1918 TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Department of computer Engineering
ati.ttu.ee

IAF0530/IAF9530

# Dependability and fault tolerance

Lectures 2 and 3
Safety, Hazards, Risks

**Gert Jervan**
gert.jervan@pld.ttu.ee

Tallinn University of Technology
Department of Computer Engineering
Estonia



## Case Studies

- List of example topics in the web
  - www.pld.ttu.ee/IAF0530
- Topic selection:
  - February 28 (via e-mail, no lecture at that day)
- Draft of the report (incl. introductory presentation of the topic):
  - March 21
- If in doubt – ASK!!





## Lecture Outline

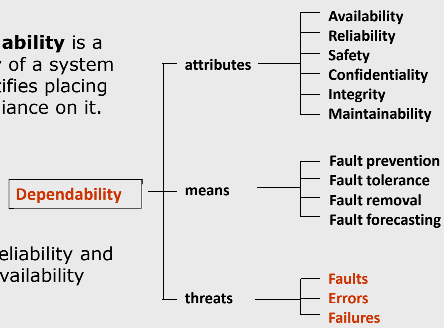
- Dependability
- Safety Requirements
- Hazards
- Hazard Analysis
- Risks
- Risk Analysis
- Risk Management
- Safety & SILs
- Risk Reduction & Design





## Dependability: an integrating concept

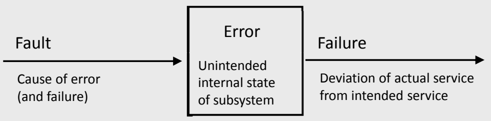
- **Dependability** is a property of a system that justifies placing one's reliance on it.
- High reliability and high availability







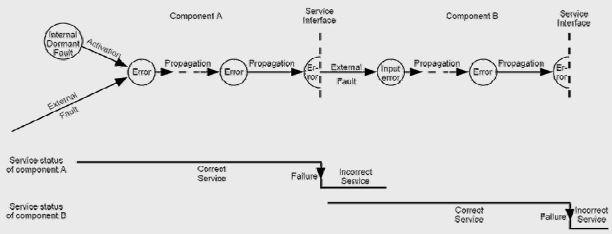

## Threats: Faults, Errors & Failures








## The pathology of failure



## Three-universe model

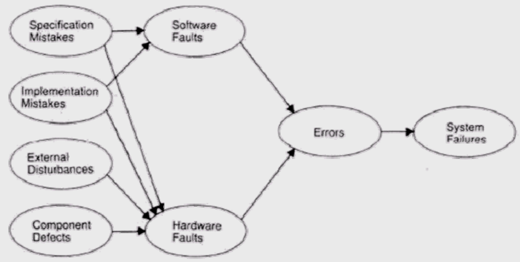
- Physical universe: where the faults occur
  - Physical entities: semiconductor devices, mechanical elements, displays, printers, power supplies
  - A fault is a physical defect or alteration of some component in the physical universe
- Informational universe: where the error occurs
  - Units of information: bits, data words
  - An error has occurred when some unit of information becomes incorrect
- External (user's universe): where failures occur
  - User sees the effects of faults and errors
  - The failure is any deviation from the desired or expected behavior





## Causes of faults

- Problems at any stages of the design process can result in faults within the system.








## Causes of faults, cont.

- Specification mistakes
  - Incorrect algorithms, architectures, hardware or software design specifications
    - Example: the designer of a digital circuit incorrectly specified the timing characteristics of some of the circuit's components
- Implementation mistakes
  - Implementation: process of turning the hardware and software designs into physical hardware and actual code
  - Poor design, poor component selection, poor construction, software coding mistakes
    - Examples: software coding error, a printed circuit board is constructed such that adjacent lines of a circuit are shorted together





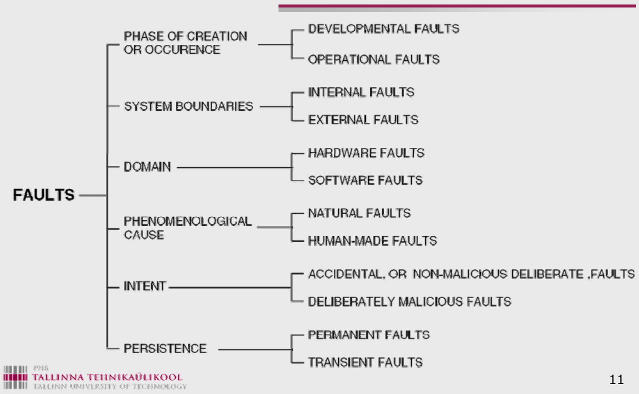
## Causes of faults, cont.

- Component defects
  - Manufacturing imperfections, random device defects, component wear-out
  - Most commonly considered causes of faults
    - Examples: bonds breaking within the circuit, corrosion of the metal
- External disturbance
  - Radiation, electromagnetic interference, operator mistakes, environmental extremes, battle damage
    - Example: lightning





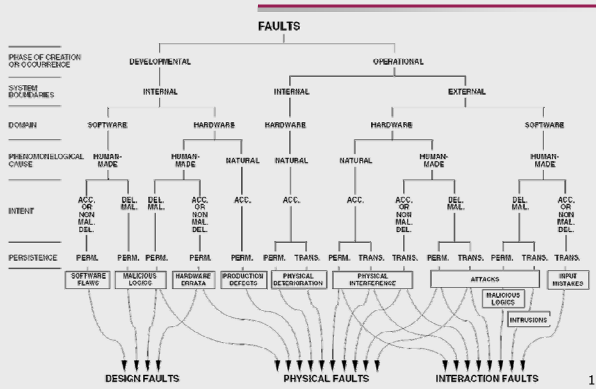
## Elementary fault classes



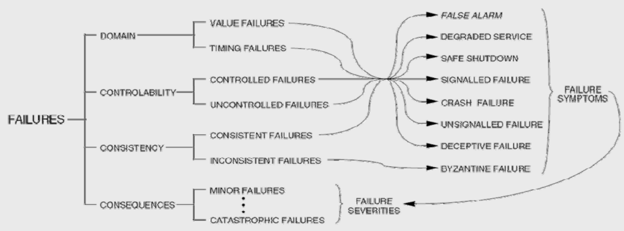





## Classification of faults



## Failure modes








## Failure modes, cont.

- Failure domain
  - Value failures : incorrect value delivered at interface
  - Timing failures : right result at the wrong time (usually late)
- Failure consistency
  - Consistent failures : all nodes see the same, possibly wrong, result
  - Inconsistent failures : different nodes see different results
- Failure consequences
  - Benign failures : essentially loss of utility of the system
  - Malign failures : significantly more than loss of utility of the system; catastrophic, e.g. airplane crash
- Failure oftenness (failure frequency and persistency)
  - Permanent failure : system ceases operation until it is repaired
  - Transient failure : system continues to operate
    - Frequently occurring transient failures are called intermittent





## Dependability

- **Dependability**
  - **attributes**
    - **Availability**
    - **Reliability**
    - **Safety**
    - **Confidentiality**
    - **Integrity**
    - **Maintainability**
  - **means**
    - **Fault prevention**
    - **Fault tolerance**
    - **Fault removal**
    - **Fault forecasting**
  - **threats**
    - **Faults**
    - **Errors**
    - **Failures**





## Dependability attributes

- **Availability**: readiness for correct service
- **Reliability**: continuity of correct service
- **Safety**: absence of catastrophic consequences on the user(s) and the environment
- **Confidentiality**: absence of unauthorized disclosure of information
- **Integrity**: absence of improper system alterations
- **Maintainability**: ability to undergo, modifications, and repairs

- **Security**: the concurrent existence of (a) availability for authorized users only, (b) confidentiality, and (c) integrity with 'improper' taken as meaning 'unauthorized'.





## Dependability

- **Dependability**
  - **attributes**
    - **Availability**
    - **Reliability**
    - **Safety**
    - **Confidentiality**
    - **Integrity**
    - **Maintainability**
  - **means**
    - **Fault prevention**
    - **Fault tolerance**
    - **Fault removal**
    - **Fault forecasting**
  - **threats**
    - **Faults**
    - **Errors**
    - **Failures**

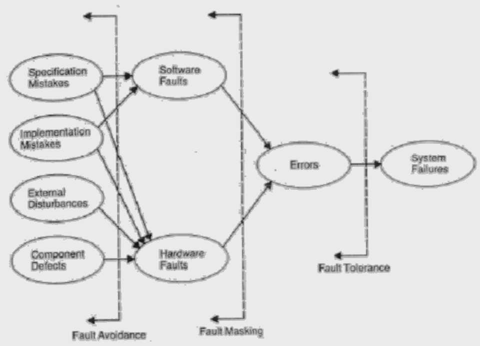




## Means to achieve dependability

- **Fault-prevention:** how to prevent, by construction, fault occurrence.
- **Fault-tolerance:** how to provide, by redundancy, service complying with the specification in spite of faults having occurred or occurring.
- **Fault-removal:** how to minimize, by verification and validation, the presence of latent faults.
- **Fault-forecasting:** how to minimize, by evaluation, the presence, the creation and the consequences of faults.

## Means to achieve dependability, cont.








## Fault prevention

- Attained by quality control techniques
  - Software
    - Structured/object oriented programming
    - Information hiding
    - Modularization
  - Hardware
    - Rigorous design rules
    - Shielding
    - Radiation hardening
    - “Foolproof” packaging
- Note:
  - Malicious faults can also be prevented; Example: firewalls





## Fault tolerance

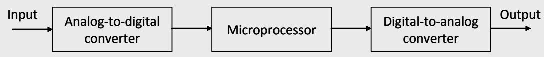
- **Fault tolerance** is the ability of a system to continue to perform its functions (deliver correct service), even when one or more components have failed.
  - **Masking**: the use of sufficient redundancy may allow recovery without explicit error detection.
  - **Reconfiguration**: eliminating a faulty entity from a system and restoring the system to some operational condition or state.
    - Error **detection**: recognizing that an error has occurred
    - Error **location**: determining which module produced the error
    - Error **containment**: preventing the errors from propagating
    - Error **recovery**: regaining operational status





## The concept of redundancy

- Definition
  - **Redundancy** is the addition of information, resources, or time beyond what is needed for normal system operation.
- Digital filter example
  - Software redundancy: lines of software to perform a validity checks
  - Hardware redundancy : if more memory needed for the software checks
  - Time redundancy: each filter calculation performed twice to detect faults
  - Information redundancy: output using with a simple parity bit








## Error detection

- Two ways to detect errors:
  - a priori knowledge about intended state
  - comparing results of two redundant computational channels
- Notes
  - Errors can happen in the **value domain** and/or in the **time domain**.
  - The probability that an error is detected, provided it is present, is called the **error detection coverage**.
  - The time interval between the start of an error and the detection of an error is the **error detection latency**.





## *A Priori* Knowledge
flexibility vs. error-detection coverage

- **Syntactic knowledge about code space**
  - Parity bits, CRC
- **Assertions and acceptance tests**
  - Valid data values, properties of the controlled object
    - Development of physical processes, plausibility of data sets
- **Activation patterns of computation**
  - Regularity in execution pattern, e.g., frequency of updates
    - Limited by the update frequency and clock synchronisation
    - Event every second, on the second -> detect missing event
- **Worst case execution time of tasks**
  - Must be known to calculate real-time schedules
  - A priory information about the execution of a task can be used for detecting task errors

## Redundant Computations

| Type of Redundancy | Implementation | Type of Detected Errors |
|---|---|---|
| Time redundancy | Same software executed on the same hardware during two different time-intervals | Errors caused by transient physical faults in hardware with a duration less than one execution time slot |
| Hardware redundancy | The same software executes on two independent hardware channels | Errors caused by transient and permanent physical hardware errors |
| Diverse software on the same hardware | Different software versions are executed on the same hardware during two different time intervals | Errors caused by independent software faults and transient physical faults in the hardware with a duration less than one execution time slot |
| Diverse software on diverse hardware | Two different versions of software are executed on two independent hardware channels | Errors caused by independent software faults and by transient and permanent physical hardware faults |





## Recovery

- Definition
  - **Recovery** transforms a system state that contains one or more errors and (possibly) faults into a state without detected errors and faults that can be activated again.
- Consists of
  - Error handling
    - **Rollback**: returning to a saved state (checkpoint)
    - **Compensation**: enough redundancy to eliminate the error
    - **Rollforward**: the state without errors is a new state
  - Fault handling
    - **Fault diagnosis**: identifies the cause of errors, location and type
    - **Fault isolation**: physical or logical exclusion of the faulty components
    - **System reconfiguration**: switches in spares or re-assigns tasks
    - **System reinitialization**: checks, updates and records the new configuration





## Fault removal

- **Verification**: "Are we building the system right?"
  - Static: does not exercise the system
    - Static analysis: inspections, walkthroughs, model checking
  - Dynamic
    - Symbolic execution: inputs are symbolic
    - Testing: actual inputs
  - Fault injection
- **Validation**: "Are we building the right system?"
  - Checking the specification





## Fault Forecasting

- Evaluation of the system behavior with respect to fault occurrence
  - **Qualitative** evaluation
    - Identifies, classifies, ranks the failure modes or the event combinations that lead to system failures
    - Example methods: Failure mode and effect analysis, fault-tree analysis
  - **Quantitative** evaluation
    - Evaluates in terms of probabilities the extent to which some of the dependability are satisfied (measures dependability)
    - Example methods: Markov chains, reliability block diagrams



TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
Department of computer Engineering
ati.ttu.ee

## Safety Requirements



## Definitions of Safety

- Informally
  - "Nothing bad will happen"
- N. Leveson, Safeware
  - "Freedom from accidents or losses"
  - But no system can be completely safe in absolute sense…
  - Focus is on making systems safe enough, given limited resources
  - Emphasis on accidents, rather than risk
- N. Storey, Safety-Critical Computer Systems:
  - "System will not endanger human life or environment"
  - More emphasis on removing hazards than actual accidents…
- Safety-critical system
  - System that has the potential to cause accidents

## Safety requirements

- In order to determine safety requirements:
  - Identification of the hazards associated with the system
  - Classification of these hazards
  - Determination of methods for dealing with the hazards
  - Assignment of appropriate reliability and availability requirements
  - Determination of an appropriate safety integrity level
  - Specification of development methods appropriate to this integrity level





## The Role of Standards

- Helping staff to ensure that a product meets a certain level of quality
- Helping to establish that a product has been developed using methods of known effectiveness
- Promoting a uniformity of approach between different teams
- Providing guidance on design and development techniques
- Providing some legal basis in the case of a dispute





## Conflicting requirements

- High performance v low cost
- Reliability ≠ safety

  BUT
- System must be reliable AND safe
- Hazard analysis and risk analysis to identify *acceptable* levels of safety and reliability



1918 TALLINNA TEHNIKAÜLIKOOL TALLINN UNIVERSITY OF TECHNOLOGY
Department of computer Engineering
ati.ttu.ee

## Hazard Analysis

**Hazards & Risk Definitions**



## Definitions

- Hazard
  - Situation with actual or potential danger to people, environment or material, of a certain severity
  - e.g. lock that prevents elevator door from opening is not activated
- Incident (near miss)
  - Unplanned event that involves no damage or loss, but has the potential to be an accident in different circumstances
  - e.g. elevator door opens while the elevator is missing but nobody is leaning against it





## Definitions (cont.)

- Accident
  - Unplanned event that results in a certain level of damage or loss to human life or the environment
  - e.g. elevator door opens and someone falls to the shaft
- Risk
  - Combination of the severity of a specified hazardous event with its probability of occurrence over a specified duration

## Risk Assessment

- Risk = penalty x likelihood
  - Penalty can be measured in money, lives, injuries, amount of deadline…
  - Likelihood is the probability that a particular hazard will be activated and result in an undesirable outcome
  - Pareto ranking: 80% of problems are from 20% of the risks…





## Risk Assessment (cont.)

- Example of risk calculation
  - Failure of a particular component results in chemical leak that could kill 500 people
  - Estimate that component will fail once every 10,000 years

    risk = penalty x (probability per year)

    = 500 x (0.0001)

    = 0.05 deaths per year
- But rare and costly events are a problem
  - E.g. infinite penalty multiplied by near-zero probability?
  - Must guard against catastrophic penalties event for near-zero probability





## Acceptability of Risk

- ALARP (As Low As is Reasonably Possible)
  - If risk can be easily reduced, it should be
  - Conversely, a system with significant risk may be acceptable if it offers sufficient benefit and if further reduction of risk is impractical
- Ethical considerations
  - Determining risk and its acceptability involves moral judgement
  - Society's view not determined by logical rules
  - Perception that accidents involving large numbers of deaths are perceived as more serious than smaller accidents, though they may occur less frequently





## Conflicting Requirements – Safety and Reliability

- A system can be unreliable but safe
  - If it does not behave according to specification but still does not cause an accident
- A system can be unsafe but reliable
  - If it can cause harm but faults occur with very low probability
- Fail Safe
  - System designed to fail in a safe state
    e.g. trains stop in case of signal failure
  - affects availability – 100% safe but 0% available..
- Fail Operational
  - System designed to keep working even if something fails
  - usually using redundancy
- Fail-over to reduced capability system
  - Mechanical backup



1918 TALLINNA TEHNIKAÜLIKOOL TALLINN UNIVERSITY OF TECHNOLOGY

Department of computer Engineering
ati.ttu.ee

# Hazards

**Hazards Overview**



## Hazards

- A Hazard is a system state that could lead to:
  - Loss of life
  - Loss of property
  - Release of energy
  - Release of dangerous materials
- Hazards are the *states* we have to avoid
- An accident is a loss event:
  - System in hazard state, ***and***
  - Change in the operating environment
- Classification
  - Severity
  - Nature

## Hazard Categories for Civil Aircraft

| DESCRIPTION | CATEGORY | DEFINITION | PROBABILITY |
|---|---|---|---|
| CATASTROPHIC | I | Loss of Lives, Loss of Aircraft | $10^{-9}$/hr |
| HAZARDOUS | II | Severe Injuries, Major aircraft Damage | $10^{-7}$/hr |
| MAJOR | III | Minor injury, minor aircraft or system damage | $10^{-5}$/hr |
| MINOR | IV | Less than minor injury, less than minor aircraft or system damage | $10^{-3}$/hr |
| NO EFFECT | V | No change to operational capability | $10^{-2}$/hr |







## Hazard Categories for Civil Aircraft

| Frequency of Occurrence | Level | Specific Item | Fleet or Inventory | Failure Probability per Flight Hour |
|---|---|---|---|---|
| Frequent | A | Likely to occur frequently | Continuously experienced | $\geq 1 \times 10^{-3}$ |
| Reasonably Probable | B | Will occur several times in the life of each item | Will occur frequently | $< 1 \times 10^{-3}$ to $\geq 1 \times 10^{-5}$ |
| Remote | C | Unlikely but possible to occur in the life of an item | Unlikely but can reasonably be expected to occur | $< 1 \times 10^{-5}$ to $\geq 1 \times 10^{-7}$ |
| Extremely Remote | D | So unlikely it can be assumed that the occurrence may not be experienced | Unlikely to occur, but possible | $< 10^{-7}$ to $\geq 1 \times 10^{-9}$ |
| Extremely Improbable | E | Should never happen in the life of all the items in the fleet | Not expected to occur during life of all aircraft of this type | $<1 \times 10^{-9}$ |



Risk from lightning is 5 x $10^{-7}$ deaths per person year





## Hazard Risk Index

| Probability | Severity Classification | | | |
|---|---|---|---|---|
| | Catastrophic | Hazardous | Major | Minor |
| Frequent | 1 | 3 | 7 | 13 |
| Reasonably Probable | 2 | 5 | 9 | 16 |
| Remote | 4 | 6 | 11 | 18 |
| Extremely Remote | 8 | 10 | 14 | 19 |
| Extremely Improbable | 12 | 15 | 17 | 20 |

- Acceptable - only ALARP actions considered
- Acceptable - use ALARP principle and consider further investigations
- Not acceptable - risk reducing measures required

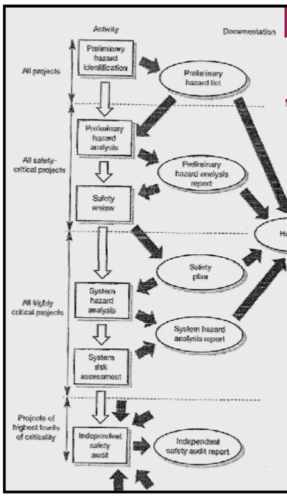


1918 TALLINNA TEHNIKAÜLIKOOL TALLINN UNIVERSITY OF TECHNOLOGY
Department of computer Engineering
ati.ttu.ee

# Hazards

### Hazard Analysis






## Hazard analysis lifecycle

- Preliminary hazard identification: is the system **safety related**?
- Preliminary hazard analysis: determine the **integrity levels** of each major function
- Safety plan: how will the safety be achieved and who is responsible (staff names!)
- System **hazard analysis**: FMEA, HAZOP, event trees, fault trees, reliability block diagrams, Markov modeling

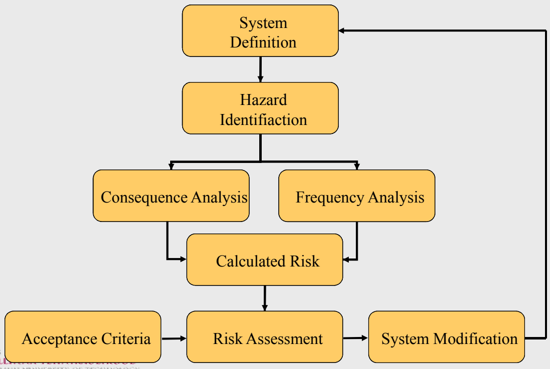




## Hazard Analysis

- The purpose
  - Identify events that may lead to accidents
  - Determine impact on system
  - Performed throughout the life cycle
- Analytical Techniques
  - Failure modes and effects analysis (FMEA)
  - FMECA: Failure modes, effects and criticality analysis (FMECA)
  - ETA: Event tree analysis (ETA)
  - FTA: Fault tree analysis (FTA)
  - HAZOP: Hazard and operability studies (HAZOP)
- Standards

## Hazard and Risk Analysis Process








## Preliminary Hazard Identification

- First activity in safety process, performed during early requirements analysis (concept definition)
- Identifies potential hazard sources and accidents
- Sources of information include
  - system concept and operational environment
  - incident data of previous in-service operation and similar systems
  - technology and domain specific analyses and checklists
- Method is group-based and dependent on experience
- Process is largely informal
- Output is Preliminary Hazard List





## Preliminary Hazard Analysis

- Refines hazards and accidents based on design proposal
- Performed using a system model that defines
  - scope and boundary of system
  - operating modes
  - system inputs, outputs and functions
  - preliminary internal structure
- Techniques for Preliminary Hazard Analysis include
  - Hazard and Operability Studies
  - Functional Failure Analysis
- Output is initial Hazard Log



1918 TALLINNA TEHNIKAÜLIKOOL TALLINN UNIVERSITY OF TECHNOLOGY

Department of computer Engineering
ati.ttu.ee

# Hazard Analysis

**Failure Mode and Effects Analysis (FMEA)**

**Failure Modes, Effects and Criticality Analysis (FMECA)**



## Failure Mode and Effects Analysis

- **Failure modes and effects analysis** (FEMA) considers the failure of any component within a system and tracks the effects of this failure to determine its ultimate consequences.
  - Probably the most commonly used technique
  - Looks for consequences of component failures (forward chaining technique)





## FMEA

- Manual analysis
  - Identify component, module or system failures
  - Determine consequences
  - Performed bottom-up
- Outputs
  - Spreadsheet noting each
    - failure mode
    - possible causes
    - consequences
    - possible remedies
  - Usually computer records kept
- Standardised by IEC (International Electrotechnical Commission)

## FMEA

- Notes
  - Can be applied at any stage of the design process and at any level within the system
  - Teams of four to eight engineers
- Limitations:
  - Lot of unnecessary work, it considers all components/failure modes
  - Requires expert knowledge to decide what to analyze
  - Usually do not consider multiple failures





## FMEA Example

FMEA for a microswitch

| *Ref No.* | *Unit* | *Failure mode* | *Possible cause* | *Local effects* | *System effects* | *Remedial action* |
|---|---|---|---|---|---|---|
| 1 | Tool guard switch | Open-circuit contacts | (a) faulty component<br>(b) excessive current<br>(c) extreme temperature | Failure to detect tool guard in place | Prevents use of machine – system fails safe | Select switch for high reliability and low probability of dangerous failure<br>Rigid quality control on switch procurement |
| 2 | | Short-circuit contacts | (a) faulty component<br>(b) excessive current | System incorrectly senses guard to be closed | Allows machine to be used when guard is absent – dangerous failure | Modify software to detect switch failure and take appropriate action |
| 3 | | Excessive switch-bounce | (a) ageing effects<br>(b) prolonged high currents | Slight delay in sensing state of guard | Negligible | Ensure hardware design prevents excessive current through switch |





## Failure Modes, Effects and Criticality Analysis

- FMECA:
  - Extension to FMEA
  - Takes into account importance of each component
  - Determines probability/frequency of occurrence of failures
- Problems
  - Measuring reliability of components difficult
  - Models often too simplistic
  - Tool support needed
- Used as input to fault tree analysis
  - Standardised





## Background

- FMECA was one of the first systematic techniques for failure analysis
- FMECA was developed by the U.S. Military. The first guideline was Military Procedure MIL-P-1629 "Procedures for performing a failure mode, effects and criticality analysis" dated November 9, 1949
- FMECA is the most widely used reliability analysis technique in the initial stages of product/system development
- FMECA is usually performed during the conceptual and initial design phases of the system in order to assure that all potential failure modes have been considered and the proper provisions have been made to eliminate these failures





## What can FMECA be used for?

- Assist in selecting design alternatives with high reliability and high safety potential during the early design phases
- Ensure that all conceivable failure modes and their effects on operational success of the system have been considered
- List potential failures and identify the severity of their effects
- Develop early criteria for test planning and requirements for test equipment
- Provide historical documentation for future reference to aid in analysis of field failures and consideration of design changes
- Provide a basis for maintenance planning
- Provide a basis for quantitative reliability and availability analyses.





## Types of FMECA

- **Design FMECA** is carried out to eliminate failures during equipment design, taking into account all types of failures during the whole life-span of the equipment
- **Process FMECA** is focused on problems stemming from how the equipment is manufactured, maintained or operated
- **System FMECA** looks for potential problems and bottlenecks in larger processes, such as entire production lines

## FME(C)A Chart

| Failure Modes and Effect Analysis | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Product Name: DeWalt Tradesman Drill | | | | Part name: Rear Vent | | | | |
| Function | Failure Mode | Effects of Failure | Causes of Failure | Current Controls | S | O | D | RPN |
| Allow Additional Air Flow | Filter Blocked | Overheated Motor | User Error | Visual Inspection | 4 | 1 | 5 | 20 |
| Prevent Dangerous Usage | Filter Not In Place | Larger Opening to Motor | User Error | Visual Inspection | 8 | 4 | 1 | 32 |
| Filter dust | Defective Filter | Additional dust flows into shell | Poor Materials | Visual Inspection | 1 | 1 | 7 | 7 |

S = Severity rating (1 to 10)
O = Occurrence frequency (1 to 10)
D = Detection Rating (1 to 10)
RPN = Risk Priority Number (1 to 1000)





## Severity Rating

| *Rank* | *Severity class* | *Description* |
|---|---|---|
| 10 | Catastrophic | Failure results in major injury or death of personnel. |
| 7-9 | Critical | Failure results in minor injury to personnel, personnel exposure to harmful chemicals or radiation, or fire or a release of chemical to the environment. |
| 4-6 | Major | Failure results in a low level of exposure to personnel, or activates facility alarm system. |
| 1-3 | Minor | Failure results in minor system damage but does not cause injury to personnel, allow any kind of exposure to operational or service personnel or allow any release of chemicals into the environment |





## Detection Rating

| *Rank* | *Description* |
|---|---|
| 1-2 | Very high probability that the defect will be detected. Verification and/or controls will almost certainly detect the existence of a deficiency or defect. |
| 3-4 | High probability that the defect will be detected. Verification and/or controls have a good chance of detecting the existence of a deficiency/defect. |
| 5-7 | Moderate probability that the defect will be detected. Verification and/or controls are likely to detect the existence of a deficiency or defect. |
| 8-9 | Low probability that the defect will be detected. Verification and/or control not likely to detect the existence of a deficiency or defect. |
| 10 | Very low (or zero) probability that the defect will be detected. Verification and/or controls will not or cannot detect the existence of a deficiency/defect. |





## Risk Ranking

- ✓ Risk Matrix
- ✓ Risk Ranking:
  - O = the rank of the occurrence of the failure mode
  - S = the rank of the severity of the failure mode
  - D = the rank of the likelihood the the failure will be detected before the system reaches the end-user/customer.
  - All ranks are given on a scale from 1 to 10. The risk priority
  - number (RPN) is defined as
    - RPN = S × O × D
  - The smaller the RPN the better – and – the larger the worse.



1918 TALLINNA TEHNIKAÜLIKOOL TALLINN UNIVERSITY OF TECHNOLOGY
Department of computer Engineering
ati.ttu.ee

# Hazard Analysis

**Hazard & Operability Analysis (HAZOP)**



## Hazard & Operability Analysis

- ✓ HAZOP:
  - Developed in Chemical industry
  - Applied successfully in other domains
  - “What if” analysis for system parameters
  - E.g., suppose “temperature” of “reactor” “rises”, what happens to system?
  - System realization of perturbation or sensitivity analysis
  - Requires flow model of operating plant

## Hazard & Operability Analysis

- Flowing items are “entities”
- Entities have characteristic properties known as “attributes”
- Analysis based on possible deviations of attribute values
- “Guide words” used to guide the analysis— designed to capture dimensions of variation
- Supplementary adjectives add temporal element
- Different word sets for different applications





## HAZOP examples

- Guide words:
  - no, more, less, early, late, before, ...

  Interpretation examples:
  - Signal arrives too late
  - Incomplete data transmitted / only part of the intended activity occurs
- Attributes:
  - Data flow, data rate, response time, ...





## HAZOP guide word interpretations

| *Guide word* | *Chemical plant* | *Computer-based system* |
|---|---|---|
| No | No part of the intended result is achieved | No data or control signal exchanged |
| More | A quantitative increase in the physical quantity | A signal magnitude or a data rate is too high |
| Less | A quantitative decrease in the physical quantity | A signal magnitude or a data rate is too low |
| As well as | The intended activity occurs, but with additional results | Redundant data sent in addition to intended value |
| Part of | Only part of the intended activity occurs | Incomplete data transmitted |
| Reverse | The opposite of what was intended occurs, for example reverse flow within a pipe | Polarity of magnitude changes reversed |
| Other than | No part of the intended activity occurs, and something else happens instead | Data complete but incorrect |
| Early | Not used | Signal arrives too early with reference to clock time |
| Late | Not used | Signal arrives too late with reference to clock time |
| Before | Not used | Signal arrives earlier than intended within a sequence |
| After | Not used | Signal arrives later than intended within a sequence |





## HAZOP attributes

| *Attribute* | *Guide word* | *Possible meaning* |
|---|---|---|
| Data flow | More | More data is passed than expected |
| | Less | Less data is passed than expected |
| Data rate | More | The data rate is too high |
| | Less | The data rate is too low |
| Data value | More | The data value is too high |
| | Less | The data value is too low |
| Repetition time | More | The time between output updates is too high |
| | Less | The time between output updates is too low |
| Response time | More | The response time is longer than required |
| | Less | The response time is shorter than required |





## HAZOP Example

| *Item* | *Inter-connection* | *Attribute* | *Guide word* | *Cause* | *Consequence* | *Recommendation* |
|---|---|---|---|---|---|---|
| 1 | Sensor supply line | Supply voltage | No | PSU, regulator or cable fault | Lack of sensor signal detected and system shuts down | |
| 2 | | | More | Regulator fault | Possible damage to sensor | Consider overvoltage protection |
| 3 | | | Less | PSU or regulator fault | Incorrect temperature reading | Include voltage monitoring |
| 4 | | Sensor current | More | Sensor fault | Incorrect temperature reading, possible loading of supply | Monitor supply current |
| 5 | | | Less | Sensor fault | Incorrect temperature reading | As above |
| 6 | Sensor output | Voltage | No | PSU, sensor or cable fault | Lack of sensor signal detected and system shuts down | |
| 7 | | | More | Sensor fault | Temperature reading too high – results in decrease in plant efficiency | Consider use of duplicate sensor |
| 8 | | | Less | Sensor mounted incorrectly or sensor failure | Temperature reading too low – could result in overheating and possible plant failure | As above |



1918 TALLINNA TEHNIKAÜLIKOOL TALLINN UNIVERSITY OF TECHNOLOGY
Department of computer Engineering
ati.ttu.ee

# Hazard Analysis

**Fault Tree Analysis (FTA)**

## Fault Tree Analysis

- Fault tree analysis (FTA) is a top-down approach to failure analysis, starting with a potential undesirable event (accident) called a TOP event, and then determining all the ways it can happen.
- The analysis proceeds by determining how the TOP event can be caused by individual or combined lower level failures or events.
- The causes of the TOP event are “connected” through logic gates
- FTA is the most commonly used technique for causal analysis in risk and reliability studies.





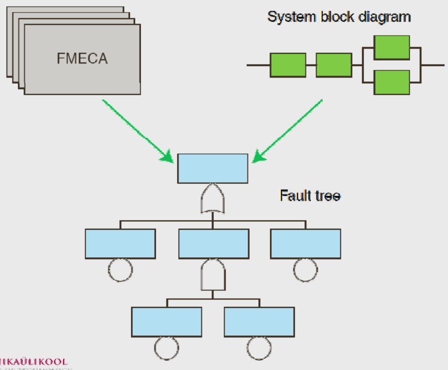
## History

- FTA was first used by Bell Telephone Laboratories in connection with the safety analysis of the Minuteman missile launch control system in 1962
- Technique improved by Boeing Company
- Extensively used and extended during the Reactor safety study (WASH 1400)





## Preparations for FTA








## Boundary Conditions

- The physical boundaries of the system (Which parts of the system are included in the analysis, and which parts are not?)
- The initial conditions (What is the operational stat of the system when the TOP event is occurring?)
- Boundary conditions with respect to external stresses (What type of external stresses should be included in the analysis – war, sabotage, earthquake, lightning, etc?)
- The level of resolution (How detailed should the analysis be?)





## Fault Tree Construction

- Define the TOP event in a clear and unambiguous way.
  Should always answer:
  What e.g., “Fire”
  Where e.g., “in the process oxidation reactor”
  When e.g., “during normal operation”
- What are the immediate, necessary, and sufficient events and conditions causing the TOP event?
- Connect via a logic gate
- Proceed in this way to an appropriate level (= basic events)
- Appropriate level:
  - Independent basic events
  - Events for which we have failure data

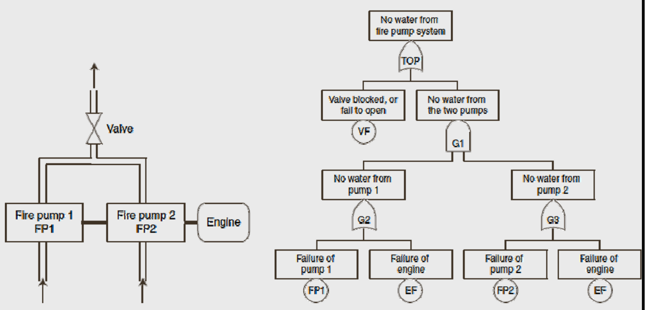




## Fault Tree Symbols

| | | |
|---|---|---|
| Logic gates | OR-gate | The OR-gate indicates that the output event occurs if any of the input events occur |
| | AND-gate | The AND-gate indicates that the output event occurs only if all the input events occur at the same time |
| Input events (states) | | The basic event represents a basic equipment failure that requires no further development of failure causes |
| | | The undeveloped event represents an event that is not examined further because information is unavailable or because its consequences are insignificant |
| Description of state | | The comment rectangle is for supplementary information |
| Transfer symbols | Transfer out<br>Transfer in | The transfer-out symbol indicates that the fault tree is developed further at the occurrence of the corresponding transfer-in symbol |

## Fault Tree Example







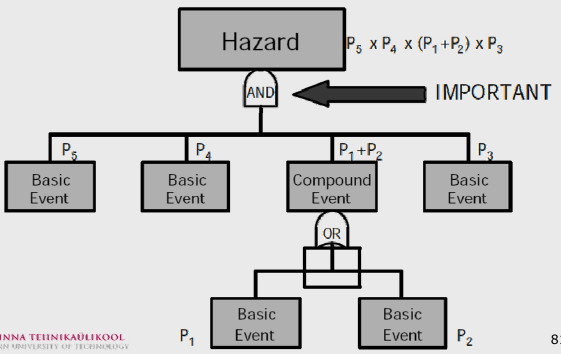

## Elementary Fault Tree Analysis

- ✓ Assignment of probabilities to specific events
- ✓ Computation of probabilities for compound events
- ✓ Sophisticated dependability analysis possible
- ✓ Extensive, elaborate, established technique
- ✓ Provides:
  - Mechanism for showing that design will meet dependability requirements





## Fault Trees and Probabilities








## Practical Fault Trees

- ✓ Developed by human analysis
- ✓ Tend to be *very* large for real systems
- ✓ Evolve as insight is gained
- ✓ Many analysis techniques possible:
  - Hazard probability can be calculated if probabilities associated with all basic events
  - Tables of probabilities available for degradation faults for common components
  - Recall, infeasible for design faults



1918 TALLINNA TEHNIKAÜLIKOOL TALLINN UNIVERSITY OF TECHNOLOGY
Department of computer Engineering
ati.ttu.ee

# Hazard Analysis

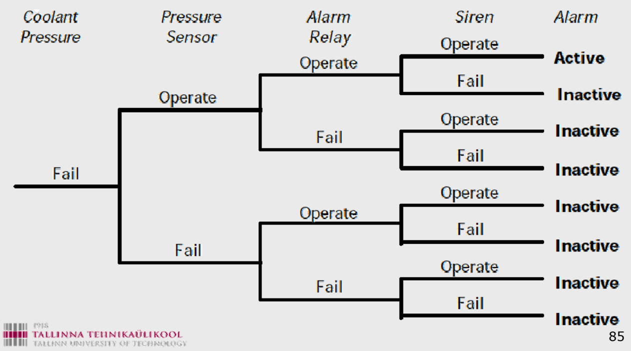
**Event Tree Analysis (ETA)**



## Event Trees

- ✓ Event sequences that follow from some initial event of interest, usually a component failure
- ✓ Downstream events follow from original event and subsequent events of other components
- ✓ E.g. Chemical plant pressure sensor sounds siren when pressure drops to unsafe level

## Event Tree






© Gert Jervan, TTÜ/ATI IAF0530 - Süsteemide usaldusväärsus ja veakindlus

## Barriers

- Most well designed systems have one or more barriers that are implemented to stop or reduce the consequences of potential accidental events. The probability that an accidental event will lead to unwanted consequences will therefore depend on whether these barriers are functioning or not.
- The consequences may also depend on additional events and factors. Examples include:
  - Whether a gas release is ignited or not
  - Whether or not there are people present when the accidental event occurs
  - The wind direction when the accidental event occurs
- Barriers may be technical and/or administrative (organizational).



© Gert Jervan, TTÜ/ATI IAF0530 - Süsteemide usaldusväärsus ja veakindlus

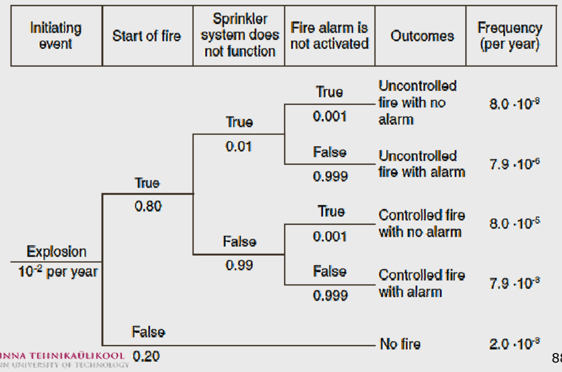
## Event Tree Analysis

- An event tree analysis (ETA) is an inductive procedure that shows all possible outcomes resulting from an accidental (initiating) event, taking into account whether installed safety barriers are functioning or not, and additional events and factors.
- By studying all relevant accidental events (that have been identified by a preliminary hazard analysis, a HAZOP, or some other technique), the ETA can be used to identify all potential accident scenarios and sequences in a complex system.
- Design and procedural weaknesses can be identified, and probabilities of the various outcomes from an accidental event can be determined.



© Gert Jervan, TTÜ/ATI IAF0530 - Süsteemide usaldusväärsus ja veakindlus

## ETA Example






© Gert Jervan, TTÜ/ATI IAF0530 - Süsteemide usaldusväärsus ja veakindlus

## ETA Pros and Cons

- Positive
  - Visualize event chains following an accidental event
  - Visualize barriers and sequence of activation
  - Good basis for evaluating the need for new / improved procedures and safety functions
- Negative
  - No standard for the graphical representation of the event tree
  - Only one initiating event can be studied in each analysis
  - Easy to overlook subtle system dependencies
  - Not well suited for handling common cause failures in the quantitative analyses
  - The event tree does not show acts of omission



© Gert Jervan, TTÜ/ATI IAF0530 - Süsteemide usaldusväärsus ja veakindlus

## Hazard Analysis in the Life Cycle

- FME(C)A
  - Used to generate event trees and fault trees
- FME(C)A, FTA, ETA
  - Appropriate when functional design complete
- Preliminary HAZOP
  - Early in the life-cycle
  - Identify hazards, take account of them in the design
- Full HAZOP
  - Later in the life-cycle
  - Identify further hazards, feed back into design design

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Department of computer Engineering
ati.ttu.ee

# Risk Analysis



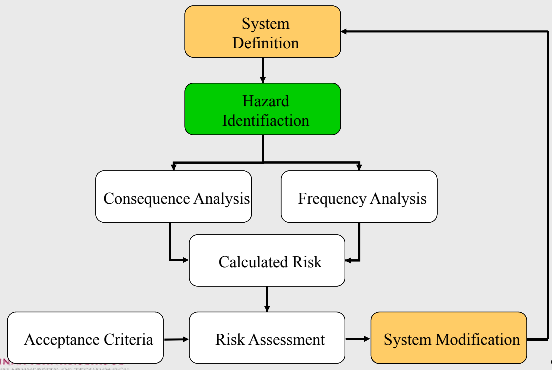
## Risk Analysis

- The purpose
  - Associate risk with given hazards
    - Consequence of malfunction - severity
    - Probability of malfunction – frequency
  - Ensure nature of risks is well understood
  - Ensure safety targets can be set and evaluated
- Techniques
  - Quantitative
  - Qualitative, risk classification
  - Integrity classification
  - Safety Integrity Levels (SILs)
  - ALARP
- Standards
  - IEC 1508, IEC 61508





## Hazard and Risk Analysis Process







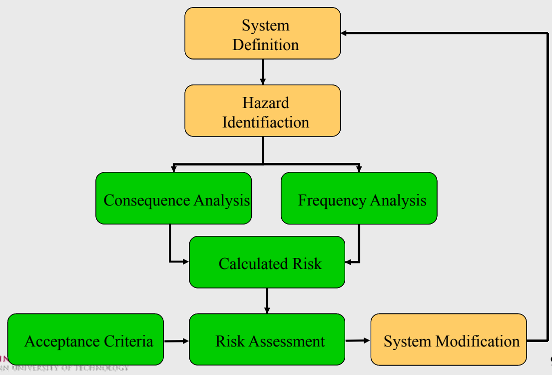

## Flashback

- A Hazard is a system state that could lead to:
  - Loss of life
  - Loss of property
  - Release of energy
  - Release of dangerous materials
- Hazards are the *states* we have to avoid
- An accident is a loss event:
  - System in hazard state, ***and***
  - Change in the operating environment
- Classification
  - Severity
  - Nature





## Hazard and Risk Analysis Process








## Introduction

- Risk is associated with every hazard
  - Hazard is a potential danger
    - i.e. possibility of being struck by lightning
  - Associated risk
- *Accident is an unintended event or sequence of events that causes death, injury, environmental or material damage*

Storey 1996

## Introduction

- Hazard analysis identifies accident scenarios: sequences of events that lead to an accident
- *Risk is a combination of the* ***severity*** *of a specified hazardous event with its* ***probability*** *of occurence over a specified* ***duration***
  - Qualitative or quantitative





## Risk Calculation

- Quantify probability/frequency of occurence:
  - number of events per hour/year of operation
  - number of events per lifetime
  - number of failures on demand
- Ex 1:
  - Failure of a particular component results in explosion that could kill 100 people. Estimate that component will fail once every 10,000 years

    1 failure per 10,000 years = 0.0001 failures per year

    **Risk** = penalty x (probability per year)

    = 100 x (0.0001)

    = 0.01 deaths per year





## Risk Calculation

- Ex 2:
  - Country with population of 50,000,000
  - Approx. 25 people are each year killed by lightning i.e. $25/50{,}000{,}000=5\text{x}10^{-7}$
  - Risk:
    - every individual has proabability of $5\text{x}10^{-7}$ to be killed by lightning at any given year
    - Population is exposed to risk of $5\text{x}10^{-7}$ deaths per person year
- Qualitative:
  - intolerable, undesirable, tolerable





## Levels of Fatal Risk

| **Risk** | **Chance per million** |
|---|---|
| Risk of being killed by a falling aircraft | 0.02 cpm |
| Risk of death by lightening | 0.1 cpm |
| Risk of being killed by an insect or snake bite | 0.1 cpm |
| Risk of death in a fire caused by a cooking appliance in the home | 1 cpm |
| Risk of death in an accident at work in the very safest parts of industry | 10 cpm |
| General risk of death in a traffic accident | 100 cpm |
| Risk of death in high risk groups within relatively risky industries such as mining | 1,000 cpm |
| Risk of fatality from smoking 20 cigarettes per day | 5,000 cpm |
| Risk of death from 5 hours of solo rock climbing every weekend | 10,000 cpm |





## The Need for Safety Targets

- Learning from mistakes is not longer acceptable
  - Disaster, review, recommendation
- Probability estimates
  - Are coarse
  - Meaning depends on duration, low/high demand, but often stated without units
- Need rigour and guidance for safety related systems
  - Standards (HSE, IEC)
  - Ensure risk reduction, not cost reduction
  - For risk assessment
  - For evaluation of designs





## Quantitative Risk Assessment

- How it works
  - Predict frequency of hardware failures
  - Compare with tolerable risk target
  - If not satisfied, modify the design
- Example
  - The probability that airbag fails when activated
  - The frequency of the interconnecting switch failing per lifetime
- Even if target met by random hardware failure
  - Hardware could have embedded software, potential for systemic failure
  - Engineer's judgment called for in IEC 61508

    (IEC 61508 – Functional Safety – www.iec.ch)

## Quantitative risk assessment

- Quantify probability/frequency of occurence:
  - number of events per hour/year of operation
  - number of events per lifetime
  - number of failures on demand
- Example:
  - Failure of a particular component results in explosion that could kill 100 people. Estimate that component will fail once every 10,000 years
    1 failure per 10,000 years = 0.0001 failures per year
    **Risk** = penalty x (probability per year)
    = 100 x (0.0001)
    = 0.01 deaths per year





## Qualitative Risk Assessment

- When cannot estimate the probability
- How it works
  - Classify risk into risk classes
  - Define tolerable/intolerable risks
  - Define tolerable/intolerable frequencies
  - Set standards and processes for evaluation and minimization of risks
- Example
  - Catastrophic, multiple deaths
  - Critical, single death
  - Marginal, single severe injury
  - Negligible, single minor injury
- Aims to deal with systemic failures





## Risk Management

| Risk | | Probability | | | | |
|---|---|---|---|---|---|---|
| | | **Very High** | **High** | **Medium** | **Low** | **Very Low** |
| **Conse-quence** | **Very High** | Very High | Very High | High | High | Medium |
| | **High** | Very High | High | Medium | Medium | Low |
| | **Medium** | High | Medium | Medium | Low | Low |
| | **Low** | High | Medium | Low | Low | Very Low |
| | **Very Low** | Medium | Low | Low | Very Low | Very Low |

Risk Ranking table





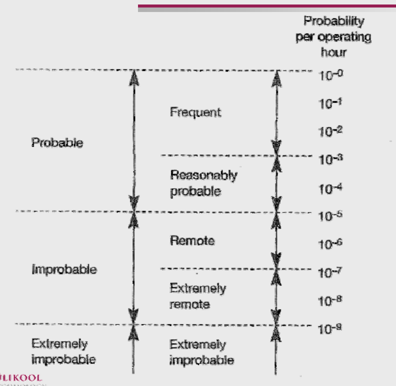
## Hazard Severity Categories for Civil Aircraft

| *Category* | *Definition* |
|---|---|
| Catastrophic | Failure condition which would prevent continued safe flight and landing |
| Hazardous | Failure conditions which would reduce the capability of the aircraft or the ability of the crew to cope with adverse operating conditions, to the extent that there would be: (1) a large reduction in safety margins or functional capabilities (2) physical distress or higher workload such that the flight crew could not be relied on to perform their tasks accurately or completely (3) adverse effects on occupants, including serious or potentially fatal injuries to a small number of those occupants |
| Major | Failure conditions which would reduce the capability of the aircraft or the ability of the crew to cope with adverse operating conditions to the extent that there would be, for example, a significant reduction in safety margins or functional capabilities, a significant increase in crew workload or in conditions impairing crew efficiency, or discomfort to occupants, possibly including injuries |
| Minor | Failure conditions which would not significantly reduce aircraft safety, and which would involve crew actions that are well within their capabilities. Minor failure conditions may include, for example, a slight reduction in safety margins or functional capabilities, a slight increase in crew workload, such as routine flight plan changes, or some inconvenience to occupants |
| No effect | Failure conditions which do not affect the operational capability of the aircraft or increase crew workload |





## Hazard Probability Classes for Aircraft Systems








## Risk Management Advice

- Identify risks and track them
  - Avoid "unknown" risks at all costs!
- Approaches to risk
  - Mitigate, i.e. perform risk reduction
    - E.g. solve the problem, obtain insurance, etc
  - Avoid
    - Use a less risky approach - not always possible
  - Accept
    - Decide that expected cost is not worth reducing further
    - Often sensible choice
- Ignore
  - Proceed ahead blindly – uninformed acceptance

## Acceptability of Risk

- Acceptability of risk is a complex issue involving
  - social factors, e.g., value of life and limb
  - legal factors, e.g., responsibility of risk
  - economic factors, e.g., cost of risk reduction
- Ideally these tasks are performed by policy makers, not engineers!
- Engineers provide the information on which such complex decisions can be made
- At beginning of project, accurate estimates of risks and costs are difficult to achieve





## Acceptability of risk

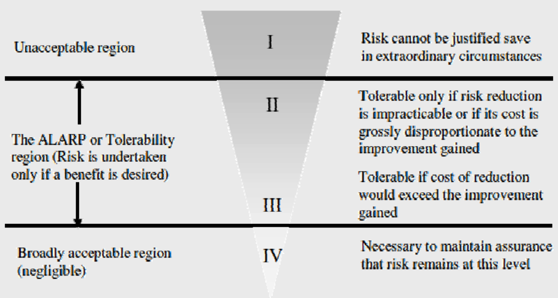
- Ethical considerations
  - Determining risk and its acceptability involves moral judgement
  - Society's view not determined by logical rules
  - Perception that accidents involving large numbers of deaths are perceived as more serious than smaller accidents, though they may occur less frequently





## Risk Reduction - ALARP

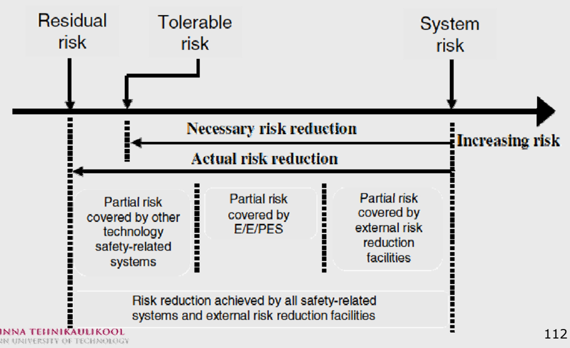
As Low As Reasonably Practicable







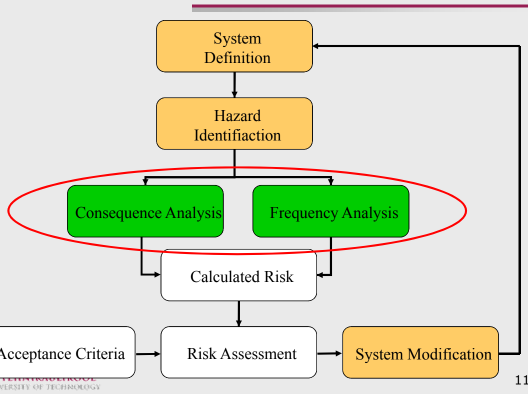

## Risk Reduction







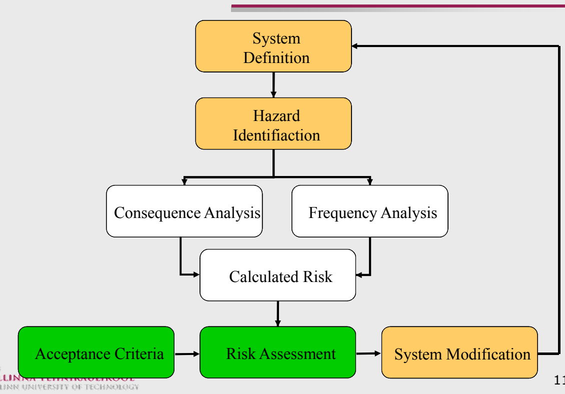

## Hazard and Risk Analysis Process








## Hazard and Risk Analysis Process



## Safety Requirements

- Once hazards are identified and assessed, safety requirements are generated to mitigate the risk
- Requirements may be
  - primary: prevent initiation of hazard
    - eliminate hazard
    - reduce hazard
  - secondary: control initiation of hazard
    - detect and protect
    - warn
- Safety requirements form basis for subsequent development





## Safety Integrity

- Safety integrity, defined by
  - Likelihood of a safety-related system satisfactorily performing the required safety functions under all stated conditions within a stated period of time
  - Hardware integrity, relating to random faults
  - Systematic integrity, relating to dangerous systematic faults
- Expressed
  - Quantitatively, or
  - As Safety Integrity Levels (SILs)
- Standards, IEC 1508, 61508
  - Define target failure rates for each level
  - Define processes to manage design & development
- Aims to deal with systemic failures





## Safety Integrity Levels (SILs)

- Tolerable failure frequency are often characterised by Safety Integrity Levels rather than likelihoods
  - SILs are a qualitative measure of the required protection against failure
- SILs are assigned to the safety requirements in accordance with target risk reduction
- Once defined, SILs are used to determine what methods and techniques should be applied (or not applied) in order to achieve the required integrity level
- Point of translation from failure frequencies to SILs may vary





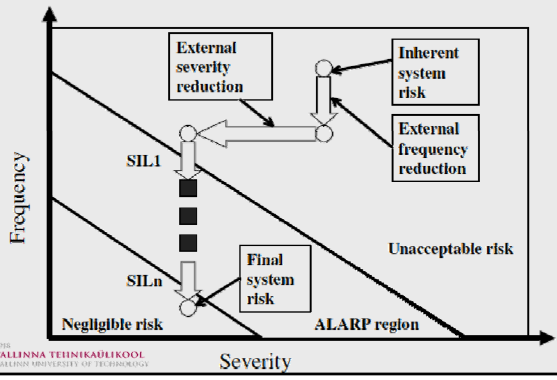
## Automotive SIL

- Uncontrollable (SIL 4), critical failure
  - No driver expected to recover (e.g. both brakes fail), extremely severe outcomes (multiple crash)
- Difficult to control (SIL 3), critical failure
  - Good driver can recover (e.g. one brake works, severe outcomes (fatal crash)
- Debilitating (SIL 2)
  - Ordinary driver can recover most of the time, usually no severe outcome
- Distracting (SIL 1)
  - Operational limitations, but minor problem
- Nuisance (SIL 0)
  - Safety is not an issue, customer satisfaction is





## Risk & SILs








## IEC 61508 Standard

- New main standard for software safety
- Can be tailored to different domains (automotive, chemical, etc)
- Comprehensive
- Includes SILs, including failure rates
- Covers recommended techniques

- IEC = International Electrotechnical Commission

- E/E/PES = electrical/electronic/programmable electronic safety related systems

## Safety-Integrity Table of IEC 61508

| Safety Integrity Level | Low demand mode of operation (Average probability of failure to perform its design function on demand) | |
|---|---|---|
| 4 | $\geq 10^{-5}$ to $< 10^{-4}$ | (> 99.99 % reliable) |
| 3 | $\geq 10^{-4}$ to $< 10^{-3}$ | (> 99.9 % reliable) |
| 2 | $\geq 10^{-3}$ to $< 10^{-2}$ | (> 99% reliable) |
| 1 | $\geq 10^{-2}$ to $< 10^{-1}$ | (> 90% reliable) |

| Safety Integrity Level | High demand mode or continuous mode of operation (Probability of dangerous failure per hour) |
|---|---|
| 4 | $\geq 10^{-9}$ to $< 10^{-8}$ |
| 3 | $\geq 10^{-8}$ to $< 10^{-7}$ |
| 2 | $\geq 10^{-7}$ to $< 10^{-6}$ |
| 1 | $\geq 10^{-6}$ to $< 10^{-5}$ |

- The higher the SIL, the harder to meet the standard
- High demand for e.g. car brakes, critical boundary SIL 3
- Low demand for e.g. airbag, critical boundary is SIL 3, one failure in 1000 activations





## SILs

- SILs 3 and 4 are critical
- SIL activities at lower levels may be needed
- SIL 1
  - Relatively easy to achieve, if ISO 9001 practices apply,
- SIL 2
  - Not dramatically harder than SIL 1, but involves more review and test, and hence cost
- SIL 3
  - Substantial increment of effort and cost
- SIL 4
  - Includes state of the art practices such as formal methods and verification, cost extremely high





## Techniques and Measures

| Clause 7.7 : Software Safety Validation | | | | | |
|---|---|---|---|---|---|
| **TECHNIQUE/MEASURE** | **Ref** | **SIL1** | **SIL2** | **SIL3** | **SIL4** |
| 1. Probabilistic Testing | B.47 | -- | R | R | HR |
| 2. Simulation/Modelling | D.6 | R | R | HR | HR |
| 3. Functional and Black-Box Testing | D.3 | HR | HR | HR | HR |
| NOTE: One or more of these techniques shall be selected to satisfy the safety integrity level being used. | | | | | |

- Implementing the recommended techniques and measures should result in software of the associated integrity level.
- For example, if the software was required to be validated to be of Integrity level 3, Simulation and Modelling are Highly Recommended Practices, as is Functional and Black-Box Testing.





## Detailed Techniques and Measures

- Related to certain entries in these tables are additional, more detailed sets of recommendations structured in the same manner. These address techniques and measures for:
  - Design and Coding Standards
  - Dynamic analysis and testing
  - Approaches to functional or black-box testing
  - Hazard Analysis
  - Choice of programming language
  - Modelling
  - Performance testing
  - Semi-formal methods
  - Static analysis
  - Modular approaches





## Modeling

| D.6 : Modelling Referenced by Clauses 7.6 | | | | | |
|---|---|---|---|---|---|
| **TECHNIQUE/MEASURE** | **Ref** | **SIL1** | **SIL2** | **SIL3** | **SIL4** |
| 1. Data Flow Diagrams | B.12 | R | R | R | R |
| 2. Finite State Machines | B.29 | -- | HR | HR | HR |
| 3. Formal Methods | B.30 | -- | R | R | HR |
| 4. Performance Modelling | B.45 | R | R | R | HR |
| 5. Time Petri Nets | B.64 | -- | HR | HR | HR |
| 6. Prototyping/Animation | B.49 | R | R | R | R |
| 7. Structure Diagrams | B.59 | R | R | R | HR |
| NOTE: One or more of the above techniques should be used. | | | | | |





## SILs

- What does it all mean?
  - SIL 4 system should have a duration of about $10^{-9}$ hours between critical failures
  - If established SIL 4 needed, used all the techniques…
  - But there is no measurement that the results actually achieves the target
  - Standard assumes that you are competent in all methods and apply everything possible
  - Except that these may be insufficient or not affordable

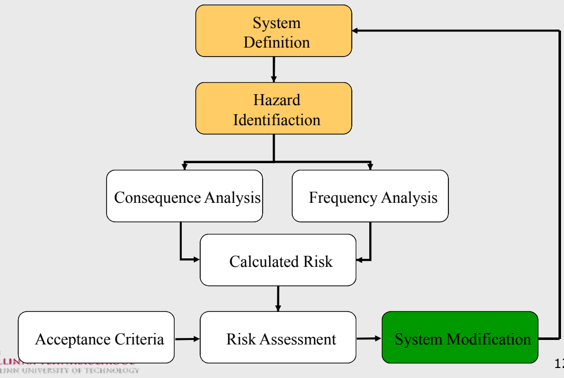
## The Engineering Council's Code of Practice on Risk Issues

| | | |
|---|---|---|
| 1 | Professional responsibility | Exercise reasonable professional skill and care |
| 2 | Law | Know about and comply with the law |
| 3 | Conduct | Act in accordance with the codes of conduct |
| 4 | Approach | Take a systematic approach to risk issues |
| 5 | Judgement | Use professional judgement and experience |
| 6 | Communication | Communicate within your organization |
| 7 | Management | Contribute effectively to corporate risk management |
| 8 | Evaluation | Assess the risk implications of alternatives |
| 9 | Professional development | Keep up to date by seeking education and training |
| 10 | Public awareness | Encourage public understanding of risk issues |





## Hazard and Risk Analysis Process








## Risk Reduction Procedures

- Four main categories of risk reduction strategies, given in the order that they should be applied:
  - Hazard Elimination
  - Hazard Reduction
  - Hazard Control
  - Damage Limitation
- Only an approximate categorisation, since many strategies belong in more than one category





## Hazard Ellimination

- Before considering safety devices, attempt to eliminate hazards altogether
  - use of different materials, e.g., non-toxic
  - use of different process, e.g., endothermic reaction
  - use of simple design
  - reduction of inventory, e.g., stockpiles in Bhopal
  - segregation, e.g., no level crossings
  - eliminate human errors, e.g., for assembly of system use colour coded connections





## Design Principles

- Familiar
  - use tried and trusted technologies, materials techniques
- Simple
  - testable (including controllable and observable)
  - portable (no use of sole manufacturer components compiler dependent features)
  - understandable (behaviour can easily be from implementation)
  - deterministic (use of resources is not random)
  - predictable (use of resources can be predicted)
  - minimal (extra features not provided)





## Design Principles (cont.)

- Structured design techniques
  - defined notation for describing behaviour
  - identification of system boundary and environment
  - problem decomposition
  - ease of review
- Design standards
  - limit complexity
  - increase modularity
- Implementation standards
  - presentation and naming conventions
  - semantic and syntactic restrictions in software

## Classes of System Failure

- Random (physical) failures
  - due to physical faults
  - e.g., wear-out, aging, corrosion
  - can be assigned quantitative failure probabilities
- Systematic (design) failures
  - due to faults in design and/or requirements
  - inevitably due to human error
  - usually measured by integrity levels
- Operator failures
  - due to human error
  - mix of random and systematic failures





## Nature of Random Failures

- Arise from random events generated during operation or manufacture
- Governed by the laws of physics and cannot be eliminated
- Modes of failure are limited and can be anticipated
- Failures occur independently in different components
- Failure rates are often predictable by statistical methods
- Sometimes exhibit graceful degradation
- Treatment is well understood





## Treating Random Failures

- Random failures cannot be eliminated and must be reduced or controlled
- Random failures can be mitigated by:
  - predicting failure modes and rates of components
  - applying redundancy to achieve overall reliability
  - performing preventative maintenance to replace components before faults arise
  - executing on-line or off-line diagnostic checks





## Nature of Systematic Failures

- Ultimately caused by human error during development, installation or maintenance
- Appear transient and random since they are triggered under unusual, random circumstances
- Systematic and will occur again if the required circumstances arise
- Failures of different components are *not* independent
- Difficult to predict mode of failure since the possible deviations in behaviour are large
- Difficult to predict the likelihood of occurrence





## Treating Systematic Failures

- In theory, design failures can be eliminated
- In practice, perfect design may be too costly
- Focus the effort on critical areas
  - identify safety requirements using hazard analysis
  - assess risk in system and operational context
- Eliminate or reduce errors using quality development processes
  - verify compliance with safety requirements
  - integrate and test against safety requirements





## Hazard Reduction

- Reduce the likelihood of hazards
- Use of barriers, physical or logical
  - Lock-ins
  - Lock-outs
  - Interlocks
- Failure minimization
  - Redundancy
  - Recovery

## Redundancy

- Hardware redundancy
  - Static redundancy, e.g. triple modular redundancy
  - Dynamic redundancy, e.g. standby spare
- Software redundancy, e.g. N-version programming
- Information redundancy, e.g., checksums, cyclic redundancy codes, error correcting codes





## Recovery

- Can reduce failures by recovering after error detected but before component or system failure occurs
- Recovery can only take place after detection of error
  - Backward recovery
  - Forward recovery





## Error Detection

- Based on check that is independent of implementation of the system
  - coding - parity checks and checksums
  - reasonableness - range and invariants
  - reversal - calculate square of square root
  - diagnostic - hardware built-in tests
  - timing - timeouts or watchdogs





## Error Detection (cont.)

- Timing of error detection important
  - early error detection can be used to prevent propagation
  - late error detection requires a check of the entire activity of system
- Checking may be in several forms
  - monitor, acting after a system function, checking outputs after production but before use
  - kernel, encapsulating (safety-critical) functions in a subsystem that allows all inputs to and outputs from the kernel to be checked





## Backward Recovery

- Corrects errors through reversing previous operations
- Return system to a previous known safe state
- Allows retry
- Requires checkpoints or saved states (and the expenses involved with producing them)
- Rollback usually impossible with real-time system





## Forward Recovery

- Corrects errors without reversing previous operations, finding safe (but possibly degraded) state for system
  - data repair, use redundancy in data to perform repairs
  - reconfiguration, use redundancy such as backup or alternate systems
  - coasting, continue operations ignoring (hopefully transient) errors
  - exception processing, only continue with selection of (safetycritical) functions
  - failsafe, achieve safe state and cease processing
    - use passive devices (e.g., deadman switch) instead of active devices (e.g., motor holding weight up)

## Hazard Control

- Detect and control hazard before damage occurs
- Reduce the level or duration of the hazard
- Hazard control mechanisms include:
  - Limiting exposure: reduce the amount of time that a system is in an unsafe state (e.g. don't leave rocket in armed state)
  - Isolation and containment
  - Fail safe design





## Damage Limitation

- In addition to eliminating hazards or employing safety devices, consider
  - warning devices
  - procedures
  - training
  - emergency planning
  - maintenance scheduling
  - protective measures





## Architectural Design

- Suitable architectures may allow a high integrity system to be built from lower integrity components
  - combinations of components must implement a safety function independently
  - overall likelihood of failure should be the same or less
  - be wary of common failure causes
- Apportionment approaches can be quantitative and/or qualitative
  - quantitative: numerical calculations
  - qualitative: judgement or rules of thumb





## Conclusions

- Hazards
- Hazard Analysis
- Risks
- Risk Analysis
- Risk Management
- Safety
- Risk Reduction



1918 TALLINNA TEHNIKAÜLIKOOL TALLINN UNIVERSITY OF TECHNOLOGY

Department of computer Engineering
ati.ttu.ee

**Questions?**