Systeme hoher Qualität und Sicherheit
Universität Bremen WS 2015/2016

Lecture 04 (02.11.2015)

Hazard Analysis

Christoph Lüth Jan Peleska Dieter Hutter

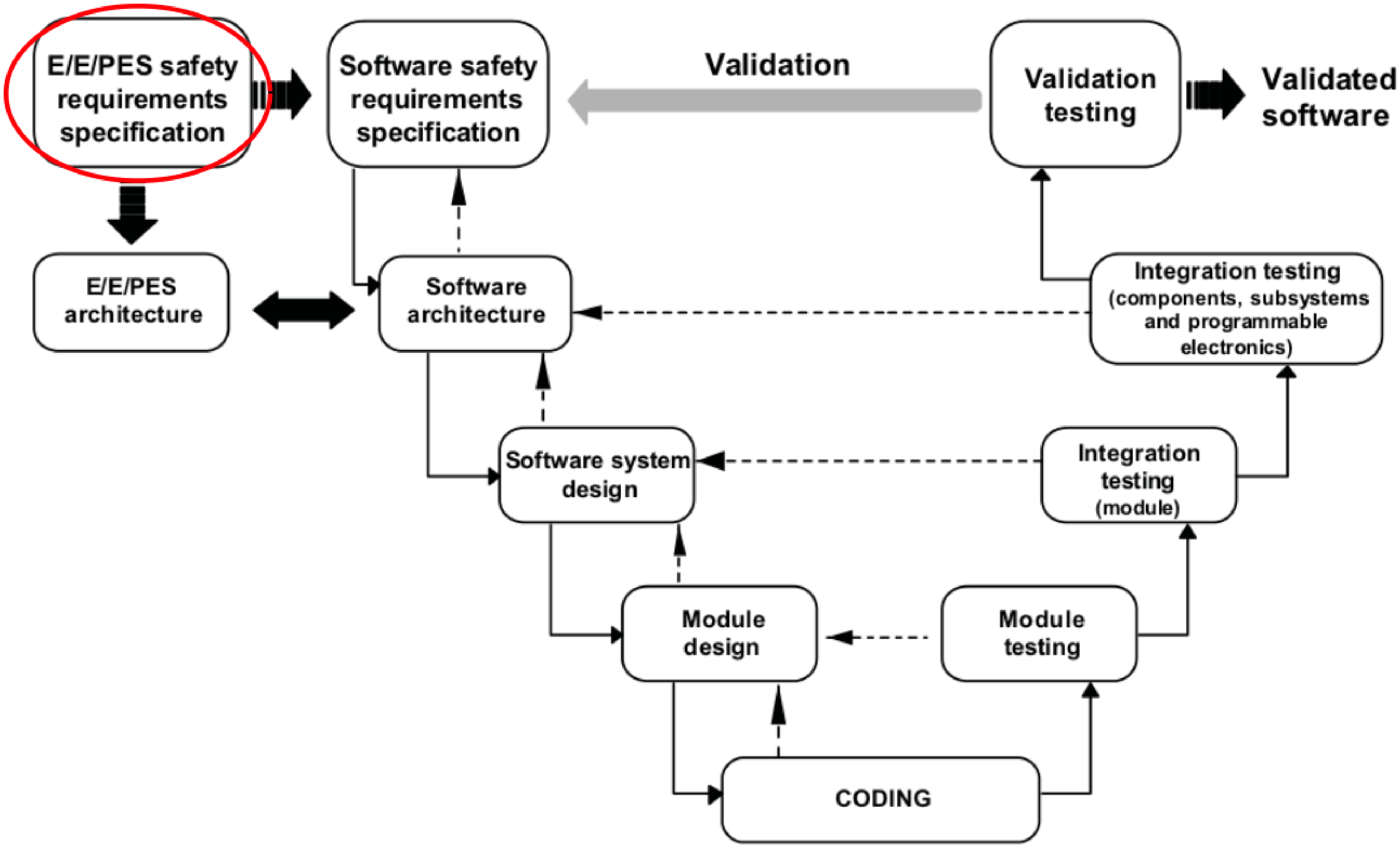
Where are we?

- 01: Concepts of Quality
- 02: Legal Requirements: Norms and Standards
- 03: The Software Development Process
- 04: Hazard Analysis
- 05: High-Level Design with SysML
- 06: Formal Modelling with SysML
- 07: Detailed Specification with SysML
- 08: Testing
- 09 and 10: Program Analysis
- 11: Model-Checking
- 12: Software Verification (Hoare-Calculus)
- 13: Software Verification (VCG)
- 14: Conclusions

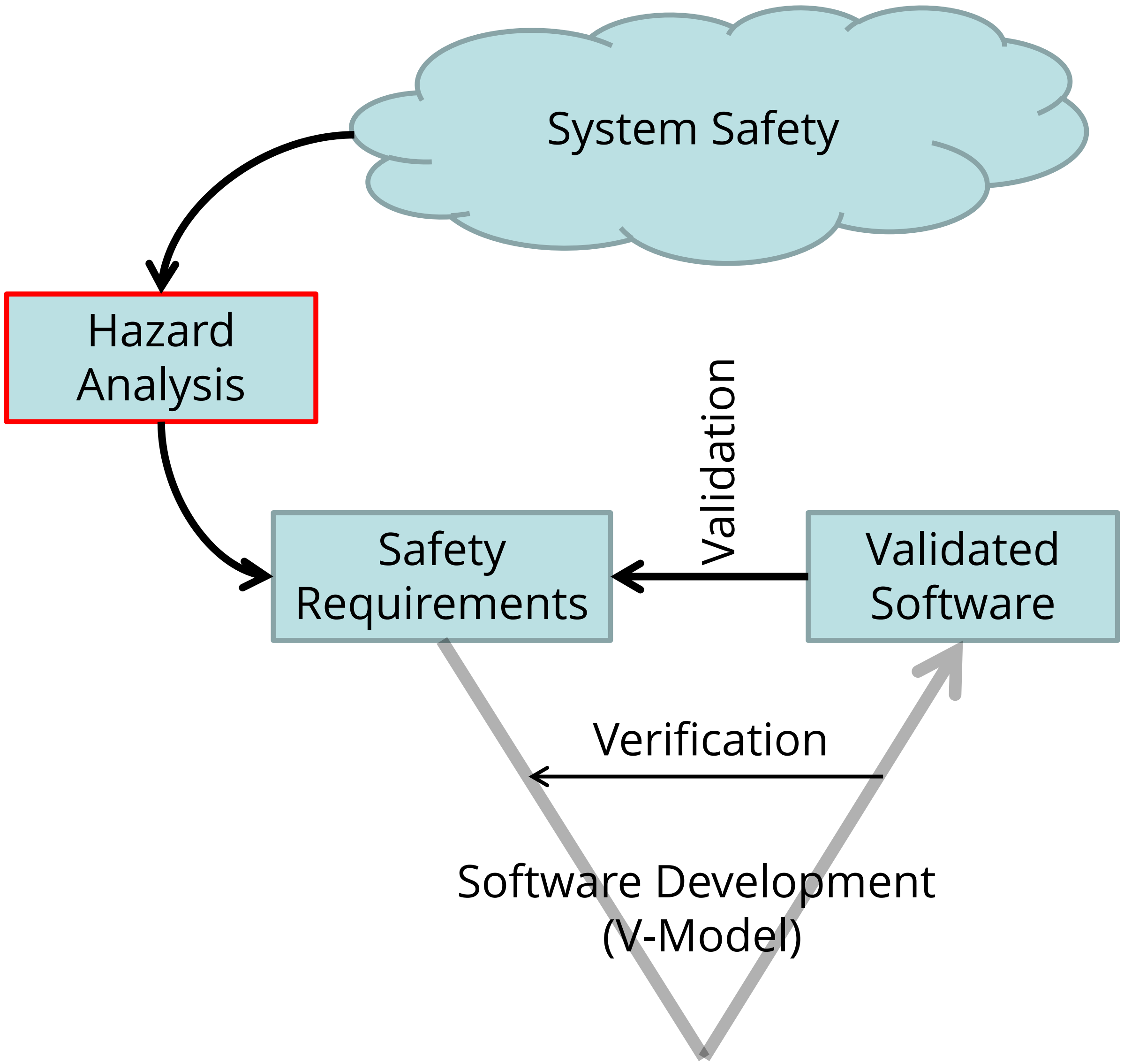
Your Daily Menu

- Hazard Analysis:
  - What‘s that?
- Different forms of hazard analysis:
  - Failure Mode andEffects Analysis (FMEA)
  - Failure Tree Analysis (FTA)
  - Event Tree Analysis (ETA)

Hazard Analysis in the Development Cycle



The Purpose of Hazard Analysis



Hazard Analysis systematically determines a list of **safety requirements**.

The realisation of the safety requirements by the software product must be **verified**.

The product must be **validated** wrt. the safety requirements.

Hazard Analysis...

- provides the basic foundations for system safety.
- is performed to identify hazards, hazard effects, and hazard causal factors.
- is used to determine system risk, to determine the signifigance of hazards, and to etablish design measures that will eliminate or mitigate the identified hazards.
- is used to **systematically** examine systems, subsystems, facilities, components, software, personnel, and their interrelationships.

Clifton Ericson: *Hazard Analysis Techniques for System Safety*.
Wiley-Interscience, 2005.

Form and Output of Hazard Analysis

- The output of Hazard Analysis is a list of safety requirements, and documents detailing how these were derived.
- Because the process is informal, it can only be **checked** by **reviewing**.
- It is therefore critical that
  - standard forms of analysis are used,
  - documents have a standard form, and
  - all assumptions are documented.

Classification of Requirements

- Requirements to ensure
  - Safety
  - Security

- Requirements for
  - Hardware
  - Software

- Characteristics / classification of requirements
  - according to the type of a property

Classification of Hazard Analysis

- **Top-down methods** start with an anticipated hazard and work back from the hazard event to potential causes for the hazard
  - Good for finding causes for hazard
  - Good for avoiding the investigation of “non-relevant” errors
  - Bad for detection of missing hazards

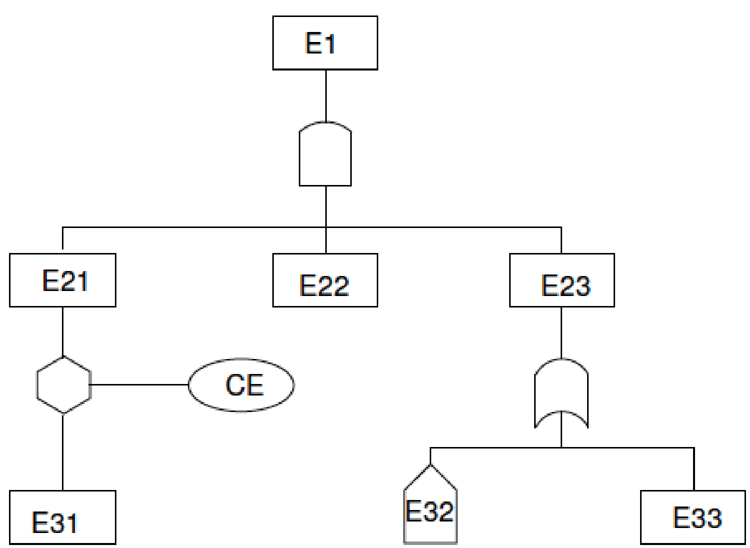
- **Bottom-up methods** consider “arbitrary” faults and resulting errors of the system, and investigate whether they may finally cause a hazard
  - Properties are complementary to top-down properties

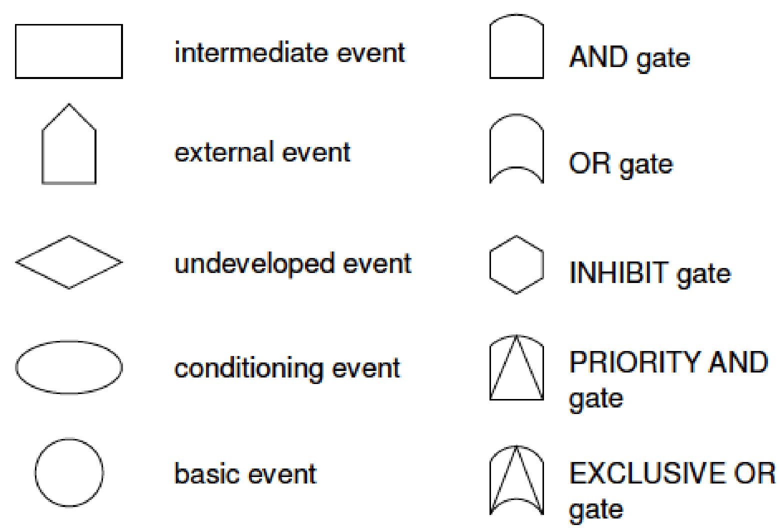
Hazard Analysis Methods

- Fault Tree Analysis (FTA) – top-down
- Failure Modes and Effects Analysis (FMEA) – bottom up
- Event Tree Analysis (ETA) – bottom-up
- Cause Consequence Analysis – bottom up
- HAZOP Analysis – bottom up

Fault Tree Analysis (FTA)

- Top-down deductive failure analysis (of undesired states)
  - Define undesired top-level event
  - Analyse all causes affecting an event to construct fault (sub)tree
  - Evaluate fault tree





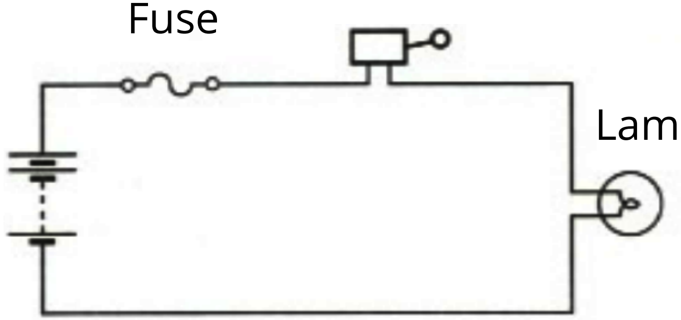
Fault-Tree Analysis: Process Overview

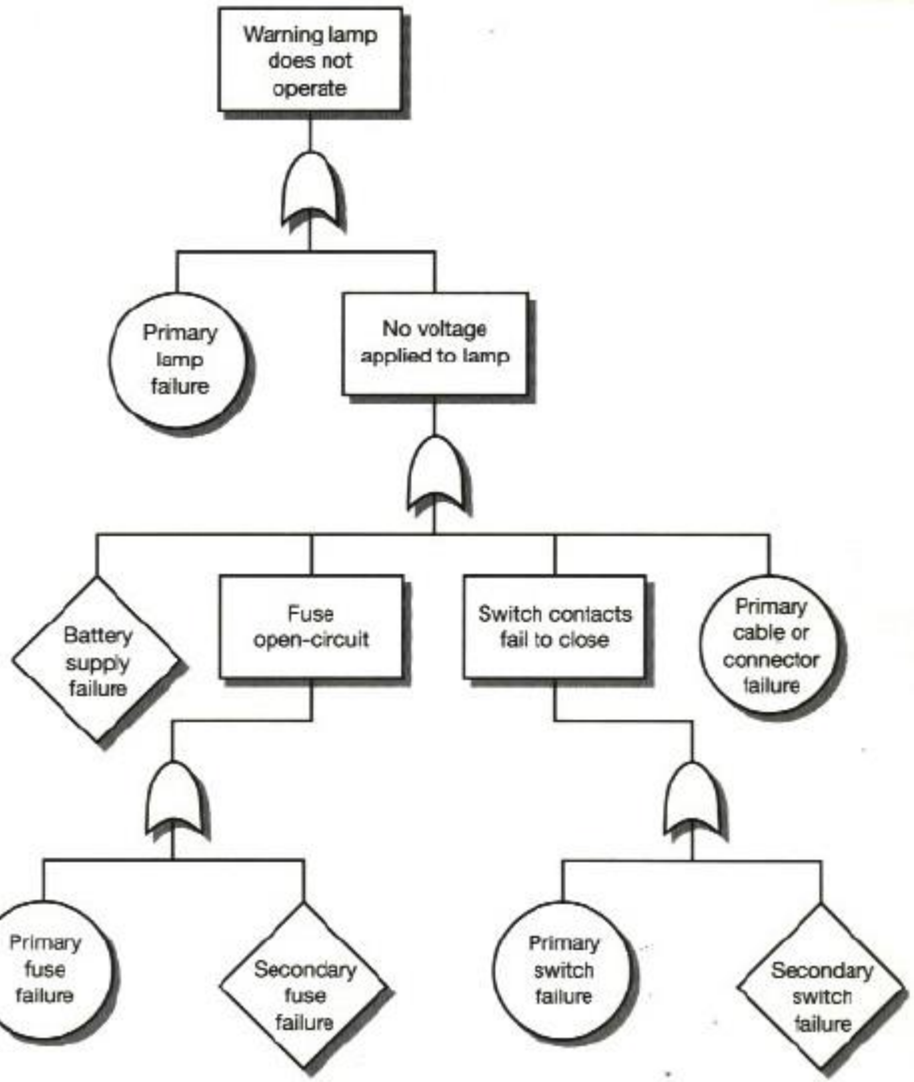
1. Understand system design
2. Define top undesired event
3. Establish boundaries (scope)
4. Construct fault tree
5. Evaluate fault tree (cut sets, probabilities)
6. Validate fault tree (check if correct and complete)
7. Modify fault tree (if required)
8. Document analysis

Fault Tree Analysis: Example 1

Example:

- A lamp warning about low level of brake fluid.
- See circuit diagram.
- Top Undesired Event: Warning lamp off despite low level of fluid.



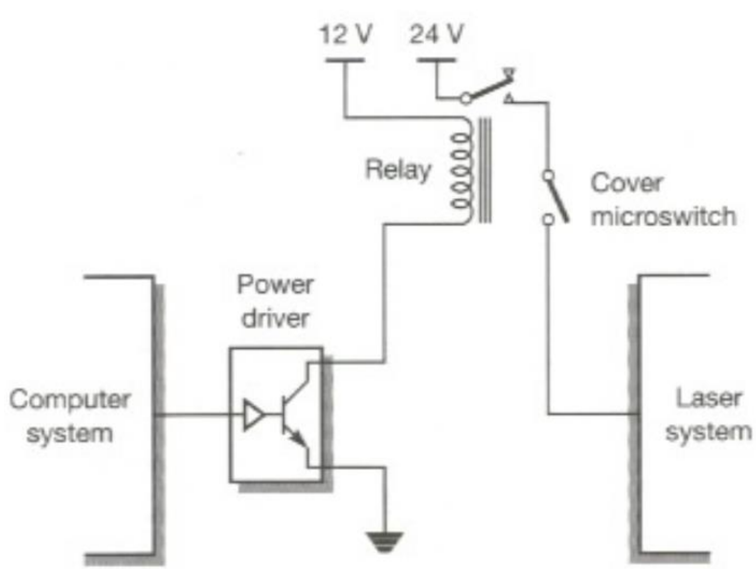


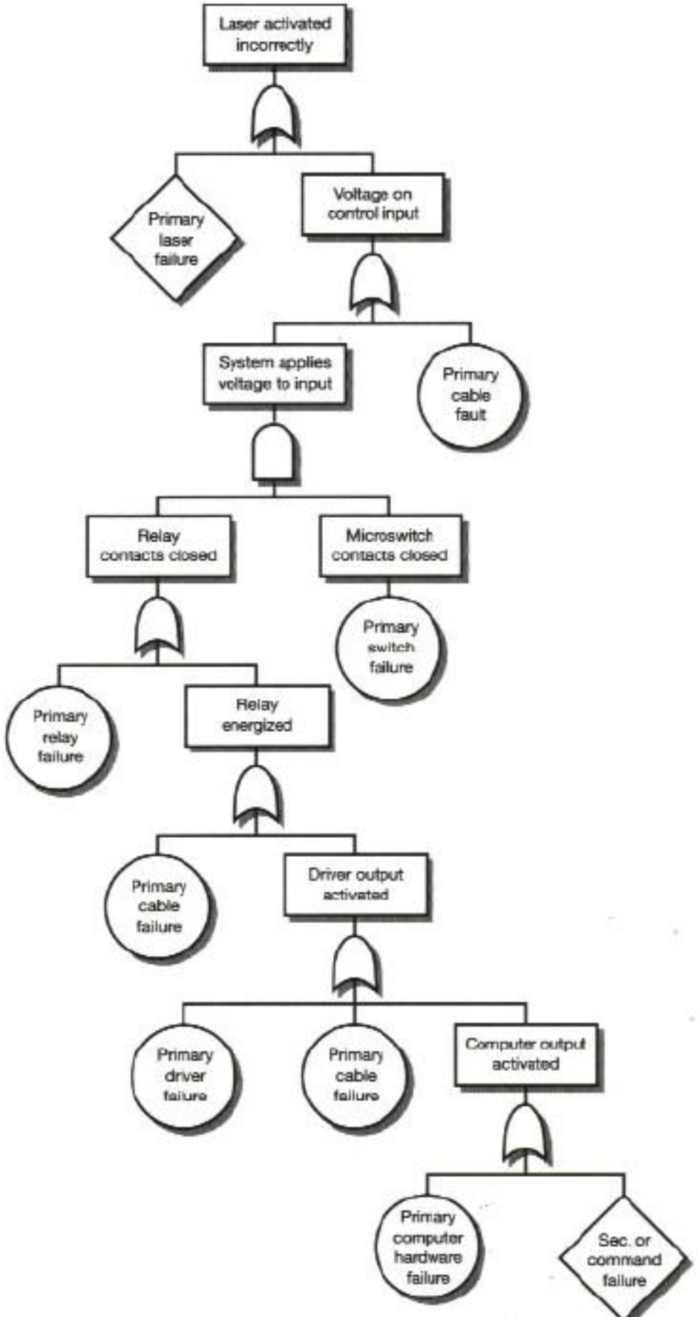
Source: N. Storey, Safety-Critical Computer Systems.

FTA: Example II

Example: A laser operated from a control computer system.

- The laser is connected via a relay and a power driver, and protected by a cover switch.
- Top Undesired Event:
  Laser activated without explicit command from computer system.





Source: N. Storey, Safety-Critical Computer Systems.

Event Tree Analysis (ETA)

- Applies to a chain of cooperating activities
- Investigates the effect of activities failing while the chain is processed
- Depicted as binary tree; each node has two leaving edges:
  - Activity operates correctly
  - Activity fails
- Useful for calculating risks by assigning probabilities to edges
- O(2^n) complexity

Event Tree Analysis Overview

Input:

- Design knowledge
- Accident histories

ETA Process:

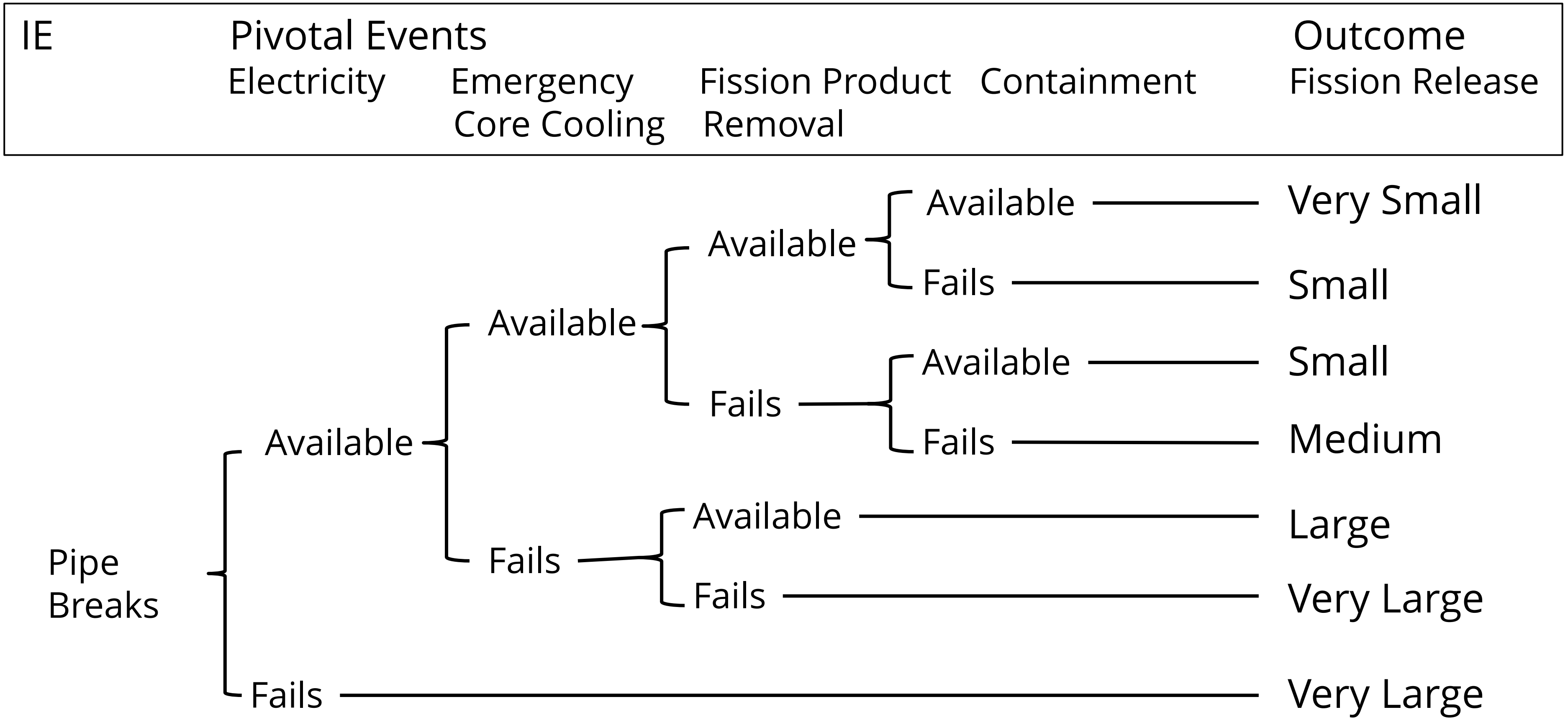
1. Identify Accident Scenarios
2. Identify IEs (Initiating Events)
3. Identify pivotal events
4. Construct event tree diagrams
5. Evaluate risk paths
6. Document process

Output:

- Mishap outcomes
- Outcome risks
- Causal sources
- Safety Requirements

Event Tree Analysis: Example 1

- Cooling System for a Nuclear Power Plant



Event Tree Analysis: Example 2

- Fire Detection/Suppression System for Office Building

IE	Pivotal Events			Outcomes	Prob.
	Fire Detection Works	Fire Alarms Works	Fire Sprinkler Works		
Fire Starts P= 0.01	YES (P= 0.9)	YES (P= 0.7)	YES (P= 0.8)	Limited damage	0.00504
			NO (P= 0.2)	Extensive damage, People escape	0.00126
		NO (P= 0.3)	YES (P= 0.8)	Limited damage, Wet people	0.00216
			NO (P= 0.2)	Death/injury, Extensive damage	0.00054
	NO (P= 0.1)			Death/injury, Extensive damage	0.001

Failure Modes and Effects Analysis (FMEA)

- Analytic approach to review potential failure modes and their causes.
- Three approaches: *functional*, *structural* or *hybrid.*
- Typically performed on hardware, but useful for software as well.
- It analyzes
  - the failure mode,
  - the failure cause,
  - the failure effect,
  - its criticality,
  - and the recommended action.

and presents them in a **standardized table**.

Software Failure Modes

Guide word	Deviation	Example Interpretation
omission	The system produces no output when it should. Applies to a single instance of a service, but may be repeated.	No output in response to change in input; periodic output missing.
commission	The system produces an output, when a perfect system would have produced none. One must consider cases with both, correct and incorrect data.	Same value sent twice in series; spurious output, when inputs have not changed.
early	Output produced before it should be.	Really only applies to periodic events; Output before input is meaningless in most systems.
late	Output produced after it should be.	Excessive latency (end-to-end delay) through the system; late periodic events.
value (detectable)	Value output is incorrect, but in a way, which can be detected by the recipient.	Out of range.
value (undetectable)	Value output is incorrect, but in a way, which cannot be detected.	Correct in range; but wrong value

Criticality Classes

- Risk as given by the *risk mishap index* (MIL-STD-882):

Severity	**Probability**
1. Catastrophic	A. Frequent
2. Critical	B. Probable
3. Marginal	C. Occasional
4. Negligible	D. Remote
	E. Improbable

- Names vary, principle remains:
  - Catastrophic – single failure
  - Critical – two failures
  - Marginal – multiple failures/may contribute

FMEA Example: Airbag Control (Struct.)

ID	Mode	Cause	Effect	Crit.	Appraisal
1	Omission	Gas cartridge empty	Airbag not released in emergency situation	C1	SR-56.3
2	Omission	Cover does not detach	Airbag not released fully in emergency situation.	C1	SR-57.9
3	Omission	Trigger signal not present in emergency.	Airbag not released in emergency situation	C1	Ref. To SW-FMEA
4	Comm.	Trigger signal present in non-emergency	Airbag released during normal vehicle operation	C2	Ref. To SW-FMEA

FMEA Example: Airbag Control (Funct.)

ID	Mode	Cause	Effect	Crit.	Appraisal
5-1	Omission	Software terminates abnormally	Airbag not released in emergency.	C1	See 1.1, 1.2.
5-1.1	Omission	- Division by 0	See 1	C1	SR-47.3 Static Analysis
5-1.2	Omission	- Memory fault	See 1	C1	SR-47.4 Static Analysis
5-2	Omision	Software does not terminate	Airbag not released in emergency.	C1	SR-47.5 Static Analysis
5-3	Late	Computation takes too long.	Airbag not released in emergency.	C1	SR-47.6
5-4	Comm.	Spurious signal generated	Airbag released in non-emergency	C2	SR-49.3
5-5	Value (u)	Software computes wrong result	Either of 5-1 or 5-4.	C1	SR-12.1 Formal Verification

The Seven Principles of Hazard Analysis

Ericson (2005)

1) Hazards, mishaps and risk are not chance events.
2) Hazards are created during design.
3) Hazards are comprised of three components.
4) Hazards and mishap risk is the core safety process.
5) Hazard analysis is the key element of hazard and mishap risk management.
6) Hazard management involves seven key hazard analysis types.
7) Hazard analysis primarily encompasses seven hazard analysis techniques.

Summary

- Hazard Analysis is the **start** of the formal development.
- Its most important output are **safety requirements**.
- Adherence to safety requirements has to be **verified** during development, and **validated** at the end.
- We distinguish different types of analysis:
  - Top-Down analysis (Fault Trees)
  - Bottom-up (FMEAs, Event Trees)
- It makes sense to combine different types of analyses, as their results are complementary.

Conclusions

- Hazard Analysis is a creative process, as it takes an informal input („system safety“) and produces a formal outout (safety requirements). Its results cannot be formally proven, merely checked and reviewed.
- Review plays a key role. Therefore,
  - documents must be readable, understandable, auditable;
  - analysis must be in well-defined and well-documented format;
  - all assumptions must be well documented.
- Next week: High-Level Specification.