'The Product Engineering Class in the
Software Safety Risk Taxonomy
for Building Safety-Critical Systems'

Janice L. Hill
NASA, Kennedy Space Center, Florida
Janice.L.Hill@nasa.gov
Daniel Victor
ManTech SRS
Daniel.C.Victor@nasa.gov

Paper to be presented at the
Australian Software Engineering Conference, ASWEC 2008
Perth, WA
March 24-28, 2008

Background

- Safety Standards contain technical and process-oriented safety requirements.
- The best time to include these requirements is early in the development lifecycle of the system.
- Software Safety requirements, such as the NASA-STD-8719.13B Software Safety Standard, can be imposed on legacy safety-critical systems.
- Retrospective safety cases need to be formulated as part of recertifying the legacy systems for further use.
- This can be a difficult task because there may be few to no artifacts available to show compliance to the software safety requirements.

The Problem

- The risks associated with not meeting safety requirements in a legacy safety-critical computer system must be addressed to give confidence for reuse.
- A problem arises when attempting to fulfill the requirements of a software safety standard in a legacy real-time safety-critical computer system.
- "How do we retrospectively make a safety case for the software, perhaps to meet new safety standards in the industry?" [1]
- A methodology is needed to accomplish this.

The Methodology - 1

- In some cases with legacy systems, it can be a difficult task to construct a safety case, because there may be few to no artifacts available to show compliance with the software safety requirements.
- Risk factors in general will be different for legacy safety-critical computer systems, and the software within them.
- These software safety risks must be addressed by project management to give confidence for reusing an existing system.
- Knowing the risks, project managers can then decide whether to try to recreate missing artifacts or accept the risks of not having certain safety documents or analyses to make the safety case.

The Methodology - 2

- A **Software Risk Evaluation** (SRE) is a practice that was developed by the Software Engineering Institute (SEI) containing a formal method for identifying, analyzing, communicating and mitigating software technical risk. [2]
- The SEI's **Software Development Risk Taxonomy** is a part of this practice.
- Scientists from the SEI developed this taxonomy in the mid 1990's and used it with new software development projects.
- They were able to collect data from several projects to show where the most risk occurred in the lifecycle of a project.

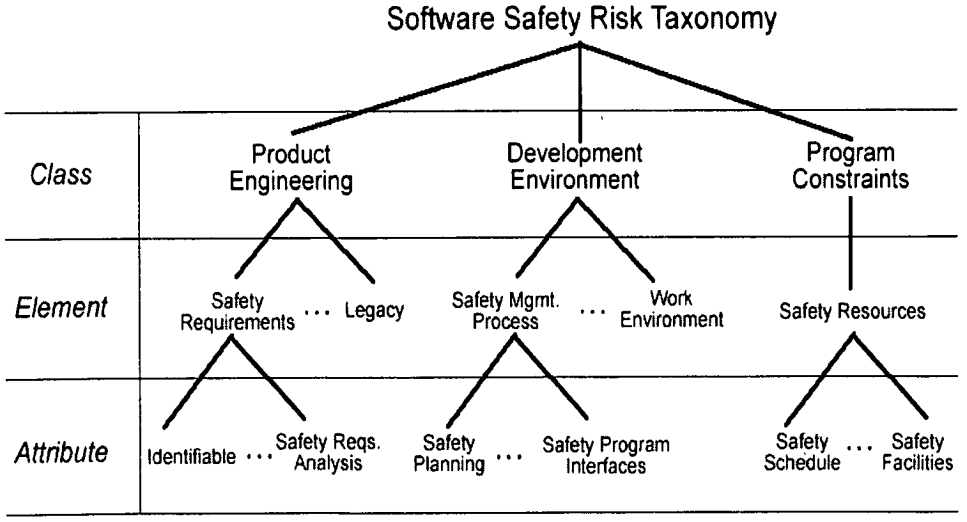
The Methodology - 3

- For our research, there is a need for a taxonomy specifically focused on identifying *software safety* risk factors.
- NASA has a requirement to re-evaluate safety-critical legacy systems for reuse.
- The ***Software Safety Risk Taxonomy*** is proposed as a partial solution for making retrospective safety cases.
- The Software Safety Risk Taxonomy was introduced at the SEW 31, IEEE/NASA Software Engineering Workshop last March.

The Methodology - 4

- In this research the ***Software Safety Risk Taxonomy*** is being used in addition to the SEI's taxonomy to generate a comprehensive list of questions for defining an inclusive set of risks for legacy safety-critical computer systems.
- Used in conjunction with the SEI taxonomy, the Software Safety Risk Taxonomy helps paint a complete risk profile for a safety-critical system.
- The ***Software Safety Risk Taxonomy*** addresses the additional safety related tasks and analyses that are required over and above traditional software engineering process activities.
- We are piloting the use of both taxonomies on several small projects at KSC.

Software Safety Risk Taxonomy



Product Engineering Class

***Safety* Requirements**
- ***Identifiable***
- **Stability**
- **Completeness**
- **Clarity**
- **Validity**
- **Feasibility**
- ***Safety requirements traceability***
- ***Safety requirements analysis***

***Safety* Design**
- ***Safety* Functionality**
- **Difficulty**
- ***Safety* Interfaces**
- ***Safety* Performance**
- ***Safety* Testability**
- **Hardware Constraints**
- **Non-Developmental Software**
- ***Safety design traceability***
- ***Safety design analysis***

***Safety* Code and Unit Test**
- **Feasibility**
- ***Safety* Testing**
- **Coding/Implementation**
- ***Safety code traceability***
- ***Safety code analysis***

***Safety* Integration and Test**
- ***Safety* Environment**
- **Product Integration**
- ***Safety test traceability***
- ***Safety test analysis***

Engineering Specialties
- ***Safety* Maintainability**
- **Reliability**
- **Security**
- **Human Factors**
- **Specifications**

Legacy
- ***Reverse engineering***
- ***Replacement***

Product Engineering Defined

- Product engineering is defined as the technical processes to define, design and construct or assemble a product. [3]
- Product engineering ***for safety*** is defined as the technical processes used ***to build a safety-critical product***.
- It refers to the system engineering and software engineering activities involved in creating a safety-critical system that satisfies specified safety requirements and customer expectations. [4]
- Activities include system hazard analysis, system and software safety requirements analysis and specification, system and software safety design and implementation, integration of hardware and software components, and software and system test for safety-critical systems.

Product Engineering Class, Element and Attributes

- In this paper, we formally define each Element and Attribute in the Product Engineering Class of the safety taxonomy.
- Additionally, we describe areas where risks may be found.
- These definitions are the foundation for the development of the questions for the Software Safety Taxonomy Based Questionnaire, TBQ.
- The Product Engineering Class was chosen first because it is the largest of the three classes in the safety taxonomy.

Product Engineering Class, Safety Design Element, Attribute Example – Safety Performance

Performance is defined as the degree to which a system or component accomplishes its designated functions within given constraints, such as speed, accuracy, or memory usage. [11] Safety performance is defined as the ability of a safety-critical system to handle periodic capacity, load and timing requirements; this is a fundamental safety property. [10] The safety performance attribute refers to time critical performance; real time response requirements, performance analyses, reliability analyses, user response requirements, 'must work' and 'must not work' requirements, failure detection, isolation and recovery requirements.

[10] NASA Office of Safety and Mission Assurance, *NASA-GB-8719.13 NASA Software Safety Guidebook*, 2004.
[11] Standards Coordinating Committee of the Computer Society of the IEEE, *IEEE Std. 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology*,The Institute of Electrical and Electronics Engineers, New York, 1990

Product Engineering Class, Safety Requirements Element, Example – Safety Requirements Analysis TBQ questions

A. Product Engineering

1. Safety Requirements

h. Safety requirements analysis

[Are safety requirements analyzed using a specified methodology?]

[1] Was a Preliminary Hazard Analysis (PHA) performed for this system?
(*Yes*) Is the PHA available for review?
(*Yes*) Is software included as a part of the PHA?

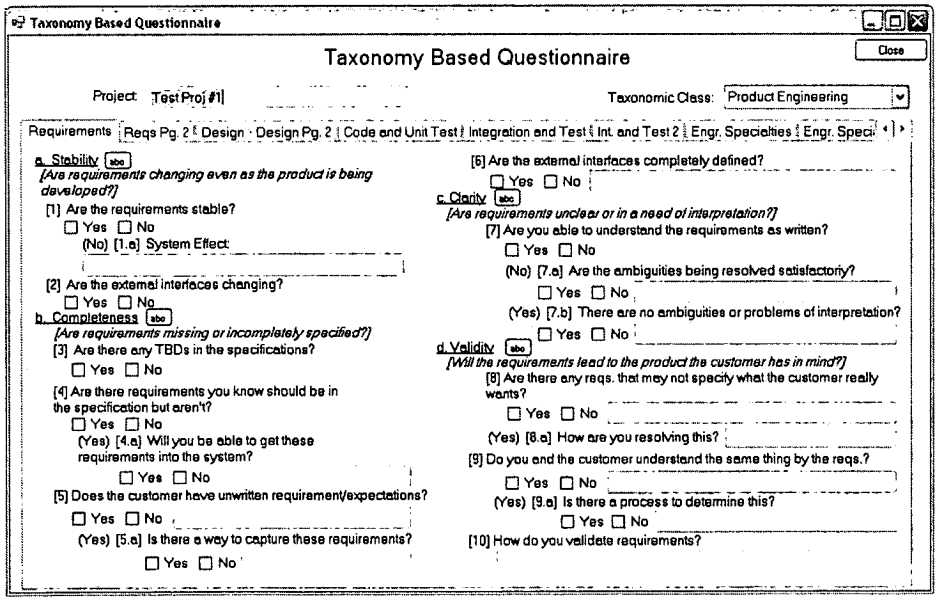
[2] Was a System Safety Analysis (SSA) performed for this system?
(*Yes*) Is the SSA available for review?
(*Yes*) Is software included as a part of the SSA?

[3] Are the system and software safety requirements analyzed for proper flow down from the system level requirements?
(No) Who is responsible for doing the safety analyses?

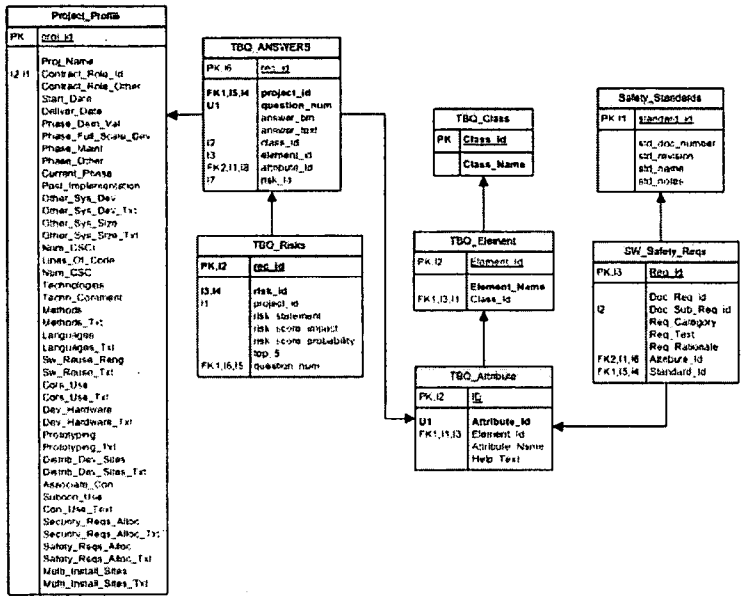
[4] What types of safety analyses are performed?
a. Requirements Criticality Analysis
b. Software Fault Tree Analysis
c. Software Safety Requirements Flow-down Analysis
d. Timing, Throughput and Sizing Analysis
e. Peer Reviews and Inspections of safety requirements
f. Traceability Analysis
g. Control Flow Analysis
h. Information Flow Analysis

[5] Are safety analyses documented?
(*Yes*) Are the documented analyses results under configuration control?

Product Engineering Class, Requirements Element, Prototype of the Software Development Risk Taxonomy



Legacy Systems Risk Database: Preliminary Data Model



- Questions?