# *Defining Requirements for and Designing Safety-Critical Software Intensive Systems*

## Course Overview

This course uses lecture and exercises to discuss the motivation, concepts, and key principles that address defining requirements for and designing safety-critical software intensive systems. The concept of software-system assurance cuts across the lifecycle phases. Software assurance must begin early starting at the system interfaces, and system and software architects must consider the risks of failures, hazards and threats systematically. Early analysis should continuously factor in verification and validation starting from planning, in addition to the development of requirement, design, implementation, and verification artifacts for each software level, including commercial off-the-shelf components. In many cases safety is also critical and architects may need to design the system to monitor, diagnose, and adapt to failures; many of these decisions result in derived requirements allocated to the software.

The design process must incorporate fault analysis and address the system as a complete entity, not just as the correct function of each component. Validation must ensure that the requirements are correct and complete to address all facets of functional and operational requirements for the system to carry out its mission; they also should include nonfunctional requirements associated with failsafe operation derived from fault and hazard analyses, as well as dependability requirements such as safety, security, reliability, and availability. Traceability of derived safety requirements and constraints should be provided bi-directionally to software design and implementation choices and the corresponding implementation-derived requirements that provide safety evidence, usually in form of verification evidence, to support the arguments for proper risk mitigation against potential system hazards.

The safety analyses identify hazards that must be mitigated through architecture, design, and requirements. Safety critical functions are assessed for normal, abnormal, and emergency conditions. Identified hazards are mitigated through the creation of a safe design and/or safety control mechanisms. Mitigations identified within the hazard analyses are implemented via requirements; sometimes derived requirements implemented by the system architects through software, firmware, and/or hardware. Therefore, system safety, like security threat management, must be designed into the overall system architecture. It is often ineffective to retrofit system safety or security into an existing system. Verification and validation efforts must ensure the operational

requirements of the system; in addition, they must ensure proper, nonintrusive operations of monitoring and diagnostic functions.

Software safety is driven by system safety. This course covers some system safety processes as the outputs from those safety-related activities provide inputs in the form of requirements and design constraints to the software processes. Traceability of the those system-derived safety requirements and constraints should be provided bi-directionally to software design and implementation choices, and the corresponding implementation derived requirements that provide safety evidence, usually in form of verification evidence, to support the arguments for proper risk mitigation against potential system hazards.

**The following provides additional details of the 3-day course:**

- Introduction, definitions and context
    - High assurance concepts and terminology
    - Lifecycle perspectives
- Inputs from System Analysis and Architecture to Software Requirements and Design
    - Preliminary Hazard Analysis
    - System Design Constraints
    - System Hazard Analysis
    - Safety Evidence
    - System Architecture Impacts on Software Assurance
- Software Requirements Analysis
    - Inputs and Outputs
    - Requirements and Implementation-Derived Requirements
    - Example Requirements Context
    - Software Safety Scenario
    - Software Safety Analysis
    - Software Hazard Analysis
    - FMEA for Software
    - Derived Safety Requirements
    - Software Failure Modes
- Software Architectural Design
    - Reuse Risk Assessment
    - Commercial-off-the-shelf (COTS) Software
    - Product Service History
    - Attributes of a Reusable Safe Component
    - Safety Critical Architectural Patterns
    - Design Principles for Safety
    - Design for Testability
    - Criticality Assessment by Element
    - Fault Tolerance
    - Evolving Architectural Trends
- Software Detailed Design
    - Design Methods
    - Derived Requirements

        - Functional Perspective
        - Object-Oriented Perspective
        - Design Principles for Software Safety
        - Detailed Design Test Planning
- Software Source Code Implementation
    - Coding Standards
    - Standards
    - Templates
    - Naming Conventions
    - Other Code Issues
- Related topics
    - Impact of requirement and design on software V&V
    - Requirement-driven modeling and automated V&V
    - Design-based modeling and automated V&V
    - COTS, Open Source, and Reuse

## Benefits:

In completing this course, attendees should be able to:

- Understand context of software requirement and design for safety-critical systems
- Describe the needed inputs from system analysis and architecture for software requirements and design
- Describe how to specifying requirements for safety-critical software
- Understand architectural aspects of design to satisfy safety requirements
- Describe design features for fault detection, isolation and recovery
- Understand detailed design techniques and principles, and understand areas of caution such as object-oriented design and implementation techniques
- Describe safe coding practices
- Describe requirement and design concept that support constructive V&V
- Describe some benefits of model-based approaches for specifying requirements and design that support automation of V&V efforts

## Contact Us about Training:

Maureen Murtha
E-mail: maureen@knowledgebytes.net
Phone: 703.999.2734