On Time...On Target
Land, Sea, Air and Space

Software Quality Assurance

Applied towards the Development of VHDL-Based Safety Critical Hardware

David A. Geremia

Principal Electrical Design Engineer

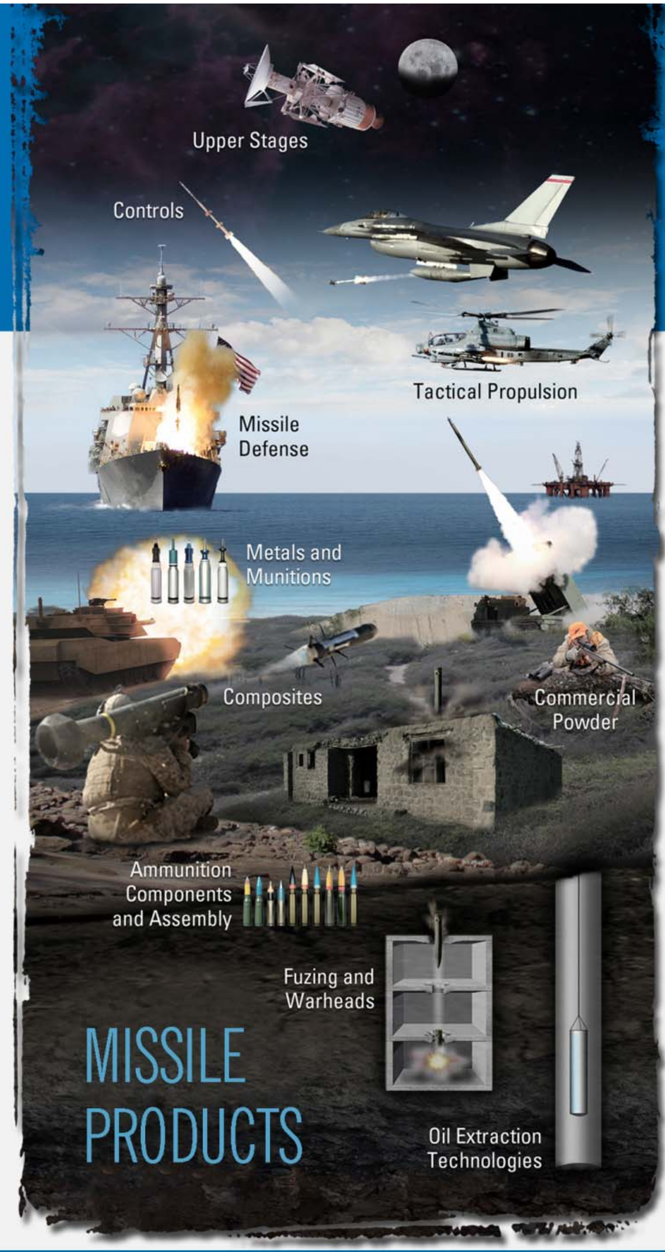
david.geremia@orbitalatk.com

61st NDIA Annual Fuze Conference

San Diego CA

May 16, 2018

NOTICE - This document does not contain export controlled technical data.



Introduction

- The software used in today's safety critical systems requires a significant amount of analysis and testing as well as traceability to the requirements
- "Software-like" languages are treated similarly by today's munition-related safety technical review panels

- Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) is one of these "software-like" languages
- Requires the generation of the appropriate Level of Rigor (LOR) and the resultant analyses
- As part of the academic pursuit on which this presentation is based, software was created in order to automate the generation of the appropriate LOR tasks, establish traceability, & provide transparency

Background

- The implementation of safety features in safety critical systems has evolved in the last few decades
- Initially, safety features were implemented using a mechanical means such as springs, setback weights, rotors and shear pins
- Recently, electronics have been used in order to implement safety features i.e. analog and/or simple digital circuits

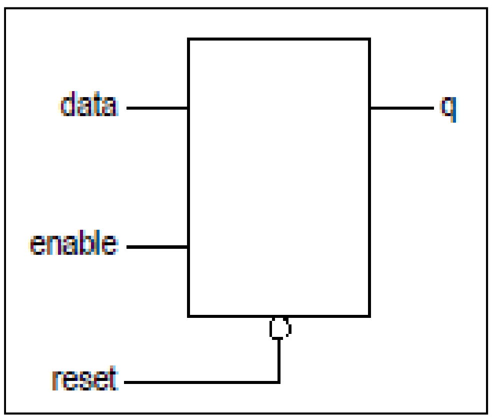
- Most recently, software and “software-like” devices are being used to implement safety features
- Field Programmable Gate Arrays (FPGAs) are hardware devices that are being used more often in today’s munition-related safety-critical applications in order to implement safety features
- A high-level language (such as VHDL) is used to design the safety features which are implemented using an FPGA.

- VHDL provides flexibility to the design engineer through being an abstract programming language
- Abstraction provides many benefits but tends to be the opposite of what a safety technical review panel desires
- Current Software System Safety analysis techniques may be applied towards the contribution of VHDL towards the total system risk.

- Behavioral VHDL allows for a high level of abstraction.
- The system is described in terms of what it does.
- Programmer is specifying the relationship between the inputs and the outputs
- The logic is described in a source code like manner using statements that are typical of conventional programming language

Schematic Entry vs. Behavior VHDL

- VHDL allows for the description of the structure of the system
- Allows for the specification of the system using familiar programming language forms



Digital Latch

```
begin
process (enable, data, reset) begin
      if (reset = '0') then
        q <= '0';
      elsif (enable = '1') then
        q <= data;
      end if;
end process;
```

VHDL Representation of Digital Latch

- Software Quality Assurance (SQA) monitors the entire process of software engineering.
- Assurance may be defined as the "Implementation of inspection and structured testing as a measure of quality."
- This research focused on the process and testing aspect of Software Quality Assurance as it applies to "software-like" hardware devices such as FPGAs.

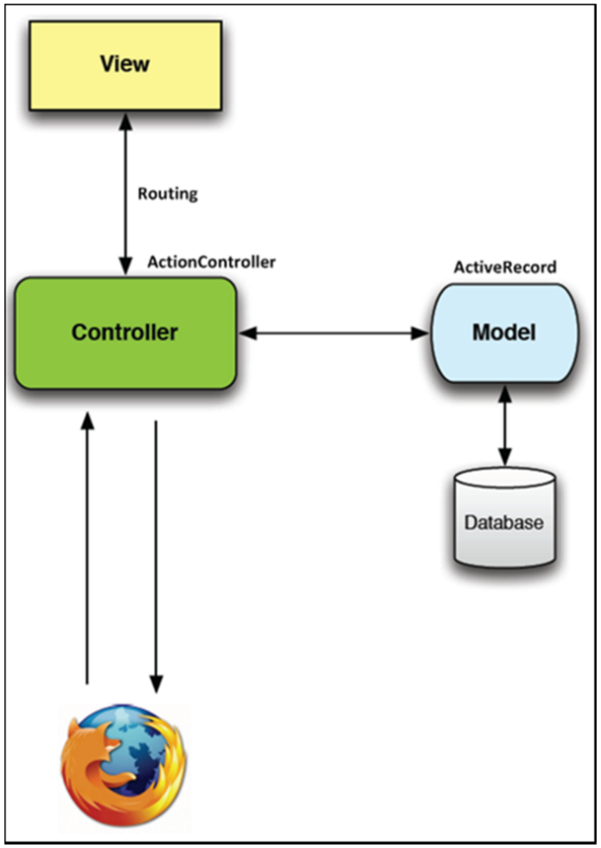
- The process flow could be increased and better traceability to the requirements provided through the use of collaborative, web-based software.
- This software is used to <u>generate the Level of Rigor tasks</u> and track the required artifacts in a real-time, multiuser environment.
- This collaborative program was created using the Ruby on Rails web-based framework. Allows for synergy among all team members.

Ruby on Rails

- Ruby on Rails was chosen as the framework for the development of the Requirements Tracking web application.
- The user would be able to take advantage of collaboration among their colleagues, decreasing the likelihood of a safety critical item being missed.
- The web application framework provides a structure that allows for the tracking of the various system safety analyses.
- Each analysis will require specific items or entities that must be entered into the database and tracked.
- These entities will also require relationships among them to be defined.
- The web application will guide the user through the Level of Rigor task selection process and create a common structure for the compliance process.

Ruby on Rails

- Ruby on Rails uses the Model View controller (MVC) architectural pattern
- Browser is routed to the Controller which translates the data from the Model into a viewable form using the View



MVC Architecture, M. Hartl

- There is no one specification that governs munition related software safety.
- Details of the use of logic devices as safety features are covered in JOTP-051.
- AOP-52 is a NATO document that provides guidance on munition-related software safety.
- The Joint Software Systems Safety Engineering Handbook (JSSSEH) is a DoD publication whose purpose is to provide guidelines to achieve a reasonable level of assurance that the software will execute within an acceptable level of risk.
- MIL-STD-882E is the Department of Defense System Safety Standard Practice document which contains an Appendix B on Software System Safety and Analysis

- Developed as a result of political pressure after several catastrophic mishaps which occurred in the 1950s, such as Atlas and Titan rockets exploding in their silos during testing
- Found during the investigations into those events that the failures were related to <u>deficiencies in the design, testing and management of the systems</u>
- Determined that the deficiencies should have been detected and corrected.

- The standard acknowledges that risk and probability cannot be the only part of the risk assessment.
- It is very difficult to determine the probability of the failure of a specific software function.
- Therefore, the potential risk severity and the degree of control that the software exercises over the hardware is used to assess the software subsystem's contribution to the system risk

Methodology

- A relational database was created in order to streamline the generation and traceability of the system software safety requirements known as the Level of Rigor.
- The database requirements were determined by reviewing the applicable standards: JOTP-51, AOP-52 & JSSSEH.

Results

- The web based framework provides an easy means by which the user can record and track safety related information for their program.
- The purpose of the software was to make it easier for the user to generate the appropriate LOR tasks.

- The index webpage identifies the initial system safety process.
- The user must begin at the first item in the list (PHL) and move downwards though the remaining analyses such as the PHA and FHA.

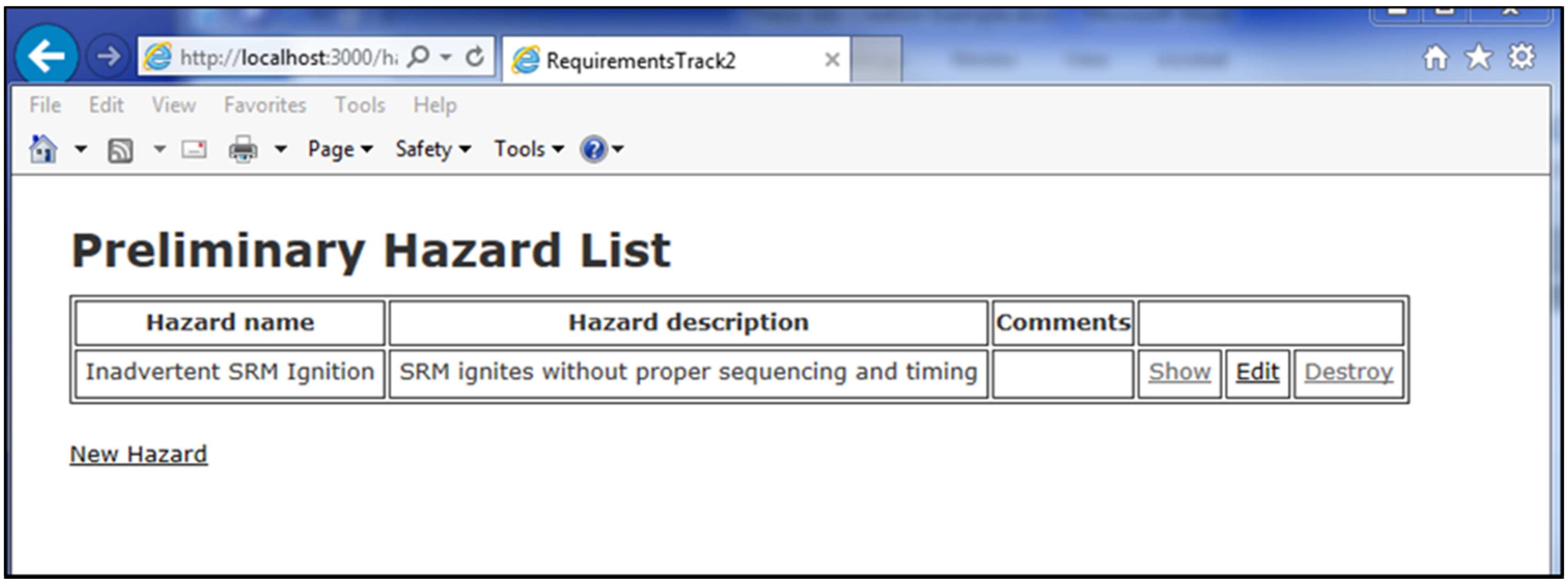
Welcome to Requirements & Hazard Tracker Database

Select Each Link and Complete the Forms to Determine the Level of Rigor for your Program

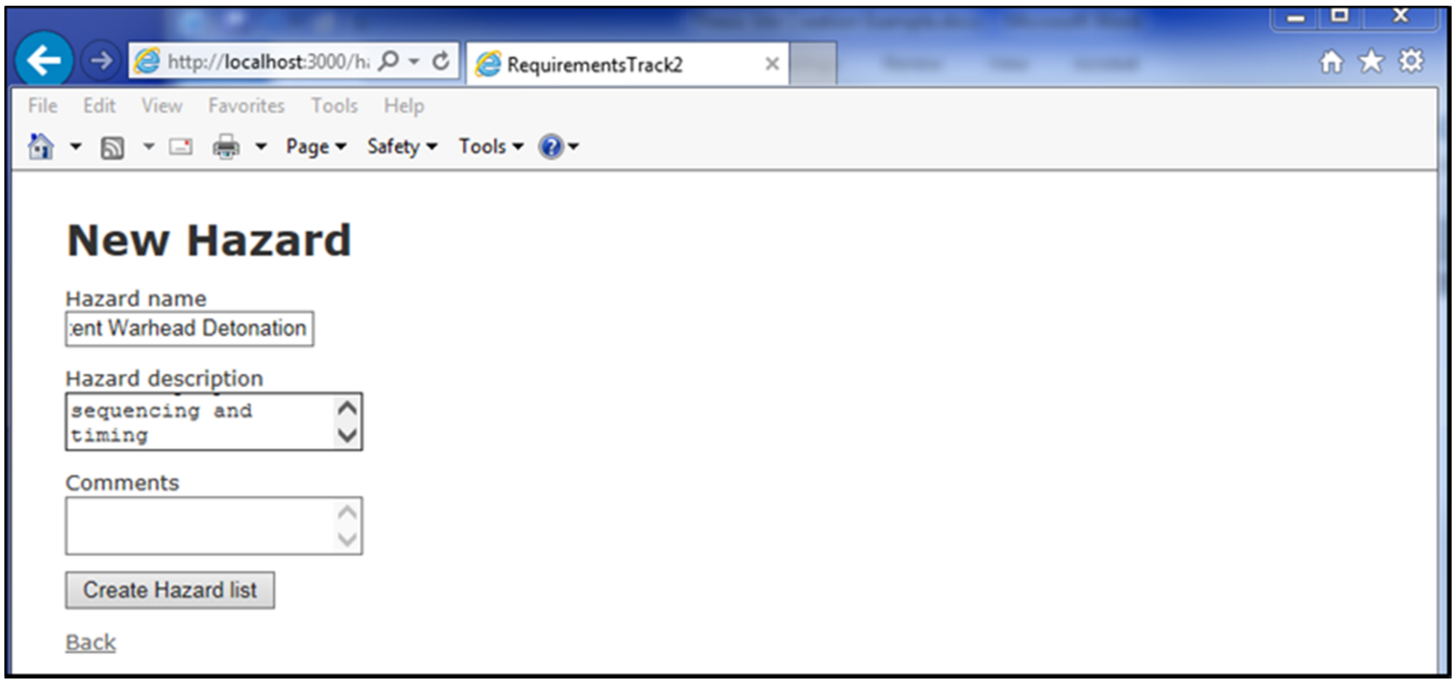
Analysis	Description
Preliminary Hazard List	The Preliminary Hazard List is a list of potential hazards identified early in the development cycle.
Preliminary Hazard Analysis	The Preliminary Hazard Analysis identifies hazards, allows for the assessment of the initial risks, and identification of potential risk mitigation efforts.
Functional Hazard Analysis	The Functional Hazard Analysis is where the decomposition of the system and/or subsystem into individual functions takes place. The functional description, failure modes and consequences of failure are all identified.
Full Level of Rigor Table	Enter all the possible LOR tasks.
Resources	Resources from the Joint Software Systems Safety Engineering Handbook and MIL-STD-882E are provided for convenience.
My Rigor	The Rigor for my program. The output of the FHA will be the RAC, which when used with JSSSEH Table 3-3, will determine the LOR.
About Requirements Tracker	About this website.

Preliminary Hazard List

- The Preliminary Hazard List is a list of potential hazards identified early in the development cycle.
- The user or users identify such hazards using the webpage.

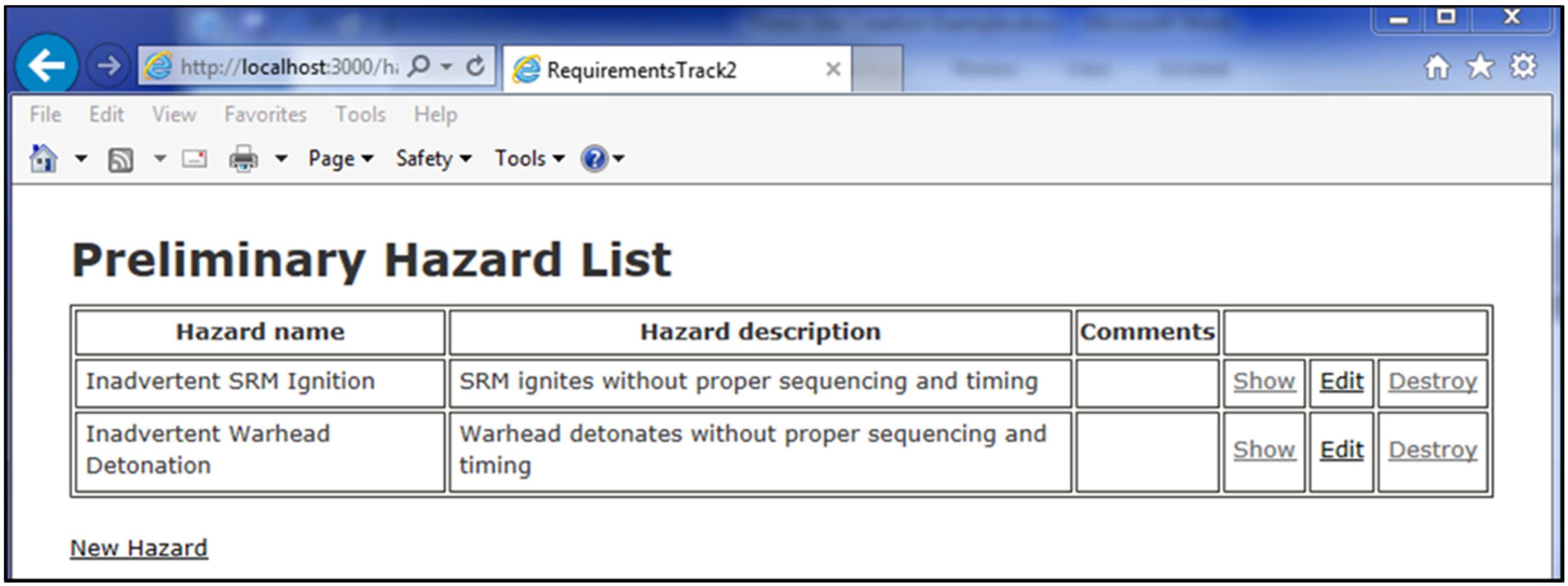


- Selecting the "New Hazard" link brings the user to a form that allows them to add a hazard to the list.



Preliminary Hazard List

- The Preliminary Hazard List has been updated with the new hazard.



Preliminary Hazard List

Hazard name	Hazard description	Comments			
Inadvertent SRM Ignition	SRM ignites without proper sequencing and timing		Show	Edit	Destroy
Inadvertent Warhead Detonation	Warhead detonates without proper sequencing and timing		Show	Edit	Destroy

New Hazard

Orbital ATK

- The Preliminary Hazard Analysis identifies hazards, allows for the assessment of the initial risks, and identification of potential risk mitigation efforts

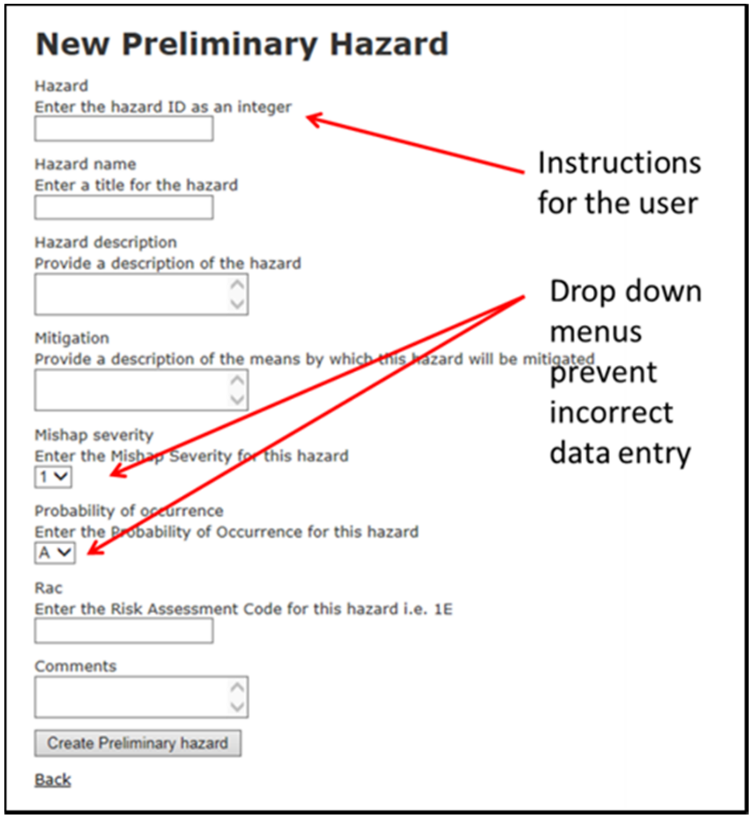
Preliminary Hazard Analysis

Hazard	Hazard name	Hazard description	Mitigation	Mishap severity	Probability of occurrence	RAC	Comments			
1	Inadvertent SRM Ignition	SRM ignites without proper sequencing and timing	Circuitry used to verify that only proper sequence will generate ARMING energy	1	E	1E		Show	Edit	Destroy
2	Inadvertent Warhead Detonation	Warhead detonates without proper sequencing and timing	Circuitry used to verify that only proper sequence will generate ARMING energy	1	E	1E		Show	Edit	Destroy

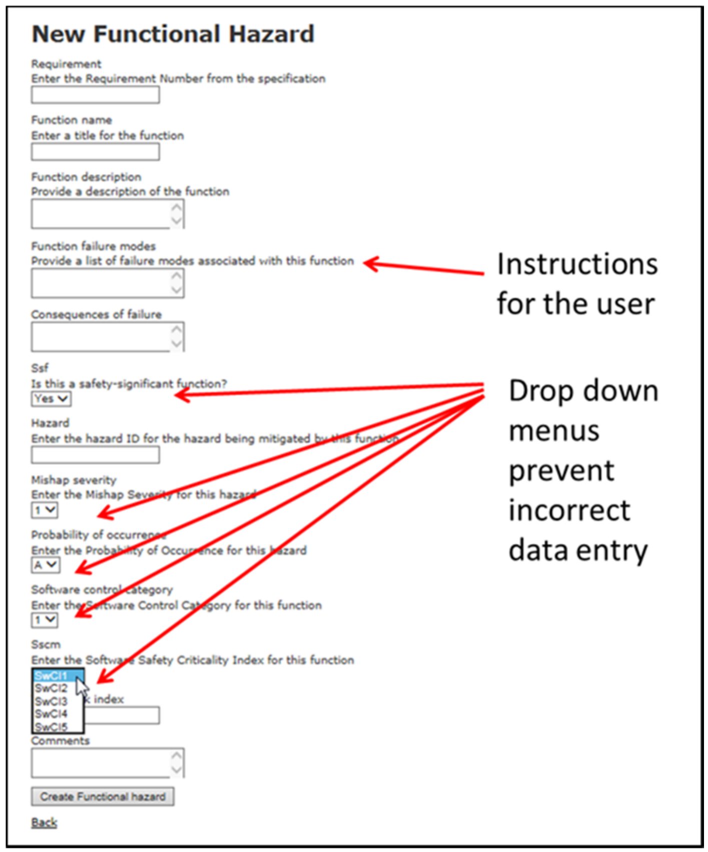
New Preliminary Hazard

Preliminary Hazard Analysis

- New Preliminary Hazards are entered into the software by using the "New Preliminary Hazard" button
- Instructions are provided to the user and drop down menus are used to improve the quality of the data

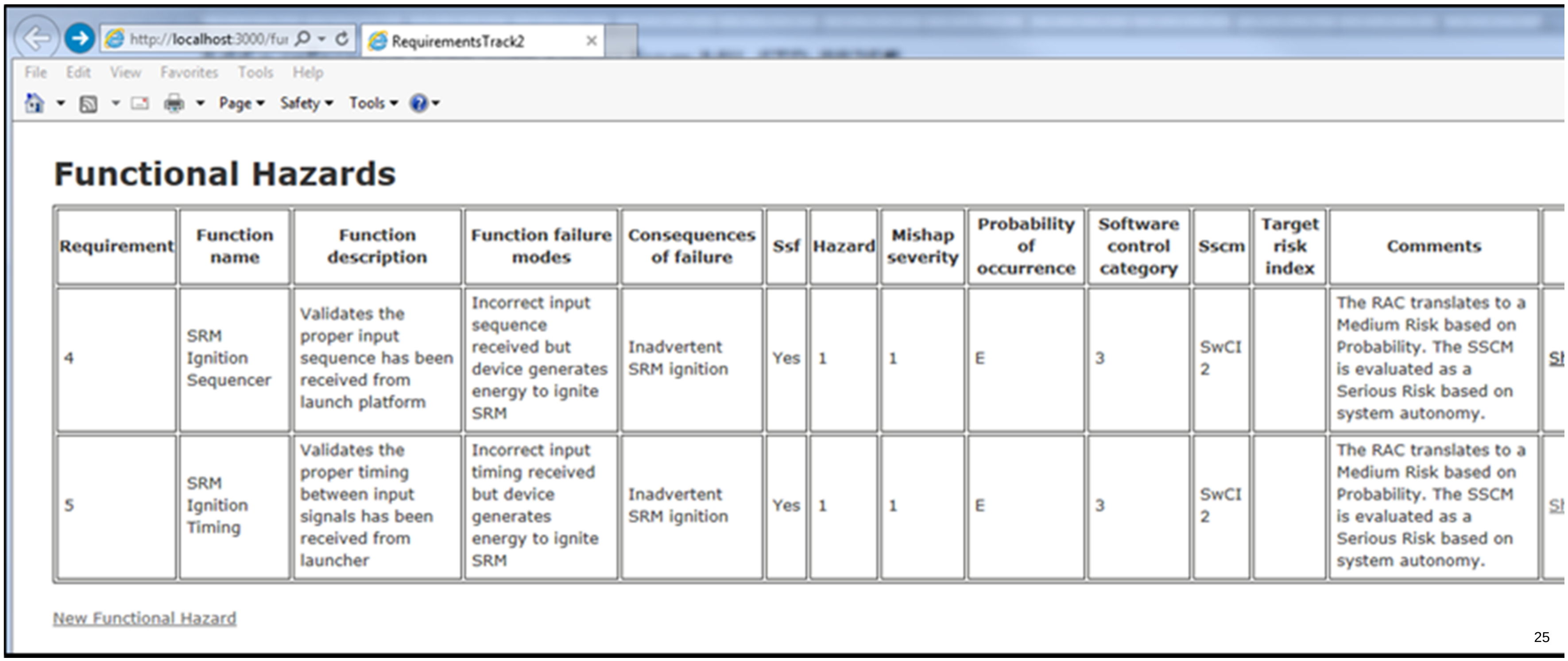


- The Functional Hazard Analysis is where the decomposition of the system and/or subsystem into individual functions occurs.
- The functional description, failure modes, and consequences-of-failure are all identified at this stage.



Functional Hazard Analysis

- The functions, which are a result of the system decomposition effort, may be associated with the hazards identified in previous analysis phases.
- Example: Both requirements 4 & 5 relate to the same hazard "Hazard 1, Inadvertent SRM Ignition."



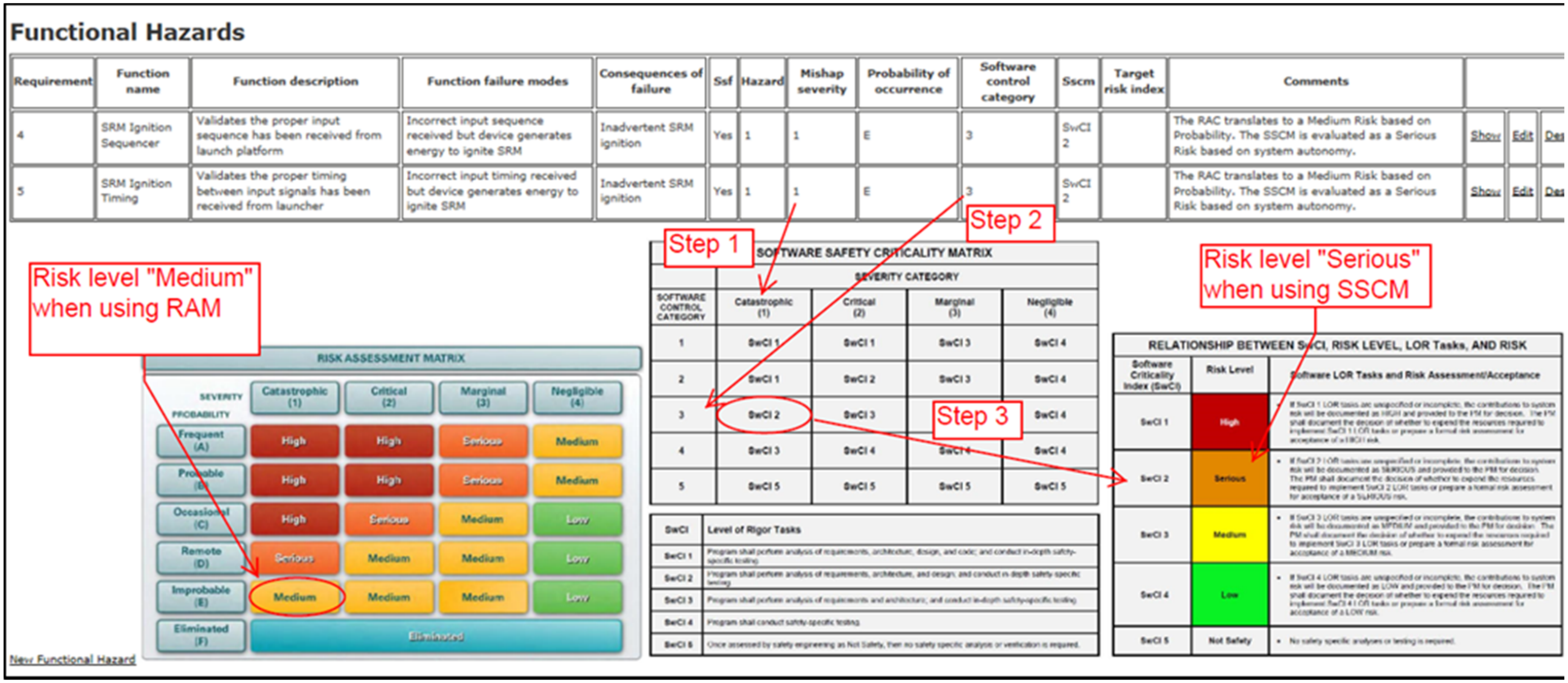
Functional Hazards

Requirement	Function name	Function description	Function failure modes	Consequences of failure	Ssf	Hazard	Mishap severity	Probability of occurrence	Software control category	Sscm	Target risk index	Comments	
4	SRM Ignition Sequencer	Validates the proper input sequence has been received from launch platform	Incorrect input sequence received but device generates energy to ignite SRM	Inadvertent SRM ignition	Yes	1	1	E	3	SwCI 2		The RAC translates to a Medium Risk based on Probability. The SSCM is evaluated as a Serious Risk based on system autonomy.	Sh
5	SRM Ignition Timing	Validates the proper timing between input signals has been received from launcher	Incorrect input timing received but device generates energy to ignite SRM	Inadvertent SRM ignition	Yes	1	1	E	3	SwCI 2		The RAC translates to a Medium Risk based on Probability. The SSCM is evaluated as a Serious Risk based on system autonomy.	Sh

New Functional Hazard

Level of Rigor Determination

- The output of the FHA will be the RAC, which when used with JSSSEH Table 3-3 and the Software Safety Criticality Matrix, will determine the Level of Rigor (LOR)



Full List of LOR Tasks

- The "My Rigor Tasks" table contains all the LOR tasks that must be accomplished as part of the System Software Safety Analysis for your program
- Automatically generated as a result of the worst case LOR
- A link is provided at the bottom of the "My Rigor Tasks" page for the purpose of adding new tasks.

My Rigor Tasks

Lor activity	Primary responsibility	Lor	Artifacts produced	Comments
Perform a Preliminary Hazard Analysis	Developer	Baseline	List of Hazards and Failure Modes PHA	
Perform a Functional Hazard Analysis	Developer	Baseline	Functional Hazard Analysis List of Safety Significant Functions	
Derive Requirements to ensure safety-significant interfaces are validated and controlled at all times	Developer	Serious	Interface Analysis	
Coordinated Safety-significant Requirements Review for correctness and completeness	Developer	Serious	Safety Requirements Review	
Perform a safety review of each test case	Developer	Medium	Safety Review Results	
Review all requirements traceability matrices for coverage and completeness	Developer	Medium	Requirements Traceability Review Results	

New My Rigor Task

Discussion

- The LOR task list was generated with the user requiring only a marginal familiarity with the safety specifications such as the JSSSEH, AOP-52, JOTP-51 or MIL-STD-882E.
- Database provides a location for the storage of artifacts
- Of course, the LOR task list will need to be checked and approved by the appropriate safety authority but a significant amount of work is generated for the user with very little effort.
- Collaboration among colleagues allows for greater safety related input to the program.

Conclusion

- The study of Software Quality Assurance techniques and its application towards the development of hardware provides a benefit to hardware developers who may now leverage decades of lessons learned from the study of safety critical software.
- The web based program developed as part of this academic pursuit provides a means by which developers can collaborate on the requirements, design and testing of safety critical software or “software-like” systems.

Questions?