Introduction into IEC 62304
Software life cycle for medical devices

Christoph Gerber
4. September 2008
SPIQ
Agenda

- Current Picture

- Regulatory requirements for medical device software

- IEC 62304 Overview

- IEC 62304 Key concepts

- Summary

- References

- Q&A
Current Picture
Current picture in medical device industry

- Dramatic increase in compliance and regulatory requirements
- Diverse regulatory requirements for different countries
- Increased number of recent recalls were software related
- Increasing number of medical devices which are pure SW products
- FDA raises expectation on software testing methodologies
- Opinion that software development in medical device industry is behind other mission critical industries such as aviation
- For vendor:

No choice be compliant!
Regulatory Requirements
FDA (US) requirements on medical device software

Quality System Requirements (QSR) aka GMP

- for medical devices
  - 21 CFR § 820.30 Design Control:
    Design validation shall include software validation and risk analysis …

- software used in manufacturing and process control
  - 21 CFR § 820.70 production and process controls:
    When computers or automated data processing systems are used as part of production or the quality system, the manufacturer shall validate computer software for its intended use according to an established protocol.

- in general
  - 21 CFR PART 11 Electronic Records & Signatures
FDA compliance

- Application of voluntary regulatory standards:
  - Once a manufacturer chooses to claim compliance with a voluntary standard, that claim is legally binding
  - Notified bodies and competent authorities use the recognized standard as a yardstick against which to measure the manufacturer against manufacturer’s method

- For medical device software
  - ISO 13485, ISO 14971, IEC 62304

- Software used in manufacturing and process control
  - ISO 13485
  - GAMP5
    - Off-the-shelf
    - Customized
    - Custom made
EU requirements on medical device software

- **Medical Devices Directive 93/42/EWG** (MDD)
- **Medical Devices Act** (MPG)
- **Conformity Assessment based on classification**
  - Notified Body
  - CE 0483
- **Harmonized Standards**
  - EN ISO 13485
    - Medical devices - Quality management systems
  - EN ISO 14971
    - Application of risk management
  - EN 60601-1-4
    - Programmable electrical medical systems
  - EN 60601-1-6 ➔ IEC 62366
    - Usability
IEC 62304 Overview
IEC 62304 – key facts

- Medical device software – software life cycle processes
- Successor of AAMI SW68 (US national standard)
- IEC since May 2006
- EN since March 2007
- Harmonized in EU standard 93/42/EWG (MDD) soon
- Plugs into IEC 60601-1 Edition 3 others will follow
- Development driven by FDA recognized by FDA
- Likely to emerge as the single global standard for medical device software engineering
- Comply once … use many times!
- Guidance paper being prepared
IEC 62304 – Relationship to other standards

- Medical device management standards
  - ISO 14971
  - ISO 13485

- Medical device process standard
  - IEC 62304
  - Gives detailed direction on how to develop and maintain safe software system

- Other sources of information
  - IEC/ISO 12207, IEC 61508-3, IEC/ISO 90003, ...
  - Gives additional guidelines, techniques, etc that may be used

- Lays out a foundation to develop a medical device

IEC 60601-1-4
- 3rd Edition
- Replaces 1-5

- Medical device product standards
  - IEC 60601-1,
  - IEC 61010-1

- Gives specific direction for creation of a safe medical device

- Implementation of medical device software

IEC 62366
- Usability

- Affects

- Requires

- Inspires
IEC 62304 – General requirement

- There is no known method to guarantee 100% SAFETY for any kind of software. There are three major principles which promote SAFETY for MEDICAL DEVICE SOFTWARE:
  
  ➢ RISK MANAGEMENT
  
  ➢ QUALITY MANAGEMENT
  
  ➢ SOFTWARE ENGINEERING
IEC 62304 – Scope

**Purpose**

This standard defines the **life cycle requirements** for medical device software. The set of

- **processes**,
- **activities**, and
- **tasks**

described in this standard establishes a common framework for medical device software life cycle processes.

**Field of Application**

This standard applies to:

- **the development and maintenance of medical device software**, 
- to the development and maintenance of medical device software when software **is itself a medical device** or when software **is an embedded or integral part** of the final medical device, 
- does not cover validation and final release of the medical device, even when the medical device consists entirely of software.
IEC 62304 – Out of scope

- Does not prescribe how to accomplish requirements
- Does not require a specific software life cycle
  - Waterfall
  - Incremental
  - Evolutionary
- Does not specify documents

- What is a medical device??
  - Also in scope supporting tools (I&C) for the medical device
  - Internal process / manufacturing software are not medical devices but process can be used as well – voluntary standard
IEC 62304 – Core processes

- Software development process
- Software maintenance process
- Software risk management process
- Software configuration management process
- Software problem resolution process
IEC 62304
Key Concepts
IEC 62304 – Key concepts

- Safety Classification of Software System and Software Items
- Software Risk Management
- Unknown Software (SOUP)
- The software life cycle doesn’t end with product release
  - Maintenance
  - Problem resolution
Software Safety Classification
IEC 62304 – Software safety classification

- **RISK**: combination of the severity of injury and the probability of its occurrence
  - no consensus on how to determine the probability of occurrence of software failures using traditional statistical methods.
  - therefore, SOFTWARE SYSTEM classification is based on the severity of the HAZARD resulting from failure of the software, assuming that the failure will occur (100% probability)

- Software safety class for **Software System** and **Software Items** according to the possible effects on the **patient**, **operator**, or other people resulting from a HAZARD to which the SOFTWARE SYSTEM can contribute.
  - 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
IEC 62304 – Software safety classification

- SERIOUS INJURY: injury or illness that directly or indirectly:
  a) is life threatening,
  b) results in permanent impairment of a body function or permanent damage to a body structure, or
  c) necessitates medical or surgical intervention to prevent permanent impairment of a body function or permanent damage to a body structure

- NOTE: Permanent impairment means an irreversible impairment or damage to a body structure or function excluding trivial impairment or damage.
IEC 62304 – Software safety classification?

- determines the PROCESSES to be used during
  - the development; and
  - maintenance of software.
- e.g., architecture for class B, C and component test for class B, C
  initialization of variables as SW verification acceptance criteria for C
- less rigor process for class A software!

- when a **software system is decomposed into software items**, such **software items shall inherit the software safety classification of the original software item** (or software system)
  - unless the manufacturer documents a rationale for classification into a different software safety class.
IEC 62304 – Assign safety class to software items

- Safety Classification Principles:
  - No adverse side effects caused by X and W.
  - No hazard contributing effects by X and W.
  - Rationale for classification of X and W required!
  - Z includes all software system contributions to hazards.
  - Software system inherits “worst” safety class.
IEC 62304 – Software safety classification – best practice

- Segregation of critical SW
  - clear communication interfaces helps for rationale
  - cluster critical SW

- Middle grained approach for item
  - to fine gets impractical
  - to coarse gets in trouble with SOUP and SW risk management

- Foster reuse of critical components
  - to save cost

- Introduce automated (unit) testing
  - i.e. with continuous integration
Software Risk Management
IEC 62304 – Software risk management

- Software risk management in addition to:
  - ISO 14971 Risk Management!

- additional requirements for RISK CONTROL for software,
  - including software that has been identified during the RISK ANALYSIS as potentially contributing to a hazardous situation,
  - or software that is used to control MEDICAL DEVICE RISKS.

- Rationale behind it:
  - SW developers needs to understand minimum requirements for RISK CONTROL measures implemented in software;
  - ISO 14971 does not address RISK CONTROL of software

- Corresponds to Design Risk Management (Bottom Up)
- Today’s practice typically Functional Risk Management (Top Down)
ISO 14971 – Risk management

Risk = combination of the likelihood (L) of occurrence of harm and the severity (S) of that harm [ISO 14971]
IEC 62304 – Software risk management principle

- same principle for HW and electronic Design-FMEA
- identify software items contributing to hazardous situation
- requires knowledge of architecture
IEC 62304 – Software risk management using FMEA tool

- Used in hardware development
- Most common risk analysis method
- Analyzes the effect of component failure
  - Bottom-up analysis
- Presented in tabular format:
  - Failure Mode
  - Effect on System
  - Cause of Failure
  - Method of Control
- Done by running many meeting with a variety of experts

<table>
<thead>
<tr>
<th>Ref.#</th>
<th>Item/Function</th>
<th>Failure Mode</th>
<th>Effect on System</th>
<th>Cause</th>
<th>Methods of Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Surgery Case</td>
<td>Does not finish in time</td>
<td>OR time exceeded</td>
<td>Unreliable interrupt source; too much computation; interference from other tasks</td>
<td>Check for loop overrun; system performance measurements; choice of operating system</td>
</tr>
<tr>
<td>2</td>
<td>Image import</td>
<td>Wrong orientation</td>
<td>Wrong location penetrated</td>
<td>Image information not readable</td>
<td>Exception if information is not available</td>
</tr>
</tbody>
</table>
IEC 62304 – Software risk management example process

1. User Needs
2. Device Risk Management
3. Software Requirements Analysis
4. Software Risk Management
5. Architectural Design
6. Detailed Design Implementation
7. Ongoing Activity
8. Software Requirements Specification (+ Risk Control Measures)
9. Architectural Design Description

Risk Analysis
1) on Functions
2) on Components

Documented in

- User Needs
- Device Risk Management
- Software Requirements Analysis
- Software Risk Management
- Architectural Design
SOUP
IEC 62304 – SOUP concept

- SOUP – Software of Unknown Provenance
  - that is already developed and generally available and that has not been developed for the purpose of being incorporated into the MEDICAL DEVICE (also known as “off the-shelf software”)
  - or software previously developed for which adequate records of the development PROCESSES are not available

- Additional requirements for SOUP
  - Configuration management of SOUP: vendor, title, version, …
  - Specify functional and performance requirements of SOUP item
  - Specify SYSTEM hardware and software required by SOUP item
  - include in software risk management
  - Evaluate list of anomalies if failure or unexpected result is contributing to a hazardous situation
IEC 62304 – SOUP best practice

- Include SOUP requirements in deal with 3rd party SW supplier
  - shall provide bug list
  - shall allow audits
  - prepare for increased requirements
  - knock out criteria for many large standard COTS suppliers

- Specify scope of SOUP
  - template for SOUP description
  - for example, also practical for declaring license policy

- Configuration management
  - add SOUP software to your SCCS
IEC 62304
QM System Mapping
IEC 62304 – QM System mapping example

<table>
<thead>
<tr>
<th>Process</th>
<th>SOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW development process</td>
<td>Software Life Cycle</td>
</tr>
<tr>
<td>SW maintenance process</td>
<td>Change Management Process / CAPA process</td>
</tr>
<tr>
<td>SW risk management process</td>
<td>Risk Management Process</td>
</tr>
<tr>
<td>SW configuration management process</td>
<td>SCCS</td>
</tr>
<tr>
<td>SW problem resolution process</td>
<td>During development ➔ Issues Management</td>
</tr>
<tr>
<td></td>
<td>After release ➔ Change Management / CAPA</td>
</tr>
</tbody>
</table>

SOP = Standard Operating Procedure
Summary
Summary

- IEC 62304 emerging as the de facto standard in medical software
- Many FDA 510(k) submission reference already IEC 62304
- Adopts safety elements from defense industry (DOD)
- Understanding not trivial at a first glance, but seems practicable
- FMEA tool from hardware engineering makes sense for SW as well
- SOUP handling implies trade-off between make or buy
- Class C requirements are hard to achieve for standard SW like mainstream OS and standard IDE’s
- Not much difference between class C and B in regard to effort
- Where does software end and electronics begin
  - firmware, device drivers, micro code, FPGA, CPLD
- Not a standalone engineering standard requires ISO ….
References

- IEC 62304:2006 Medical device software -- Software life cycle processes
- IEC 62366:2007 Medical devices -- Application of usability engineering to medical devices
- Code of Federal Regulations TITLE 21 PART 820 QUALITY SYSTEM REGULATION
- 21 CFR Part 11: Electronic Records and Signatures
- Design Control Guidance For Medical Device Manufacturers
  (March 11, 1997) [http://www.fda.gov/cdrh/comp/designgd.html]
- General Principles of Software Validation
  (January 11, 2002), [http://www.fda.gov/cdrh/comp/guidance/938.html]
- Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices
  (May 11, 2005), [http://www.fda.gov/cdrh/ode/guidance/337.html]
- Off-The-Shelf Software Use in Medical Devices
  (Sep. 9, 1999), [http://www.fda.gov/cdrh/ode/guidance/585.html]
- Cybersecurity for Networked Medical Devices Containing Off-the-Shelf (OTS) Software
- Computerized Systems Used in Clinical Trials
  (April 1999) [http://www.fda.gov/ora/compliance_ref/bimo/ffinalcct.htm];
  (May 2007) [http://www.fda.gov/cber/gdlns/compclintrial.htm]
Thank you!

Christoph Gerber
Manager Software Engineering
Stryker Navigation
Stryker Leibinger GmbH & Co. KG
Bötzinger Str. 41
D-79111 Freiburg, Germany
t: + 49 761 4512 362
f: + 49 761 4512 449 362
c: + 49 160 369 2203
christoph.gerber@stryker.com