Using Design FMEAs to Improve Software Design

Definition of Risk
- Two Components of Risk
  - Probability of Exposure To Harm When Objectionable Incident Occurs
  - Severity of Harm

Classic FMEA Versus Modern FMEA
- Classic FMEA Primary Focus
  - Mitigation of Harm When Objectionable Incident Occurs
  - Root Cause(s) of Objectionable Incident Not Required
- Modern FMEA Primary Focus
  - Prevention of Objectionable Incident
  - Root Cause(s) of Objectionable Incident Required
Fault Tree Does Not Equal FMEA

- Fault Tree
- All Causes are Not Root Causes
- FMEA
- All Causes Must Be Root Causes
- 1st Edition AIAG-VDA FMEA Handbook FMEA Methodology
- Adoption of 20+ Year-Old Software Driven FMEA Methodology
- Based on Fault Tree Equals FMEA Premise
- ASQ September Webinar on Design FMEA and Process FMEA Methodologies Contained In 1st Edition AIAG-VDA FMEA Methodology

Root Causes Of Software Objectionable Incidents

- Definition of Software Objectionable Incident
- Potential Root Causes
  - Incorrect Customer Requirements
  - Competing
  - Conflicting
  - Limits of Technology
  - Incorrect Software Design Requirements
  - Incorrect Code/Calibration Factors
  - Incorrect Sprint Task Requirements (Agile Development Only)
  - Software Not Used As Intended

Software Risk Management Tools

- Incorrect Customer Requirements
  - Requirements Risk Assessment® (RRA®)
- Incorrect Software Design Requirements
  - Requirements Risk Assessment® (RRA®)
- Incorrect Code/Calibration Factors
- Software Design FMEA (SDFMEA)
- Incorrect Sprint Task Requirements
- Sprint Task FMEA™ (STFMEA™)
- Software Not Used As Intended – Software Application FMEA

Software Risk Management Tools (Non-Agile Development)
Software Design FMEA

- Objective
- Risk Assessment of Releasing Code/Calibration Factors in Current Form
- Track Risk Reduction Activities
- Important Deliverables
  - Clear definition of design requirements.
  - Design Verification Plan
  - Prioritization of Risk Issues

Why Software Design FMEAs Have Not Been Effectively Used

- Common Misperceptions
  - SFMEA Bottom-Up
  - Improper Focus on Faults/Failure Modes
  - Analysis to Determine Impacts of Faults on System
  - Concentration on Mitigation of Effects of Failure Rather Than Failure Prevention
- How The Modern SFMEA Solves the Problem
  - With Existing Software Design FMEA Methodologies
  - Assessment of Risk of Releasing Software and Calibration Factors in Current Form

Software Design FMEA Example

- Oil Application System

SDFMEA Entries – Design Requirements Column

- Derived From RRA® (if it exists)
- Multiple Categories of Requirements
- Significance of Design Requirements Column on DVP
- Level of Detail Required - Environmental and Machine Conditions of Performance
SDFMEA Entries – Failure Modes, Effects and Sev Columns

- Failures To Include
  - (Sev) Severity of Harm Rating

SDFMEA Entries – Typical Severity of Harm Rating Table

<table>
<thead>
<tr>
<th>Description</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Possibility of injury or violation of law</td>
<td>10</td>
</tr>
<tr>
<td>Possibility of injury or violation of law with warning</td>
<td>9</td>
</tr>
<tr>
<td>Loss of primary function</td>
<td>8</td>
</tr>
<tr>
<td>Reimbursement of secondary function</td>
<td>7</td>
</tr>
<tr>
<td>Reimbursement of secondary function</td>
<td>6</td>
</tr>
<tr>
<td>Noise or appearance issue detected by customer that results in returns</td>
<td>5</td>
</tr>
<tr>
<td>Noise or appearance issue detected by customer that does not result in returns</td>
<td>4</td>
</tr>
<tr>
<td>Noise or appearance issues typically not detected by customer</td>
<td>3</td>
</tr>
<tr>
<td>No effect</td>
<td>2</td>
</tr>
<tr>
<td>No effect</td>
<td>1</td>
</tr>
</tbody>
</table>

SDFMEA Entries – Remaining Columns

- Failure Cause Similarity With Hardware Design FMEA
- Hardware Spec Versus Software Code and Calibration Factors
- (OCC) Probability of Objectionable Incident Exposure due To Cause (versus Harm) – Determined Using DVP
- Why RPN Should Not Be Used – No Containment in Design FMEA
- Class Column (aka Residual Risk) - Risk Matrix and Risk Policy

SDFMEA Entries – Typical Probability of Objectionable Incident Exposure

<table>
<thead>
<tr>
<th>Description</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10; Confidence Level: &lt;=70%</td>
<td>10</td>
</tr>
<tr>
<td>1 ln 20; Confidence Level: 70%</td>
<td>9</td>
</tr>
<tr>
<td>1 ln 50; Confidence Level: 75%</td>
<td>8</td>
</tr>
<tr>
<td>1 ln 100; Confidence Level: 80%</td>
<td>7</td>
</tr>
<tr>
<td>1 ln 500; Confidence Level: 85%</td>
<td>6</td>
</tr>
<tr>
<td>1 ln 2,000; Confidence Level: 90%</td>
<td>5</td>
</tr>
<tr>
<td>1 ln 10,000; Confidence Level: 95%</td>
<td>4</td>
</tr>
<tr>
<td>1 ln 100,000; Confidence Level: 99%</td>
<td>3</td>
</tr>
<tr>
<td>1 ln 1,000,000; Confidence Level: 99.9%</td>
<td>2</td>
</tr>
<tr>
<td>Failure is eliminated</td>
<td>1</td>
</tr>
</tbody>
</table>
### What Sources of Risk Must Be Worked On – Automotive Design

#### Risk Matrix (Auto Industry Design Process)

<table>
<thead>
<tr>
<th>SEV/OCC</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
</tr>
<tr>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
</tr>
</tbody>
</table>

Symbol is assigned based on SEV and OCC.

---

### What Sources of Risk Must Be Worked On – Medical Device

#### Risk Matrix - Medical Industry

<table>
<thead>
<tr>
<th>SEV/OCC</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
</tr>
<tr>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
</tr>
</tbody>
</table>

---

### When To Release – Automotive Design Risk Policy

<table>
<thead>
<tr>
<th>SEV/OCC</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
</tr>
<tr>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
</tr>
</tbody>
</table>

* = Do Not Release

---

### When To Release – Medical Device Risk Policy

<table>
<thead>
<tr>
<th>SEV/OCC</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
<td>YC</td>
</tr>
<tr>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
<td>YS</td>
</tr>
</tbody>
</table>

* = Do Not Release

---

### Areas of Acceptability in Risk Table for release (aka Risk Policy).
Different products can use same Risk Matrix but have different Risk Policies.

---

### Risk Policy (*) = Do Not Release

- Spinal Implant
- Late Stage Cancer Treatment

---

Copyright © 2019 Harpco® Systems Inc. All rights reserved.
What Is Agile Software Development

- Group of software development methodologies (i.e. YSRUM) based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.
- Design Requirements Broken Down Into Sprint Tasks

Using FMEAs With Agile Software Development – Sprint Task Requirement FMEA

- Failure Cause Columns
- Incorrect Sprint Task Requirement Versus Incorrect Software Code and Calibration Factors
- Determining Probability of Harm Exposure – DVP (Pre and Post Sprint)
- Class Column - Risk Matrix and Risk Policy

Using FMEAs With Agile Software Development – Sprint Task Code FMEA

- Design Requirements and Design Control Differences
- Hardware Conditions
- Environmental Conditions
- Important Design Control Considerations
- Hardware Conditions
- Environmental Conditions
Summary

- Objectives of FMEAs When Used for Software Development
  - Provides Clear Definition of Software Design Requirements
  - Software Is Typically Tail of Development Process When Hardware and Software Involved
  - Prevent Cause of Risk Exposure Rather Than Mitigation of Effects
  - Defines Design Verification Plan
  - Provides Structure to Software Design Process

For More Information

Richard Harpster E-mail: richard.harpster@harpcosystems.com
Phone: 248-374-1718 Office
248-767-7557 Cell