ABSTRACT
Many software-intensive systems have significant safety ramifications and need to have their associated safety-related requirements properly engineered. However, there is little effective interaction and collaboration between the requirements and safety teams on most projects. This tutorial is intended to improve such collaboration by providing clear definitions of the different kinds of safety-related requirements, examples of such requirements, and a generic process for producing them.

Categories and Subject Descriptors
D.2.1 [Software Engineering]: Requirements/Specifications – Elicitation Methods and Methodologies.

General Terms
Documentation, Design, Reliability, Standardization.

1. INTRODUCTION
It has been observed by multiple consultants, researchers, and authors that inadequate requirements are a major cause of accidents involving software-intensive systems. Yet in practice, there is very little interaction between the requirements and safety disciplines and little collaboration between their respective communities. Most requirements engineers know little about safety engineering, and most safety engineers know little about requirements engineering. Also, safety engineering typically concentrates on architectures and designs rather than requirements engineering because hazard analysis typically depends on the identification of hardware and software components, the failure of which can cause accidents. This leads to safety-related requirements that are often ambiguous, incomplete, and even missing.

2. GOAL AND OBJECTIVES
The overall goal of this tutorial is to teach attendees how to engineer safety-related requirements for software-intensive systems. Specific objectives of this tutorial that support this goal include teaching:

- The basic concepts underlying safety engineering.
- The different types of safety-related requirements including their different purposes, form, and content.
- The relationship between safety quality subfactors and the quality criteria of safety requirements.
- Basic tasks of safety engineering that are related to the engineering of safety-related requirements.
- How to apply these concepts and processes to a realistic example application.

3. INTENDED AUDIENCE
The intended audience for this tutorial includes:

- Requirements engineers who must collaborate with safety engineers to engineer safety-related requirements.
- Safety engineers who must perform the hazard and risk analysis that drives the safety-related requirements and who must collaborate with requirements engineers to engineer these requirements.
- Stakeholders in safety-related requirements including subject matter experts, customer representatives, architects, software engineers, and testers.

Although the tutorial level is intermediate, it does not assume any prerequisites. However, basic familiarity with requirements engineering and safety engineering would be beneficial.

4. TUTORIAL CONTENTS
The tutorial covers the basic concepts of safety engineering relevant to requirements engineering, the different types of safety-related requirements, a basic process for engineering safety requirements, and an example application where these ideas have been applied.

4.1 Basic Concepts
In order to address the lack of understanding and collaboration between requirements engineers and safety engineers, one must begin by ensuring that requirements engineers understand the many basic concepts of safety engineering related to requirements engineering. In brief, safety engineering is concerned with protecting valuable assets from accidental harm. Thus, safety engineers are concerned with preventing, detecting, and reacting to safety incidents such as accidents and near misses (close calls). These safety incidents are made possible by the existence of hazards (collections of hazardous conditions). By addressing the potential severity of the resulting harm to valuable assets and the
likelihood of hazards/accidents causing harm, safety engineers identify safety risks. These risks are categorized into safety integrity levels (SILs) requiring developers to achieve associated safety assurance evidence levels (SEALs) to properly manage the risks. The result of addressing these safety concepts leads to the production of safety goals, safety policies, and safety-related requirements. Safety engineers meet these goals, policies, and requirements by means of safety mechanisms and safeguards, which eliminate or mitigate the existence of hazards and associated safety vulnerabilities.

Similarly, safety engineers need to understand the characteristics of well specified requirements so that relatively vague safety policies are not confused with verifiable safety-related requirements. They should also understand the basics of requirements elicitation, analysis, specification, and management.

4.2 Types of Safety-Related Requirements
When engineering safety-related requirements, stakeholders must realize that these requirements come in four distinct types, which need to be analyzed and specified differently. First of all, there are pure safety requirements, which are a kind of quality requirement that views safety as a quality factor within a quality model. As such, safety requirements are typically of the form of a quality criterion (a system-specific statement about the existence of a subfactor of safety) combined with a minimum or maximum required threshold along some quality measure. They directly specify how safe the system must be. Second are safety-significant requirements, which are normal functional, data, interface, and non-safety quality requirements that are relevant to the achievement of the safety requirements. In other words, safety-significant requirements can lead to hazards and accidents when not implemented correctly. When most people think of safety-critical systems, they are thinking of systems, the required functionality of which makes them subject to serious accidents. Third are system safety requirements, which are the requirements for safety systems or safety components of safety-related systems. A canonical example of which would be requirements for the emergency core cooling system of a nuclear power plant. Requirements for an aircraft’s fire detection and suppression system would also be safety system requirements. Finally, safety constraints are architecture or design constraints mandating the use of specific safety mechanism or safeguards. Many industries including petrochemicals, nuclear power, and automated people movers have industry safety standards requiring specific safeguards.

4.3 Requirements Engineering Process
Currently, typical processes for performing requirements engineering and safety engineering are discipline-specific, each with its own tasks, using its own techniques, and producing its own work products. For example, requirements engineering uses various techniques for the requirements elicitation, requirements analysis, requirements specification, and requirements management tasks, eventually resulting in a repository of requirements and associated requirements specification documents. Similarly, safety engineering often uses fault trees and event trees for hazard analysis to produce safety policies and safety compliance repositories to achieve safety certification.

A generic subset of safety engineering tasks can be combined with requirements engineering tasks to produce an integrated process for engineering the different kinds of safety-related requirements. Safety program planning produces safety goals to be included in requirements ConOps documents or vision statements. Safety engineers can then perform safety analysis consisting of asset analysis, safety incident analysis, hazard analysis, safety risk analysis, safety significance analysis, and safety control analysis. During these tasks, safety engineers can collaborate with requirements engineers to identify, analyze, and specify the four kinds of safety-related requirements, which are then stored in requirements repositories and documented in requirements specifications. The remaining requirements-related aspects of safety engineering are then performed during the remaining safety tasks including safety monitoring, safety incidence investigation, safety compliance assessment, and safety certification.

4.4 Example System
The tutorial concludes with an example of a safety-critical system. An automated people mover system is described, and the tutorial attendees are led through the development of several of the different kinds of safety-related requirements.

5. REFERENCES