# SOFTWARE RISK MANAGEMENT PROCEDURE

# TRAINING EXAMPLE

# DRAFT

Revision History

| Date | Revision | Author | Description |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Note: This example is conceived as the body of a procedure.
Explanatory comments are included in << comment >>.
Other text is example definition that outlines an example of Software Risk Management Process.
In the appendix, an example template for Software Risk Analysis report is provided.
This is not a complete RA, just a training example to guide in the development of a RA for a certain type of device.

# TABLE OF CONTENTS

# 1 Introduction

## 1.1 Purpose and Scope

This procedure defines the Software Risk Management process and activities within the <TBD> <<state the company, department, etc. to which this document applies. >>
This procedure describes the way software related risks will be identified, evaluated, classified, documented and controlled.
Software Risk Management is part of System Risk Management……. << describe how these two processes interrelate to each other. >>
This procedure will be used with risk analysis tools such as Hazard Analysis, Fault Tree Analysis and Failure Mode Effects and Criticality Analysis.
This procedure applies to:

- All new product software developments
- All software changes through product maintenance
- ……..
- ……..

<<Detail applicability>>
This procedure does not apply to: …….. << If necessary, detail where the procedure is not applicable >>
Intended audience for this document are……..<<Define intended audience>>

## 1.2 Definitions

For department common definitions, please refer to <TBD> Common Definitions.
Additional specific terms are defined in Table 1**Error! Reference source not found.** below.

| | |
|---|---|
| ALARP | Acronym for "As Low As Reasonably Practicable" |
| Failure Path | The causes or combinations of causes that can lead to the top level event (in the case of an FTA) or the event of interest, i.e. results in hazard. |
| Foreseeable Misuse | Use of the medical device in a manner the manufacturer did not intend but could have reasonably predicted as a consequence of human behavior. |
| Residual Risk | Risk remaining after protective measures have been taken |
| Risk Control Effectiveness (RCE) | Capability of Risk Control Measure to reduce or eliminate the risk of the hazard it is meant to mitigate. |

**Table 1**

<< Define any term used in the document that may be unfamiliar to readers. >>

## 1.3 References

ISO 14971:2007, Medical devices— Application of risk management to medical devices.
IEC 62304:2006, Medical device software—Software life cycle processes.
AAMI TIR32:2004, Medical device software risk management. - << replace with IEC 80002 when released >>

# 2 Roles and Responsibilities

The QA Manager is responsible for supervising the application of this procedure to every project……..
The Project Manager of each project is responsible to ensure that Risk Management activities are properly performed in every phase of the software development and maintenance lifecycle……..
<< Define specific responsibilities for application of the procedure and approvals >>

# 3 Process Overview

The risk management process is composed of 4 main phases:

- Risk analysis to identify the potential risks and the causes that could lead to hazardous situations.
- Risk evaluation to evaluate the possible consequences of risks
- Risk control to define and apply mitigations that allow to reduce the risks
- Residual risk evaluation to estimate the levels of risk after applying the mitigations and decide on its acceptability.

The process is repeated cyclically during all the software lifecycle phases and is reapplied after changes or anomaly fixes…..
<< Modify/Expand to describe the entire process. >>

# 4 Process Description

## 4.1 Risk Analysis

### 4.1.1 Identify Intended Use

Describe <device> software intended use and software foreseeable misuse.

### 4.1.2 Identify SW Related Hazards

Compile a list of all known and foreseeable hazards associated with <device> software in both normal and abnormal (faulty) conditions.
These could be derived from the hazards identified in the <device> Risk Analysis, applying a filter to isolate those that could be software related.
Additional specific hazards can be investigated and added to the list, by using a variety of sources, including:

- Historical field information for similar products such as review of complaints.
- Software related Medical Device Reports (MDRs) and product recalls reported to the FDA
- Standards and guidance documents, such as TIR32
- ……..

<< Detail the process and specify all the sources to consider >>

### 4.1.3 Identify SW Causes

Once the hazards for the intended use have been identified, evaluate the possible software related causes for each of them. These could be software causes, or hardware causes and use causes handled by software. I n this sense it is important to identify the chain of events that can lead to the harm to occur.
Fault Tree Analysis tool could be used to help identifying the possible failure paths.
An adequate number of software engineers and system engineers should be involved in this analysis so that multiple levels of analysis occur, ranging from the hardware interface layer to the user interface layer.
TIR32 Annex A could be used to help in this analysis.

In addition to function-specific failure causes, software indirect/common causes for the specific application must be taken into consideration. These are software initiating causes that could lead to hazards through an unpredictable chain of events. TIR32 Annex B could be used to help in this analysis.
SOUP items unexpected failures shall be included in this analysis. If SOUP unexpected behavior could be a contributor to the risk, published anomaly lists shall also be examined.
<< Detail the methods used for identifying SW causes. >>

## 4.2 Risk Evaluation

Risk evaluation will be an iterative process. Risk will be evaluated prior to any risk control measures being applied and then re-evaluated any time a risk control measure is identified and applied.
For software related causes initial risk evaluation shall only be based on severity of the risk, while probability of occurrence is not considered, because software errors are considered systematic.

In subsequent evaluations after application of Risk Control Measures, reductions in likelihood of harm based on the effectiveness of the risk control measures taken are utilized based on risk control effectiveness rationales as explained below.
<< Further detail according to your process. >>

The severity is evaluated according to the following table: << this is just an example, replace with your own levels. Severity levels could also be unique to the intended use of the device either by clarification of types of harm in the definitions or in the severity levels themselves (e.g., radiation overdose, infection, minor burn, temporary cosmetic …)>>

| Severity Rank | Severity Level | Severity Description |
|---|---|---|
| 1 | Negligible | No potential for direct or indirect harm. |
| 2 | Marginal | Potential of minor direct or indirect harm not necessitating medical intervention.<br>May cause reversible damage to the system, reagents or consumable materials, other property, or the environment. |
| 3 | Serious | Potential for significant direct or indirect harm necessitating medical intervention.<br>May cause major, irreversible damage to the system, reagents or consumable materials, other property, or the environment. |
| 4 | Critical | Potential death, serious injury, or serious deterioration in state of health. |

**Table 2**

## 4.3 Risk Control

For each cause that requires risk reduction, appropriate risk control measures will be identified and applied. Risk control measures can be applied at different stages of the casual chain, therefore could be different for different causes and sub-causes, or sometimes, if applied to the last point of control, could be common to several causes and sub-causes.
Again, Fault Tree Analysis could be used as a powerful tool to identify risk control measure and apply them in the most effective point in the failure path.

Depending on the type of risk control measures applied their effectiveness in reducing the risk will be different.

The most effective risk control measures are those that are "inherently safe," that is, the possibility of the hazard is eliminated altogether. Early in the product concept phase of the project, the designers should consider the potential hazards and how they might be eliminated through inherent design. An example could be only use of static memory allocation to eliminate the possibility of memory fragmentation or memory leaks.

When inherently safe design is not possible, design or process mitigations will be applied. General mitigation will be applied to indirect\common causes, where the casual chain of events is not defined, while specific mitigations will be applied to specific causes and failure paths. Effectiveness of the mitigations will depend on the type of mitigation and on its reliability. Diverse RCMs are more effective than simple RCM. Diversity depends on the specific design: in some case could be use of a different processor to implement the mitigation, in other cases could be execution of the main code and of the controlling code as totally independent processes on the same processor so that one cannot affect the other.

Another level of mitigation is labeling and training. By providing the user with instructions for use, warning, cautions, training, and other information, the user can become a key element in reducing the risk,

but the goal should always be to reduce risk prior to the device's last point of control. Effectiveness of this type of mitigations should always be evaluated with the assistance of medical/clinical experts.

Multiple risk control measures can sometimes combine to increase risk control effectiveness (RCE).

Software failures being analyzed will be assigned a RCE rating in order to determine residual risk and risk acceptability.
The following table can be used to evaluate the RCE. << this is just an example, replace with your own levels >>

| RCE Rank | RCE Level | RCE Description |
|---|---|---|
| 1 | Safe Design | Inherently Safe Design – failure can not happen |
| 2 | Diverse RCM | Diverse RCM(s) considered highly reliable (cannot fail in a latent manner), independent, redundant, highly effective. |
| 3 | Simple RCM | Effective but non-diverse RCM. |
| 4 | Labeling | Only labeling, training and/or clinical practice to mitigate the concern. |

**Table 3**

Each risk control measure will be verified to be correctly designed and implemented and will be traceable to one or more requirements in the requirement specification and to one or more test procedures or test cases.

Risk Control Measures shall be further analyzed to verify they do not introduce new failure paths. In case they do, further analysis shall be applied to document the additional causes and to verify if additional controls need to be introduced

<<Add/Replace as needed with your process >>

## 4.4 Overall Residual Risk Evaluation

The residual risk evaluation is an iterative process. For each failure path (cause and sub-causes), residual risk will be evaluated after every risk control measure is applied.

The RCE residual risk acceptability can be evaluated according to the following table: << this is just an example, replace with your own levels >>

| Severity><br><br>RCE | Negligible | Marginal | Serious | Critical |
|---|---|---|---|---|
| **Safe Design** | Acceptable | Acceptable | Acceptable | Acceptable |
| **Diverse RCM** | Acceptable | Acceptable | Acceptable | Further Evaluation |
| **Simple RCM** | Acceptable | Acceptable | Further Evaluation | Further Evaluation |
| **Labeling** | Acceptable | Further Evaluation | Further Evaluation | Further Evaluation |

**Table 4**

For items that fall in the cautionary "Further Evaluation" Category attempts will be made to identify alternative risk control measures to result in the Acceptable range. If this is not practicable then the acceptability of the residual risk will be evaluated based on the overall device risk analysis which includes the likelihood that if such failures occurred the hazardous situation would occur and if so then the likelihood of actual harm occurring as part of the resulting residual risk rating.

A final overall residual risk acceptability will be determined and documented. This will be based on all the different risk control measures applied (although each one may not be fully effective in itself, they can combine to result in a very effective risk mitigation) and will also take into account external factors, such as intended use and common clinical practice.
Whenever residual risk acceptability does not result immediately Acceptable according to the above table, a detailed justification for final acceptability of the risk is required.
<< This last portion of the process could be part of the overall device risk analysis and could be omitted from the Software Risk Analysis, or could be used to support the overall device risk analysis. >>

# 5 Report

A Software Risk Analysis report will be compiled as a result of this process.
This will be a document growing through the phases of the process.
An example template for this document is provided as an attachment to this procedure.
The main features in the report will be the list of the identified hazards, and, for each hazard, a table illustrating causes, sub-causes, risk control measures, and risk evaluation.

Each table could be structured with the following columns, as illustrated in the template:

- **Cause** Various ways the hazard can be caused; these are high-level causes.
- **Sub-Cause** This column lists various ways the hazard can be caused at a more detailed level: indicate here the initiating cause, or the chain of events that could lead to the high level cause and the hazard. If fault tree has been used to document failure paths, reference here the branch illustrating the chain of events.
- **Severity** Risk Severity level according to Table 2 above
- **Internal RCM** This column lists various controls built into the device or its software that control or mitigate the hazard. These could include also labeling (manuals, labels, user screens) and specific user training.
- **Traceability** This column has linkage information either to the requirement implementing the RCM or to the verification test verifying its implementation (or both). << alternatively a separate trace table could be maintained >>
- **RCE (risk control effectiveness)** For each risk control an effectiveness rate attributed according to Table 3 above.
- **Initial Residual Risk rating** Acceptability level of risk mitigation for any single RCM, according to Table 4
- **External RCM** These are related to the actual way the instrument is expected to be used. Could be accepted clinical practice or intended use limitation.
- **Final Residual Risk Rating** This is the acceptability level of risk mitigation assigned to each sub-cause. It takes into account all the different internal RCMs applied and the external RCMs to evaluate the overall effectiveness of all mitigations applied. Even if none of the single mitigations taken by itself could be sufficient to reduce the risk to an acceptable level, they could when considered all together to increase effectiveness.
- **Justification for acceptance** The rationale to accept the residual risk without further mitigation. Required only if none of the internal RCMs leads to Initial Residual risk rank = ACCEPTABLE.

# 6 APPENDIX: report template example.

# SOFTWARE RISK ANALYSIS REPORT

# TEMPLATE

# DRAFT

Revision History

| Date | Revision | Author | Description |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Note: This template is conceived as a partial example template for a generic small device with embedded real time control. Explanatory comments are included in << comment >>.
Other text is example definition that you should replace with your own text.

# TABLE OF CONTENTS

<< Insert Report TOC >>

# 1 Introduction

## 1.1 Purpose

Purpose of this Software RISK ANALYSIS REPORT is to describe the risk analysis and risk control activities performed related to the software of <device> . <<state the version of the software to which this document applies. >>

## 1.2 Scope

The <device> is meant to ……<< Describe intended use of the device, >>
The software in the <device> executes the following tasks…….. <<Describe shortly the main software tasks and software intended use >>
This report documents the following activities related to the software of <device> :

- Risk analysis.
- Risk evaluation.
- Risk control.
- Residual risk evaluation

This report is limited to software; it does not document any of the hardware or overall device risk analysis activities.
<<The information in this report could be incorporated into the overall device risk analysis report if desired.>>

## 1.3 Definitions, acronyms, and abbreviations

| | |
|---|---|
| RA | Risk Analysis |
| RC | Risk Control |
| RCE | Risk Control Effectiveness |
| RCM | Risk Control Measure |

<< Define any acronyms, abbreviations, or terms used in the document that may be unfamiliar to readers. If a project level definitions document exists, it can be referenced here and this section limited to specific terms used only in this document. >>

## 1.4 References

**Standards:**
ISO 14971:2007, Medical devices— Application of risk management to medical devices.
IEC 62304:2006, Medical device software—Software life cycle processes.
AAMI TIR32:2004, Medical device software risk management. - << replace with IEC 80002 when released >>

……..
<< Include any **relevant** standard >>

**Project Documents:**
<device> Software Requirement Specifications, Rev. x.y, …….<<you could add any information useful to locate the document>>

……..
<< Include **relevant** project documents such as device risk analysis, device and software risk management procedures, software architecture/design specs >>

## 1.5 Overview

This document is organized in 3 main sections.
Section 1 (this section) puts the document into its context and gives an overview of it.
Section 2 gives a high level summary of the identified software risks.

Section 3 details the risk analysis for each identified risks and for the software indirect/common causes considered.

<< Add anything apt to describe content and organization of this document >>

# 2 Hazards and Risk Summary

- Wrong parameter result presented to the user……
- Therapy dosage delivery speed too high ……

<< State the main risks of the device and as the project progresses adding how these are handled. These are explained in a general way, and generally how they are mitigated (within the device or externally once these are defined). >>

# 3 Hazards Analysis

The tables below describe in detail the causes and methods of control for each software hazard.

Each cause, sub-cause and RCM are numbered as C*n*, SC*n.n*, RC*n*, where *n* is a consecutively assigned number.

<< The following are just examples of possible hazards in a medical device. Replace with your own. Consider also possible failures of SOUP components >>

## 3.1 Specific hazards

### 3.1.1 Wrong parameter result presented to the user.

| Cause | Sub-Cause | Severity | Internal RCM | Traceability | RCE | Initial Residual Risk rating | External RCM | Final Residual Risk Rating | Justification for acceptance. |
|---|---|---|---|---|---|---|---|---|---|
| C1 – The value stored in Patient DB is wrong | SC1.1 The value was correctly stored, but got corrupted after storing. | Critical | RC1 – Multiple (3) copies handled by different process for critical info. | SRS111 TC063 | Diverse RCM | ALARP | The clinical practice is never to use single parameter data… | Acceptable | Highly effective risk control measures including multiple methods of error detection together with clinical practice/labeling warning |
| | | | RC2 – CRC on data stored in Patient DB…… | SRS1234 TC159 | Simple RCM | Not Acceptable | | | |
| | | | RC3 – Check on defined limits when retrieving data from DB….. | SRS2468 | Simple RCM | Not Acceptable | | | |
| | | | RC4 -…… | …. | ….. | ….. | | | |
| | SC1.2 Wrong A/D conversion… . | Critical | RC5 – Periodic check on A/D using known reference values…. | SRS3721 | Simple RCM | Not Acceptable | The clinical practice is never to use single parameter data… | …… | …… |
| | | | RC6 - …….. | …… | …… | ……. | | | |
| C2 – Malfunction in data formatting algorithm during presentation | | | | | | | | | |
| | | | | | | | | | |

<< Complete with all identified causes and all identified paths and RCMs. >>

### 3.1.2 Therapy dosage delivery speed too high.

| Cause | Sub-Cause | Severity | Internal RCM | Traceability | RCE | Initial Residual Risk rating | External RCM | Final Residual Risk Rating | Justification for acceptance. |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |
| | | | | | | | | | |

### 3.1.3 ……..

| Cause | Sub-Cause | Severity | Internal RCM | Traceability | RCE | Initial Residual Risk rating | External RCM | Final Residual Risk Rating | Justification for acceptance. |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |
| | | | | | | | | | |

### 3.1.4 ……..

### 3.1.5 ……..

## 3.2 Indirect/Common Causes

The following table lists the software failure causes that are not tied to specific functionality of the device and individual hazards. They can cause unpredictable effects:

| Cause | Sub-Cause | Severity | Internal RCM | Traceability | RCE | Initial Residual Risk rating | External RCM | Final Residual Risk Rating | Justification for acceptance. |
|---|---|---|---|---|---|---|---|---|---|
| C33 - Arithmetic error | SC33.1 Division by zero | Critical | RC33 - Error reported by arithmetic coprocessor | | Safe Design << Provided error is correctly handled >> | Acceptable | | | |
| | SC33.2 Numeric Overflow | Critical | RC34 – Range checks are used …… | | Simple RCM | | | | |
| | | | RC35 - User manual indicate to verify all results outside limits….. | | Labeling | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

<< Complete this table with all the common causes that have been considered in software risk analysis. Use Annex B of AAMI TIR32 for reference. Explicitly consider also possible failures of SOUP components (e.g. operating system failures) >>