Traditional Approaches to Risk Management and Medical Device Software

Are They Good Enough?
Can We Do Better?

David A. Vogel, Ph.D.
President
Intertech Engineering Associates, Inc.
dav@inea.com
David A. Vogel, Ph.D.
President and Founder
Intertech Engineering Associates, Inc

Dr. David Vogel is the founder and president of Intertech Engineering Associates, Inc., of Westwood, Massachusetts. Founded in 1982, Intertech has served the medical device industry by providing electronics hardware and software development services. Additionally, Dave and his Intertech engineering team have developed engineering processes for Intertech that facilitate product design compliance with FDA Quality System Regulations (QSR). The product development, verification and validation services provided by Intertech have ensured that software based instrumentation developed and/or tested by Intertech have had a 100% safety record.

Dave was selected to participate with a joint AAMI/FDA workgroup to develop a standard for Critical Device Software Validation which was subsequently included in the IEC 62304 Software Lifecycle Standard. He was also a participant on the joint AAMI/FDA workgroup to develop a Technical Information Report (TIR32:2004) for Medical Device Software Risk Management. Most recently his work with the AAMI/FDA workgroup concluded in the development of TIR36:2007 on the Validation of Software for Regulated Processes. Both are available at http://www.aami.org/publications.

A textbook covering some of the topics of today’s workshop was written by Dr. Vogel and was released in December 2010. The title of the book is “Medical Device Software: Verification, Validation, and Compliance” and is available from Artech Publishing at http://www.artechhouse.com/Detail.aspx?strBookId=2106, Amazon, and other booksellers. Details are available at www.softwarevalidationtext.com or from Intertech’s website www.inea.com.

As an AAMI instructor, Dave authored a 3 day workshop on Medical Software Validation that is presented to the public several times a year by AAMI (http://www.aami.org/meetings/courses/software.workshop.html)

Dr. Vogel received a bachelor’s degree in electrical engineering from Massachusetts Institute of Technology. He earned a master’s degree in bioengineering, a master’s degree in electrical and computer engineering, and a doctorate in biomedical engineering from the University of Michigan. He was named by MD&DI magazine as one of the “100 Notable People in the Medical Device Industry” in 2008.
Much of the content of this talk is covered in more detail in Chapters 8 and 18 of this textbook.

For more information, see [www.validationtext.com](http://www.validationtext.com)
I Don’t Trust Software
Opening Thoughts

• Do we as an industry *really* know what we are doing with Risk Management?

• With all the regulatory “help”, standards, technical information reports, guidances, webinars, workshops, assurance cases, software tools … has the industry concept and execution of risk management really improved?
Why Formal Risk Management Is Needed

1. As a design and development tool
2. As a validation tool
3. As a focus for continued post-market attention to RM
4. As a communication tool
   a. Internally
   b. With regulators

Which makes the devices safer?
Which is necessary for compliance?
Which do many companies worry about more?
Relationship Validation & Risk Management

Reduces likelihood of design flaw

Validation

Risk Management

Increases confidence that risks were analyzed and controlled
A Very Brief History

• FMEA had its roots in NASA in the 60’s and the auto industry in the 70’s and 80’s
• Heavily used in process FMEA originally, and for BOM level FMEA
• Standard 14971 for Risk and Medical Devices
• FDA recently added requirement for Safety Assurance Cases for infusion pump submissions.
Non-medical Listeners

QUICK BACKGROUND ON RISK MANAGEMENT IN MEDICAL DEVICES
Medical devices — Application of risk management to medical devices
- Analysis of Risk
- Evaluation of Acceptability
- Control of Unacceptable Risk
- Repeat Process Until Acceptable

Device Risk Management
- 80002-1 Application to Device Software
Harm, Hazards, and Risk

- Two components of probabilities
- Harm dictates the Severity
- Risk is the product of the two
- 14971 describes a method of Risk Management one Hazard at a time.

\[ R = P_1 \times P_2 \]

**NOTE**
- \( P_1 \) is the probability of a HAZARDOUS SITUATION occurring.
- \( P_2 \) is the probability of a HAZARDOUS SITUATION leading to a HARM.

From ISO14971 Medical devices – Application of risk management to medical devices 2007-03-01

No consensus exists for a method of estimating the probability of occurrence of a software failure. When software is present in a sequence of events leading to a HAZARDOUS SITUATION, the probability of the software failure occurring cannot be considered in estimating the RISK for the HAZARDOUS SITUATION. In such cases, considering a worse case probability is appropriate, and the probability for the software failure occurring should be set to 1. When it is possible to estimate the probability for the remaining events in the sequence (as it may be if they are not software) that probability may be used for the probability of the HAZARDOUS SITUATION occurring (P₁ in Figure 1). If this is not possible, the probability of the HAZARDOUS SITUATION occurring should be set to 1.

Estimates of probability of a HAZARDOUS SITUATION leading to HARM (P₂ in Figure 1) generally require clinical knowledge to distinguish between HAZARDOUS SITUATIONS where clinical practice would be likely to prevent HARM, and HAZARDOUS SITUATIONS that would be more likely to cause HARM.

From 80002-1: Sub-Clause 4.4.3 Probability
What Happens If We Ignore Probability?

• If Severity cannot be controlled, only Probability reduction can reduce risk
  – Testing
  – Engineering Practices
• Not a one time event – throughout the lifecycle of the software
• Diversity of tools and techniques

AAMI TIR32:2004 (R2016)
Medical device software risk management
Risk Analysis Often Treated as FMEA Only

<table>
<thead>
<tr>
<th>CODE MODULE / FUNCTION</th>
<th>POTENTIAL FAILURE MODE</th>
<th>POTENTIAL EFFECT(S) OF FAILURE</th>
<th>S E V</th>
<th>POTENTIAL CAUSE(S) OF FAILURE</th>
<th>O C C</th>
<th>DETECTION METHOD</th>
<th>D E T</th>
<th>R P N</th>
<th>RECOMMENDED ACTION</th>
<th>INDIVIDUAL/AREA RESPONSIBLE &amp; COMPLETION DATE</th>
<th>ACTION RESULTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 CODE MODULE NAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Returns software version</td>
<td>Returns nothing</td>
<td>no communication with instrument</td>
<td>5</td>
<td>Comm cable disconnected</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>corrupted eeprom or code</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Returns zero</td>
<td>Returns zero</td>
<td>communication failure</td>
<td>5</td>
<td></td>
<td>1</td>
<td>5</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101 REVISION DATE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Return Code build date</td>
<td>Returns nothing</td>
<td>no code date displayed</td>
<td>3</td>
<td>transmission error</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110 SYSTEM ON/OFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Get System Status</td>
<td>unknown / error status returned</td>
<td>no communication with instrument</td>
<td>3</td>
<td>Comm cable disconnected</td>
<td>1</td>
<td>7</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status OFF when should read ON</td>
<td>may reduce component life</td>
<td>transmission error</td>
<td>1</td>
<td>7</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status ON when should read OFF</td>
<td>Instrument non operational</td>
<td>transmission error</td>
<td>1</td>
<td>7</td>
<td>49</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note use of “OCC” for Occurrence to “solve” the Probability problem
Safety Assurance Cases

Goal 1
The Device is Safe to Use

Strategy 1
Argue that processes are followed that would lead to safe design, development, and production of the device

Strategy 2
Argue that all identifiable or anticipated hazards are sufficiently controlled for safe operation

Strategy 3
Argue over all Operational Hazards

Strategy 4
Argue over all Environmental Hazards

Goal 2
Operational Hazard 1 is controlled acceptably

Solution 1
Labeling

Solution 2
Active control measure 1

Goal 4
Operational Hazard 2 is controlled acceptably

Solution 3
Labeling

Solution 4
Active Control Measure 1
Questions for You

• How many of you are happy with your risk management process?

• Have you ever heard of someone “fudging” the risk priority numbers?
What’s Wrong with Risk Management?

1. Often compartmentalized and done in isolation
2. Seldom (until recently) incorporates usability information
3. Seldom utilizes diversity of tools throughout lifecycle
4. Difficulty dealing with overall residual risk
5. FMEA model is stretched beyond usefulness
6. Post-market data is seldom factored in
7. Most common models don’t deal well with hierarchical nature of harm/hazard/cause/…
1. Compartmentalization: Islands of Good RM Thought

- **Risk Management**
- **FMEA**
- **FTA**
- **Competitive Experience**
- **Post Market Experience**
- **Usability Data**
- **Historical Experience**

**EE**
**ME**
**SW**

RM – top down
FTA/FMEA – bottom up

2. Usability Data Seldom Used

- Viewed as distinct from other activities
- Often outsourced, not included in RM
- Often gathered late in development lifecycle
- Use error finally being recognized as a major contributor to hazards, and controls being treated seriously
3. Diversity of Tools Throughout Lifecycle

- Risk Management
- Fault Tree Analysis
- Safety Assurance cases
- Different tools for different disciplines
- What tools are used when?
- How are results merged?
Diversity of Activities

System Design
  • Past Product Experience
  • Clinical Input

Software Requirements
  • PRD
  • PHL
  • PHA

Software Architecture
  • SAD
  • SRHA
  • TDHA

Software Design
  • SAD
  • SRS
  • SRHA
  • TDHA

Code & Test
  • SDD
  • SAD
  • SRS
  • SDHA
  • DDHA

Integration
  • SST
  • CSHA
  • CSHA

System Test
  • TDHA
  • TRHA

Validation
  • SST

Traceability

Diversity of Activities
4. Dealing with Overall Risk Assessment

• 14971 – “There is no preferred method for evaluating overall residual risk and the manufacturer is responsible for determining an appropriate method”
  That is … you’re on your own.

• Hazard by hazard approach of 14971 does not lend itself to easy overall residual risk
5. The FMEA Model is Overused

- Often the only method used
- Risk = “combination of Severity (S) and Probability (P)”
- Often Detectability (D) also included
- Quantitative FMEA calculates Risk Priority Number (RPN) as
  \[ RPN = S \times P \times f(D) \]
- Risk controls are prioritized based on RPN
What’s Wrong With Severity?

Common Severity Scale

1. Discomfort
2. Reversible Injury
3. Permanent Injury
4. Death

- Scaling is linear

  If you would $1 not to be pinched, would you only pay $4 not to be killed?
What’s Wrong with Probability?

- Historically used for probability of process failure, mechanical failure, electrical failure, where failure is random (vs. systematic), and where probability has real meaning based on measurements or experience.

- Design failure (esp. software) is the probability that the engineer made a mistake … what kind of number do we put on that?
What is a Risk & Risk Control?

Risk
Combination of:
- **Probability** of Occurrence of Harm
- **Severity** of Resulting Harm

Risk Control
Mechanism to reduce:
- Probability of Occurrence of Harm
- Severity of Resulting Harm
Is Testing a Risk Control Measure?

- Reduces likelihood by unquantifiable amount
- Possible to assess residual risk?

Rely on testing after you have exhausted all other possible risk control measures.
Can’t:
• Quantify probability of software failure
• Quantify how much that probability is reduced

We Do Know:
• What makes software more likely to fail
• What to do to reduce the likelihood

Suggestion: Minimize arguments based on probability, because even severity control is often just shifting probabilities.
For high risk software (items)

A. Control measures to **reduce severity**
B. Control measures (largely validation activities) to **address each probability factor** that **increased** the **probability** of failure
Probability Factor?

- Not a “standard” term
- Proposing alternative methods
Risk is “Combination” of Likelihood and Severity. For example:

\[(P_1 P_2 P_3 P_4)S\]

where \(P_n\) is probability and \(S\) is severity

Consider Therac 25

- \(P_1\) – probability software fails (=1?)
- \(P_2\) – probability the failure results in radiation dosage error
- \(P_3\) – probability the dosage error is hazardous level
- \(P_4\) – probability the error is not prevented and results in harm
• Note the longer the causal chain
  – The more probabilities (P) are involved in sequence
  – Thus the P’s multiply
  – Since all P’s are ≤ 1, the more P’s involved the lower the overall probability P.
What Else Determines Likelihood of Software Failure?

Probability Factors (not a 14971 term):

- Number Independent Events Needed in Sequence Leading to Harm
- Size
- Complexity
- Amount of and quality of Testing
- Development and Validation Control Process
- Skill & Experience Level of Designer
- Skill & Experience Level of Tester
- History of Software
- Regression History
- Changes Late in Lifecycle
- Understanding of Functionality
- Complexity of User Interface
Detectability

• Holdover from Process FMEA, with built in assumption that trained process operators know remedial action upon detection.

• Very often misapplied to devices
  – Detectability is worthless *unless* a corrective risk control is triggered in response.
  – A car rolling over a cliff is highly detectable, but …
6. Post Market Data Not Used
7. Hierarchical Nature of Harms/Hazards/Causes

Layers of Harms/Hazards/Causes possible in real applications.
Software risk control measures:

- Make hazardous condition impossible by **design** (inherently safe design)
- **prevent** system hazards or their causes. (preventative)
- **detect and take corrective action** if system hazard is about to occur (corrective)
- **mitigate** (i.e. reduces severity of) damage if a hazard occurs. (mitigative)
- **Soft controls**: labeling, training, instructions for use.

Why is it important to understand the differences?
Layers of Harms/Hazards/Causes and risk control measures possible in real applications.
Example – Infusion Pump

**HARM**

- Overdose

**HAZARD**

- Pump won’t stop
- Pump runs too long
- Pump runs too fast

**CAUSES/CONTRIBUTING FACTORS (SW)**

- Runaway Processor
- Rx calculated wrong
- Timer error
- Encoder calc error
- Exception exit error
### Partial Risk Management Table

<table>
<thead>
<tr>
<th>Harm</th>
<th>Hazard</th>
<th>Cause</th>
<th>Risk Control Measure</th>
<th>Trace to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overdose</td>
<td></td>
<td>Severity limited by small reservoir</td>
<td>PRD</td>
<td></td>
</tr>
<tr>
<td>Pump won’t stop</td>
<td></td>
<td>Power from limited charge maxcap</td>
<td>Electrical</td>
<td></td>
</tr>
<tr>
<td>Runaway Processor</td>
<td></td>
<td>Watchdog Timer</td>
<td>System, SW, Electrical</td>
<td></td>
</tr>
<tr>
<td>Runaway Processor</td>
<td></td>
<td>Safety Processor Monitor</td>
<td>System, SW</td>
<td></td>
</tr>
<tr>
<td>Pump runs too long</td>
<td>Runaway Processor</td>
<td>Watchdog Timer</td>
<td>System, SW, Electrical</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rx calculated wrong</td>
<td>Safety Processor Redundant Calculation</td>
<td>System, (SW)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Timer wrong</td>
<td>Safety Processor Timer Cross-check</td>
<td>System, (SW)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Encoder calc error</td>
<td>Safety Processor Redundant Calculation</td>
<td>System, (SW)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exception exit error</td>
<td>Safety Processor Monitor</td>
<td>System, (SW)</td>
<td></td>
</tr>
<tr>
<td>Pump runs too fast</td>
<td>Rx calculated wrong</td>
<td>Safety Processor Redundant Calculation</td>
<td>System, (SW)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Timer wrong</td>
<td>Safety Processor Timer Cross-check</td>
<td>System, (SW)</td>
<td></td>
</tr>
</tbody>
</table>
Risk Management
Top Down Traceability Example

- Historical Data
- Competitive Data
- Preliminary Hazard Analysis
- System Fault Tree Analysis
- xxx FMEA’s
- Reviews

Risk Management Table

Product / System Requirements

Software Requirements
Software Design
Software Verification Tests

Electrical Requirements
Electrical Design
Electrical Verification Tests

Risk Management File
1. How can an operator of this software, or an end user of the device you produce be harmed by a failure of the device or component? (harms)

2. How can a failure of the software item lead to that hazardous result? (hazards)

3. What specific failures of the software item could cause each hazard? (causes)

4. What risk control measures are already in place, or need to be in place to assure that hazardous situations won’t occur or won’t lead to a harmful conclusion? (risk control)
Risk Control Measures

• Think “globally” (outside your software item). Are cross-checks in place that would detect, correct, or mitigate failures of the software item?
• Can verification processes or procedures be implemented outside the software item to verify its operation on a result by result basis?
• Will a hazardous failure be detectable? What can you do to make it detectable?
• Control risk as high on the Harm-to-Cause tree as possible.

Risk Management IS a Validation Activity
Is Risk Management Worthwhile with All These Problems?

YES
RISK AND NON-DEVICE
(820.70) SOFTWARE?
Documenting Risk Analysis

• Tell the story
  – As with I.U., describe process or part of process that is automated by the software.

• Step Through the Process and Surrounding QC Activities
  – How could failures of the software ultimately result in harm (safety, process, regulatory, business …)?
  – What else would have to happen for harm to result (how many in causal chain)?
  – Is there anything you can do to prevent, detect, or correct risk related failures?
Recall from Device Risk: the longer the causal chain of P’s the lower the overall Probability. Non-device software by its very nature will always have a longer causal chain.

Figure 18.2 Relationship of software failure to harm.
Figure 18.3 regulatory risk maps to safety risk.
Risk Management Table for Non-Device Software

• Similar to approach for Device Software
• Unless custom software, much less FMEA sourced info simply because you don’t have visibility at that level
• Process FMEA and effect of software is applicable
• Consider Qualitative Risk Acceptability method.

• Describe
  • Harms
  • Severity
  • Likelihood
    • Remember “causal chain” to cause harm
    • Is it acceptable?
    • If not, add controls, reassess

Summary

• Do we know what we’re doing?
  – Maybe it’s time to take fresh look.

• Are things getting better?
  – Maybe. With more “burn time” on risk we are likely to stumble into controls for safer devices … but there should be better ways.

• I’ve only mentioned concerns about “under controlling risk”. Examples also exist for using RM that leads to “over-controlling” almost non-existent risk!
Thank you for your kind attention.

If you have questions or would like more information on our development, validation, consulting, or training services, please don’t hesitate to call or email:

David Vogel
Intertech Engineering Associates, Inc.
100 Lowder Brook Drive, Suite 2500
Westwood, MA 02090

dav@inea.com

www.inea.com