A Methodology for Exposing Software Safety Risk in Early Development Phases

Lucas Layman, Victor R. Basili, Marvin V. Zelkowitz
Fraunhofer Center for Experimental Software Engineering

NDIA Systems Engineering Conference
October 24, 2012

Fraunhofer
USA
Center for Experimental Software Engineering

Outline

1. The challenge: measuring software safety risk early in the lifecycle
2. Apply the Process Risk Assessment (PRA) methodology to NASA’s Constellation program
3. Common software safety process risks, lessons learned and recommendations

What is software safety risk?

A safety risk is a hazard – any real or potential condition that can cause:

- injury, illness, or death to personnel;
- damage to or loss of a system, equipment, or property; or
- damage to the environment

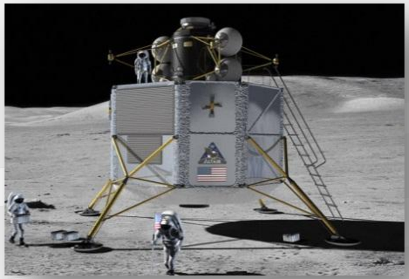
Software safety risks have become a greater concern in systems development as many traditionally hardware-centric systems become more reliant on software

Software safety is an example of an emergent system property that cannot be fully tested until the system is operational.

Constellation – why measure software safety risk?

CONSTELLATION

Ares rockets

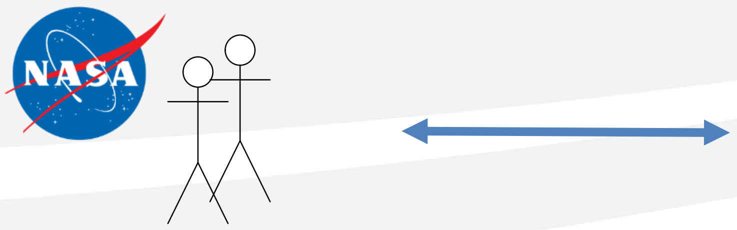

Altair lunar lander

Orion crew vehicle

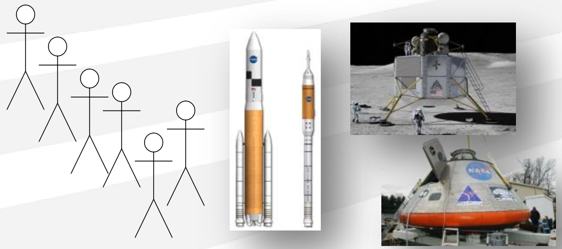
- Hundreds of suppliers, thousands of engineers
- NASA oversight on quality, safety, reliability

The challenge - What is the state of software safety in my system?

Software Safety Assurance



Project Safety Engineers



"What is software's impact on system safety?"

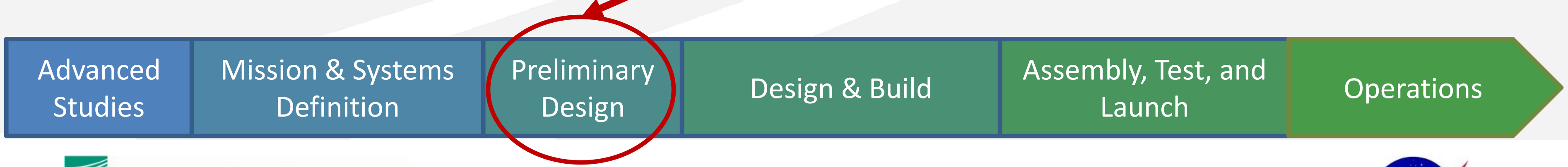
"Which subsystems have the most software safety risks?"

"Are our software safety processes appropriate?

"How does software fail?"

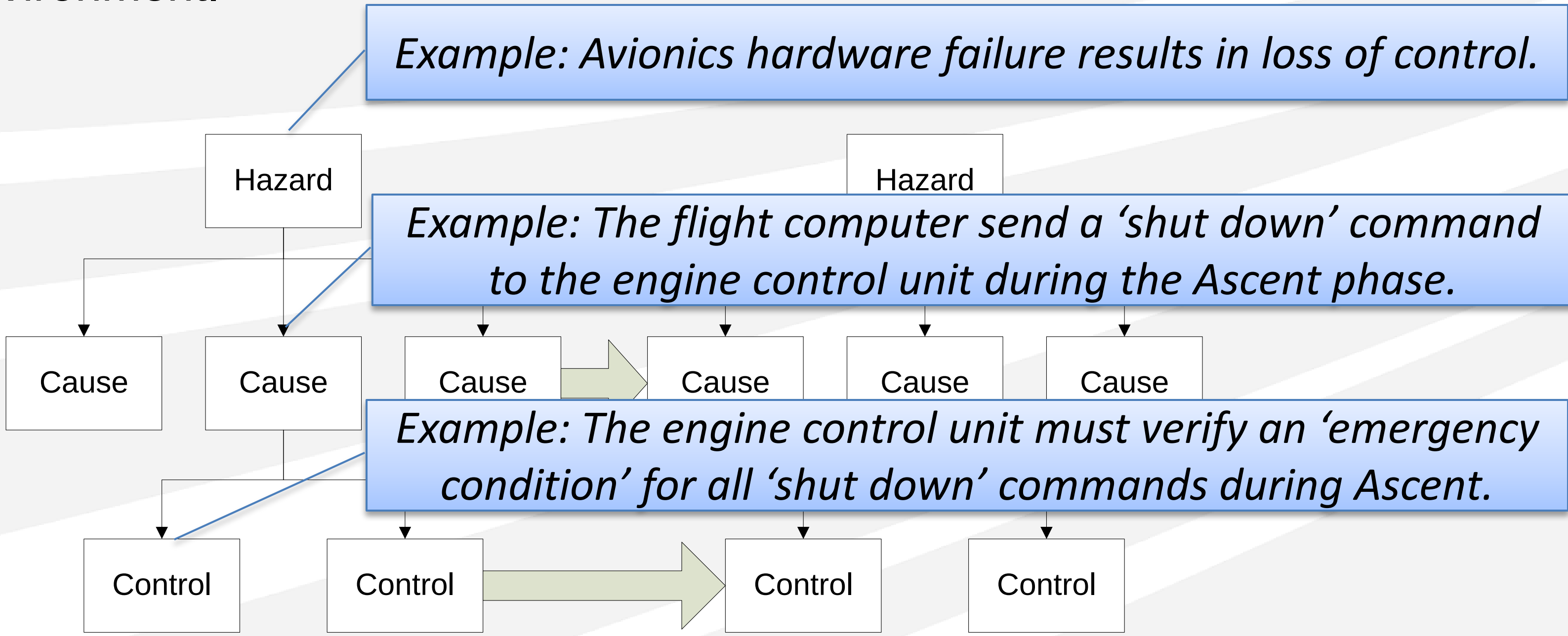
"How do I talk about software in relation to the rest of the system?"

Early visibility: How do I assess software safety when there is no software?



Managing safety risk through process – hazard analysis

Hazard analysis is a top-down approach to system safety that identifies potential conditions that could lead to loss of life, injury, damage to equipment or the environment.



Example: Avionics hardware failure results in loss of control.
Hazard
Hazard
Example: The flight computer send a 'shut down' command to the engine control unit during the Ascent phase.
Cause
Cause
Cause
Cause
Cause
Cause
Example: The engine control unit must verify an 'emergency condition' for all 'shut down' commands during Ascent.
Control
Control
Control
Control

- Hazard reports are created by safety engineers and stored in a hazard tracking system
- Hazard reports are reviewed at development milestones before development proceeds
- Hazard analysis is governed by a process document

Examining safety assurance processes

The processes for achieving safety must satisfy three assumptions:

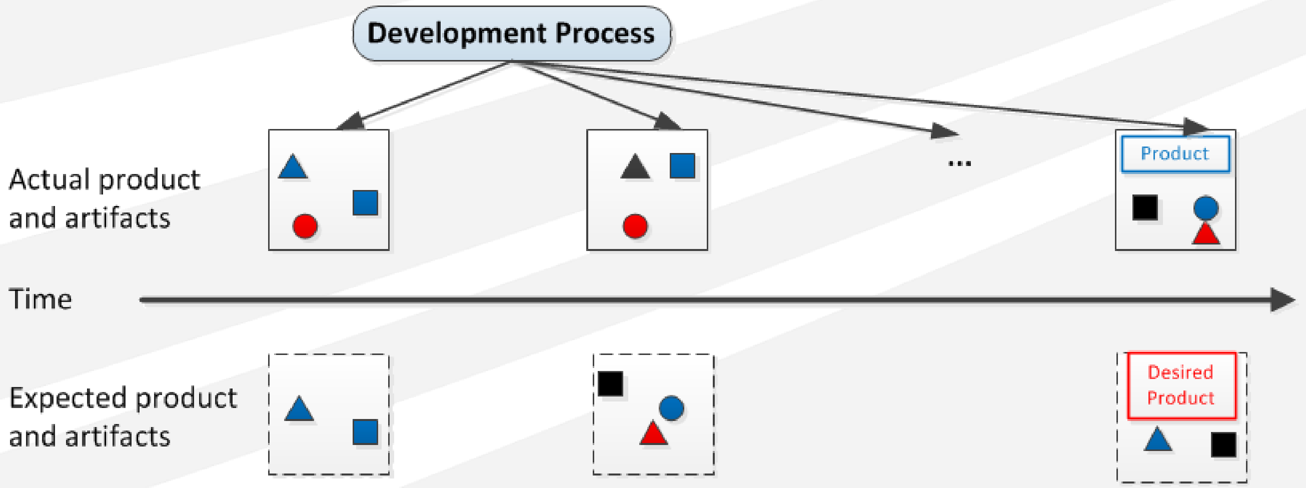
1. The process is capable of achieving safety or mitigating the risk of not being safe;
2. The process is appropriate for the development context;
3. The process is followed correctly

If a process fails to meet any of these three assumptions, there is a **risk** that the product will not achieve acceptable levels of safety

Safety risk measurement approach

Approach: Measure process artifacts with respect to the safety risks they are meant to mitigate.

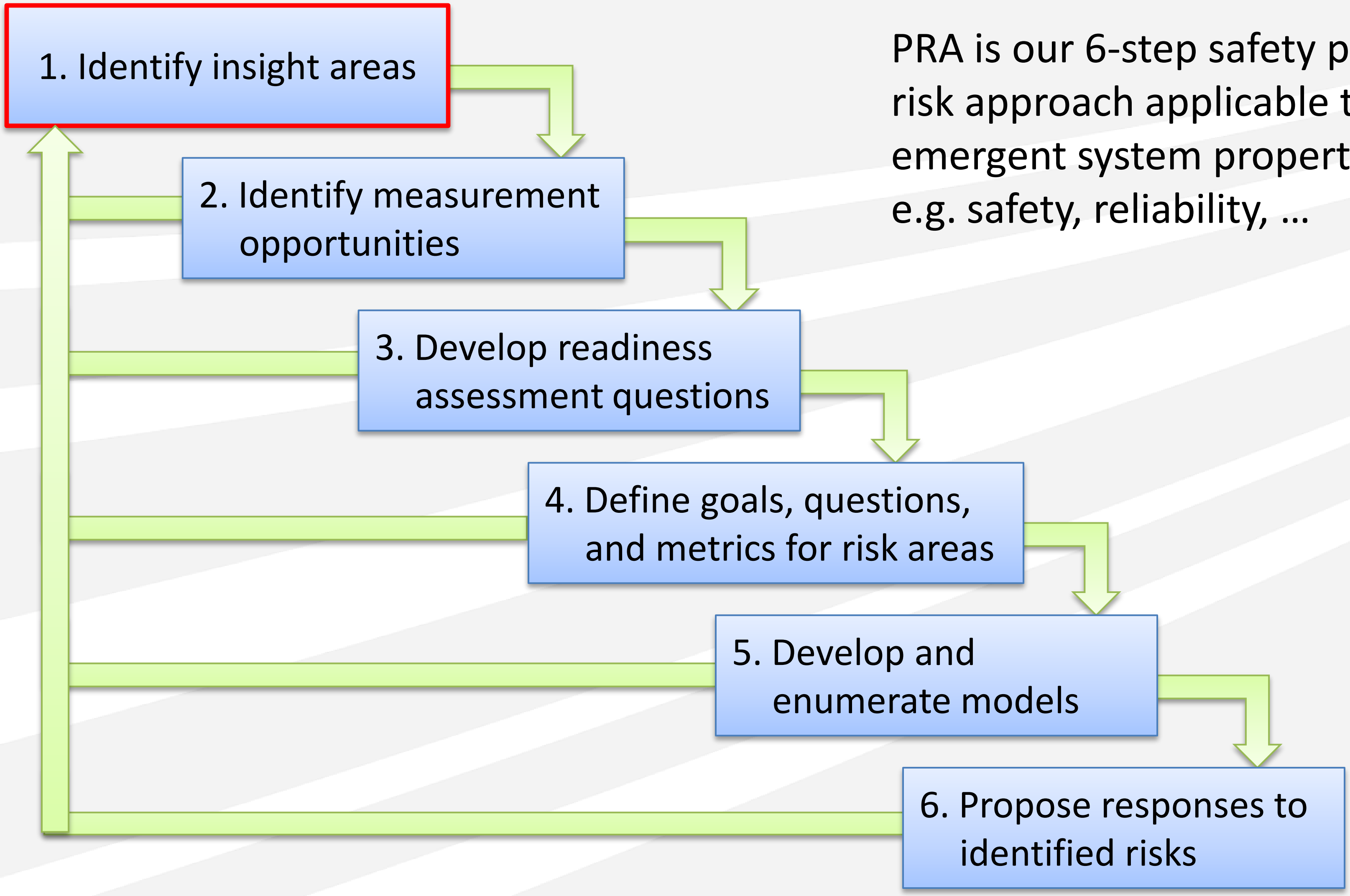
- Process artifacts contain indicators of potential risk.
- Processes and process artifacts are available throughout development.
- Quantifiable measures for trend analysis, baselines and comparison.



Outline

1. The challenge: measuring software safety risk early in the lifecycle
2. Apply the Process Risk Assessment (PRA) methodology to NASA's Constellation program
3. Common software safety process risks, lessons learned and recommendations

Process Risk Assessment Method


1. Identify insight areas
2. Identify measurement opportunities
3. Develop readiness assessment questions
4. Define goals, questions, and metrics for risk areas
5. Develop and enumerate models
6. Propose responses to identified risks

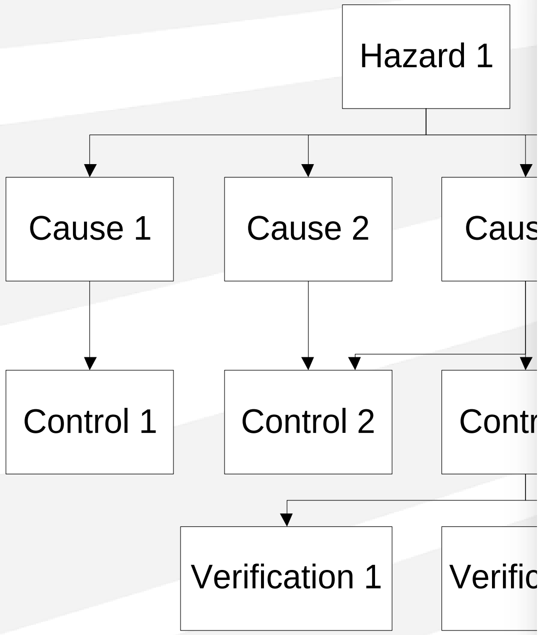
PRA is our 6-step safety process risk approach applicable to emergent system properties, e.g. safety, reliability, …

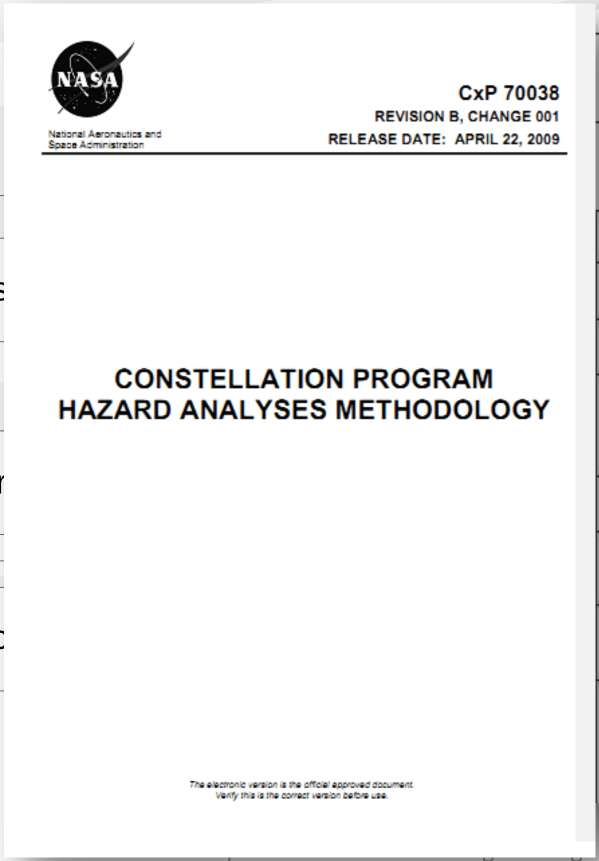
Step 1: Identify insight areas

Identify **intermediate outputs** of a process that can provide insights into process conformance and effectiveness

The set of hazards, with its causes, controls, and verifications

The relationship between hazard causes, controls, and verifications over time

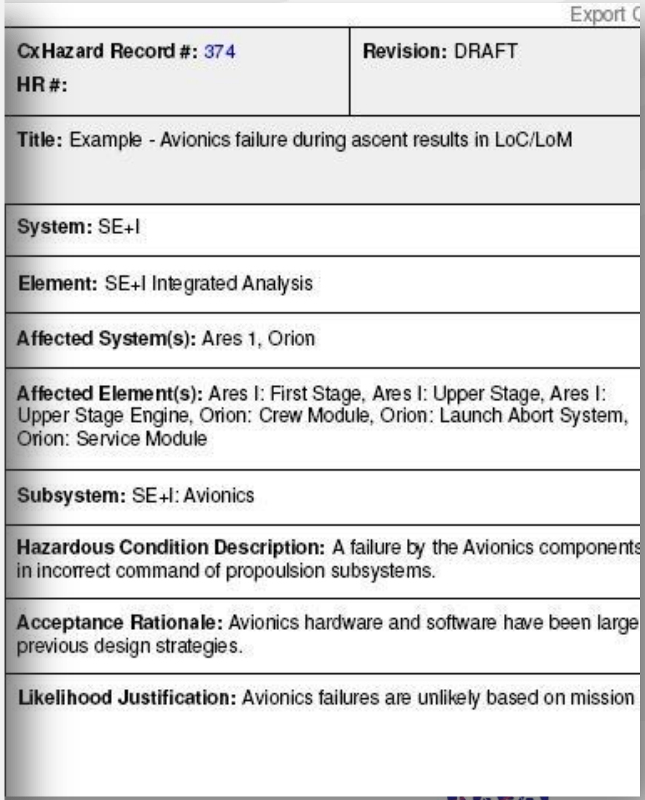



National Aeronautics and Space Administration

CxP 70038
REVISION B, CHANGE 001
RELEASE DATE: APRIL 22, 2009

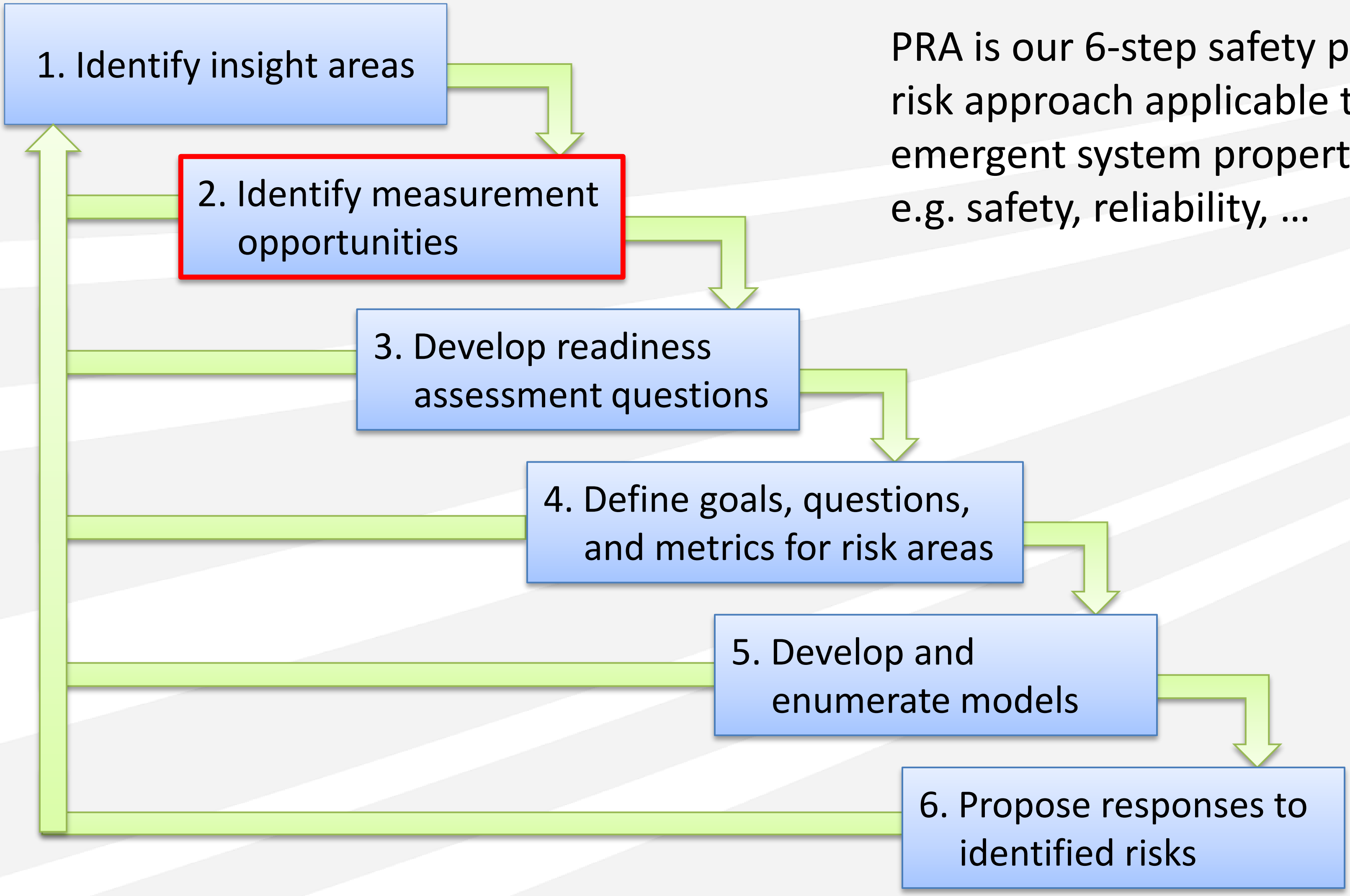
CONSTELLATION PROGRAM
HAZARD ANALYSES METHODOLOGY

The electronic version is the official approved document.
Verify this is the correct version before use.


Export C

CxHazard Record #: 374 **HR #:**	**Revision:** DRAFT
Title: Example - Avionics failure during ascent results in LoC/LoM	
System: SE+I	
Element: SE+I Integrated Analysis	
Affected System(s): Ares 1, Orion	
Affected Element(s): Ares I: First Stage, Ares I: Upper Stage, Ares I: Upper Stage Engine, Orion: Crew Module, Orion: Launch Abort System, Orion: Service Module	
Subsystem: SE+I: Avionics	
Hazardous Condition Description: A failure by the Avionics components in incorrect command of propoulsion subsystems.	
Acceptance Rationale: Avionics hardware and software have been large previous design strategies.	
Likelihood Justification: Avionics failures are unlikely based on mission	

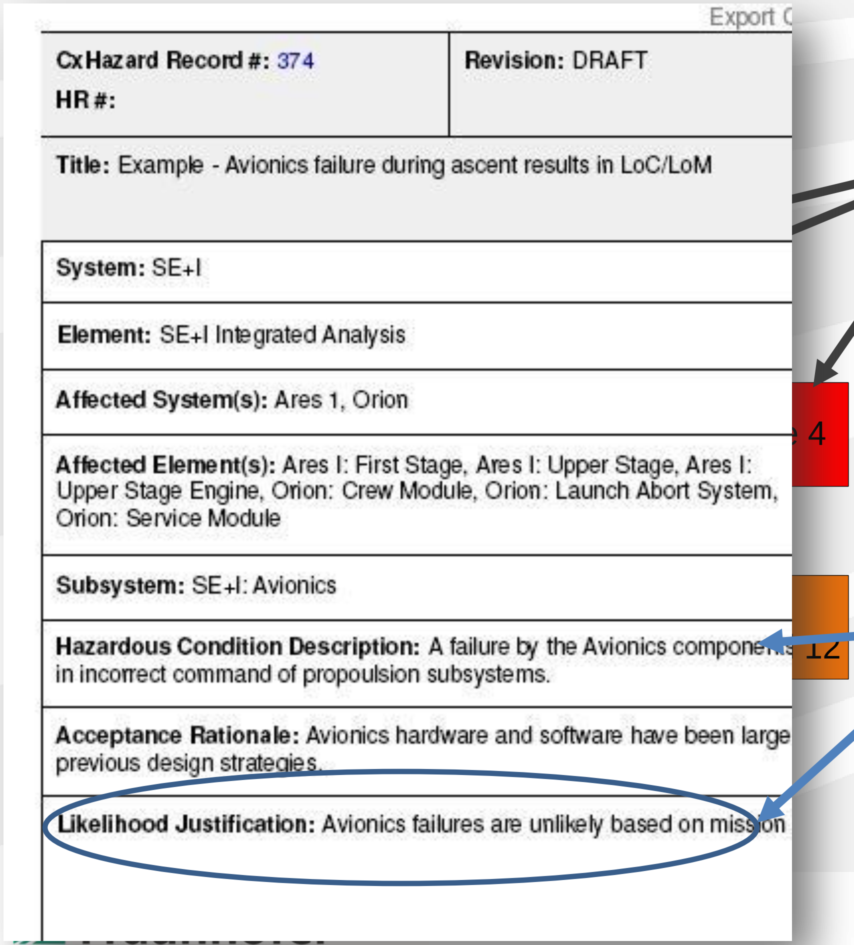
Process Risk Assessment Method


1. Identify insight areas
2. Identify measurement opportunities
3. Develop readiness assessment questions
4. Define goals, questions, and metrics for risk areas
5. Develop and enumerate models
6. Propose responses to identified risks

PRA is our 6-step safety process risk approach applicable to emergent system properties, e.g. safety, reliability, …

Step 2: Identify measurement opportunities

Evaluate each insight area for information that **could be used to measure** technical or process risks



Export C

CxHazard Record #: 374 **HR #:**	**Revision:** DRAFT
Title: Example - Avionics failure during ascent results in LoC/LoM	
System: SE+I	
Element: SE+I Integrated Analysis	
Affected System(s): Ares 1, Orion	
Affected Element(s): Ares I: First Stage, Ares I: Upper Stage, Ares I: Upper Stage Engine, Orion: Crew Module, Orion: Launch Abort System, Orion: Service Module	
Subsystem: SE+I: Avionics	
Hazardous Condition Description: A failure by the Avionics compone in incorrect command of propoulsion subsystems.	
Acceptance Rationale: Avionics hardware and software have been large previous design strategies.	
Likelihood Justification: Avionics failures are unlikely based on mission	

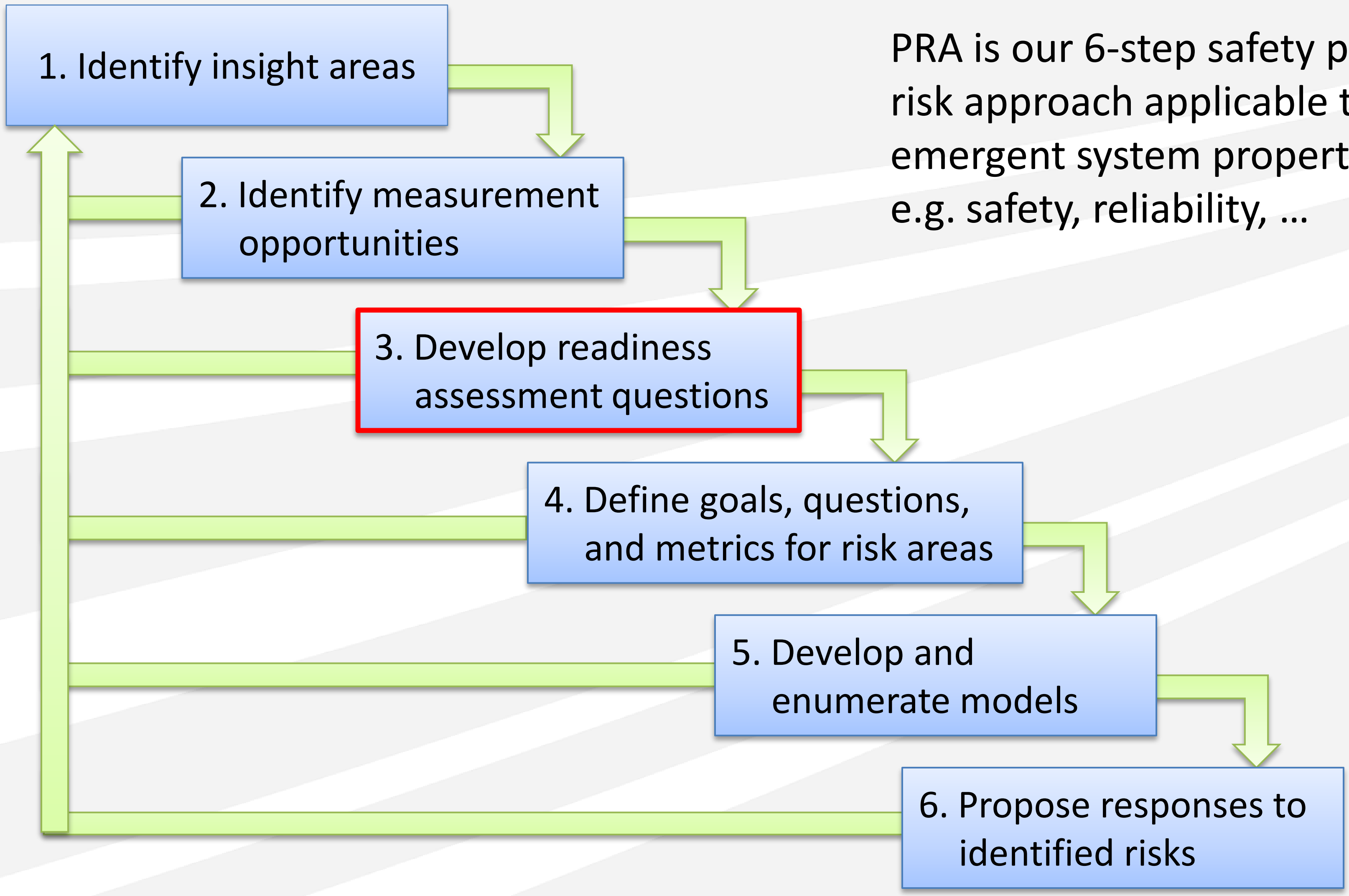
Quantify software's prevalence in hazards by counting hazards, causes and controls with software

Find the subsystems with the most software risks

What is the quality of this information?

Is the information complete and syntactically correct?

Process Risk Assessment Method

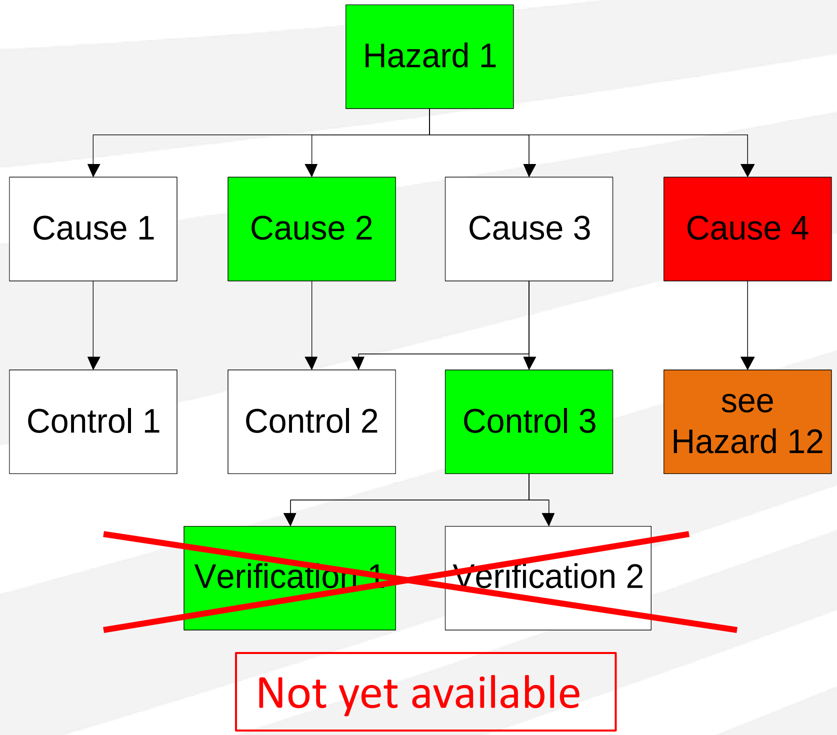

1. Identify insight areas
2. Identify measurement opportunities
3. Develop readiness assessment questions
4. Define goals, questions, and metrics for risk areas
5. Develop and enumerate models
6. Propose responses to identified risks

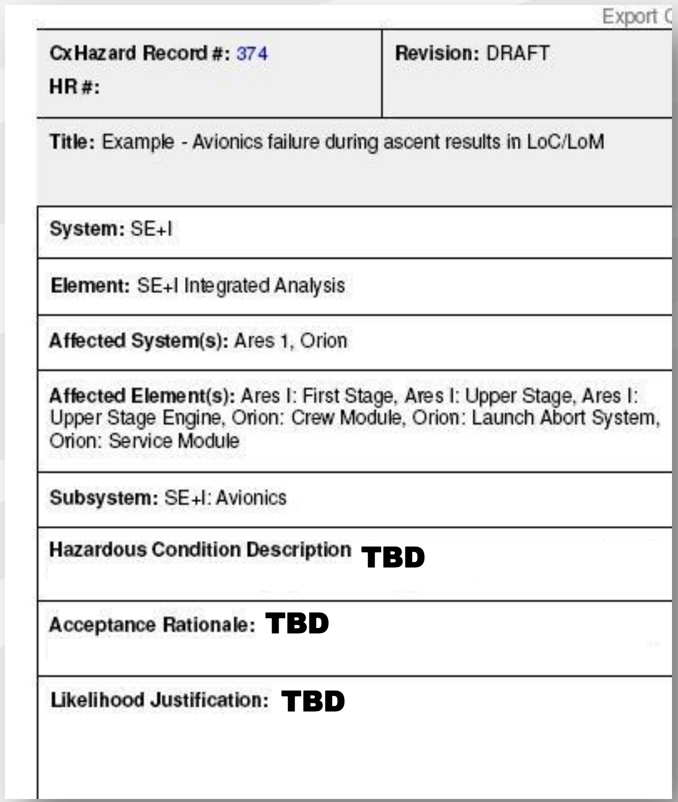
PRA is our 6-step safety process risk approach applicable to emergent system properties, e.g. safety, reliability, …

Step 3: Develop readiness assessment questions

Determine if it is possible to **delve deeper** into the area

e.g., Are the cause, control and verification data available, up to date, and complete enough for analysis?

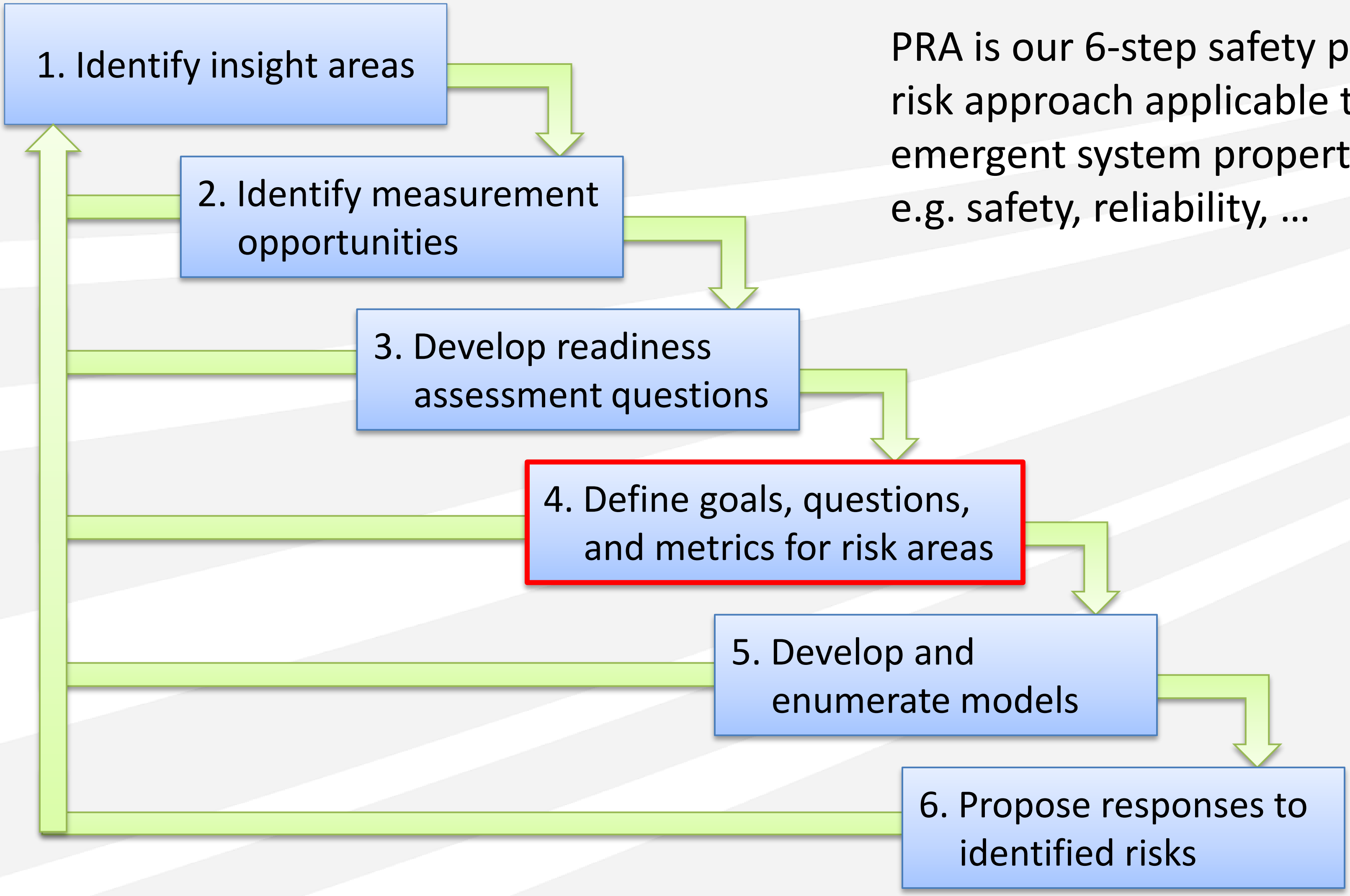




Export C

CxHazard Record #: 374 **HR #:**	**Revision:** DRAFT
Title: Example - Avionics failure during ascent results in LoC/LoM	
System: SE+I	
Element: SE+I Integrated Analysis	
Affected System(s): Ares 1, Orion	
Affected Element(s): Ares I: First Stage, Ares I: Upper Stage, Ares I: Upper Stage Engine, Orion: Crew Module, Orion: Launch Abort System, Orion: Service Module	
Subsystem: SE+I: Avionics	
Hazardous Condition Description **TBD**	
Acceptance Rationale: **TBD**	
Likelihood Justification: **TBD**	

Process Risk Assessment Method


1. Identify insight areas
2. Identify measurement opportunities
3. Develop readiness assessment questions
4. Define goals, questions, and metrics for risk areas
5. Develop and enumerate models
6. Propose responses to identified risks

PRA is our 6-step safety process risk approach applicable to emergent system properties, e.g. safety, reliability, …

Step 4: Define goals, questions and metrics

We used the Goal-Question-Metrics (GQM) Approach to formalize the goals from which we derived the questions and measures

We define our goals to be:

- *Goal 1:* Quantify the importance of software with respect to system safety; and
- *Goal 2:* Quantify the level of risk due to software by leveraging the hazard analysis process.

Goal 1 - Prevalence of software

Analyze the available set of hazards in order to characterize them with respect to the prevalence of software in *hazards*, *causes*, and *controls* from the point of view of NASA quality assurance personnel in the context of the Constellation program

Example Questions

What percentage of the hazards is software-related? A software-related hazard has at least one software cause or software control.

What percentage of hazard causes have software controls?

What percentage of hazard causes are non-software causes (e.g., hardware, operational error, procedural error) with software controls? These causes represent potentially "hidden" software risks.

Example Metrics

The number and percentage of software-related hazards

The number and percentage of software causes

The number and percentage of software controls

Goal 2 - Specificity of software causes

Analyze the *software causes* in a sample set of hazard reports in order to evaluate them with respect to the specificity of those software causes and hazards from the point of view of NASA quality assurance personnel in the context of the Constellation program.

Example Questions

What number and percentage of software causes is *well-specified*, *partially-specified,* or *generically-defined* according to the Constellation hazard analysis methodology requirements?

A well-specified software cause describes all of the following:

Origin – the CSCI (e.g., software component) that fails to perform its operation correctly

Erratum – a description of the erroneous command, command sequence or failed operation of the CSCI

Impact – the effect of the erratum which results in the hazardous condition, and if known, the specific CSCI(s) or hardware subsystem(s) affected

Example Metrics

- Count the software causes that are well-specified

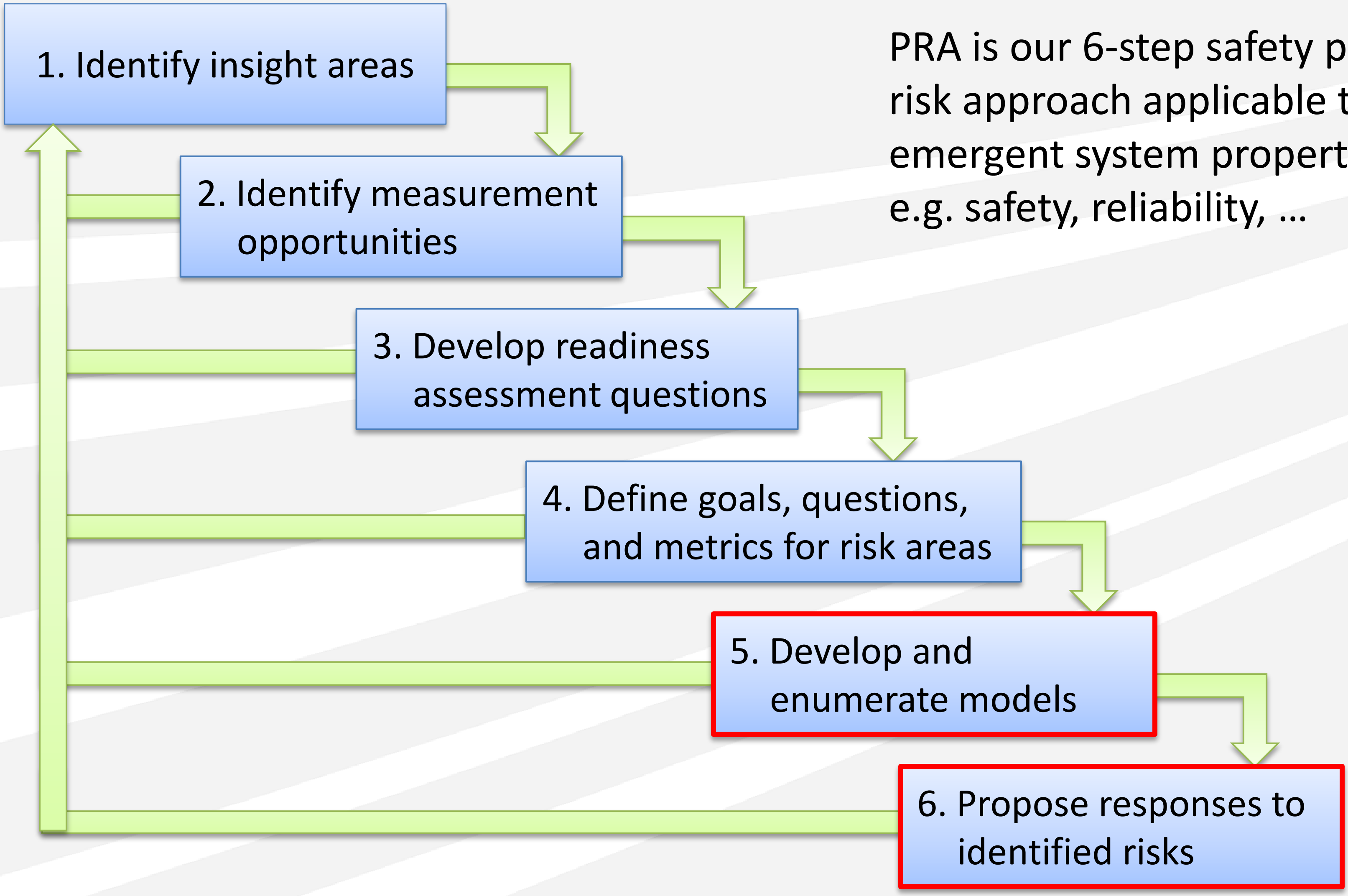
Step 4: Metric values

GQM for each risk area to expose risks associated with process artifacts

	Question	Sys A	Sys B	Sys C
1	What percentage of the hazards is software-related?	45%	67%	70%
2	What percentage of the hazard causes are software causes?	15%	12%	17%
3	What percentage of hazard causes are hardware causes with software controls (hidden software related hazards)?	14%	11%	-
4	What percentage of hardware causes has software controls?	16%	12%	-
5	What percentage of the causes has software controls?	29%	23%	-
6	What percentage of causes is transferred?	31%	22%	37%
7	What percentage of controls is transferred?	22%	11%	-
8	What percentage of the non-transferred hazard controls are specific software controls?	12%	14%	-
9	What percentage of the non-transferred hazard controls are references to “generic” software hazards?	5%	2%	-

* System C controls are in a format that prevented accurate assessment of whether the control is software or not.

Process Risk Assessment Method


1. Identify insight areas
2. Identify measurement opportunities
3. Develop readiness assessment questions
4. Define goals, questions, and metrics for risk areas
5. Develop and enumerate models
6. Propose responses to identified risks

PRA is our 6-step safety process risk approach applicable to emergent system properties, e.g. safety, reliability, …

Steps 5 and 6

- Step 5: Develop interpretation models and define threshold values
  - For each metric that was measured, define values that represent appropriate process conformance and those values that represent potential risk that the process is not being followed

- Step 6: Propose responses to identified risks, e.g., decisions and actions
  - Propose responses to the early risk identifications that can and should be taken as soon as possible to alleviate the risk

Fraunhofer USA
Center for Experimental Software Engineering

Steps 5 & 6: Interpreting and responding

Step 5 – **develop and enumerate models** of how the measures will be interpreted via **threshold values**.

Subsystem	HRs	SW HRs	SW related	% SW HRs	% SW related
Affected subsystem – Top 3 out of 52					
Avionics	26	13	13	50%	50%
Propulsion	34	12	18	35%	53%
Command & Data Handling	29	9	14	31%	48%

> 50% indicates increased software risk

Step 6 – propose responses to identified risks, e.g., decisions and actions.

1. Allocate additional Software Assurance personnel to design teams and product reviews to evaluate software risk.
2. Require dissimilar command monitoring software on separate partition for all software commands issued to this subsystem

Outline

1. The challenge: measuring software safety risk early in the lifecycle
2. Apply the Process Risk Assessment (PRA) methodology to NASA's Constellation program
3. Common software safety process risks, lessons learned and recommendations

Common software safety process risks

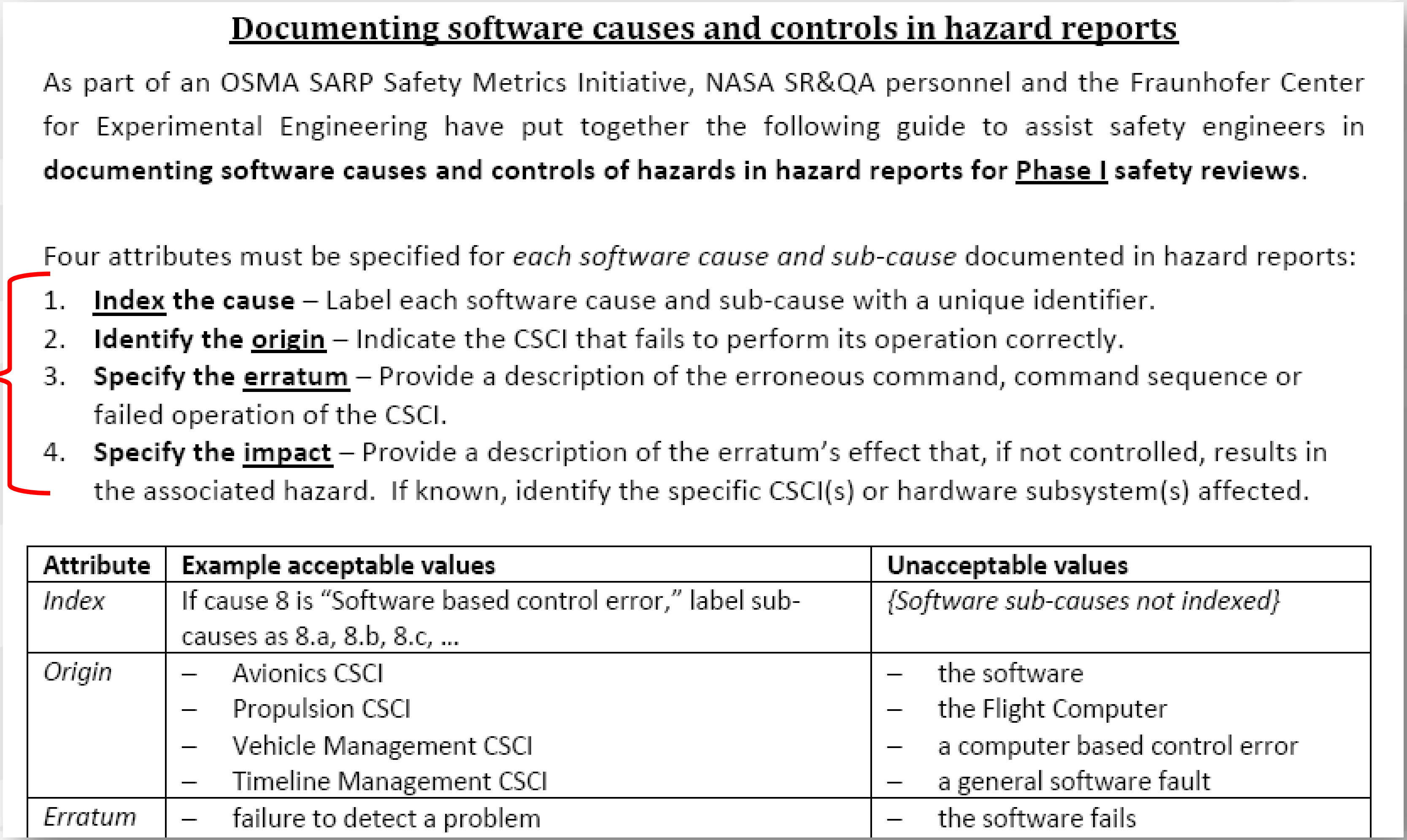
We applied the PRA method to evaluate software safety on three projects to date:

1. NASA Constellation project
2. A large, network-centric US Department of Defense system-of-systems
3. NASA satellite

Three process risks were common across the projects

1. Inability to track software safety hazards and requirements – software safety risks were often not specifically marked in the hazard reports
2. Inadequate traceability – No bi-directional traceability between safety requirements, hazards, causes and controls
3. Inconsistent scope and unstructured details – safety engineers on each project wrote their hazards, causes and controls in unique ways

Software cause "user guide"



Documenting software causes and controls in hazard reports

As part of an OSMA SARP Safety Metrics Initiative, NASA SR&QA personnel and the Fraunhofer Center for Experimental Engineering have put together the following guide to assist safety engineers in **documenting software causes and controls of hazards in hazard reports for Phase I safety reviews.**

Four attributes must be specified for *each software cause and sub-cause* documented in hazard reports:

Specificity

1. **Index the cause** – Label each software cause and sub-cause with a unique identifier.
2. **Identify the origin** – Indicate the CSCI that fails to perform its operation correctly.
3. **Specify the erratum** – Provide a description of the erroneous command, command sequence or failed operation of the CSCI.
4. **Specify the impact** – Provide a description of the erratum's effect that, if not controlled, results in the associated hazard. If known, identify the specific CSCI(s) or hardware subsystem(s) affected.

Attribute	Example acceptable values	Unacceptable values
Index	If cause 8 is "Software based control error," label sub-causes as 8.a, 8.b, 8.c, …	*{Software sub-causes not indexed}*
Origin	– Avionics CSCI – Propulsion CSCI – Vehicle Management CSCI – Timeline Management CSCI	– the software – the Flight Computer – a computer based control error – a general software fault
Erratum	– failure to detect a problem	– the software fails

Common software safety risks (2)

Institutional challenges:

- Integrating software safety with traditional safety processes that originated in hardware and system reliability.
- Defining how software should be incorporated into traditionally hardware-oriented analyses (such as hazard analysis) is still very much a work in progress
- Elevating software safety to a level of importance equivalent to hardware and system safety was challenging.

Lessons learned for future programs

1. Need to provide explicit guidance for applying safety analyses to software.

2. Need to plan for automated analysis and traceability and promote usage of the hazard tracking system capabilities.

3. Need to require software safety management and measurement in the acquisition process in order for appropriate data to be made available for safety analysis during development

Thanks and acknowledgement

Contact: Lucas Layman (llayman@fc-md.umd.edu)
Victor Basili (basili@cs.umd.edu)
Marv Zelkowitz (mvz@cs.umd.edu)

Full-length technical report: http://goo.gl/fEKc2

Acknowledgements

- This research was supported by NASA OSMA SARP grant NNX08AZ60G
- Thanks to Karen Fisher and Risha George at Goddard Space Flight Center

Some Publications

1. Basili V. R., Zelkowitz M. V., Layman L., and Dangle K., Obtaining valid safety data for software safety measurement and process improvement, Empirical Software Engineering and Measurement, Bolzano Italy, September 2010.

2. Layman L., Basili V. R., Zelkowitz M. V., and Fisher K. L., A case study of measuring process risk for early insights into software safety, ACM and IEEE Int. Conf. on Software Engineering, Honolulu, HI, May 2011.

3. Basili V. R., Layman L., Zelkowitz M. V., A Methodology for Exposing Software Development Risk in Emergent System Properties**,** Technical Report 11-101, Fraunhofer Center for Experimental Software Engineering College Park, Maryland 20740, April 21, 2011

4. Basili V. R., Layman L., Zelkowitz M. V., A Methodology for Exposing Software Development Risk in Emergent System Properties (Fraunhofer Technical Report).

BACKUP SLIDES

Fraunhofer USA Center for Experimental Software Engineering

Process Risk Assessment Method

PRA is our 6-step safety process risk approach applicable to emergent system properties, e.g. safety, reliability, …

I. Identifying insight opportunities	1. **Identify insight areas** from the development process that provide insight into risk areas. 2. **Identify measurement opportunities** that provide insight into each risk area.
II. Evaluating the quality of information	3. **Develop readiness assessment questions** to identify if it is possible to delve deeper into the area.
III. Measuring, interpreting, and providing advice	4. **Define goals, questions, and measures for each risk area** to expose risks associated with process artifacts. 5. **Develop and enumerate models** of how the measures will be interpreted via threshold values. 6. **Propose responses to identified risks**, e.g., decisions and actions.

Step 1: Details

- **Inputs**
  - The property you want to measure
  - The processes associated with achieving that property
  - The intermediate outputs of each step for each process
- **Outputs**
  - The set of process outputs or artifacts that should give us the most information about the effectiveness of the process for achieving the property, including:
  - The format of the output
  - Rationale as to how these outputs are of value for identifying the risk of non-conformance or evaluating the effectiveness of the process
- **Sample Activities or Questions to ask**
  - What are the process outputs created during application of the process?
  - How does that information grow or change over time?
  - Can I use this information to gain insight into whether the process is being performed appropriately and if the process is achieving its goals?

Step 2: Details

- **Inputs**
  - Process outputs/artifacts identified in step one

- **Outputs**
  - Potential metrics based on process outputs/artifacts

- **Sample Activities or Questions to ask**
  - What can I measure to determine if the desired product property (e.g. safety, reliability) is being achieved?
  - What can I measure to evaluate if the process is sufficient for achieving the desired property?
  - Can we identify potential bounds that provide insight for our goals? What is good or bad?

Step 3: Details

- **Inputs**
  - Proposed measurement opportunities and the associated risks they measure
- **Outputs**
  - Advice on how the intermediate outputs and metrics can be used to identify process risk
  - A high-level assessment of process conformance risk, i.e. are the processes producing meaningful outputs?
  - If step 3 fails, this is a likely indicator of process risk
  - If step 3 succeeds, then continue with the remainder of PRA to measure risk
- **Sample Activities or Questions to ask**
  - Examine the process artifacts and try to apply the proposed metrics. Can I apply the metric?
    - Is the information accessible and available?
    - Is the information in good enough form that it can be measured?
  - If I cannot apply a metric, why not?

Step 4: Details

- **Inputs**
  - A set of proposed metrics that have passed the readiness assessment check
- **Outputs**
  - A GQM structure with specific goals, questions and metrics
- **Activities or Questions to ask**
  - Apply the GQM method to derive a goal template, the questions, and what measures are needed.
    - What is the object of study?
    - What is the specific focus of the measure?
    - What is the purpose of the measure?
    - Who is the person who needs to make a decision about the results of this measure?
    - What are the context variables that might influence the interpretation of the results?
    - Given the goals and questions, what are the metrics?

Step 5: Details

- **Inputs**
  - A set of goals, questions and metrics to be collected
- **Outputs**
  - A set of models that provides indication that there may be a risk
- **Activities or Questions to ask**
  - Define a set of measures and interpretation models for those metrics, based upon what data is available or can be assumed, to provide indicators that there is a risk that the process is not being followed and the product is at risk of not satisfying the particular property.
    - What is the expected value of that metric and possible margin of error, i.e. what is the range of values that would be acceptable?
    - Do historical data exist for any of the metrics?
    - Are there proxies for the bounds on these metrics?
    - Can we gather any expert opinion on the bounds?

Step 6: Details

- **Inputs**
  - Metrics and an interpretation model
  - Data from intermediate project artifacts

- **Outputs**
  - Advice on what the project should do if we are outside the acceptable bounds and there is a risk

- **Activities or Questions to ask**
  - Provide expert safety engineer advice on what to do under the circumstances

Fraunhofer USA
Center for Experimental Software Engineering

Summary

- PRA identified metrics based on hazard data to **quantify risk early** in the lifecycle
  - Identified **early risk** in three systems
  - Created a **baseline** for comparison with future review milestones and projects
  - Measures used as input into a **Software Risk Dashboard** to identify subsystems and mission phases with highest software risk
- PRA was able to **identify process problems** and improve the processes
  - Identified where and why the **process** was **not** being **followed**
  - Developed **guidelines** developed for safety engineers to describe **software causes** of hazardous conditions
  - Developed a **draft "Handbook"** for incorporating and improving **software assurance oversight for acquisitions** (RFP process)
- PRA demonstrated the benefits of **automated analysis** of early data
  - Built a prototype Hazard Tracking System tool that demonstrated the benefits of additional data and traceability when analyzing software safety risk