Software Safety Course

Outline

Introduction
- Software Safety
  - Overview, Benefits
- Myths
- Software-Caused Accidents
  - Examples
  - Lessons Learned – First Hand
- Safety Loopholes
- Ergonomic Factors
  - Their Nature
  - Why Haven’t We Seen More?
  - Their Cause
  - Source of Errors in Systems
    - Complexity Issues
    - Simplicity, Determinism
  - Personnel
    - Independence
    - Why V & V Fails
  - Minimizing Them in Already Commissioned Systems
- Software Safety Incentives
  - Accidents - Devastating Effects
  - Software Liability
  - Software Engineering Malpractice?
- Safety And Reliability Concepts
  - Definitions
  - Dependability Concepts
  - Safety Integrity Levels
    - Common Mistake
    - Systematic & Random Failure Integrity
    - Software SILs
  - Robustness
  - System/Software
    - Designing In Safety
    - Validating Safety
      - Can We Always Validate Safety?
      - How Can We Validate Safety:
        - When Our System Contains COTS Elements?
        - When Little or No Documentation Exists?
        - When We Are Given Only the Software?
  - Expected Probability of Failure of Systems
- Risk Concepts
  - Risk Engineering
  - Socioeconomic Factors
  - Definitions
  - Severities & Probabilities
    - Defined By Standards
  - System Risk Assessment
  - Risk Assessment Matrix/RACs
  - Risk Classes
  - Safety Integrity Level (SIL) Determination
    - System, Software
  - Reducing Software Integrity Levels
  - Software Criticality Assessment
  - Software Control Categories (SCCs)
  - Software Criticality Indexes (SwCIs)
  - Same As Software Integrity Levels?
  - Software Safety Criticality Matrix (SSCM)
  - Software Development Assurance Levels (SDALs)
    - With Respect to RTCA DO-178
      - Same As Software Integrity Levels?
      - Same As Software Criticality Indexes?
    - Software Assurance Levels (SWALs)
  - Determination
  - Basic Approaches to Safe Design
  - Software Safety Stds., Guidelines & Regulations
  - Defense
    - Joint Services Software Safety Engineering Handbook
    - MIL-STD-882E(System Safety)
    - Relevance to Software Safety
    - AMCOM 385-17
    - AOP-52
    - STANAG 4404
  - Aerospace
    - NASA Software Safety Standard
    - NASA Guidebook
    - FAA System Safety Handbook
    - SAE ARP4754A/4761
    - Relevance to Software Safety
    - RTCA DO-178
    - Relevance to Software Safety
    - ESARR 3, ESARR 4, ESARR 6
    - ED-153
  - Rail
    - EN 50128
    - IEEE 1483
  - General
    - IEEE 1228 (Software Safety Plans)
    - IEC 61508
    - ISO/IEC 1526
    - System & Software Integrity Levels
    - UL 1998 (Safety-Related Software)
    - MISRA Guidelines
  - Formal Methods
    - Introduction
    - Study of Industrial Experience
      - Program Function Table Analysis
    - Relevance
    - Formalism
  - Fault Tolerant Techniques
    - N Version Programming, Recovery Blocks
    - Other Techniques
    - Data Redundancy
Safe Design Techniques
  Security Kernels, Safety Kernels, Firewalls
  Barriers
    Lockins, Lockouts - Baton Passing
    Interlocks - Types, Precautions
  Checks
    Hardware, Assertions
    Audit, Supervisory
  Fail Safe, Fail Soft
    Fail Operational, Passive, Active
    Recovery Techniques
Safety Assurance Concepts
  Software Assertions
  Many Others
Software Requirements Checklist
Software Design Checklist
Programming Languages
  Importance?
  Language Subsets
  Reality?
System Safety Programs (SSP)
  Objectives
  General Requirements
  Tailoring
  Flow-Down of Safety Requirements
  Safety Integration
  Safety Requirements Traceability
  Tools
  Design/Implementation/Testing Influence
  Chronology
Safety Program Results
System Safety Program Plans (SSPP)
  Dangers Lurking
  Guidelines
Software Safety Program Plans (SwSPP)
  Guidelines
Software Safety Working Group (SwSWG)
Hazard Mitigation Precedence
Hazard Tracking
Preliminary Hazard Analysis (PHA)
  Objectives
  System Boundary
  Analyst Credentials
  Format
    Life-Cycle, Post-Design
    Guidelines - Keys To Success
    In-Class Assignment
Functional Hazard Analysis (FHA)
  Determining/Lowering Software Criticality
  Degree Of Rigor In Software Development
  Subsystem Hazard Analysis (SSHA)
System Hazard Analysis (SHA)
Software Safety Analysis Process
  Software Requirements Analysis
    Types of Analysis
  Software Design Analysis
    Types of Analysis
  Software Code Analysis
    Types of Analysis
  Software Change Analysis
Tools
  Static Code Analyzers
  Many Others
Software FMEA
  Types
  Examples
  Guidelines
Software FMECA?
Fault Tree Analysis (FTA)
  History
    Qualitative/Quantitative Human Failure Rate Derivation
    Versus FMEA/FMECA
    Advantages/Disadvantages
    Fault Tree Symbols and Terminology
    Definitions, Special Symbols
    Examples
    Software FTA
    Software Failure Rate Derivation
    Immediate, Necessary and Sufficient Concept
    Basic Rules
    System Operational Modes
    Guidelines - Keys to Success
    Increased Accuracy, Consistency, Economy
    Best Kept Secrets?
    Maintainability
    Fault Tree Notes
    Step Size Precautions
    Similar Subtrees
    Improving Fault Tree Size, Sharing Subtrees
    Limiting Fault Tree Production
    Class Exercise
    Other Analysis Techniques
Software Sneak Analysis (SSA)?
Petri Nets
Other Techniques
Software Safety Cases
Dealing with COTS Elements
RTOS's
  VxWorks, Integrity, LynxOS
  OSE, QNX, Linux
  Windows?
  And more
Safety Verification
Testing
Now, Let Us Step Back
What Is Really Do-able?
Avoiding The Monetary Sink Hole

HCRQ, Inc.
7151 Richmond Road, Suite 201
Williamsburg, VA 23188
Tel: (757) 564-7703
Fax: (757) 564-7704
web: http://www.hcrq.com/Training.html
e-mail: training@hcrq.com