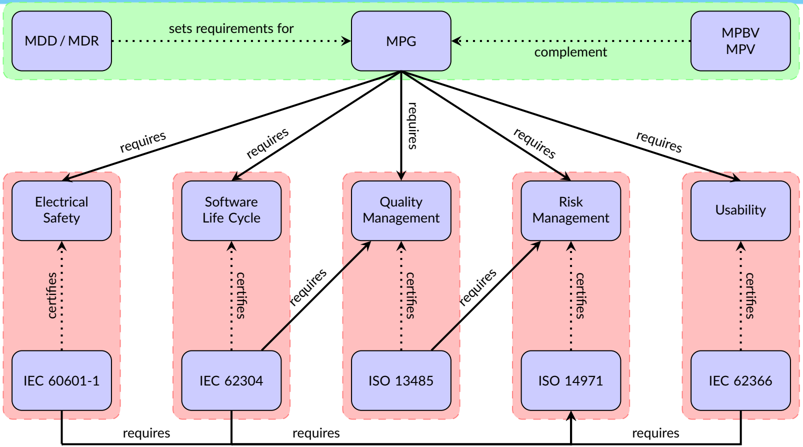
Andreas Böhler

COTIDIANA

Agenda

1. Legal Background
2. Software Safety Classification
3. EN 62304
4. Software Risk Management

Legal Background



Software Safety Classes

- The EN 62304 defines three different safety classes
- Similar to Medical Device Class, based on the possible effects for
  - patient
  - operator
  - other people
- resulting from a hazard to which the software system can contribute to

Software Safety Classes

- Many people confuse Medical Device Class and Software Safety Class
- Here, classification is done using **letters**, not numbers
  - A No injury or damage to health is possible
  - B Non-serious injury is possible
  - C Death or serious injury is possible
- Initially, before qualification, the software is treated as Class C
- External risk control measures can reduce the safety class by one level

External Risk Control Measures

can be either hardware or another software system or health care procedures that reduce the risk!

Software Safety Classes: Amendment 1

- Amendment 1 made an important change to the safety classification system
- Amendment 1: Allows to take the **probability** into account, thus is risk-based
- Risk control measures **outside** of the software system can be included in the decision
- There is now a decision tree for the classification, the definitions of the safety classes have also changed

Probability of Software Failure

Attention: The probability of a software failure should still be assumed with 100 %!

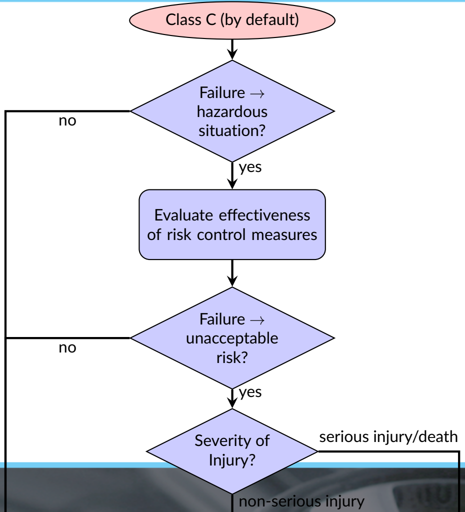
Software Safety Classes: Amendment 1

A 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

B 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

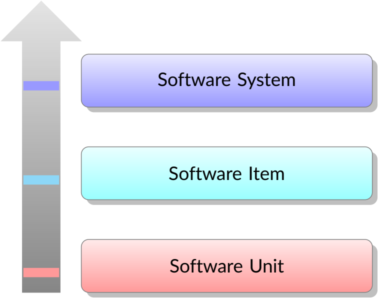
C the software system can contribute to a hazardous situation which results in unacceptable risk after consideration of risk control measures external to the software system and the resulting possible harm is death or serious injury

Software Safety Classes: Amendment 1


Class C (by default)
Failure → hazardous situation?
no
yes
Evaluate effectiveness of risk control measures
Failure → unacceptable risk?
no
yes
Severity of Injury?
serious injury/death
non-serious injury

Software System, Items and Units

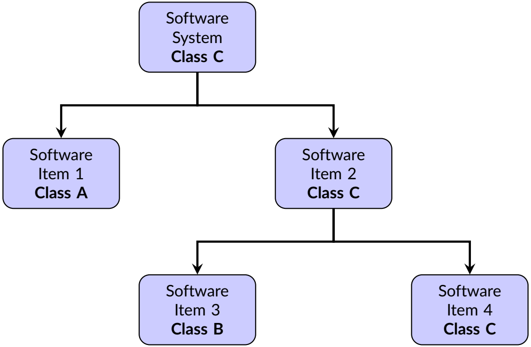
- It makes sense to decompose bigger systems into smaller parts
- A Software Systems consists of Software Items
- A Software Item consists of Software Units
- Manufacturer can define the granularity



Software Safety Classes

- Documentation about safety classification is part of the risk management file
- By default, each Software Item or Software Unit inherits parent's safety classification
- Manufacturer **can** classify a Software Item or a Software Unit differently (Segregation)
- Needs to be documented
- Good reason for it

Software Segregation


Software System **Class C**
Software Item 1 **Class A**
Software Item 2 **Class C**
Software Item 3 **Class B**
Software Item 4 **Class C**

Consequence

Software Safety Classification leads to

- Certain parts of the standard EN 62304 are only applicable to certain safety classes
- Annex contains a summary table, some examples:
  - Unit/Integration Testing not required for Class A
  - Software development methods, standards and tools planning only required for Class C
  - Detailed design verification only required for Class C
- Class C requires all aspects of the standard
- Class B requires most aspect of the standard
- Class A requires only minimal aspects of the standards

EN 62304: Editions

- The first edition was released in 2006 as IEC 62304:2006
- It was immediately harmonized as EN 62304:2006
- In 2008, a corrigendum was made
- In 2015, amendment 1 (A1) was released and published
- Currently, the second edition of EN 62304 is (still) in draft status and was expected for 2017

Current Edition

The current edition is EN 62304:2006+A1:2015

EN 62304: Introduction

- Lists reasons for the creation of the standard
- Details relationship to EN ISO 14971 and Quality Management Systems (Annex C)
- Clarifies some words
  - shall: compliance is mandatory
  - should: compliance is recommended, but not mandatory
  - may: permissible way to achieve compliance
  - establish: define, document and implement
  - “as appropriate”: manufacturer needs to do so unless he can document a justification for not doing so

EN 62304: Scope

- Purpose: “provide a development PROCESS that will consistently produce high quality, safe MEDICAL DEVICE SOFTWARE”
- Describes
  - Processes: Biggest sequence of “things to do”
  - Activities: Smaller part of a Process
  - Tasks: Smallest part of an Activity

EN 62304: Processes

The are five main processes:

- Software Development Process
- Software Maintenance Process
- Software Risk Management Process
- Software Configuration Management Process
- Software Problem Resolution Process

EN 62304: Annex A

- Annex A details the rationale for the standard
- Summarises the requirements by class

Table A.1

This table is one of the most important tables, as it summarises the applicable parts of the standard depending on the Software Safety Class!

EN 62304: Annex B

- Annex B gives guidance on how to provision the standard
- Gives more in-depth and background information than the standard's text
- Lists in a very detailed way how to handle **legacy software**

Legacy Software

is software that has been brought to the market before March 2010, i.e. before the standard was absolutely required. If a manufacturer needs to work with the old software, he has to deal with it as **legacy software**.

EN 62304: Annex C

- Annex C lists the relationship to other standards
- The list includes, but is not limited to
  - EN ISO 14971
  - EN ISO 13485
  - EN 60601-1
  - EN 62366-1
  - EN 82304-1
- A table lists the relationship to the requirements of the EN 60601-1

EN 62304: Annex D

- Annex D gives further implementation help
- Contains a checklist for small manufacturers without a certified Quality Management System

Software Documentation

- The EN 62304 requires, among others, the documentation of
  - Requirements
  - Architecture [Class B and C]
  - Detailed Design [Class C]
  - Verification
  - Validation
  - Risk Management Activities → EN ISO 14971
  - Usability Engineering Activities → EN 62366-1
- Every step need step needs to be planned (e.g. Software Development Plan)
- and documented (e.g. Software Test Report)

Software Configuration Management

- All artefacts from software development need to be clearly identified and traceable
- Use of Version Control System is strongly recommended
- Documentation can be done in an electronic system (issue tracker, Wiki ...) as long as QM requirements are met

Traceability

The traceability of all artefacts is one of the key factors for successful medical device software development

Motivation

Failure to use risk management.

Source: http://www.wainwright.army.mil/safety/risk_management.htm

Legal Background

There is a separate standard regarding risk management for medical devices:

Risk Management

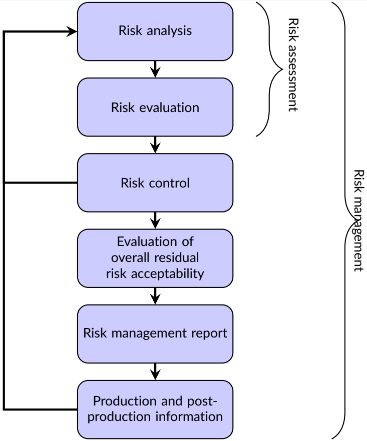
EN ISO 14971 Application of risk management to medical devices

Furthermore, there is a technical report on the application of ISO 14971 to medical device software

Guidance

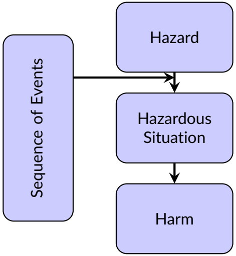
IEC/TR 80002-1

Risk management process



Hazard, Hazardous Situation and Harm

Harm can only occur, if a **hazardous situation** is present which results in **harm** by a **Sequence of events**.



A hazard alone cannot lead to harm without a sequence of events!

Software

- Software is considered a special case in terms of risk management
- Software **cannot** result in **direct** harm:
  - Software cannot be touched
  - Software is not poisonous
- However, Software can play a role in the series of events

Software

Software can only contribute to a hazardous situation, but can never be a hazard!

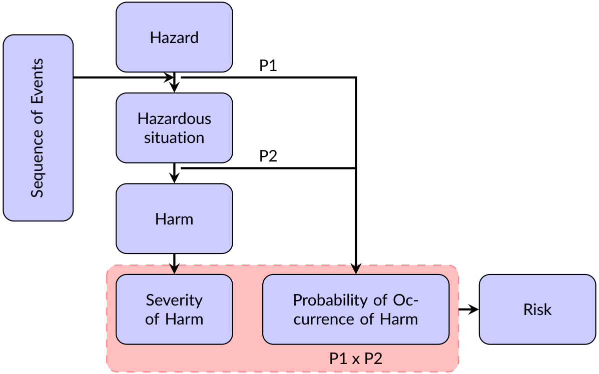
Probability

- Software-Errors in a specific version occur in all copies of the software
- Probability can be very difficult to estimate, as there are a lot of possible inputs and states
- There is **no satisfactory** way to estimate the probability of a software error
- Software-Errors in a series of events should therefore be considered with 100 % (1) → Worst-Case-Scenario

Probability

If the probability of occurrence of harm cannot be estimated, only the severity can be taken into account for estimating the risk

Interplay



With probability *P1* a hazardous situation results, with probability *P2* this results in actual harm.

Questions?

Questions

Feel free to ask questions any time!

Contact

R'n'B Consulting GmbH
Grillparzerstraße 2/29
4020 Linz
Austria

office@rnb-consulting.at
www.rnb-consulting.at