POLITECNICO
MILANO 1863

Dependable Systems

Basic Concepts & Terminology

Luca Cassano
luca.cassano@polimi.it
cassano.faculty.polimi.it/ds.html

Most of the material of these slides has been provided by Prof. Cristiana Bolchini, Politecnico di Milano, Italy

Dependability

“Dependability is that property of a computer system such that reliance can justifiably be placed on the service it delivers.”

Avizienis, A., Laprie, J. C., Randell, B., & Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, *1*(1), 11-33.

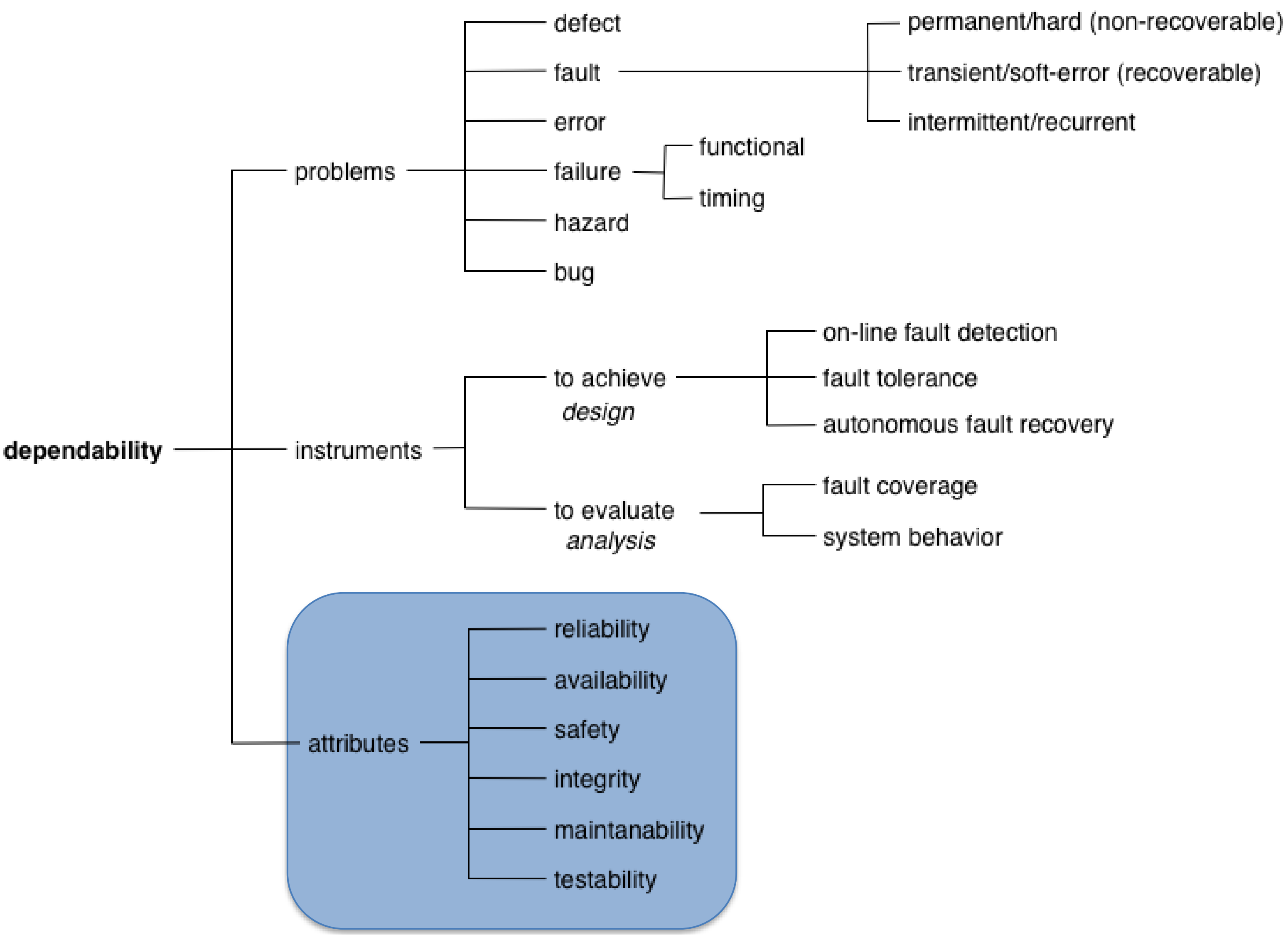
Dependability concept

For critical systems, often the most important system property is the dependability of the system

The dependability of a system reflects the user degree of trust in that system.

Usefulness and trustworthiness are not the same thing. A system does not have to be trusted to be useful

The scenario



Dependability

Term used to encapsulate the concepts of

- Reliability
- Availability
- Safety
- Security
- Maintainability
- Performability
- Testability

… measures used to quantify the dependability of a system …

Dependability attributes

Reliability

Availability

Safety

Integrity

Security

Maintainability

Testability

when expressing the system specification and requirements it is necessary to identify which properties are desirable/mandatory

Dependability attributes

These are “non-functional properties”
they do not relate to any specific functionality of the system

Some or all of these attributes may be more important than specific system functionality

Reliability

The ability of a system or component to perform its required functions under stated conditions for a specified period of time [IEEE610]

[IEEE610]: IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990 (R2002).

definition

R(t): probability that the system will operate correctly in a specified operating environment until time *t*

R(t) = P(not failed during [0, *t*])

assuming it was operating at time t = 0

t is important

If a system needs to work for slots of ten hours at a time, then that is the target

characteristics

1 – R(*t*): unreliability, also denoted Q(*t*)

R(*t*) is a non-increasing function varying from 1 to 0 over [0,+∞)

$$\lim_{x \to +\infty} R(t) = 0$$

adoption

Often used to characterize systems in which even small periods of incorrect behavior are unacceptable

- Performance requirements
- Timing requirements
- Extreme safety requirements
- Impossibility or difficulty to repair

Availability

The degree to which a system or component is operational and accessible when required for use
[IEEE610]

Availability = Uptime / (Uptime + Downtime)

definition

$A(t)$: probability that the system will be operational at time t

$$A(t) = P(\text{not failed at time } t)$$

Literally, readiness for service

Admits the possibility of brief outages

Fundamentally different from reliability

characteristics

1 – A(t): unavailability

When the system is not repairable?

characteristics

1 – A(*t*): unavailability

When the system is not repairable: A(*t*) = R(*t*)

In general (repairable systems): A(*t*) ≥ R(*t*)

Some numbers

Availability

Availability as a function of the "number of 9's"

Number of 9's	Availability	Downtime (mins/year)	Practical meaning
1	90%	52596.00	~5 weeks per year
2	99%	5259.60	~4 days per year
3	99.9%	525.96	~9 hours per year
4	99.99%	52.60	~1 hour per year
5	99.999%	5.26	~5 minutes per year
6	99.9999%	0.53	~30 secs per year
7	99.99999%	0.05	~3 secs per year

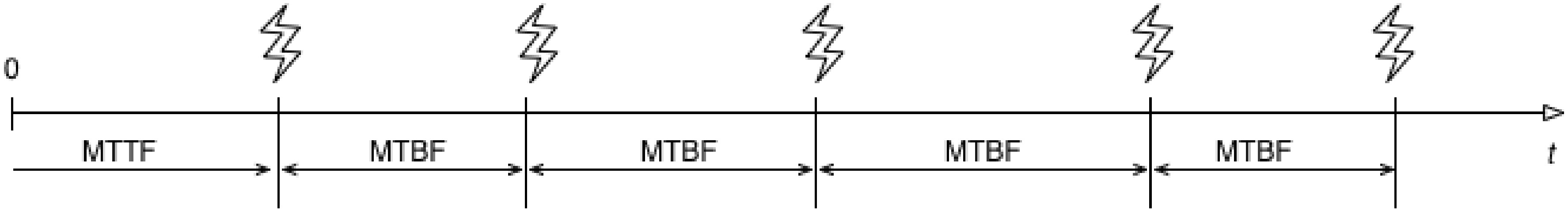
Some example

Number of 9’s	Availability	Downtime/year	System
2	99%	~4 days	Generic web site
3	99.9%	~9 hours	Amazon.com
4	99.99%	~1 hour	Enterprise server
5	99.999%	~5 minutes	Telephone system
6	99.9999%	~30 seconds	Phone switches

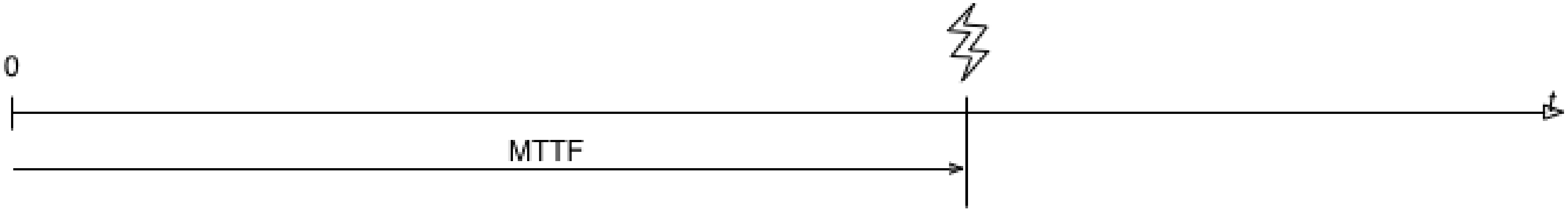
R(t) & A(t) related indices

MTTF (Mean Time To Failure): mean time before *any* failure will occur

MTBF (Mean Time Between Failures): mean time between two failures



hypothesis: negligible repair time



R(t) & A(t) related indices

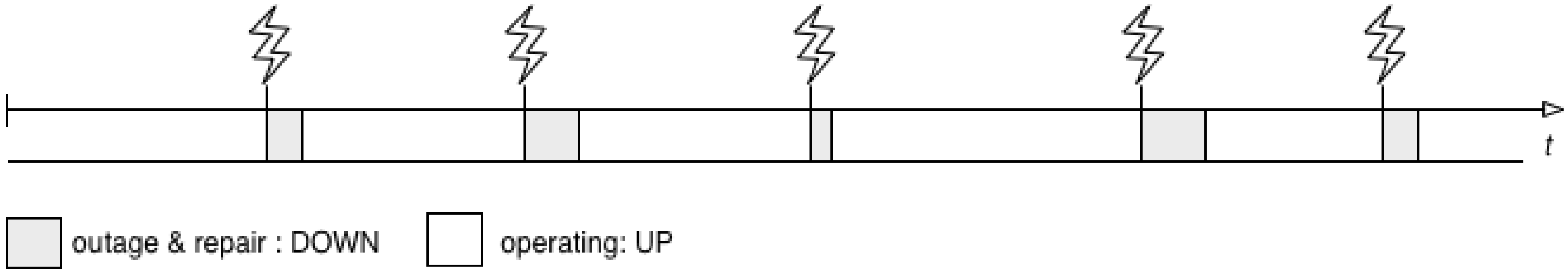
MUP: **mean up time**

– The device is operational (either correctly or not)

MDT: **mean down time**

– The device is not working: Time to repair + Time to recover

hypothesis: negligible detection time



R(t) & A(t) related indices

MTTR: **mean time to repair**

- Time to detect the fault + time to repair + time to recover

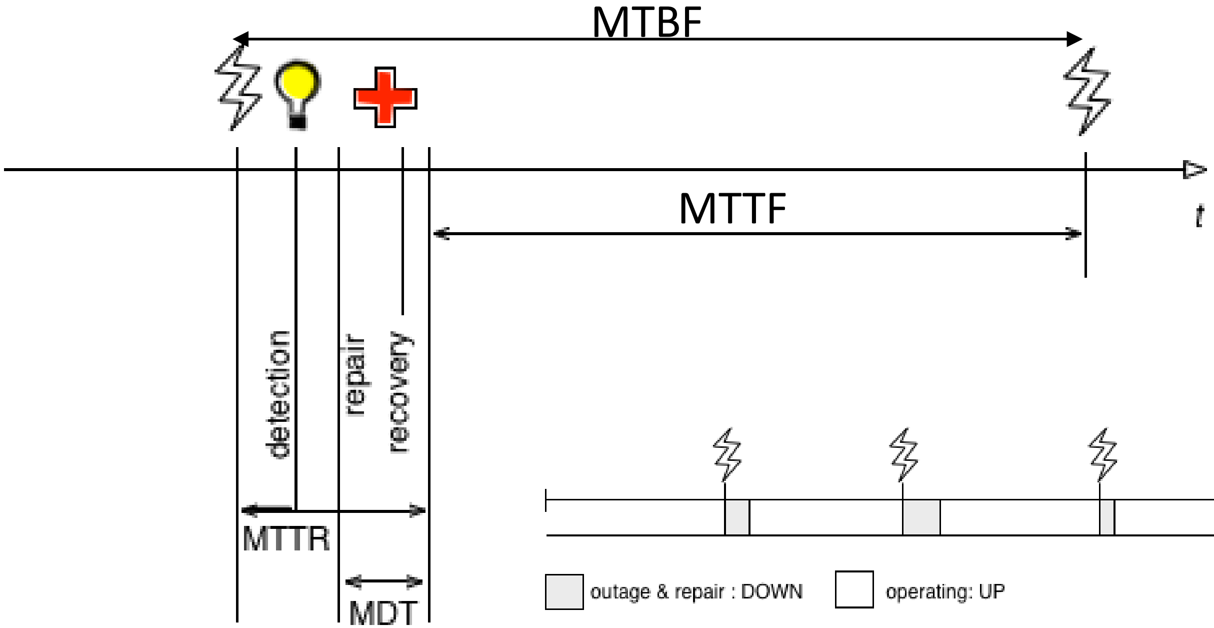
MTTR may not be the same as MDT because:

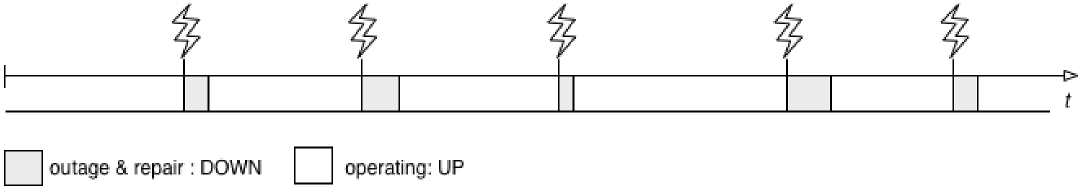
- The failure may not be noticed for some time after it has occurred
- It may be decided not to repair the equipment immediately
- The equipment may not be put back in service immediately after it is repaired

R(t) & A(t) related indices

Other indices

MTBF, MTTF, MTTR, MDT





$$\text{MTBF} = \frac{\text{total operating time}}{\text{number of failures}}$$

$$\text{MTBF} = \text{MTTF} + \text{MTTR}$$

R(t) & A(t) related indices

MTTF: mean time to (first) failure, the up time before the first failure

MTBF: mean time between failures

$$\text{MTBF} = \frac{\text{total operating time}}{\text{number of failures}}$$

R(t) & A(t) related indices

Other indices

MTTF: mean time to (first) failure, the up time before the first failure

MTBF: mean time between failures

$$\text{MTBF} = \frac{\text{total operating time}}{\text{number of failures}}$$

FIT: failures in time

$$\lambda = \frac{\text{number of failures}}{\text{total operating time}}$$

- another way of reporting MTBF
- the number of expected failures per one billion hours (10^9) of operation for a device
- MTBF (in h) $=10^9/\text{FIT}$

$$\text{MTBF} = \frac{1}{\lambda}$$

Reliability & Availability

Two different points of view

“**reliability**: does not break down …”

“**availability**: even if it breaks down, it is working when needed …”

Could you provide an example of system with high availability but low reliability?

Reliability & Availability

Two different points of view

“**reliability**: does not break down …”

“**availability**: even if it breaks down, it is working when needed …”

Example:
a system that fails, on average, once per hour but which restarts automatically in ten milliseconds is not very reliable but is highly available

A(t)=0.9999972

Two points of view

Of course they are related:

- if a system is unavailable it is not delivering the specified system services

It is possible to have systems with low reliability that must be available

- system failures can be repaired quickly and do not damage data, low reliability may not be a problem (for example a database management system)

The opposite is generally more difficult…

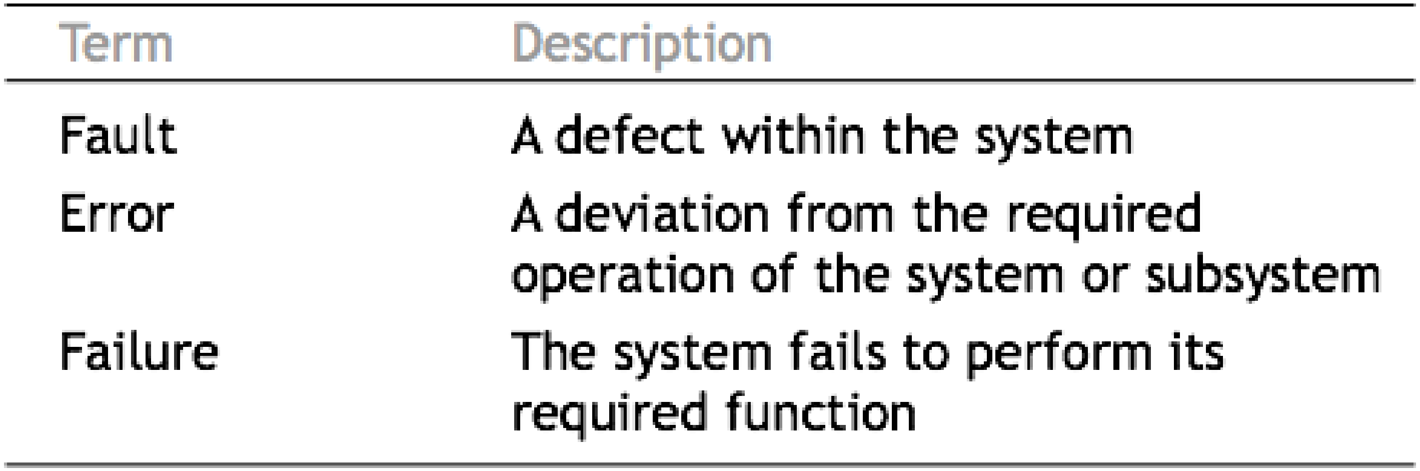
R(t) … what to do?

Exploitation of R(t) information is used to compute, for a complex system, its reliability in time, that is the expected lifetime

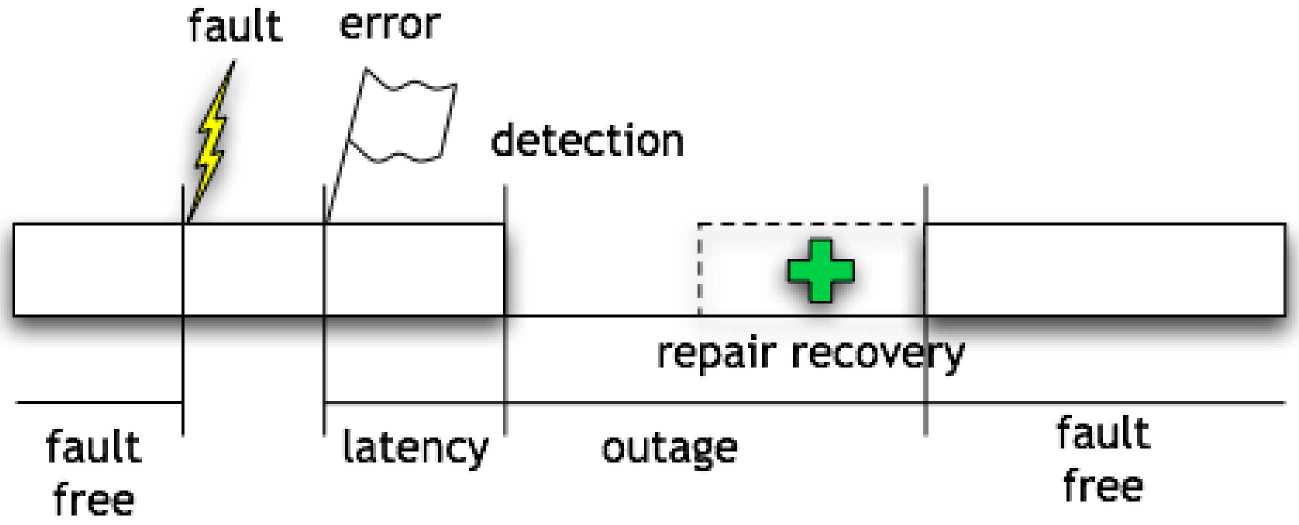
- computation of the MTTF

Computation of the overall reliability starting from the components’ one

Reliability terminology



Term	Description
Fault	A defect within the system
Error	A deviation from the required operation of the system or subsystem
Failure	The system fails to perform its required function



Reliability terminology

An example: a flying drone with an automatic radar-guided landing system

Fault: electromagnetic disturbances interfere with a radar measurement

Error: the radar-guided landing system calculates a wrong trajectory

Failure: the drone crashes to the ground

Reliability terminology

Another example: a tele-surgery system

Fault: radioactive ions make some memory cells change value (bitflip)

Error: some frames of the video stream are corrupted

Failure: the surgeon kills the patient

Reliability terminology

Not always the *fault – error – failure chain* closes

example: a tele-surgery system

Fault: radioactive ions make some memory cells change value (bitflip) but the corrupted memory does not involve the video stream

Error: no frames are corrupted

Failure: the surgeon carries out the procedure

Reliability terminology

Not always the *fault – error – failure chain* closes

example: a tele-surgery system

Fault: radioactive ions make some memory cells change value (bitflip) but the corrupted memory does not involve the video stream

Error: no frames are corrupte

Failure: the surgeon carries out the pro

Non activated fault

Reliability terminology

Not always the *fault – error – failure chain* closes

example: a flying drone with automatic radar-guided landing

Fault: electromagnetic disturbances interfere with a radar measurement

Error: the radar-guided landing system calculates a wrong trajectory, but then, based on subsequent correct radar measurements it is able to recover the right trajectory

Failure: the drone safely lands

Reliability terminology

Not always the *fault – error – failure chain* closes

example: a flying drone with automatic radar-guided landing

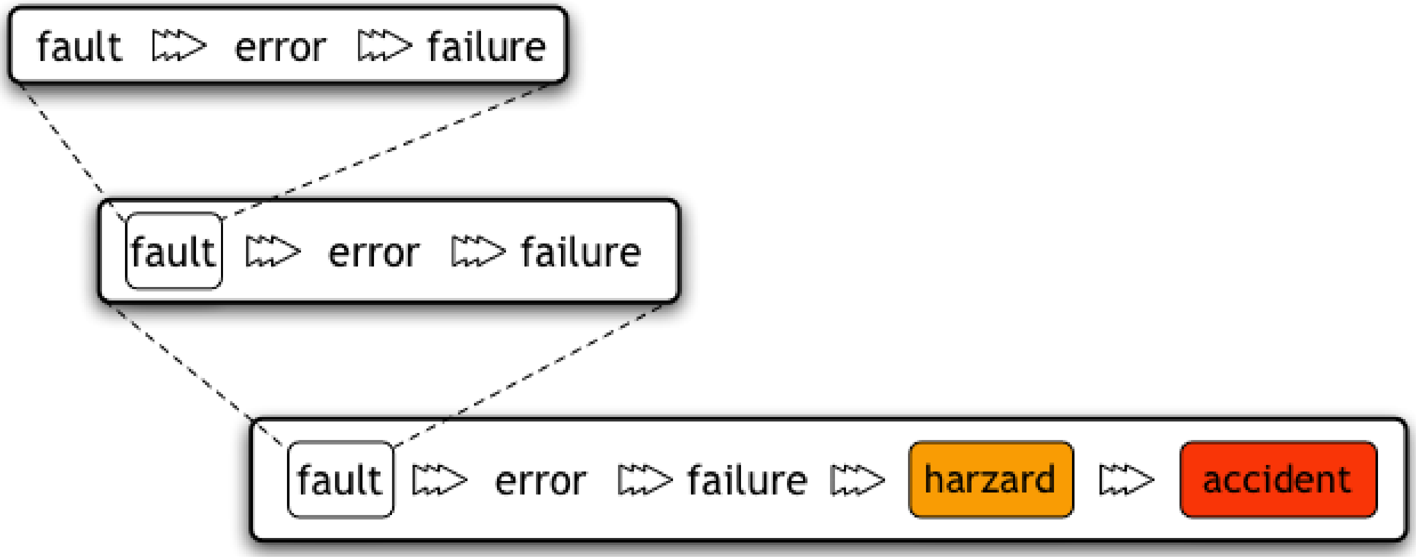
Fault: electromagnetic disturbances interfere with a radar measurement

Error: the radar-guided landing system calculates a wrong trajectory, but then, based on subsequent radar measu right trajectory

Non propagated (or absorbed) error

Failure: the drone safely lands

Fault hierarchy



Fault-error-failure cascades can lead to life-threatening hazards

Maintainability

Ability to undergo repairs and modifications

Ease of repairing the system after a failure has been discovered or changing the system to include new features

definition

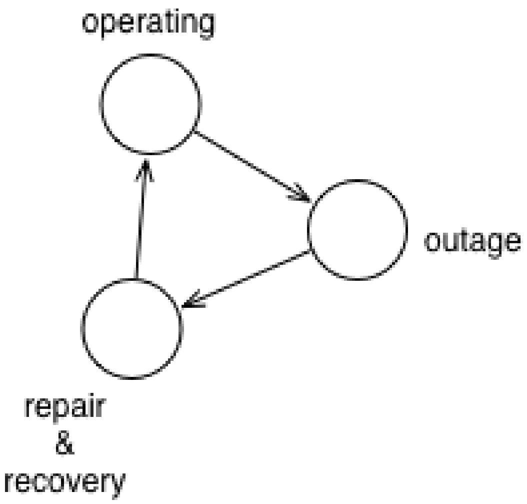
Maintanability

M(*t*): probability that a failed system can be repaired within time *t*

$$M(t) = P(\text{repaired in } [0, t])$$

M(*t*) is a non-decreasing function varying from 0 to 1 over [0,+∞)

$$\lim_{x \to +\infty} M(t) = 1$$



Performability

$P(L,t)$: probability that the system performance will be at, or above, some level L, at time t

A subset of the functions are performed correctly

Graceful degradation

Ability of a system to automatically decrease its level of performance to compensate for hardware and software failures

Hazard

A set of conditions (state of the system) that in certain environmental situations may lead to an incident

Hazard is the ***potential*** to cause harm

It determines a certain risk …

Hazard

It is fundamental at design-time to identify all the possible hazards

Risk

Based on the identified hazard, risk is the likelihood of harm

$\text{Risk}(t) = \sum p(\text{accident}) * \text{cost}(\text{accident})$

Risk is the expected loss per unit of time
(in defined circumstances, and usually qualified by some statement of the severity of the harm)

Risk

Based on the identified hazard, risk is the likelihood of harm

$\text{Risk}(t) = \sum p(\text{accident}) * \text{cost}(\text{accident})$

Risk is the expected loss per unit of time
(in defined circumstances, and usually qualified by some statement of the severity of the harm)

The designer, based on design standards, has to identify acceptable risks:

Risk

Based on the identified hazard, risk is the likelihood of harm

$\text{Risk}(t) = \sum p(\text{accident}) * \text{cost}(\text{accident})$

Risk is the expected loss per unit of time
(in defined circumstances, and usually qualified by some statement of the severity of the harm)

The designer, based on design standards, has to identify acceptable risks:

Case 1: high accident cost but negligible accident probability

Is the risk affordable?

Risk

Based on the identified hazard, risk is the likelihood of harm

Risk(t) = ∑p(accident) * cost(accident)

Risk is the expected loss per unit of time
(in defined circumstances, and usually qualified by some statement of the severity of the harm)

The designer, based on design standards, has to identify acceptable risks:

Case 2: reduced accident cost but high accident probability

Is the risk affordable?

Risk

It is fundamental at design time to quantify the acceptable risk

Safety is expressed as an acceptable level of loss

Safety

The absence of catastrophic consequences on the users or the environment

Are commercial aircraft “safe”?

- They seldom crash
- What is acceptable?

Are cars safe?

- They crash a lot …

Safety

Who does define the level of safety a system must guarantee?

Safety

Who does define the level of safety a system must guarantee?

Companies, scientists and governments define **safety standards**

Safety

Who does define the level of safety a system must guarantee?

Safety standards classify subsystems based on *Safety Integrity Levels (SILs)*

For example, the automotive ISO 26262 classifies subsystems into four *Automotive Safety Integrity Levels (ASILs)*

Safety

Who does define the level of safety a system must guarantee?

- ASIL A: failures cause no injuries - the radio for example
- ASIL B: failures cause light to moderate injuries
- ASIL C: failures cause severe injuries (survival probable)
- ASIL D: failures cause life-threatening and fatal injuries - breaking and steering for example

Safety property

A safety-related system is one by which the safety of equipment or plant is assured

Safety for computer systems:

- Computer hardware
  - ➧ primary safety
- Equipment directly controlled by the computer
  - ➧functional safety
- Indirect consequences of a computer failure or incorrect information production
  - ➧indirect safety

Safety, Reliability and Availability

Note that safety, reliability and availability are not always strongly connected….

….do you have any idea?

Safety, Reliability and Availability

Note that safety, reliability and availability are not always strongly connected….

A safety-critical system MUST be highly reliable but not necessarily highly available

Safety, Reliability and Availability

Note that safety, reliability and availability are not always strongly connected….

A safety-critical system MUST be highly reliable but not necessarily highly available

A failure of a train may cause huge human and monetary loss

BUT

After a failure, a train may be deeply analysed and maintained so its availability may be strongly reduced

Safety, Reliability and Availability

Note that safety, reliability and availability are not always strongly connected….

A system may need to be highly available but users may tolerate short down periods (low reliability) not exposing safety-criticality

Safety, Reliability and Availability

Note that safety, reliability and availability are not always strongly connected….

A system may need to be highly available but users may tolerate short down periods (low reliability) not exposing safety-criticality

Web and email servers for examples

Safety, Reliability and Availability

Note that safety, reliability and availability are not always strongly connected….

A system may need to be highly reliable but users may tolerate relatively long maintainance periods (low availability) not exposing safety-criticality

Safety, Reliability and Availability

Note that safety, reliability and availability are not always strongly connected….

A system may need to be highly reliable but users may tolerate relatively long maintainance periods (low availability) not exposing safety-criticality

Smarphones for examples

Reliability & Availability & Safety

a system that is turned off is not very reliable, not very available, probably safe….

Security

Systems should protect themselves and their data from external interference

- Resist to accidental or deliberate intrusions and denial-of-service attacks (survivability)
- Prohibit unsupported actions (Functionality integrity)
- Hide sensitive data (Confidentiality)
- Prevent data alteration (Data integrity)
- Distinguish between fresh and legitimate data and forged ones (Non repudiation)

Testability

Ability to test for certain attributes within a system

Related to maintainability ➢ importance of minimizing time required to identify and locate specific problems (diagnosis)

Dependability requirements

Telecommunications

- Availability, maintainability

Transportation

- Reliability, availability, safety

Weapons

- Safety

Nuclear systems

- Safety

Pervasive computing

- Reliability, availability, maintainability, safety

References

[IEEE610]: IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990(R2002).

D. K. Pradhan, “Fault-tolerant Computer System Design,” Computer Science Press, 2003

J. C. Knight, “An Introduction To Computing System Dependability”, Proc. 26th Int. Conf. on Software Engineering (ICSE’04)

A. Villemeur, “Reliability, Availability, Maintainability and Safety Assessment,” vols. 1 & 2, John Wiley and sons, 1991

Ian Sommerville, “Software Engineering”, 9th edition, 2010