IEC 62304 for medical device software development – Steps to Compliance

May 15, 2013

Marty Bakal, Electronics Industry Leader, IBM Rational software
# Five forces affecting the industry today

<table>
<thead>
<tr>
<th>Legal issues</th>
<th>Markets and financial issues</th>
<th>Technology</th>
<th>Regulations</th>
<th>New product development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient lawsuits</td>
<td>Consolidation</td>
<td>Systems integration</td>
<td>FDA</td>
<td>Speed to market</td>
</tr>
<tr>
<td>Patents</td>
<td>Overseas marketplace expansion</td>
<td>New technologies</td>
<td>New forms of regulation</td>
<td>Strategic agreements</td>
</tr>
<tr>
<td>Product recalls</td>
<td>Slowing marketplaces</td>
<td>Compliance</td>
<td>Patient privacy, HIPAA</td>
<td>Significant R &amp; D spend</td>
</tr>
<tr>
<td>Liability</td>
<td></td>
<td></td>
<td>SOX</td>
<td></td>
</tr>
<tr>
<td>Data retention</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Healthcare is shifting to a patient-centric model
IEC 62304 Overview – Stronger focus on Software

- **IEC 62304:2006 Medical device software – Software life cycle processes**

- Focused on software development processes for medical devices but does not specify the methodologies, artifacts or life cycle models themselves

- Derived from ISO/IEC 12207, a general standard for software processes

- **Adoption**
  - FDA Consensus Standard since September 2008
  - FDA regards complying with IEC62304 as fulfilling “Software Development Environment Description” section of the Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices
  - Normative standard in Europe for conformance marking

- Standard available for purchase from ISO website (~$225 USD)
IEC 62304 Structure

- **Processes**: Set of interrelated or interacting activities that transforms inputs into outputs

- **Activities**: Set of interrelated or interacting tasks

- **Tasks**: Single piece of work that needs to be done and results in a deliverable
What IEC 62304 does **not** do

- **Does not** specify an organizational structure
  - You can have a hierarchical, matrix, or mixed organization

- **Does not** specify the content of the documentation to be developed
  - You need to show traceability through all the artifacts but not in some set format

- **Does not** prescribe a specific lifecycle model
  - Waterfall, Iterative, Agile, … it is all up to you
Standards Landscape and Process

- Quality management system
- RISK MANAGEMENT
- Software safety classification
- Software development PROCESS
  - Software development planning
  - Software requirements analysis
  - Software ARCHITECTURAL design
  - Software detailed design
  - SOFTWARE UNIT implementation and verification
  - Software integration and integration testing
  - SOFTWARE SYSTEM testing
  - Software release
- Software maintenance PROCESS
- Software RISK MANAGEMENT PROCESS
- Software configuration management PROCESS
- Software problem resolution PROCESS
- Documentation Requirements

Source: European Medical Device & Technology, June 2010
Overview of SW Development / Maintenance

PROCESSES and ACTIVITIES – as defined in the IEC 62304 standard

- Development / Maintenance Planning
- Requirements / Problem & Modification Analysis
  - ARCHITECTURAL design
  - Detailed design
- UNIT Implementation and VERIFICATION
- Integration and integration testing
- SYSTEM testing
- Software release

Software configuration management

Software problem resolution

Customer Needs / Maintenance Satisfied

Customer Needs / Maintenance
### Product Development and Verification Life Cycle (Process)

#### DEFINITION / DEVELOPMENT

- **Requirements Capture and Analysis**
  - DOORS
- **System Analysis and Design**
  - Rhapsody
  - DOORS
- **Software Design**
  - Rhapsody
- **Implementation**
  - Rhapsody
- **System Analysis and Design**
  - Rhapsody
- **Software Design**
  - Rhapsody

#### TEST / VERIFICATION

- **Validating the Product**
  - Traceability for Test Coverage
- **Verifying the System**
- **System Acceptance**
  - RQM
- **System/Subsystem Integration and Test**
  - RQM, Rhapsody

---

**Integrated coverage of the software and systems engineering lifecycle (includes compliance coverage for SOUP)**

**Configuration Management**
- RTC

**Change Management and Problem Reporting**
- RTC

**Automate Document Generation**
- RPE
Step 1 - Gap Analysis: Rational DOORS

Determine compliance with IEC 62304 by performing gap analysis

1. Capturing your existing processes in a solution for tagging (Rational DOORS)
2. Identify and tag each of the respective control points in your existing process
3. Capture the IEC 62304 standard as the yardstick to evaluate each of "your processes" control points (Rational DOORS)
4. Display a traceability matrix between the process standard (IEC 62304) and your process
   - Identify and remediate process gaps
   - Document updated process into Rational Method Composer
     - Allows change control, version control, publication, and general oversight of process changes as the process matures.
Quality Management System

Make quality management a continuous lifecycle activity

"The manufacturer of medical device software shall demonstrate the ability to provide medical device software that consistently meets customer requirements and applicable regulatory requirements."

Demonstration of this ability can be by the use of a quality management system that complies with ISO 13485

- Unify the entire team with a shared view of quality assets
- Integrates with Requirements Management to insure Customer needs are met
- Intelligent automation to improve accuracy and efficiency
- Automated reporting to enhance project decision-making and compliance
Step 3 - Risk Management: Rational DOORS

- Anticipate possible failures of the system
  - Define control measures
    - Inherently safe, Preventive, Corrective, Informative
- Systematic risk analysis is to anticipate failures
  - Top-down: Function analysis - ISO 14971
    - Hazard Analysis
  - Bottom-up: Design Analysis – FMEA, FTA
    - Failure Modes and Effects Analysis
- Each failure leads to risk control (RCM) measures
- Each RCM leads to requirements implemented in product hardware, software or documentation
- Risk Management File documents traceably risk to control measure, to verification of control measure
- Risk Management Activities continue after release

Document TRACEABILITY of software HAZARDS
- From hazardous situation to the SOFTWARE ITEM
- From SOFTWARE ITEM to the specific software cause
- From the software cause to RISK CONTROL measure
- From RISK CONTROL measure to VERIFICATION of
- RISK CONTROL measure
Step 4 - Safety Critical with MDD: Rational Rhapsody

Software safety classification

- Typical Safety Critical Workflow
  - Implement code from textual requirements
  - Test only on target late in development cycle
- Safety Critical with Model-Driven Development
  - Consistent Design, Code and Documentation
  - Visualization of complex requirements
  - High quality code generation (tool dependent)
  - Test Driven Development support
    - Early functional verification on host, detect bugs early in development
  - Harmony for Embedded RealTime™ process defines a safety workflow and provides guidance
  - Safety analysis profile supports FTA, FMEA, FMECA and Hazard analysis – supports safety classification and compliance to section 4.3 of IEC 62304
Traceability - Encourage Collaboration Across the Lifecycle

- Multi-level graphical analysis

- Dynamic traceability in columns
Agile Development

- Grifols Case study
• Grifols: Agile Case Study
  - Plasma derivatives, in vitro diagnostic products and pharmaceutical products.
  - €2,600M total turnover (40% US, 40% EU, 20% ROW).
  - €130M diagnostic division turnover.
Waterfall development – Agile development

- Moves in stages.
- Proceeds when stage is completed.
- Allows some feedback.

- Emphasis in working software.
- Time boxed iterations.
- Continuous change and adaptation.

Scrum Diagram Source: Scott Ambler
### Making both worlds work together: conflicts and trade-offs

<table>
<thead>
<tr>
<th>Conflicts</th>
<th>Trade-offs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Inputs → Design Outputs</td>
<td><img src="Diagram1.png" alt="Diagram" /></td>
</tr>
<tr>
<td><img src="Diagram2.png" alt="Diagram" /></td>
<td><img src="Diagram3.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

- Guidance from AAMI TIR 45 on applying agile in the development of medical device software.
Grifols’ agile cycle

- Release plan
- SRS (spec) - Product Backlog
- Architectural design - Development environment

- Test Design
- Architecture & detailed design
- Code + unit & integration test
- Regression testing
- User story

- Daily meeting
- Daily build
  - Code audit
  - Unit test
  - Integration test
  - Smoke test

- System test procedures
- Smoke test procedures
- Regression test procedures
- SRS analysis, service definition
- Backlog prioritization

- SRS (formal approval)
- Architecture & detailed design (formal approval)
- Test procedures (formal approval)

- Defect fixing
- Sprint review & retrospective

- SPRINT 0
- Form approval

- Release planning

- SPRINT 1

- HARDENING SPRINT

- SPRINT 2

- HARDENING SPRINT (Q)

- SW release
Grifols - Agile success story

• Planning – building – hardening.
• Off-sprint tasks:
  – Backlog “grooming”.
  – Formal test procedures.
• Leveraged DOORS

- System test procedures
- Smoke test procedures
- Regression test procedures
- SRS analysis, stories definition
- Backlog prioritization

HARDENING SPRINT(s)

- Defect fixing
- System & regression test
- Final review & approval of inputs and outputs

SW release
Questions