

# Software Reliability and Safety

## CSE 8317 — Spring 2005

Prof. Jeff Tian, [tian@engr.smu.edu](mailto:tian@engr.smu.edu)  
CSE, SMU, Dallas, TX 75275  
(214) 768-2861; Fax: (214) 768-3085  
[www.engr.smu.edu/~tian/class/8317.05s](http://www.engr.smu.edu/~tian/class/8317.05s)

### **SSE.2: Hazard Analysis & Resolution**

- Hazard Analyses and Techniques
- Hazard Analysis Techniques:
  - FTA and ETA
- Hazard Resolution and Damage Reduction

---

## Safety Analysis

---

- Hazard and risk identification:
  - ▷ Accident scenarios: actual/hypothetical
    - starting points for safety
  - ▷ Focus: operations and operational env.
  
- Hazard analysis and assessment:
  - ▷ Fault trees: (static) logical conditions
  - ▷ Event trees: dynamic sequences
  - ▷ Other analyses/assessment techniques
  
- Hazard and risk resolution
  - ▷ Hazard elimination
  - ▷ Hazard reduction
  - ▷ Hazard control
  - ▷ Damage control

---

## Hazard Analyses: Types

---

- Sub-system hazard analyses (SSHA)
  - ▷ Hazard within individual sub-system
  - ▷ Component/sub-system in isolation
  
- System hazard analyses (SHA)
  - ▷ Focus: interface and interaction
  - ▷ Subsys/env/human effect on system
  - ▷ Throughout development process
  - ▷ Focus on early phases to provide info. for other activities (hazard resolution and safety verification)
  
- SHA/SSHA in software process
  - ▷ Throughout development process
  - ▷ Focus on early phases to provide info. for other activities (hazard resolution and safety verification)

---

## Hazard Analyses: Techniques

---

- Primary techniques for SHA/SSHA:
  - ▷ Fault-tree analyses (FTA)
  - ▷ Event-tree analyses (ETA)
  - ▷ SQE Ch.16 (slides!) and Safeware Ch.14.
  
- Other techniques:
  - ▷ Design reviews & checklists
  - ▷ Hazard indices
  - ▷ Risk trees
  - ▷ Cause-consequence analysis (CCA)
  - ▷ Hazard & operability analysis (HAZOP)
  - ▷ Failure modes and effect analysis, etc.
  - ▷ Above: “Safeware” Ch.14.
  - ▷ Specific to software: “Safeware” Ch.15.
  
- FTA and ETA slides from SQE Ch.16 next.

---

## Hazard Analysis: SFTA

---

- SFTA: Software FTA
    - ▷ Same concept applied to software
    - ▷ Actual implementation (white-box)
    - ▷ Language elements (high-level):
      - assignment and function calls
      - branching statement, loops, etc.
    - ▷ Also for specification/architecture
      - black-box control flow diagram
      - equivalent language representation
  
  - SFTA construction:
    - ▷ Templates/examples for diff. statements
    - ▷ Safeware 18.2.2 (pp.497-507)
- ⇒ Additional work needed,  
especially for system design/architecture

---

## Hazard Analysis: ETA & CCA

---

- ETA alone: trace of accident.  
May desire explanation also (from FTA)
  
- Cause-consequence diagram (CCA):
  - ▷ Combine ETA with FTA
  - ▷ Explaining decisions in ET
  
- Using ETA and CCA:
  - ▷ Partial vs. total ETA
  - ▷ Focus on main consequences
  - ▷ Details:  
    “Safeware” 14.5-14.6 (pp.327-pp.335)

---

## Hazard and Risk Resolution

---

- Generic hazard resolution techniques (in order of their precedence):
  - ▷ Hazard elimination (pre-failure):
    - eliminate hazard sources
  - ▷ Hazard reduction (pre-failure):
    - reduce hazard likelihood
  - ▷ Hazard control (post-failure):
    - control hazard severity/scope (accident↓)
  
- Related issues:
  - ▷ Basis: hazard identification and analysis via FTA, ETA, CCA, etc.
  - ▷ Many specific techniques
  - ▷ Related to QA and SRE
  - ▷ Risk resolution: damage reduction ( + hazard resolution above)

---

## Hazard Elimination

---

- Elimination of hazard
  - ▷ Intrinsically safe (sub-)system
  - ▷ All eliminated: feasibility & cost?
  - ▷ Certain types of hazard eliminated
  - ▷ Direct application of hazard identification and analysis results.
  
- Specific techniques: “Good SE & SSE”
  - ▷ Component substitution ( $\Leftarrow$  FTA)
  - ▷ No single point of failure ( $\Leftarrow$  ETA)
  - ▷ Simplification of building blocks
  - ▷ Decoupling of system architecture
  - ▷ Human errors/hazardous material elim.
  - ▷ Component safety certification:
    - formal verification
    - components identified by FTA etc.
  - ▷ Link to testing/FT/QA activities



---

## Hazard Reduction

---

- Hazard reduction:
  - ▷ Severity reduction:
    - change failure characteristics
    - various locks/barriers
  - ▷ Likelihood reduction:
    - reduce failure probability
    - in combination with above
    - also: most QA/SRE related techniques
  
- Specific techniques:
  - ▷ Control devices (active)
  - ▷ Barriers and locks (passive)
  - ▷ Failure/hazard probability/severity ↓  
(accident probability↓)

---

## Hazard Reduction: Techniques

---

- Control devices (active):
  - ▷ Control: fail  $\Rightarrow$  action  $\Rightarrow$  safe
  - ▷ Observability: observation of failure
  - ▷ Controllability:
    - system design/structure limit
    - energy/capacity limit
  - ▷ E.g., auto pressure relief
  - ▷ Partial control solution:
    - reduce the severity
    - bring to a neighboring state
  
- Locks and barriers (passive)
  - ▷ Lock-outs (preventing hazard)
  - ▷ Lock-ins (maintaining safety conditions)
  - ▷ Interlocks (correct order/combinations)
  - ▷ Other barriers (extra capacity, etc.)

## Hazard Reduction

---

- Hazard probability minimization:
  - ▷ Design with extra capacity:
    - safety factors/margins example
    - melt temp.  $T_m$  and margin  $M$
    - $\Rightarrow$  safety bound  $T_s = T_m - M$
  - ▷ QA and SRE: failure  $\downarrow$ 
    - focused hazard probability min.
    - with FTA/ETA/etc. help
  
- Redundancy (FT etc.)  $\Rightarrow$  prob(hazard)  $\downarrow$ :
  - ▷ Hardware redundancy/backup
  - ▷ Software redundancy:
    - fault tolerance (NVP, & (?) RB)
    - anticipated input/env. enlargement
    - “fool-proof” software
  - ▷ Hardware/software interlocks

## Hazard Resolution: Hw/Sw Interlock

---

- Interlock software
  - ▷ Software used as safety interlock
    - (s/w usage: data/control/safety)
    - example: emergency shut-down s/w
  - ▷ More stringent safety requirement:
    - most s/w function safety-related
    - should not rely solely on s/w
    - Therac-25 accident lessons
  
- Hardware/software interlock
  - ▷ Limitation of s/w backups:
    - diversity and independence problems
  - ▷ Hardware backups and interlocks:
    - different characteristics
    - different failure mechanisms
    - more likely to be *independent*
    - passive/active safety devices
  - ▷ Combine the advantages  $\Rightarrow$  safety  $\uparrow$

---

## Hazard Control

---

- Hazard control:
  - ▷ Limit hazard scope
  - ▷ Redesigning/re-structuring system
  - ▷ System augmentation
  
- Internal system change:
  - ▷ Hazard exposure reduction
  - ▷ Isolation of hazard event
  - ▷ Containment around hazard event
  - ▷ Fail-safe design (passive)
  
- System augmentation:
  - ▷ Protection system (PS) added on:
    - failure  $\Rightarrow$  PS action  $\Rightarrow$  shut-down (safe)
  - ▷ Similar to some redundancy situations
  - ▷ prescription monitor in Module 8

## Risk Resolution: Damage Reduction

---

- Damage reduction: Why?
  - ▷ Risk factors:  
 $f(\text{prob}(\text{haz}), \text{prob}(\text{haz} \rightarrow \text{acc}), \text{damage})$
  - ▷ All the hazard resolution techniques  
 $\Rightarrow$  risk  $\neq$  0 still!
  - ▷ Damage reduction needed
  - ▷ Passed “point of no return”
  
- Specific techniques:
  - ▷ Escape routes (lifeboats, fire escapes, evacuation plans, etc.)
  - ▷ Safe abandonment (haz. waste disposal)
  - ▷ Devices for limiting damage:
    - auto safety devices
    - limited melt-down
    - collapsible signpost, etc.

## Perspectives

---

- SSE: Augment S/w Eng.
  - ▷ Analysis to identify hazard
  - ▷ Design for safety
  - ▷ Verify safety constraints (next module)
  - ▷ Leveson's s/w safety program
  
- Dealing with hazard/risk in SSE:
  - ▷ Hazard identification and analysis
  - ▷ Design for safety/hazard resolution:
    - Hazard elimination/reduction/control
  - ▷ Damage reduction
  - ▷ Safety verification
  - ▷ All in SSE context: hazard focus.