Lecture 10 – Critical Systems 2 (Sommerville Ch. 17)

Karl R. Wilcox
K.R.Wilcox@reading.ac.uk

Functional and non-functional requirements

- System functional requirements may be generated to define error checking and recovery facilities and features that provide protection against system failures.
- Non-functional requirements may be generated to specify the required reliability and availability of the system.

System reliability specification

- ***Hardware reliability***
  - What is the probability of a hardware component failing and how long does it take to repair that component?
- ***Software reliability***
  - How likely is it that a software component will produce an incorrect output. Software failures are different from hardware failures in that software does not wear out. It can continue in operation even after an incorrect result has been produced.
- ***Operator reliability***
  - How likely is it that the operator of a system will make an error?

System reliability engineering

- Sub-discipline of systems engineering that is concerned with making judgements on system reliability
- It takes into account the probabilities of failure of different components in the system and their combinations
  - Consider a system with 2 components A and B where the probability of failure of A is P (A) and the probability of failure of B is P (B).

Failure probabilities

- If there are 2 components and the operation of the system depends on both of them then the probability of system failure is
  - $P(S) = P(A) + P(B)$
- Therefore, as the number of components increase then the probability of system failure increases
- If components are replicated then the probability of failure is
  - $P(S) = P(A)^n$ (all components must fail)

Functional reliability requirements

- A predefined range for all values that are input by the operator shall be defined and the system shall check that all operator inputs fall within this predefined range.
- The system shall check all disks for bad blocks when it is initialised.
- The system must use N-version programming to implement the braking control system.
- The system must be implemented in a safe subset of Ada and checked using static analysis

Non-functional reliability specification

- The required level of system reliability required should be expressed in quantitatively
- Reliability is a dynamic system attribute- reliability specifications related to the source code are meaningless.
  - No more than N faults/1000 lines.
  - This is only useful for a post-delivery process analysis where you are trying to assess how good your development techniques are.
- An appropriate reliability metric should be chosen to specify the overall system reliability

Reliability metrics

- Reliability metrics are units of measurement of system reliability
- System reliability is measured by counting the number of operational failures and, where appropriate, relating these to the demands made on the system and the time that the system has been operational
- A long-term measurement programme is required to assess the reliability of critical systems

Reliability metrics

Metric	Explanation
POFOD Probability of failure on demand	The likelihood that the system will fail when a service request is made. For example, a POFOD of 0.001 means that 1 out of a thousand service requests may result in failure.
ROCOF Rate of failure occurrence	The frequency of occurrence with which unexpected behaviour is likely to occur. For example, a ROCOF of 2/100 means that 2 failures are likely to occur in each 100 operational time units. This metric is sometimes called the failure intensity.
MTTF Mean time to failure	The average time between observed system failures. For example, an MTTF of 500 means that 1 failure can be expected every 500 time units.
MTTR Mean time to repair	The average time between a system failure and the return of that system to service.
AVAIL Availability	The probability that the system is available for use at a given time. For example, an availability of 0.998 means that in every 1000 time units, the system is likely to be available for 998 of these.

Availability

- Measure of the fraction of the time that the system is available for use
- Takes repair and restart time into account
- Availability of 0.998 means software is available for 998 out of 1000 time units
- Relevant for non-stop, continuously running systems
  - telephone switching systems, railway signalling systems

Probability of failure on demand

- This is the probability that the system will fail when a service request is made. Useful when demands for service are intermittent and relatively infrequent
- Appropriate for protection systems where services are demanded occasionally and where there are serious consequence if the service is not delivered
- Relevant for many safety-critical systems with exception management components
  - Emergency shutdown system in a chemical plant

Rate of fault occurrence (ROCOF)

- Reflects the rate of occurrence of failure in the system
- ROCOF of 0.002 means 2 failures are likely in each 1000 operational time units e.g. 2 failures per 1000 hours of operation
- Relevant for operating systems, transaction processing systems where the system has to process a large number of similar requests that are relatively frequesnt
  - Credit card processing system, airline booking system

Mean time to failure

- Measure of the time between observed failures of the system. Is the reciprocal of ROCOF for stable systems
- MTTF of 500 means that the mean time between failures is 500 time units
- Relevant for systems with long transactions i.e. where system processing takes a long time. MTTF should be longer than transaction length
  - Computer-aided design systems where a designer will work on a design for several hours, word processor systems

Failure consequences

- Reliability measurements do NOT take the consequences of failure into account
- Transient faults may have no real consequences but other faults may cause data loss or corruption and loss of system service
- May be necessary to identify different failure classes and use different metrics for each of these. The reliability specification must be structured.

Failure consequences

- When specifying reliability, it is not just the number of system failures that matter but the consequences of these failures
- Failures that have serious consequences are clearly more damaging than those where repair and recovery is straightforward
- In some cases, therefore, different reliability specifications for different types of failure may be defined

Failure classification

Failure class	**Description**
Transient	Occurs only with certain inputs
Permanent	Occurs with all inputs
Recoverable	System can recover without operator intervention
Unrecoverable	Operator intervention needed to recover from failure
Non-corrupting	Failure does not corrupt system state or data
Corrupting	Failure corrupts system state or data

Steps to a reliability specification

- For each sub-system, analyse the consequences of possible system failures.
- From the system failure analysis, partition failures into appropriate classes.
- For each failure class identified, set out the reliability using an appropriate metric. Different metrics may be used for different reliability requirements
- Identify functional reliability requirements to reduce the chances of critical failures

Bank auto-teller system

- Each machine in a network is used 300 times a day
- Bank has 1000 machines
- Lifetime of software release is 2 years
- Each machine handles about 200, 000 transactions
- About 300, 000 database transactions in total per day

Examples of a reliability spec.

Failure class	Example	Reliability metric
Permanent, non-corrupting.	The system fails to operate with any card which is input. Software must be restarted to correct failure.	ROCOF 1 occurrence/1000 days
Transient, non-corrupting	The magnetic stripe data cannot be read on an undamaged card which is input.	POFOD 1 in 1000 transactions
Transient, corrupting	A pattern of transactions across the network causes database corruption.	Unquantifiable! Should never happen in the lifetime of the system

Specification validation

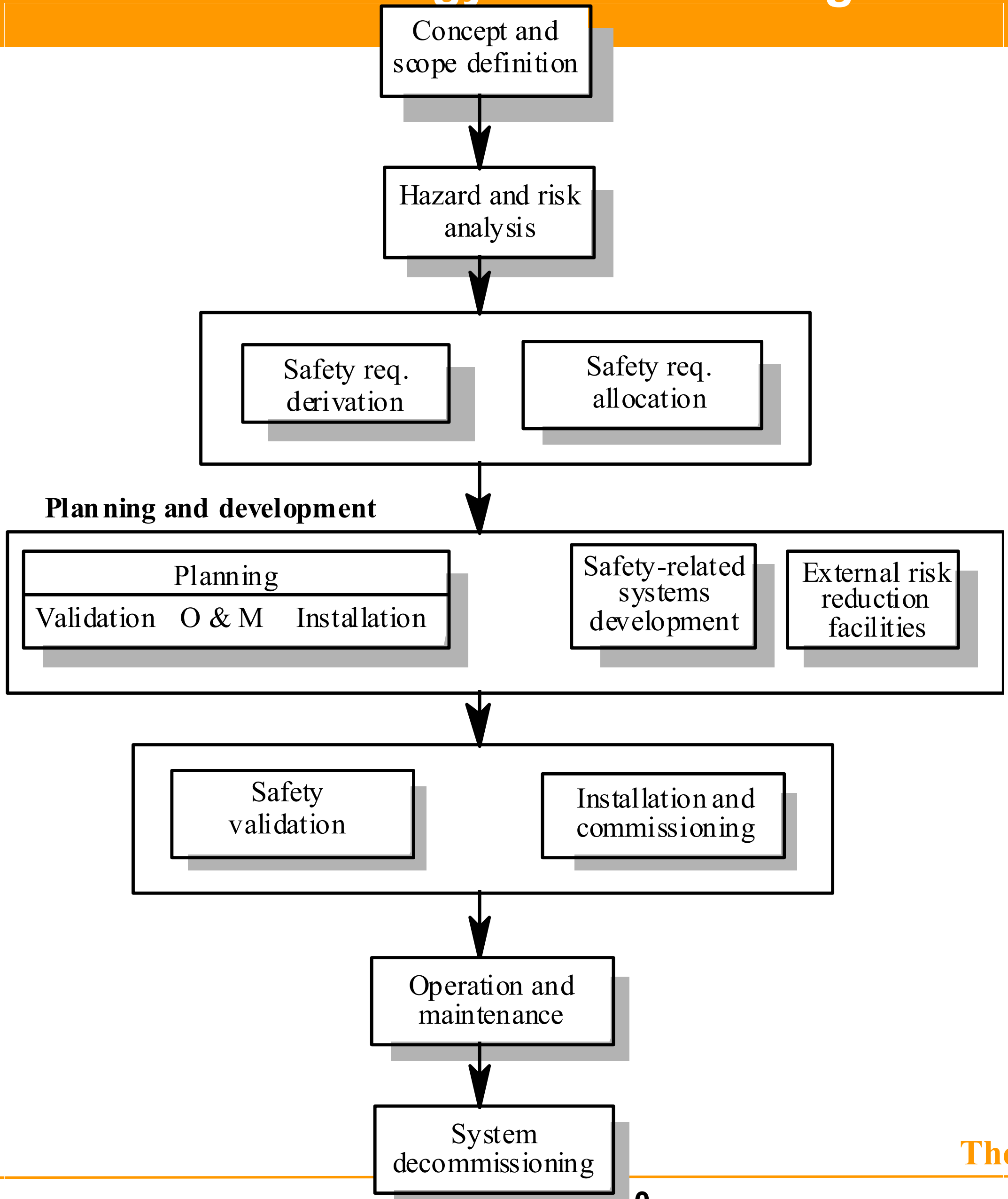
- It is impossible to empirically validate very high reliability specifications
- No database corruptions means POFOD of less than 1 in 200 million
- If a transaction takes 1 second, then simulating one day’s transactions takes 3.5 days
- It would take longer than the system’s lifetime to test it for reliability

Key points

- There are both functional and non-functional dependability requirements
- Non-functional availability and reliability requirements should be specified quantitatively
- Metrics that may be used are AVAIL, POFOD, ROCOF and MTTF
- When deriving a reliability specification, the consequences of different types of fault should be taken into account

Safety specification

- The safety requirements of a system should be separately specified
- These requirements should be based on an analysis of the possible hazards and risks
- Safety requirements usually apply to the system as a whole rather than to individual sub-systems. In systems engineering terms, the safety of a system is an emergent property

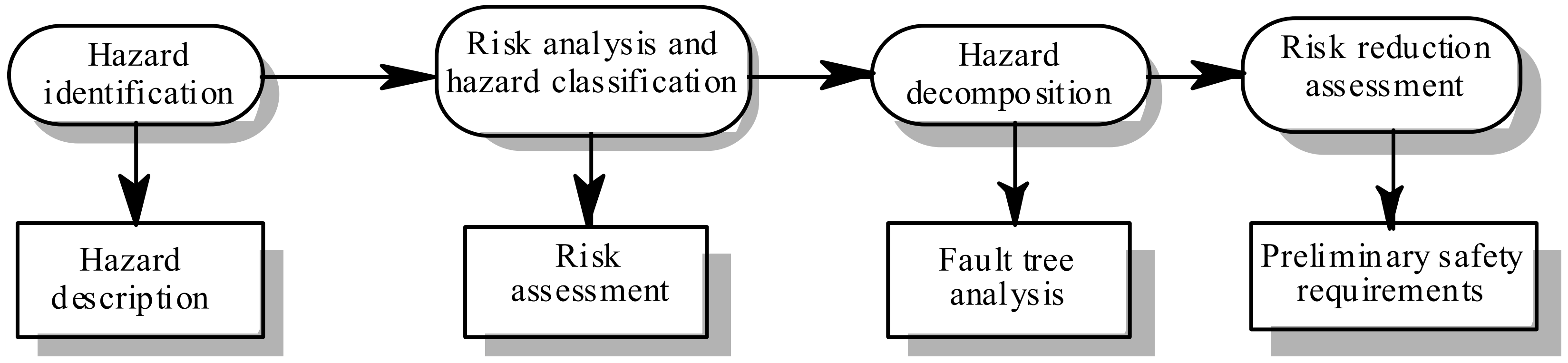


The safety life-cycle

Safety processes

- Hazard and risk analysis
  - Assess the hazards and the risks of damage associated with the system
- Safety requirements specification
  - Specify a set of safety requirements which apply to the system
- Designation of safety-critical systems
  - Identify the sub-systems whose incorrect operation may compromise system safety. Ideally, these should be as small a part as possible of the whole system.
- Safety validation
  - Check the overall system safety

Hazard and risk analysis


Hazard identification
Risk analysis and hazard classification
Hazard decomposition
Risk reduction assessment
Hazard description
Risk assessment
Fault tree analysis
Preliminary safety requirements

Hazard and risk analysis

- Identification of hazards which can arise which compromise the safety of the system and assessing the risks associated with these hazards
- Structured into various classes of hazard analysis and carried out throughout software process from specification to implementation
- A risk analysis should be carried out and documented for each identified hazard and actions taken to ensure the most serious/likely hazards do not result in accidents

Hazard analysis stages

- Hazard identification
  - Identify potential hazards which may arise
- Risk analysis and hazard classification
  - Assess the risk associated with each hazard
- Hazard decomposition
  - Decompose hazards to discover their potential root causes
- Risk reduction assessment
  - Define how each hazard must be taken into account when the system is designed

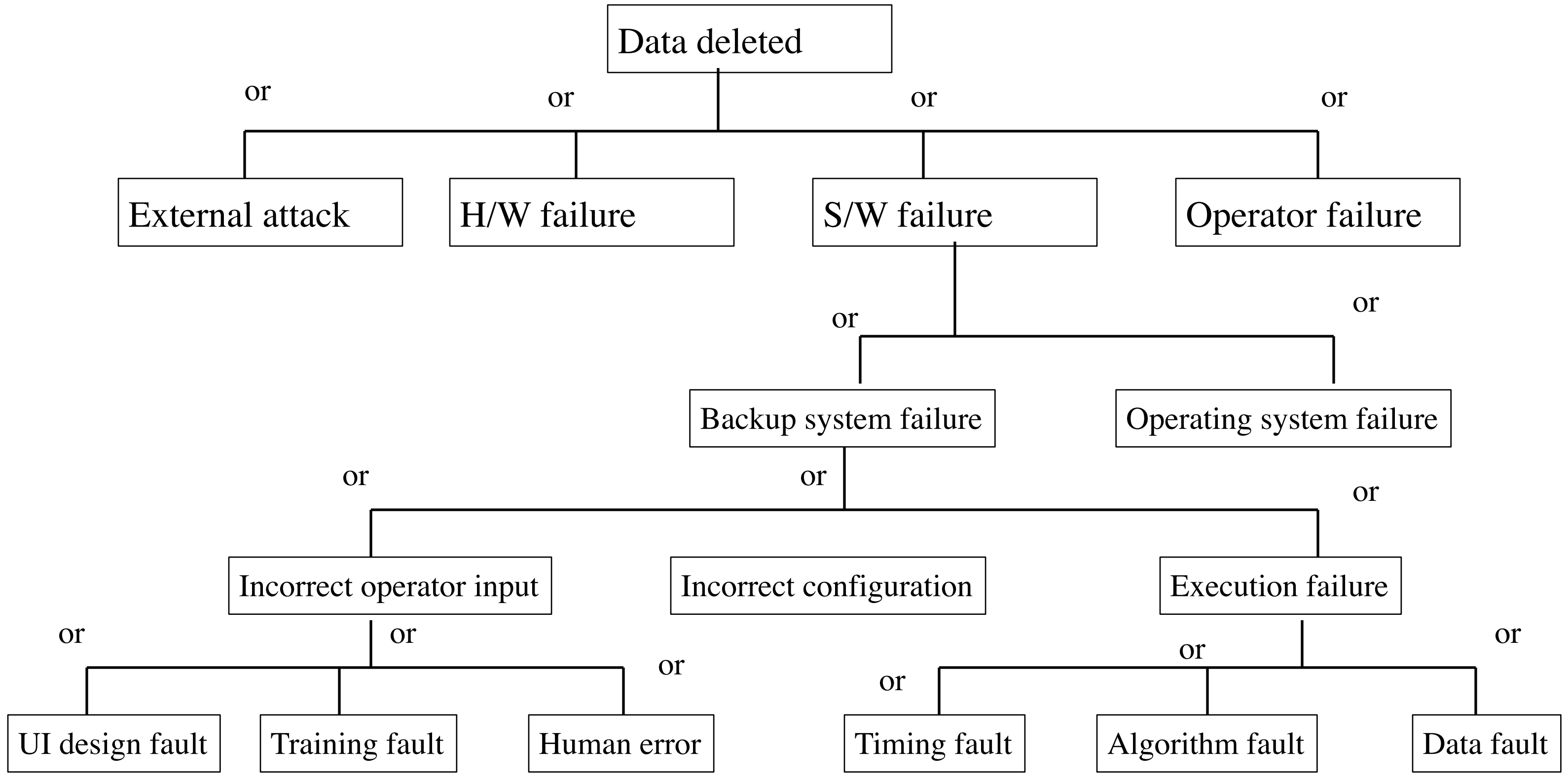
Fault-tree analysis

- Method of hazard analysis which starts with an identified fault and works backward to the causes of the fault.
- Can be used at all stages of hazard analysis from preliminary analysis through to detailed software checking
- Top-down hazard analysis method. May be combined with bottom-up methods which start with system failures and lead to hazards

Fault- tree analysis

- Identify hazard
- Identify potential causes of the hazard. Usually there will be a number of alternative causes. Link these on the fault-tree with ‘or’ or ‘and’ symbols
- Continue process until root causes are identified
- Consider the following example which considers how data might be lost in some system where a backup process is running

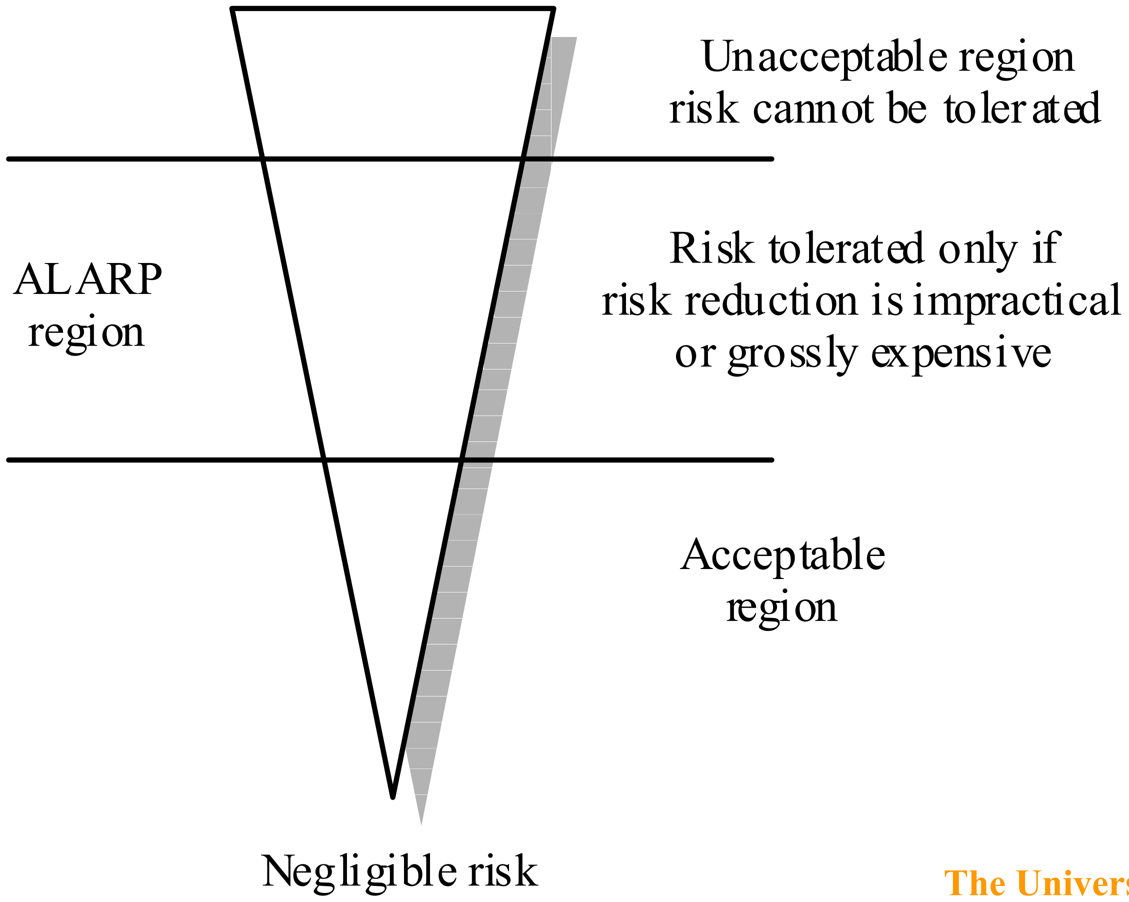
Fault tree


Data deleted
or
or
or
or
External attack
H/W failure
S/W failure
Operator failure
or
or
Backup system failure
Operating system failure
or
or
or
Incorrect operator input
Incorrect configuration
Execution failure
or
or
or
or
or
or
UI design fault
Training fault
Human error
Timing fault
Algorithm fault
Data fault

Risk assessment

- Assesses hazard severity, hazard probability and accident probability
- Outcome of risk assessment is a statement of acceptability
  - Intolerable. Must never arise or result in an accident
  - As low as reasonably practical(ALARP) Must minimise possibility of hazard given cost and schedule constraints
  - Acceptable. Consequences of hazard are acceptable and no extra costs should be incurred to reduce hazard probability

Levels of risk



Risk acceptability

- The acceptability of a risk is determined by human, social and political considerations
- In most societies, the boundaries between the regions are pushed upwards with time i.e. society is less willing to accept risk
  - For example, the costs of cleaning up pollution may be less than the costs of preventing it but this may not be socially acceptable
- Risk assessment is subjective
  - Risks are identified as probable, unlikely, etc. This depends on who is making the assessment

Risk reduction

- System should be specified so that hazards do not arise or result in an accident
- Hazard avoidance
  – The system should be designed so that the hazard can never arise during correct system operation
- Hazard detection and removal
  – The system should be designed so that hazards are detected and neutralised before they result in an accident
- Damage limitation
  – The system is designed in such a way that the consequences of an accident are minimised

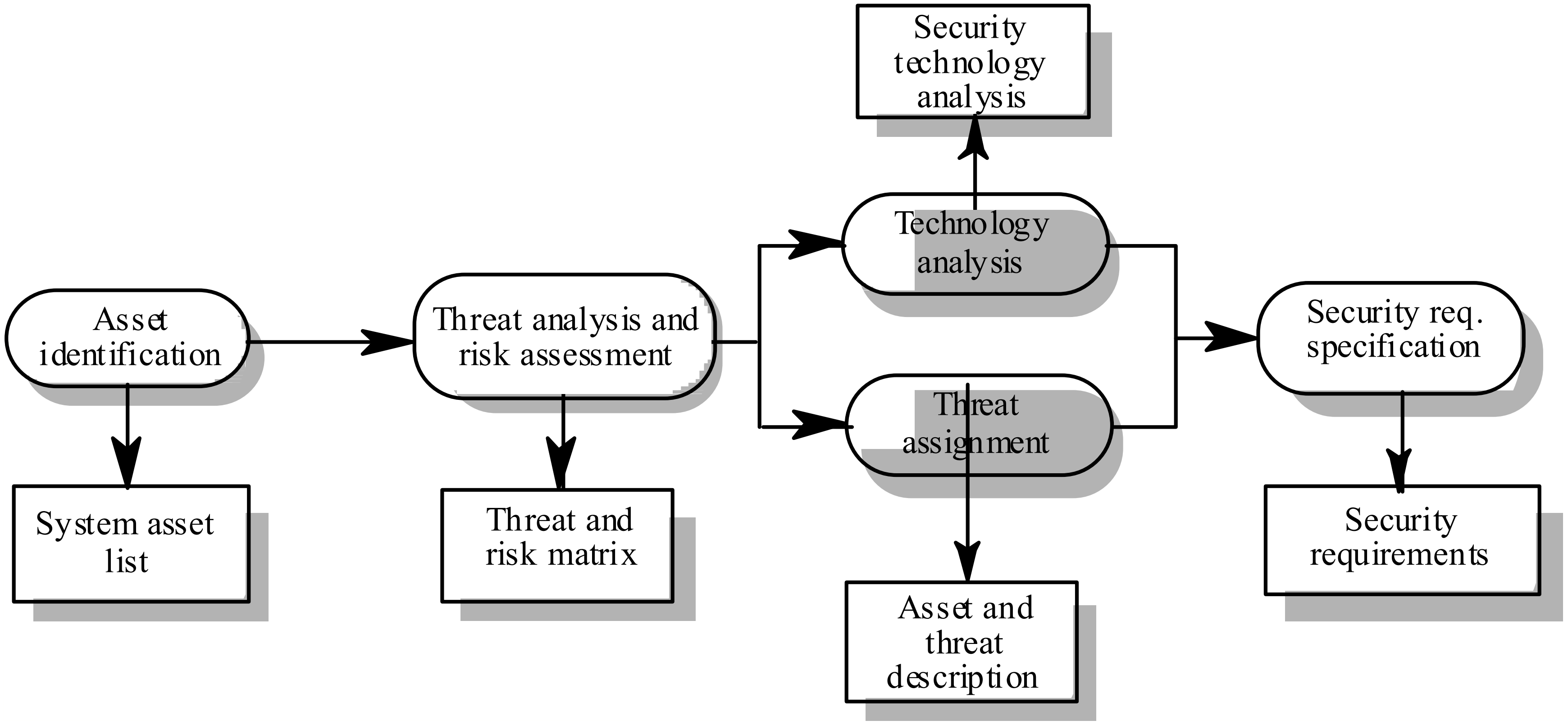
Specifying forbidden behaviour

- The system shall not allow users to modify access permissions on any files that they have not created (security)
- The system shall not allow reverse thrust mode to be selected when the aircraft is in flight (safety)
- The system shall not allow the simultaneous activation of more than three alarm signals (safety)

Security specification

- Has some similarities to safety specification
  - Not possible to specify security requirements quantitatively
  - The requirements are often ‘shall not’ rather than ‘shall’ requirements
- Differences
  - No well-defined notion of a security life cycle for security management
  - Generic threats rather than system specific hazards
  - Mature security technology (encryption, etc.). However, there are problems in transferring this into general use

The security specification process


Asset identification
System asset list
Threat analysis and risk assessment
Threat and risk matrix
Technology analysis
Security technology analysis
Threat assignment
Asset and threat description
Security req. specification
Security requirements

Stages in security specification

- ***Asset identification and evaluation***
  - The assets (data and programs) and their required degree of protection are identified. The degree of required protection depends on the asset value so that a password file (say) is more valuable than a set of public web pages.
- ***Threat analysis and risk assessment***
  - Possible security threats are identified and the risks associated with each of these threats is estimated.
- ***Threat assignment***
  - Identified threats are related to the assets so that, for each identified asset, there is a list of associated threats.

Stages in security specification

- ***Technology analysis***
  - **Available security technologies and their applicability against the identified threats are assessed.**
- ***Security requirements specification***
  - **The security requirements are specified. Where appropriate, these will explicitly identified the security technologies that may be used to protect against different threats to the system.**

Key points

- Hazard analysis is a key activity in the safety specification process.
- Fault-tree analysis is a technique which can be used in the hazard analysis process.
- Risk analysis is the process of assessing the likelihood that a hazard will result in an accident. Risk analysis identifies critical hazards and classifies risks according to their seriousness.
- To specify security requirements, you should identify the assets that are to be protected and define how security techniques should be used to protect them.