SYSTEM SAFETY SOCIETY
Organized 1962
Incorporated 1973
Professionals Dedicated to the Safety of Systems, Products & Services
System Safety Society, Singapore Chapter
http://www.systemsafety.org.sg/

Software Safety Engineering
17th to 21st September 2012, 8.30am – 5.00pm

Course Outline

Session Number	5-Day Software Safety Engineering Course DESCRIPTION	Schedule & Time
	DAY ONE	
1	**Lecture: General Introduction** Introduction of Trainer, Introduction of Students, Training Format and Expectations; Desired Learning Objectives; Introduction to Common Terminology; Introduction to Mishaps, Hazards, Failure Modes, Causal Factors, and Failure Pathway Events; Introduction to How Software Contributes to Failure and Safety Risk.	0830-0930 1.0 hours
2	**Lecture: System Safety Standards and Compliance** Existing Compliance Standards and Common Best Practice; Evaluating the Contractual Requirements; Common Elements of All Standards; Prioritizing the Breadth and Depth of the System Safety and the Software Safety Engineering Program, Introduction to Draft Mil-Std 882E; STANAG 4404; MOD 00-55 and 00-56 Motivations of the Stakeholders and the Contractors.	0930-1000 0.5 hours
Break		1000-1015 0.25 hours
2 continued	**Lecture: System Safety Standards and Compliance (continued)**	1015-1200 1.75 hours
Lunch		1200-1300 1.0 hours
3	**Lecture: Functional Hazard Analysis** Purpose of Functional Hazard Analysis; SAE ARP 4761 Criteria; Safety-Critical Functions Identification; Physical and Functional Decomposition of the System. Identification of Safety-Critical or Safety-Significant Functions; Mapping/Allocating Safety-Related (critical and significant) Functions to the Software Design Architecture; Assessing Safety-Related Functions Against Software Control Categories; Assignment of Software Criticality Indices; Assignment of Level of Rigor for Each Software Safety-Related Functions.	1300-1430 1.5 hours
	Discussion: Introduction to Course Exercise Introduction to the "System" to be analyzed in the course group exercise activity. Discussion of the purpose and the objectives of performing the analysis required as defined by the exercise objectives. Introduction to the group concept of system analysis. Introduction to the first task in performing the FHA analysis for the course exercise.	1430-1500 0.5 hours
Break		1500-1515 0.25 hours
	Exercise One: Functional Hazard Analysis Group exercise to begin the Functional Hazard Analysis on the target "System"; Functional and physical decomposition of the system; Determining the functions of the system that require system safety and software safety assessment.	1515-1700 1.0 hours

Session Number	5-Day Software Safety Engineering Course DESCRIPTION	Schedule & Time
	DAY TWO	
	Discussion – Day One Review, FHA Exercise One Review	0830-0900 0.5 hours
4	**Lecture: Designing Safety Into the System/Software** Safety Order of Precedence; Design objectives and Goals; Introduction to the software development life cycle models; Determining when and how to integrate software safety into the system and software development life cycle model being used on the program. Definition of software development phases; Introduction to software safety requirements.	0900-1000 1.0 hours
Break		1000-1015 0.25 hours
5	**Lecture: Software Safety Assurance/Integrity in the Software Development and Test Process** Introduction to the two separate and complementary processes of software safety engineering; Functional Safety Analysis; Defining the Level of Rigor Tasks; Process-Related Tasks; Design-Related Tasks; Test Related Tasks; Verification Related Tasks; Assignment of Level of Rigor Tasks to Safety Engineering, Software Developers/Coders, Software Testers, and Software Quality Assurance.	1015-1200 1.75 hours
Lunch		1200-1300 1.0 hours
6	**Lecture: Performing the Hazard Analyses** An introduction to the common hazard analyses required on development programs; Functional Hazard Analysis (FHA); Preliminary Hazard Analysis (PHA); Safety Requirements Criteria Analysis (SRCA); System Hazard Analysis (SHA); Software FMECA (SFMECA); Fault Tree Analysis (FTA to include the software contribution to failure); Software Safety Code-Level Analyses; Safety-Critical Functional Modeling (IMPACT Analysis); Safety Risk Assessment (SAR).	1300-1500 2.0 hours
	Exercise Two – Functional Hazard Assessment Objective: Finalize the functional and physical decomposition of the system; Document the function(s) of each physical subsystem and major component and identify the consequence of loss of function or malfunction. Identify Safety-Critical Functions and Safety-Significant Functions.	1500-1700 2.0 hours
	DAY THREE	
	Discussion – Day Two Review, FHA Exercise Two Review	0830-0900 0.5 hours
7	**Lecture: Mishap/Hazard Elimination, Mitigation, and/or Control** Safety Requirements for Software Development and Test Processes and Tasks; System/Software Design Features (Kernels, Firewalls, Barriers, Functional Partitioning, Physical Partitioning; Lock-Outs/Lock-Ins, Baton Passing, Interlocks, BIT Checks, Parity Checks); Fault Management (Fault Detection, Fault Annunciation, Fault Isolation, Fault Tolerance, Fault Recovery); Fail Safe, Fail Operational, Fail Passive, Fail Active; Fail Catastrophic?; Language Specific Software Safety Issues.	0900-1000 1.0 hours
Break		1000-1015 0.25 hours
8	**Lecture: Software System Safety Processes** Software System Safety Summit Results – The Summit Weaved Software Safety Process; The Apogen Technologies Defined and Documented 20-Step Software Safety Process to include (a) the Software Assurance and Integrity Process, and (b) the Software Safety Hazard Analysis Process; The Differences Between Software System Safety and Software Reliability.	1015-1200 1.75 hours
Lunch		1200-1300 1.0 hours
8	continued **Lecture: The System Safety Process (continued)**	1300-1500 2.0 hours

Session Number	5-Day Software Safety Engineering Course DESCRIPTION	Schedule & Time
Break		1515-1530 .25 hours
	Exercise Three – Hazard Analysis Objective: Identify the mishaps and the hazards associated with the SPTS.	1530-1700 1.5hours
	DAY FOUR	
	Discussion – Day Three review, Exercise Three review	0830-0900 0.5 hours
9	**Lecture: Requirements/Criteria Analysis** Introduction to the Mil-Std 882 objectives of performing a SR/CA. Introduction to preliminary safety requirements; contributing safety requirements, and mitigating safety requirements. The importance of safety requirements in the system specifications including the software specification. Introduction to fault detection, isolation, annunciation, tolerance, and recovery. Defining software-specific safety requirements; Best practices and Generic software safety lists.	0900-1000 1.0 hours
Break		1000-1015 0.25 hours
	Exercise Four – Defining the Software Safety Requirements for the System Objective: Define "best practice" software safety requirements for the system; Define contributing software safety requirements for the system; Define the hazard mitigation software safety requirements of the system.	1015-1200 1.75 hours
Lunch		1200-1300 1.0 hours
	Discussion: Exercise four results; Meet the "Murder Board" and the rationale, justification, and defense of the software safety requirements for the system.	1300-1400 1.0 hours
10	**Lecture: Software Safety Testing** Test Planning and Purpose; Models and Simulations; Generic Safety Tests; Functional Safety Tests; Testing Techniques; Formal Testing; Incremental Testing; Safety Testing Exit Criteria; Collecting the Evidence of Hazard Elimination, Mitigation, and/or Control.	1400-1500 1.0 hours
Break		1500-1515 0.25 hours
10 continued	**Lecture: Software Safety Testing (continued)**	1515-1600 0.75 hours
	Exercise Five – FHA Level-or-Rigor Assignments Objective: Prepare a "Tailored" Level-of-Rigor Table for the system. Assign LOR to each of the functions that are deemed safety-critical or safety-significant.	1600-1700 1.0 hours
	DAY FIVE	
	Discussion – Day four review, Exercise five review.	0830-0900 0.5 hours
11	**The Safety Case – Meeting/Exceeding the Criteria of the Contract** Assessing the Contractual Requirements and Specific Product Delivery Requirements; Software Safety Engineering Checklist for Contractual Compliance; Collecting the Evidences of Software Safety Contractual Performance; Integrating the Software Safety Evidences into The Safety Case Deliverable; Preparing for a Software System Safety Technical Review Panel (SSSTRP).	0900-1000 1.0 hours
Break		1000-1015 0.25 hours
12	**Lecture: Software Safety Lessons Learned** Documented Lessons Learned from A Variety of Sources – Integrating Lessons Learned into the Software System Safety Processes of the Stakeholder and Contractor (the Program).	1015-1200 1.75 hours

Session Number	5-Day Software Safety Engineering Course DESCRIPTION	Schedule & Time
Lunch		1200-1300 1.0 hours
	Discussion: Special Topics Based upon the Desires, Needs, Expectations, and Requirements of the Customer. Examples Include, but are not limited to, Software Safety Requirements for Unmanned Systems; Battlefield Environments; Software Safety in the OSSE Process; Software Trouble Reports and Managing the Requirements of System and Software Changes (functional change, physical change, mission change, performance change) from a Safety Perspective.	1300-1400 1.0 hours
	Discussion: Final Discussion and Wrap-up Objective: Review of each exercise, the analyses produced the methods or techniques used, and the perceived return on investment. Discussion of how to meet or exceed customer expectations and assessment of residual safety risk.	1400-1500 1.0 hours
Break		1500-1515 0.15 hours
	Course Critique Preparation: Objective: Allow students to provide constructive feedback to the instructor and the University of Cincinnati pertaining to the course content, issues or improvements, and methods of teaching employed by the instructor.	1515-1600 1.0 hours
	Singapore Chapter Closing Instructor remains to provide one-on-one discussions	1600-1700 1.0 hours
Exercise	The "running" exercise that begins on Day One and goes through to Day Five is the heart and soul of the course. It provides the student with the practical application and real-time experience of performing analysis on a system. The exercise will consist of a defined "system" that will require the following analyses to be accomplished during the group exercise time. The performed analysis will include…but, not be limited to: • Functional Hazard Analysis • Identification of Safety Critical Functions • Allocation of Safety Critical Functions to Hardware, Software, or Human Interfaces • Preliminary Hazard Analysis • Definition of Preliminary (Generic) Safety Requirements • Fault Tree Analysis • Definition of Derived Safety Requirements • Safety Risk Assessment	

Who Should Attend

- Program managers responsible for the overall safety of the system from design, to fabrication, test, operations, maintenance, and disposal.
- System safety managers responsible for the establishment, management, and implementation of a system safety engineering program through any or all phases of the system acquisition life cycle.
- System safety engineers responsible for performing the analyses required for a system safety program.
- Individuals responsible for, or interested in, product loss prevention and the reduction of product liability risk.

Course Details

Course Venue: (Location in Singapore to be determined)

Course Fee	
System Safety Society or Singapore Chapter Affiliate / Member	**Others**
S$ 2,050	**S$ 2,150**

Certification:
A course certificate from the University of Cincinnati will be issued to the course participants on condition that their attendance exceeds 75%. The course participants will also earn a total of 4.0 Continuing Education Units (CEU's) if they attended the full 40 hours of the course.

For more information, please contact:

- Course Administrator via email: ***admin@systemsafety.org.sg***

Software Safety Engineering
17th to 21st September 2012

REGISTRATION FORM		
Organisation Name:		
Address:		
Contact Person (for payment):	Mr / Ms.	
Fax:	Telephone:	
Email:		
Participants Details		
Name: Mr / Ms.		Contact No.
Designation:		
Email:		
Name: Mr / Ms.		Contact No.
Designation:		
Email:		
Name: Mr / Ms.		Contact No.
Designation:		
Email:		
Name: Mr / Ms.		Contact No.
Designation:		
Email:		
Name: Mr / Ms.		Contact No.
Designation:		
Email:		
Name: Mr / Ms.		Contact No.
Designation:		
Email:		
Name: Mr / Ms.		Contact No.
Designation:		
Email:		
Name: Mr / Ms.		Contact No.
Designation:		
Email:		
Name: Mr / Ms.		Contact No.
Designation:		
Email:		
Name: Mr / Ms.		Contact No.
Designation:		
Email:		

* Please delete wherever appropriate.

ADMINSTRATIVE DETAILS

Registration should be done via **email only**. Please fill in the registration form, scan and email to ***admin@systemsafety.org.sg***.

For payment, please indicate the **invoice number**, **organisation name** and **contact person** at the back of the cheque. Cheques should be **made payable to "System Safety Society (Singapore Chapter)"** and mail to:

System Safety Society (Singapore Chapter)
(c/o Management Systems & Processes)
Singapore Technologies Kinetics Ltd
249 Jalan Boon Lay
Singapore 619523

Confirmation will be sent to the participants within 3 days before the commencement of the course. No refunds will be granted for cancellation; however a replacement is acceptable. The System Safety Society (Singapore Chapter) reserves the right to cancel, modify or postpone the course.

Bastion Technologies
Biography

Steven F. Mattern, Senior Software Safety Engineer

Mr. Mattern has accumulated a broad range of government, commercial and education-related experience in acquisition management and systems engineering on major system procurements. He possesses over 29 years of USAF active-duty, DOD, and private sector experience on the development and test of high-profile, safety-critical systems of national importance.

Mr. Mattern began his military career with the United States Air Force in September 1971 as an Airman, specializing in the installation, operations, and maintenance of HVAC equipment within military base facilities. He was honorably discharged in 1979 to attend the University of Wyoming through the USAF ROTC program where he was commissioned a Second Lieutenant 1981. Throughout his commissioned career in the USAF, Steven chose to work as an Acquisition Specialist in the System Safety Engineering career field. On active duty, he was assigned to work on many major acquisition programs to include, but not limited to, the Peacekeeper (Missile-X) intercontinental ballistic missile (ICBM) program, Small ICBM, F-15E, C-17, X-30 National Aerospace Plane, and Air Force One Replacement for the President of the United States. He also was the Explosive Safety Officer that performed the Quantity Distance (QD) explosive testing for the basing of Peacekeeper missiles in Minuteman silos. During Steven's commissioned career he was certified as a Level III Acquisition Professional in Program Management, and Systems Planning, Research, Development, and Engineering of the Acquisition Professional Development Program (APDP).

In September 1994, Steven joined Science and Engineering Associates (later to become Apogen Technologies and then QinetiQ-North America) where he was specifically tasked to stand up and manage a systems engineering analysis directorate for the company that focused on both systems and software safety engineering. Since 1994, Steven has secured and successfully executed contracts with government agencies, government prime contractors, and commercial companies on programs such as the Airborne Laser (ABL) Program, Crusader, Future Combat Systems, Ballistic Missile Defense, SH-2G(A) Helicopter, Standard Missiles 2-, -3, and -6, Wide Area Augmentation System, Shuttle Orbiter Cockpit Avionics Upgrade, SWY-3 Ship Self-Defense System, C-5 Avionics Modernization Program, and many others.

Mr. Mattern joined Bastion Technologies in June of 2010 as a subject matter expert (SME) and currently provides system safety and software safety engineering support to the US Army Aviation and Missile Command (AMCOM) Safety Office on unmanned air vehicle programs.

Steven holds a BS in Industrial/Electronic Technology from the University of Wyoming and a MA in Computer Resource Management from Webster University. He teaches two Continuing Engineering Education courses for the University of Cincinnati; System Safety Management and Software Safety Engineering. He is member of the Armed Forces Communications and Electronics Association, a Fellow Member of the international System Safety Society, and a Lifetime Member of the Veterans of Foreign Wars.