Software in Medical Devices

Module 3:

• Software Risk Management

Pat Baird
Head of Global Software Standards
Philips
Disclosure: Philips has applications pending before FDA. I am not involved in the submission or direct support of those applications.

Pat Baird
Head of Global Software Standards
Philips
Risk Management =
Managing Murphy’s Law

Murphy’s Law:
“Anything that can go wrong will go wrong”
After we taught Risk Management principles to a Clinical co-worker, she created a risk analysis for her daughter’s wedding…

TAKEAWAY: If you care about the outcome, do Risk Management.
Agenda

Baseline fundamental concepts
ISO/IEC 62304 Software Standard & Risk Classification
Unique Aspects for Software
Baseline fundamental concepts
ISO/IEC 62304 Software Standard & Risk Classification
Unique Aspects for Software
Standards

ISO 14971 is a widely recognized risk management standard for medical devices

- Standard itself is short
- Informative annexes make up the bulk of the document

ISO 24971 gives additional direction / tips
IEC/TR 80002-1 guidance for applying 14971 to software
Risk Management, In a Nutshell

1. What are you trying to do?
2. What can go wrong?
3. What are you going to do about it?
4. Did it work?

Note: Although this presentation uses ISO 14971 to discuss “risk”, the concepts are universal. For example, the UK NHS “Clinical Risk Management: Agile Development Implementation Guidance” has similar features.
ISO 14971 Walkthrough

1. Management Responsibility & Personnel Qualifications
2. Risk Management Plan
3. Risk Management File
   3.1 Analysis
   3.2 Evaluation
   3.3 Implementation & Verification
   3.4 Assessment of acceptable residual risk
Analysis – but first, some Definitions

Intended Use:

ISO 14971: “Use for which a product, process, or service is intended according to the specification, instructions, and information provided by the manufacturer.”

The Intended Use statement is the cornerstone of risk management activities…
**Intended Use Statement for Infusion Pumps**

If you have the Intended Use statement, you can identify key elements, even without knowing the design.

What are the implications of “delivering fluid incorrectly”? What happens if there is too much? Too little? None?

What are the implications of the route?

What are the implications of the environment?

What are the implications of the patients?

What are the implications of the users?

“Pump is intended to deliver ___ fluids, via ___ routes, in ___ environments, for ___ patients, by ___ users.”

Is there something unique about the patients?

Or the users?
Definitions

Intended Use:
ISO 14971: “Use for which a product, process, or service is intended according to the specification, instructions, and information provided by the manufacturer.”

Hazard:
Potential Source of Harm
Note: Sources are always there, somewhere.

Hazardous Situation:
Exposure to a Hazard
Note: It is the bridge between a hazard and harm.

Harm:
Physical injury or damage to health of people, property or environment.
Hazard, Hazardous Situation, Harm Examples

Hazard:
Potential Source of Harm.
Always there... but just because it exists doesn’t mean there will be Harm..

Hazardous Situation:
Exposure to a Hazard.
It’s a situation where Harm could happen... It still hasn’t happened though..

Harm:
Physical injury or damage to health of people, property or environment.
Ouch.
Quiz: Hazard, Hazardous Situation, or Harm?
In addition to the Hazards, Hazardous Situations, and Harms identified by the design team, they can also leverage existing data sources

1. Software Specific Sources: IEC/TR 80002-1 Annex B & C
2. Clinical literature
3. MAUDE reports
4. Recalls
5. 14971 Annex E
Why do commercial aircraft tell us not to smoke, but then provide an ashtray?

Because they know someone will try to sneak a few puffs and then will try to hide the evidence by putting the cigarette in the trash bin, possibly starting a fire.

As part of Risk Management, we have to acknowledge that sometimes people won’t do what they are told to do, or will do what we tell them not to.
Cumulus Risk Example

• Sensor could fail, giving bad readings
  – Software could check for sensor values being in a reasonable range

• Cloud account not properly set up
  – ? Prompt user to complete on-line setup activity?
  – ? Prompt caregiver to setup user account?

• What if Grandmother AND Grandfather both have this product – what safeguards are there that the information won’t be mixed up?
“Risk” is combination of the probability of occurrence of harm and the severity of that harm [ISO 14971.]
What do we mean by probability? What do we mean by severity?
### Table D.4 — Example of semi-quantitative probability levels

<table>
<thead>
<tr>
<th>Common terms</th>
<th>Examples of probability range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequent</td>
<td>$\geq 10^{-3}$</td>
</tr>
<tr>
<td>Probable</td>
<td>$&lt; 10^{-3}$ and $\geq 10^{-4}$</td>
</tr>
<tr>
<td>Occasional</td>
<td>$&lt; 10^{-4}$ and $\geq 10^{-5}$</td>
</tr>
<tr>
<td>Remote</td>
<td>$&lt; 10^{-5}$ and $\geq 10^{-6}$</td>
</tr>
<tr>
<td>Improbable</td>
<td>$&lt; 10^{-6}$</td>
</tr>
</tbody>
</table>

Source: ISO 14971:2007
### Severity

**Table D.3 — Example of five qualitative severity levels**

<table>
<thead>
<tr>
<th>Common terms</th>
<th>Possible description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catastrophic</td>
<td>Results in patient death</td>
</tr>
<tr>
<td>Critical</td>
<td>Results in permanent impairment or life-threatening injury</td>
</tr>
<tr>
<td>Serious</td>
<td>Results in injury or impairment requiring professional medical intervention</td>
</tr>
<tr>
<td>Minor</td>
<td>Results in temporary injury or impairment not requiring professional medical intervention</td>
</tr>
<tr>
<td>Negligible</td>
<td>Inconvenience or temporary discomfort</td>
</tr>
</tbody>
</table>

Source: ISO 14971:2007
Does software have a probability?

Probabilities for mechanical / electrical issues are often based on reliability – failure is usually associated with a broken mechanical component or damaged electrical component – like a worn-out transmission or a blown resistor.

But Software doesn’t typically wear out – a bug that was just discovered has actually always been there – so what does probability mean in this context?

{we will talk about this a little later}
1. Management Responsibility & Personnel Qualifications
2. Risk Management Plan
3. Risk Management File
   3.1 Analysis
   3.2 Evaluation
   3.3 Implementation & Verification
   3.4 Assessment of acceptable residual risk
### ISO 14971 Walkthrough

Use the Severity and Probability rankings to determine what is acceptable, unacceptable, and needs investigation.

People commonly use a matrix of some sort.

<table>
<thead>
<tr>
<th>Probability of Harm</th>
<th>Severity of Harm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Negl. – 1</td>
</tr>
<tr>
<td></td>
<td>Minor – 2</td>
</tr>
<tr>
<td></td>
<td>Serious – 3</td>
</tr>
<tr>
<td></td>
<td>Critical - 4</td>
</tr>
<tr>
<td></td>
<td>Cat. – 5</td>
</tr>
<tr>
<td>Frequent – 5</td>
<td>Yellow</td>
</tr>
<tr>
<td>Probable – 4</td>
<td>Green</td>
</tr>
<tr>
<td>Occ. – 3</td>
<td>Yellow</td>
</tr>
<tr>
<td>Remote – 2</td>
<td>Green</td>
</tr>
<tr>
<td>Improb. – 1</td>
<td>Green</td>
</tr>
</tbody>
</table>
Things that complicate Software Risk Management

- Do software failures have a probability?
- Common mode failures vs. specific failures
  - Stack overflow
  - Memory corruption
  - Task starvation
  - Uninitialized variables
- Software Of Unknown Provenance (SOUP) & Commercial Off The Shelf Software (COTS) (covered in next session)
14971 Walkthrough

1. Management Responsibility & Personnel Qualifications
2. Risk Management Plan
3. Risk Management File
   3.1 Analysis
   3.2 Evaluation
   **3.3 Implementation & Verification**
   3.4 Assessment of acceptable residual risk
Mitigations are called “risk controls”
There are different kinds of risk controls. ISO 14971 has a priority order:
1. Inherent Safety – eliminate the problem
2. Protective measure – protect people from the problem
3. Information for Safety – tell someone about the problem
Mitigations are called “risk controls”
There are different kinds of risk controls. ISO 14971 has a priority order:

1. Inherent Safety – eliminate the problem
2. Protective measure – protect people from the problem
3. Information for Safety – tell someone about the problem
Flixborough disaster: cracked pipe leaked 40 tons of cyclohexane in less than a minute. When ignited, the fuel-air explosion was the equivalent of 15 tons of TNT.

1800 buildings damaged; 28 people died. Fires took 10 days to put out.

Maybe keeping a large inventory of a dangerous substance isn’t such a good idea..

Source: “What you don’t have can’t leak” by Trevor Kletz.
Inherent Safety continued...

What can you do to totally eliminate the issue?

Center for Chemical Process Safety has 4 principles:

1. **Minimize**: make smaller batches that won’t explode

2. **Substitute**: a wind-up infusion pump will never give an electric shock

3. **Moderate**: battery powered devices don’t have high-voltages

4. **Simplify**: fewer options = fewer mistakes
Mitigations are called “risk controls”

There are different kinds of risk controls. ISO 14971 has a priority order:

1. Inherent Safety – eliminate the problem
2. Protective measure – protect people from the problem
3. Information for Safety – tell someone about the problem
Protective Measures

What can you do to protect the user / patient?

1. Barriers: physical (or software) interventions such as shut-off valves, fences, software lockouts

2. Detection: Detect and notify an out-of-bounds conditions such as high-temperature alarms, data entry validation, etc.
Protective Measures: Examples
Risk Control Methods

Mitigations are called “risk controls”
There are different kinds of risk controls. ISO 14971 has a priority order:

1. Inherent Safety – eliminate the problem
2. Protective measure – protect people from the problem
3. Information for Safety – tell someone about the problem
If the hazardous situation cannot be removed by design, and if there are no protective measures, then and only then can you use information as a risk control.
Quiz: Inherent, Protect, or Inform?

CAUTION
THIS SIGN HAS SHARP EDGES
DO NOT TOUCH THE EDGES OF THIS SIGN
ALSO, THE BRIDGE IS OUT AHEAD
Boomerang Effects – what are unintended consequences?

Attempts to mitigate sometimes results in even more dangerous behavior. We are required to consider the consequences of the mitigations.

- Example 1: Weather radar for North Atlantic fishermen was intended to give them more time to leave a dangerous area.
- Instead, the captain know exactly when a storm will arrive, and stays until the last minute.
- Example 2: When anti-lock brake systems were first introduced, the number of car crashes in the US declined.
- However, once ABS was the norm, the number of crashes resumed it’s previous level. Why?
- People started pressing on the brake pedal later; sometimes not giving themselves enough time to stop.

Why Things Bite Back, Edward Tenner
Verification / Validation

Need to verify & validate that the risk control was effective.
1. Management Responsibility & Personnel Qualifications

2. Risk Management Plan

3. Risk Management File
   3.1 Analysis
   3.2 Evaluation
   3.3 Implementation & Verification
   **3.4 Assessment of acceptable residual risk**
Acceptable Residual Risk

- ISO 14971 defines Residual Risk as:
  - “risk remaining after risk control measures have been taken”
- After verification, manufacturers need to assess if the Overall Residual Risk (ORR) is acceptable using the criteria in the Risk Management Plan.
- If not, manufacturer may conduct a risk-benefit analysis.
- The results should be documented in the risk file.
- Regardless, the manufacturer needs to disclose the ORR to users (Annex J provides guidance)
Risk Management and Product Lifecycle

- Risk Management (RM) can start before the concept phase – as soon as you have an Intended Use, you can have inherent risks irrespective of design.
- RM unfolds as the product requirements and design evolves; Design decisions introduce new hazards and new risk controls.
- RM continues into post-market activities as customers use and misuse your products.
- Even after stop-sales and stop-service, RM continues – you always continue to monitor risk.
Post-market Risk Management

Manufacturers are required to have a system in place to collect and review post-production data.

Need to look for previously unrecognized Haz/HazSit and estimated risks outside of predictions. The risk file must be updated, as appropriate.
Event-Based Reviews

These “event-based” reviews could be triggered by a variety of reasons:

- Complaints
- MDRs
- CAPAs
- Trending of manufacturing or repair data
- Design changes
- Standard change

Although these events might not have resulted in Harm, but could still be relevant.
Periodic Risk File Reviews

Depending on your product, there are also requirements for periodic risk file reviews (e.g. EU Periodic Safety Update Report (PSUR))

Possible data sources include:

- Literature reviews (adverse events, off-label use, etc.)
- MDRs of similar / predicate devices
- Recalls on similar / predicate devices
- Complaints / CAPAs,
- Manufacturing/Repair
Many well-known disasters can be traced back to design changes.

Although the root cause for the source of the fire has never been determined, there are some lessons that still apply today.

Recap: After crossing the Atlantic, the Hindenburg attempted to land in Lakehurst New Jersey. Trivia: This flight was actually a partnership with American Airlines.

During the landing procedure, a fire started on the aircraft.

The aircraft was destroyed in 30 seconds.
Lesson 1: The Hindenburg design contained performance and safety improvements over the previous best-in-class airships from both Germany and the UK. **Even “better” can fail catastrophically.**

Lesson 2: An obvious assumption is that the fire was the hydrogen burning, the flames are yellow – indicating the skin and internal containment bags are burning. **The obvious root cause might not be correct.**

Lesson 3: The designers originally specified Rolls-Royce gasoline engines, but the manufacturer refused to buy from a UK supplier. Instead, they installed Benz diesels, with a lower power-to-weight ratio. As a result, the designers had to quickly lighten the airship. They chose to skip the flame-retardant coating on the original skin. **Change has risk.**
Cumulus Risk Example

• Mobile apps are updated all of the time – how does this impact the patient information file stored on the phone?

• What should we do with patient information on the phone if the app is deleted? What should we do with the patient information in the cloud?
Agenda

Baseline fundamental concepts
ISO/IEC 62304 Software Standard & Risk Classification
Unique Aspects for Software
ISO/IEC 62304 Software Classifications

Medical devices have varying degrees of risk associated with them. The standard ISO/IEC 62304 “Medical Device Software – Software life-cycle Processes” takes this into account and the amount of work that needs to be done is related to software risk classification.

- 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

In determining the software safety classification of the software system:
- Probability of a software failure shall be assumed to be 1.
- Only risk control measures not implemented within (external to) the software system shall be considered.

NOTE: Such risk control measures may reduce the probability that a software failure will cause harm, and/or the severity of that harm.

Note: A software system which implements risk control measure may fail, and this may contribute to a hazardous situation. The resulting harm may include the harm which the risk control measure is designed to prevent (see 7.2.2b).
Agenda

Baseline fundamental concepts
ISO/IEC 62304 Software Standard & Risk Classification
Unique Aspects for Software
ISO/TR 80002-1 points out that software hazards are different than other hazards in that software alone doesn’t typically hurt people (vs. electrical exposure.) Software has to work in a sequence of other events to cause harm.

Unique Aspects for Software

Software gets blamed for non-software design issues. Particularly in post-market feedback, since software is the most visible aspect of the design, when things go wrong, it is software’s fault.

• Hardware failures are often detected by power-on self tests that the software executes. If a failure is detected, the software will issue an error code (“Error 123”), and this gets reported as a “software problem.” The software did was it was supposed to do – detect a hardware problem, but the problem is initially listed as a software issue.

• Usability issues are also assigned to software. For example, the workflow that was implemented on the device doesn’t match with the user’s expectations, and software is blamed. Although this isn’t the result of a coding error, and is rather a result of not understanding user needs, this is sometimes listed as software issue.
Unique Aspects for Software

Question: “Mechanical and Electrical failure rates are often a function of wear out... but software doesn’t wear. So what is the basis for probability? If there is no probability, then what do you do?”

Sometimes stated as “Software has no probability. If there’s a bug, then the probability of that bug existing is 100%”

While this may be true, this isn’t particularly useful. A bug in a rarely used feature does not mean that every time the software is executed results in a software failure. The probability CAN be estimated.
Additional Reading

Collection of papers regarding safety, regardless of industry
http://www.dependablesos.org/category/dsos

A podcast (start with episode 8 as an example)
http://disastercast.co.uk/?feed=podcast\
Questions?