

# System and Software Safety



# Accidents & Incidents

## Accident or Mishap

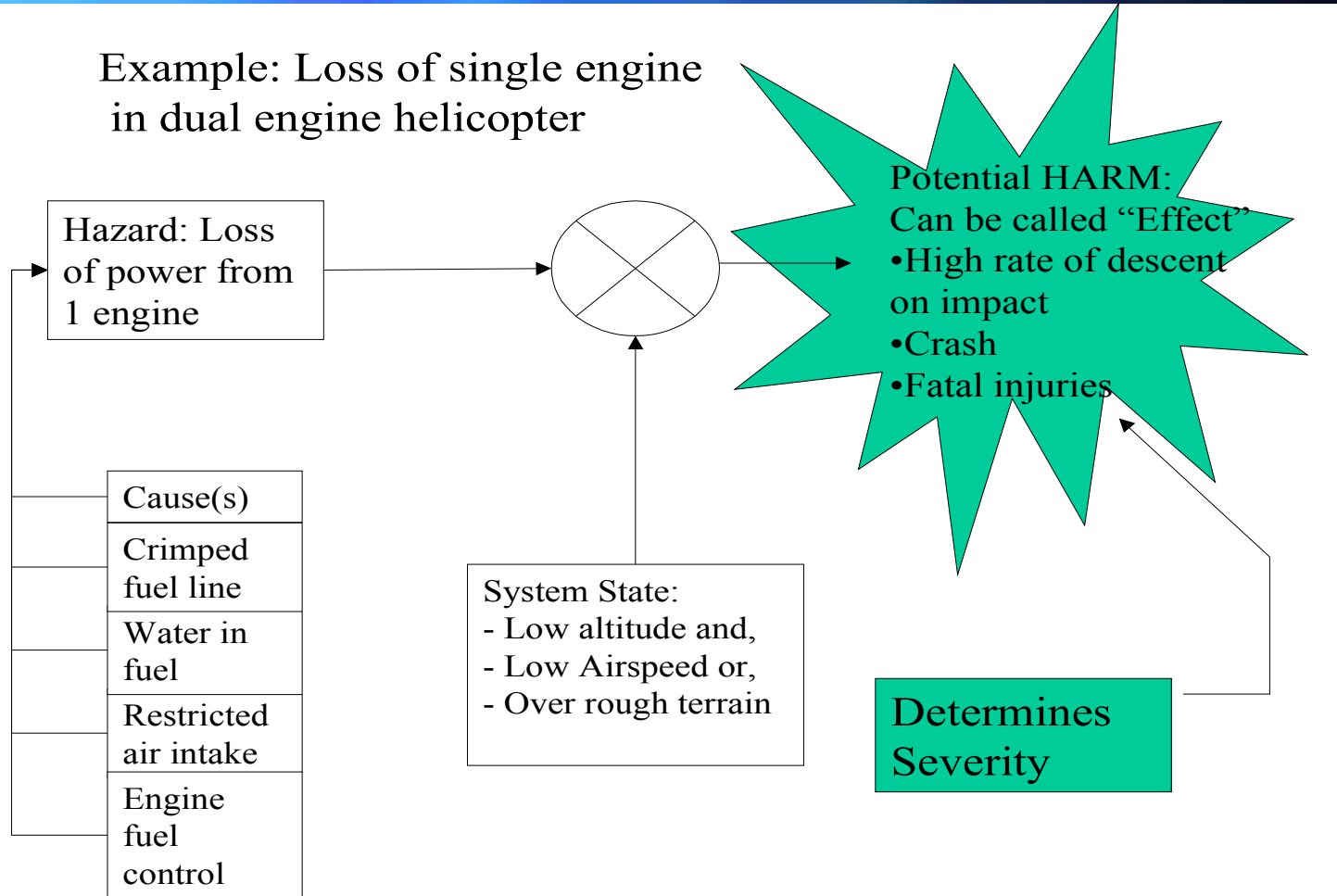
- An unplanned event or series of events that results in:
  - death, injury, illness
  - damage to or loss of equipment or property, or
  - environmental harm

## Incident

- A near miss
- Not to be used in place of “accident”

# Progression To Accidents

Example: Loss of single engine  
in dual engine helicopter



# Accidents Nature & Cause

---

A particular, although unusual, event or combination of events is typically the catalyst that is required

Many serious accidents are caused by complex interactions between system components and by multiple failures

We are usually very good at addressing single point failures

The “who would have thought?” syndrome

# Accidents Nature & Cause

---

A system can enter an unsafe state in several ways:

- Hardware component failures
- Interfacing (i.e., communication and timing) problems between system components
- Human error in operation and maintenance
- Environmental stress
- Software control errors

# Safety, Safety-Critical, Safety-Related

## Safety

- Freedom from those conditions that can cause death, injury, illness, damage to or loss of equipment or property, or environmental harm

## Safety-Critical

- Term applied to a condition, event, operation, process or item of whose proper recognition, control, performance or tolerance is essential to safe system operation or use
- e.g., safety-critical function, safety-critical path, safety-critical component

## Safety-Related

- A term encompassing all levels of safety criticality including safety-critical

# Myths Regarding Safety

---

Our system (or software) is safe

That can't happen

Safety engineers are not adept at reliability

Good engineering practices ensure safety

Longevity of systems in the field with no accidents implies that these systems are safe

# Myths Regarding Safety

---

Software cannot fail

An individual is not professionally liable for the products he/she develops

ISO-9000 certification implies something about system safety

# System Safety Significant Contributors

---

Reliability

Instrumentation & Measurement

Human Factors

System Engineering

Software

# System Safety Significant Contributors

## Reliability

- e.g., increasing safety can decrease reliability
- Drives quantitative safety targets

e.g.,  $P < 10^{-9}$  for catastrophic hazards

- Also relevant where:

Certification Maintenance Requirements, and  
Minimal Equipment Lists are involved

# System Safety Significant Contributors

## Instrumentation & Measurement

- Huge contributor to overall system safety
- Can be the root cause of accidents
- Can have an excellent control system but if the integrity of the data from the sensors is not dependable, then what?
- Redundancy and diversity of electrical and electronic instruments is a prime consideration
- Unique knowledge of failure modes is essential  
Can only come from subject area experts

# System Safety Significant Contributors

## Human Factors

- The operator can either use or abuse your system!
- The design of the Human-Machine Interface (HMI) is extremely important
- There is a strong link between Human Factors Engineering and System Safety

One good link is via Operating & Support Hazard Analysis

- Human reliability analysis

# System Safety Significant Contributors

Type of Human Behavior	Human Error Probability
Extraordinary errors - it is difficult to conceive how they could occur. Stress free, with powerful cues pointing to success.	$10^{-5}$
Errors in regularly performed, commonplace simple tasks with minimum stress.	$10^{-4}$
Errors of commission such as pressing the wrong button or reading the wrong display. Reasonably complex tasks, little time available, some cues necessary.	$10^{-3}$
Errors of omission where dependence is placed on situation and memory. Complex, unfamiliar task with little feedback and some distraction.	$10^{-2}$
Highly complex task, considerable stress, little time available.	$10^{-1}$
Process involving creative thinking, unfamiliar, complex operations where time is short and stress is high.	$1 - 10^{-1}$

# System Safety Significant Contributors

## System Engineering

- Bi-directional path between system engineering and system safety
- First level of safety scrutiny is performed by engineering subsystem area specialists
- System safety sets safety-related reliability targets, safety integrity levels (development assurance levels) for systems, subsystems, items, hardware, software, and influences system architecture and testing
- System safety defines additional requirements (derived ones) for system engineering

# System Safety

## Significant Contributors

### Software

---

#### Software Cannot Harm You?

- Software itself typically cannot cause harm
- Hardware which it controls can kill or injure
- A moot point though

#### Exceptions

- Stand alone diagnostic systems
- Air traffic systems

# System Safety

## Significant Contributors

### Software

---

Safe software?

- “System” safety is the issue
- Safety is an emergent property

e.g., a bolt – on its own its safety cannot be assessed; we need to know what it is being used for; and what stresses are being applied to it

This is why we must be very cautious regarding software re-use in safety-critical applications

# System Safety

## Significant Contributors

### Software

#### Software-Related Accidents (examples)

- Saab JAS39 Gripen fighter plane crash
- Computer-controlled fuel system problems
- Zaragoza Spain cancer radiation, at least 3 died
- Patriot missile system, 27 killed, 97 wounded
- Aircraft crashes into Mount Erebus, 257 killed
- F-18 missile thrust while clamped on, plane lost 20,000 feet
- F-14 lost to uncontrollable spin, traced to tactical software

# System Safety

## Significant Contributors

### Software

---

#### Software-Related Accidents (their cause)

- Failures to sense hazardous conditions requiring corrective action
- Producing incorrect responses to hazardous conditions
- Failures to perform required functions
- Performing unintended functions
  - BUHF (Built-in Unintended Hazardous Functions)
- Performing functions at wrong time or in wrong order

# Safety-Critical System Design

## Ideal Goals

- Simplicity
- Determinism
- Dependability

Safety  
Reliability  
Security

# System Safety Nemesis Complexity

Complexity becomes a problem when it decreases understandability or verifiability of system

## Functional Complexity

- Difficulty of problem being solved
  - Functions that system is expected to provide
  - Interactions among those functions

## Structural Complexity (Design Complexity)

- Complexity of system architecture and design
- Design should not be more complex than necessary

# System Safety Nemesis Complexity

## Code Complexity

- Easiest to measure

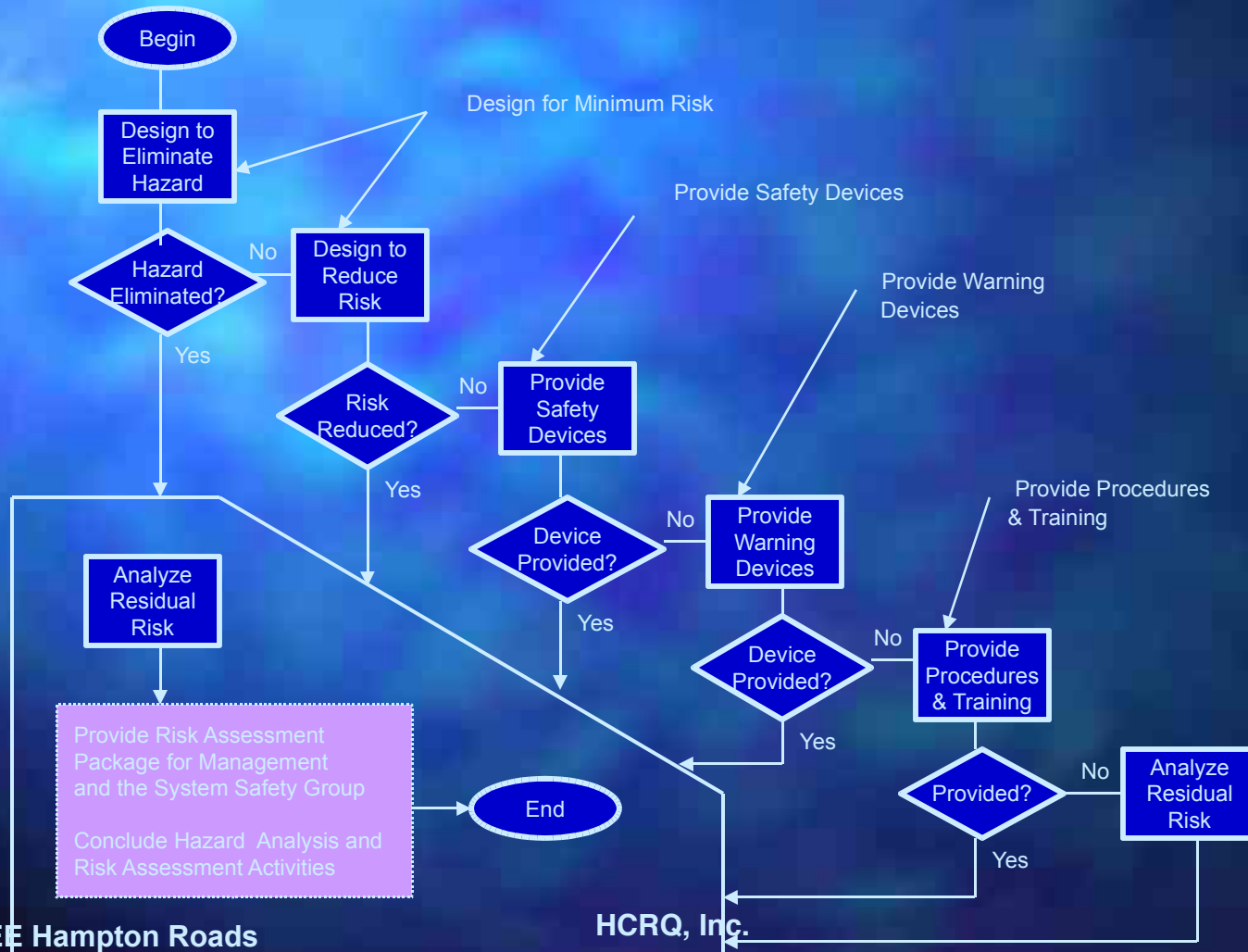
Generally receives largest share of attention

- Cyclomatic Complexity

Demands that code modules have metric less than 10

- Likely to result in an overly complex structure of modules

# System Safety Precedence



# System Safety Approaches

---

## Proactive (life cycle)

- System Safety Program

## Post-design

- Wait until system is designed then try to identify and mitigate hazards

## Reactive approach

- Wait until catastrophe strikes, then identify and mitigate
- Very sobering to analyze a system that has killed

# System Safety

## Approaches To Safe Design

---

- Two-pronged simultaneous approach
  1. Apply:
    - Checklists
    - Guidelines
    - Standards
  2. Design Guided By Hazard Analyses

# System Safety Difficult Aspects

---

COTS (Commercial-Off-The-Shelf) Software

Subcontractor safety management

System safety integration

Client with little safety engineering knowledge

Client with over-zealous safety consultants

# Safety & Engineering Management

---

Good engineering management and safety management are crucial to the success of system safety programs

Key groups include system safety and software safety groups, System Safety Working Group, Software Safety Working Group, Safety Committees

Safety management oversees System Safety Program which encompasses the subcontractor System Safety Programs

Signs off on overall system safety

- Potentially accepts risk associated with some hazards

Required during design, construction, test & commissioning, operations & maintenance

# Safety Consulting

## Knowing When You Have Done Enough

- We deploy safety-critical systems  
when we have enough assurance  
not when we know the system is safe
- Experienced safety engineers “know” when they have gone far enough
  - Subjective
  - Frustrating for those less skilled
- If one has insufficient experience or wants to assume little risk
  - Approaches tend to be over-zealous
    - Leading to increased costs and delays

# Safety Consulting

---

Everyone has an opinion about safety

- A little bit of knowledge is a dangerous thing
- You hear a lot of stupid ideas

Safety can easily become a “make work” exercise

# Interesting Quotes From Industry

---

Weapons system are inherently unsafe

Our pilots are used to assuming risk

Statistically speaking, the number of accidents over the combined lifetimes of our products isn't bad

The patient did not have long to live anyhow

We comply with the standard

But we test it

I've been hired as a safety engineer to deflect our customer's safety concerns