Industrial Embedded Systems
- Design for Harsh Environment -

Dr. Alexander Walsch
alexander.walsch@ge.com

Part II

WS 2011/12

Technical University Munich (TUM)

Recap: Practical Example

- Two PCBs (Printed Circuit Board)
- What do you remember from last lecture?

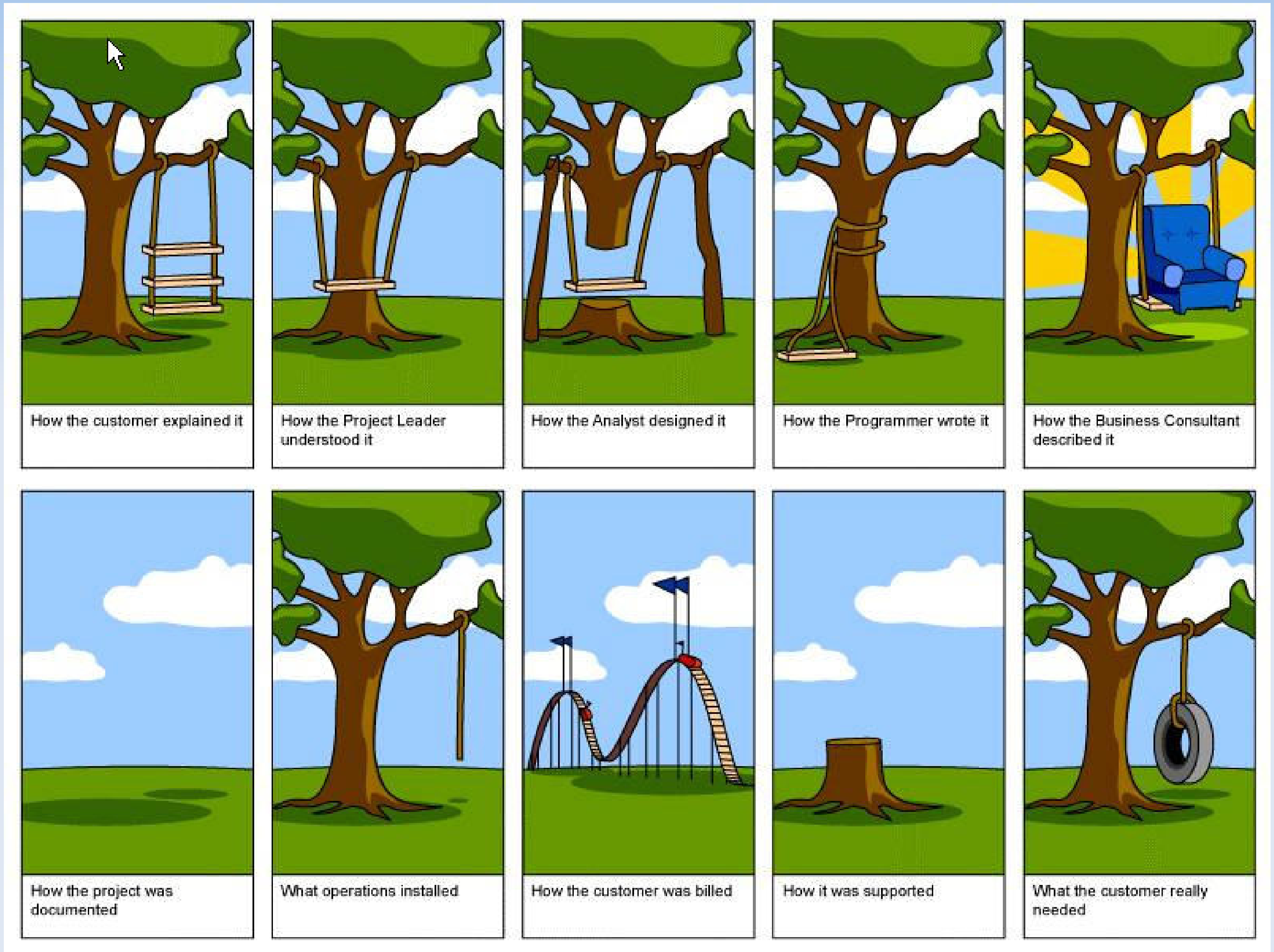
Recap: Practical Example

- Two PCBs (Printed Circuit Board)
- What do you remember from last lecture?
  - Nothing?
  - A CPU?
  - An OpAmp
  - Some more that looks familiar?

Part II: Requirements Engineering

Requirements Analysis
Requirements Specification

Motivation


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Source: http://www.bowdoin.edu/~disrael/what-the-customer-really-needed/what-the-customer-really-needed.jpg

Motivation II

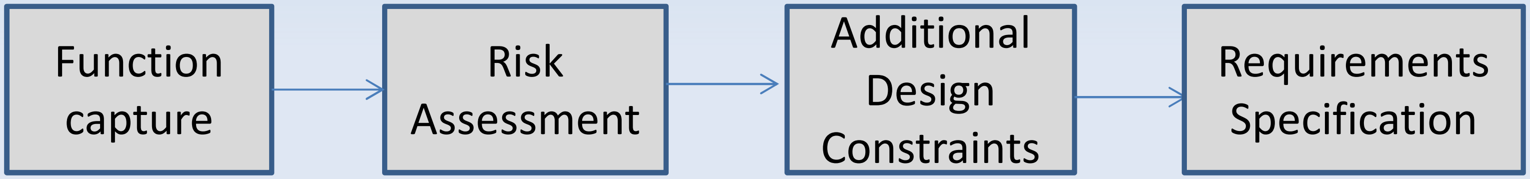
- Requirement
  Features of a system or system function used to fulfill the system purpose.
- Reliability
  the probability that a system will perform a required function under specified conditions for a specific period of time
- Safety
  the freedom from unacceptable risk of physical injury or of damage to the health of people
- Risk
  a measure of the probability and consequence of a specified undesired event

The Big Picture

- The requirements analysis phase of embedded system development is about:
  - Getting all system functions together
  - Showing scope, usage, and constraints (performance, environment, regulation, threats, etc.) of the proposed system
  - Get a good understanding on effort and system architecture (risk reduction)
- Wrong (e.g. missing, contradicting) information will make us fail at a very cost intensive level
- Once all information is available the requirements definition (outcome of analysis) is translated into a requirements specification which is a technical document for further development (metrics on all requirements)

The Big Picture II

- There are basically two kinds of developments:
  - Safety-related (we will define this later)
  - No safety concern



Function capture

Risk Assessment

Additional Design Constraints

Requirements Specification

- We need to figure out very early if we need to cope with additional design constraints like
  - Reliability – this could influence the system architecture and/or cost
  - Safety – this could complicate the overall effort

Standards and Certifications

Important domain specific standards and quality metrics:

- General/Industrial: IEC61508 – Safety Integrity Level (SIL)
- Automotive: ISO CD 26262 – Automotive Safety Integrity Level (ASIL)
- Aviation: DO178/DO254 – Design Assurance Level (DAL)
- Rail: EN 50126/50128/50129 – Safety Integrity Level (SIL)
- Healthcare: IEC 62304 (SW)

Requirements Analysis

How do we get all these requirements?

- Involves technical staff working with customers or users to find out about the application domain (field technicians), the services that the system should provide and the system's operational constraints.
- May involve end-users, our customers, managers, engineers involved in prior development and/or maintenance, domain experts, certification bodies, etc. These are called stakeholders.

Challenges in Requirements Analysis

- Stakeholders don’t know what they really want.
- Stakeholders express requirements in their own terminology – maybe not precise.
- Different stakeholders may have conflicting requirements.
- Political factors may influence the system requirements (e.g. disasters).
- The requirements change during the analysis process.
- Some requirements might be common sense and not explicitly mentioned.

Feasibility Study

Feasibility Study

A feasibility study decides whether or not the proposed system or component is worthwhile. Usually a study on the most risky elements of a new development.

A short focused study (simulation or setup) that checks

- If the proposed system can be engineered using current technology and within budget (technical and economic feasibility);
- If the proposed system can be integrated with other systems that are used (interoperability).
- If the proposed system can meet the requirements (especially non-functional like reliability, e.g.)

Fault, Error, Failure

Fault

abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function

Error

a deviation from the correct value or state

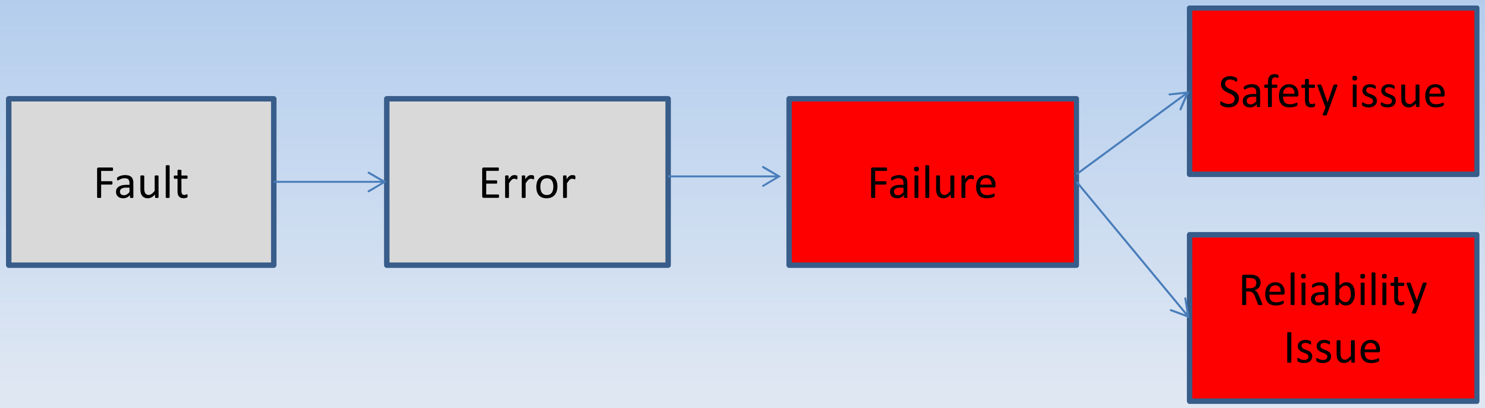
Failure

Failure is defined as deviation from the specification. The designed function can not be executed anymore as specified.

Failure Mode

A component (or system) can fail in various ways. In our analysis we pick the failure mode that leads to the failure we investigate.

Fault, Error, Failure II



- Faults can be though of as physical faults, e.g. a bit flips, a wire breaks
- Faults are dormant until the faulty component (memory cell, etc.) is used (think of a software task that executes cyclically)
- Once it is used it will cause an error which is a deviation from the expected
- The error will make the system deviate from its specification. It is running outside its intended use

Quantitative Failure Example

Function:

A process variable is measured and the reading transmitted using a 4 – 20 mA data communication interface.

The following failure modes and occurrences are known. What failure modes do influence our design most?

Failure Mode	Failure occurance
4 – 20 mA current signal stuck fail	Low
4 – 20 mA current signal low fail	Low
Sensor head fail	Medium
Power failure	High
Other	low

Failure Modes and Effect Analysis (FMEA)

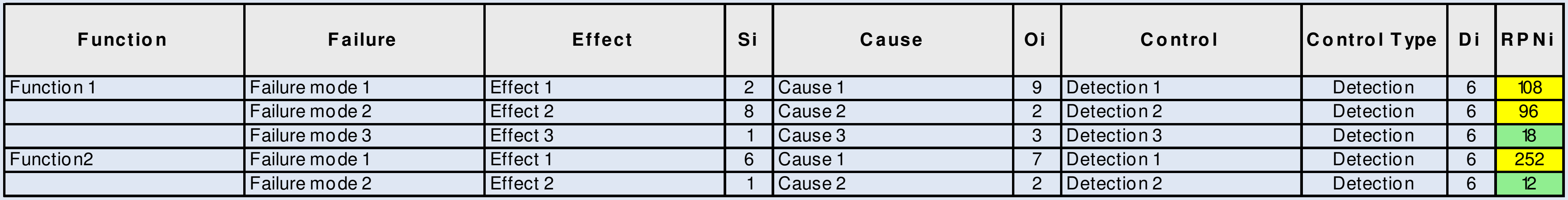
- System FMEA in requirements analysis (proposed system)
  - Also: Design FMEA (existing system)
- What are the failure modes and what is the effect:
  - System failure (e.g. power, communication, timeliness, erroneous) mode assessment
  - Plan how to prevent the failures
- How does it work?
  - Identify potential failure modes and rate the severity (team activity)
  - Evaluate objectively the probability of occurrence of causes and the ability to detect the cause when it occurs
  - Rank failure modes and isolate the most critical ones

FMEA II

- FMEA tools
  - Spreadsheet, proprietary (e.g. Reliasoft)
- Risk ratings: 1 (best) to 10 (worst)
  - Severity (SEV) – how significant is the impact
  - Occurance (OCC) – likelihood of occurance
  - Detection (DET) – how likely will the current system detect the failure mode
- Risk Priority Number (RPN)
  - A numerical calculation of the relative risk of a particular failure mode
  - RPN = SEV x OCC x DET
  - Used to isolate the most risky functions and their failure modes
  - Qualtitative approach (risk ratings are relative numbers)

FMEA III

- Function – What is the system going to do?
- Failure – How could the function fail?
- Effect – What could be the outcome of the failure?
- Cause – What could be the cause of the failure?



Function	Failure	Effect	Si	Cause	Oi	Control	Control Type	Di	RPNi
Function 1	Failure mode 1	Effect 1	2	Cause 1	9	Detection 1	Detection	6	108
	Failure mode 2	Effect 2	8	Cause 2	2	Detection 2	Detection	6	96
	Failure mode 3	Effect 3	1	Cause 3	3	Detection 3	Detection	6	18
Function2	Failure mode 1	Effect 1	6	Cause 1	7	Detection 1	Detection	6	252
	Failure mode 2	Effect 2	1	Cause 2	2	Detection 2	Detection	6	12

Fault Tree Analysis (FTA)

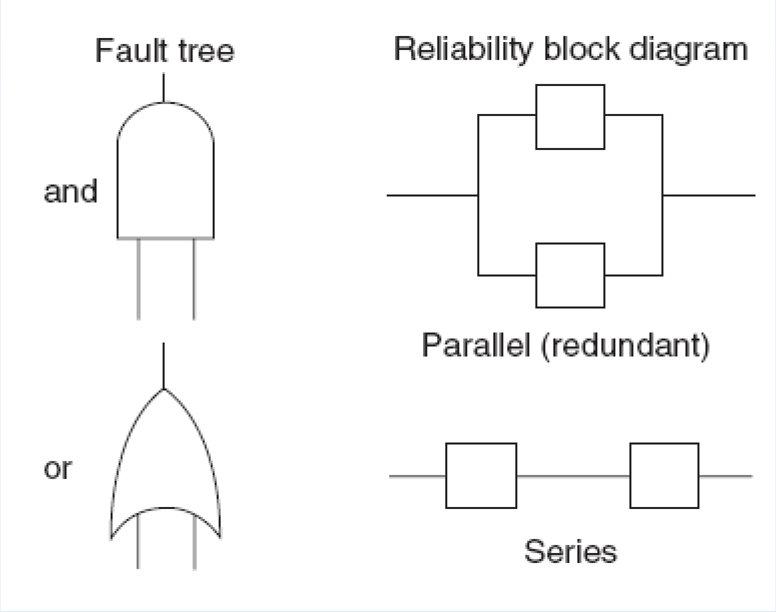
- Top event is failure mode
- Devide system into components
- Look into combinations of faults (strength of FTA)
- Tree like structure using combinatorical logic
- Paths of Failure

Outcome:

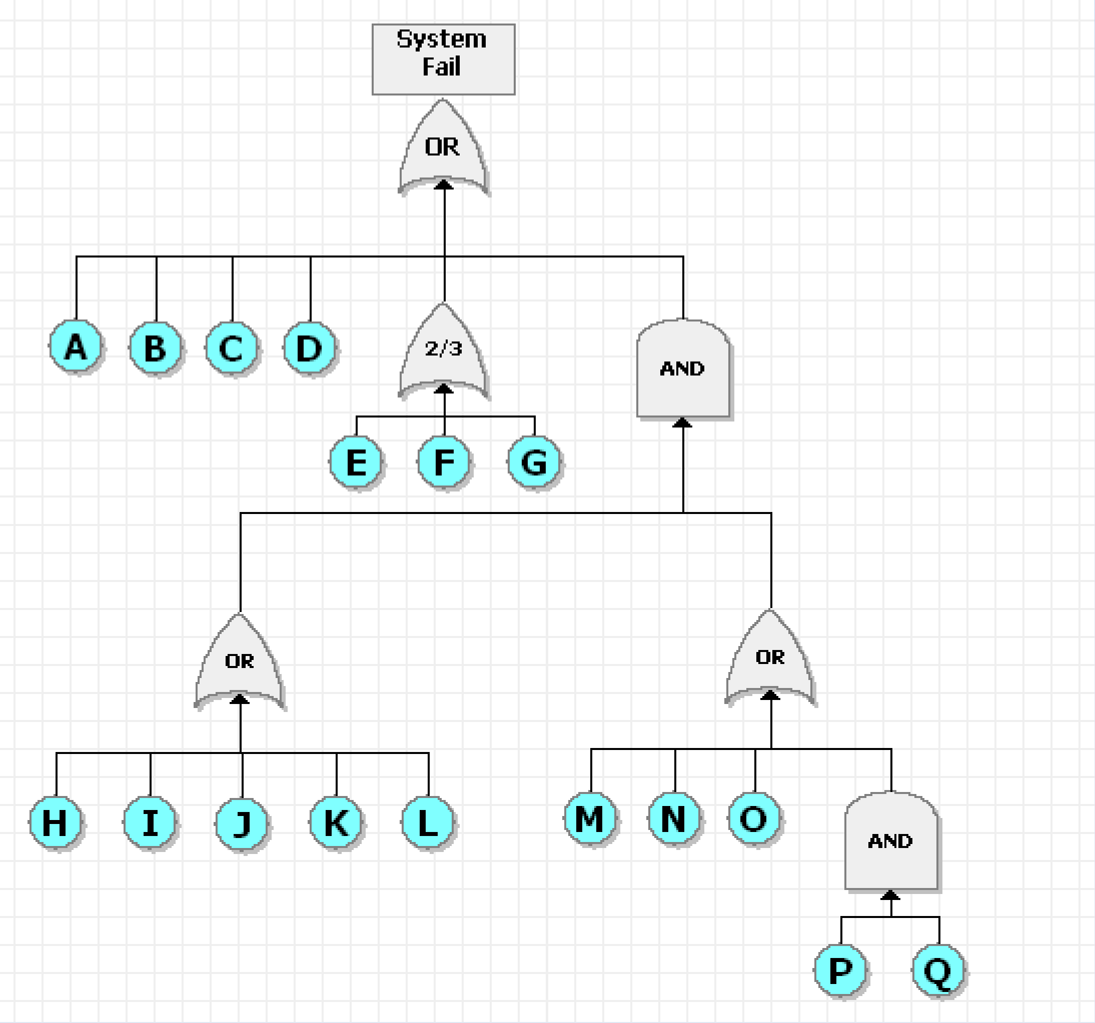
- Root cause event (external, internal) that (in combination) will lead to top event
- Good system understanding – very useful if applied to existing systems to isolate reliability issues

FTA II

- FTA is semantically equivalent to RBD which we will examine later



Source:
Smith, Functional Safety



Where are we?

- We know that there are critical requirements that influence our proposed system:
  - What can we do about that?
    Are there any architecturural or technology decisions we should make early on?
  - What metrics do we have?
    At this point we have only used categories but no real numbers. We need some metrics.

Failure Rate

Failure Rate

A time dependent measure of #failures/time. Commonly only random failures are considered. The symbol for failure rate is $\lambda(t)$. A failure rate is tied to a failure mode. This is a hardware related metric.

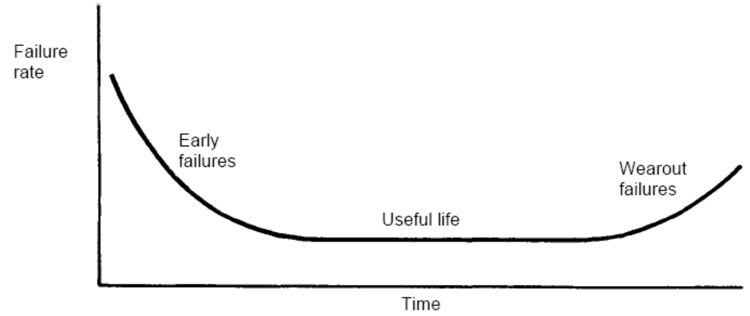
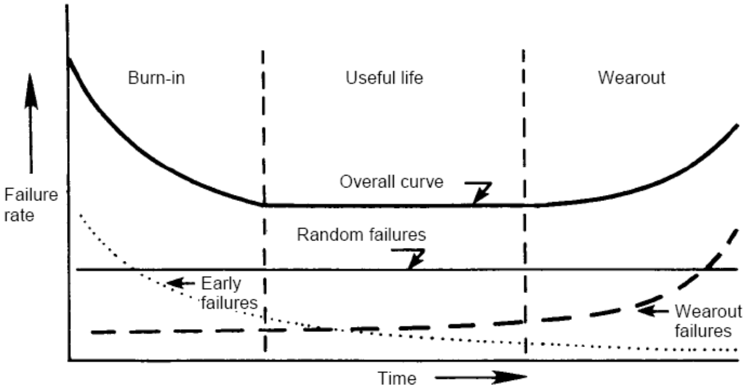


Figure 2.4



Source:

Smith: Reliability, Maintainability and Risk

Reliability

Reliability

Reliability of a system or component is defined to be the probability that a given system or component will perform a required function under specified conditions for a specified period of time.

- "probability of non-failure (survival) in a given period"
- Reliability of a system function is modeled as: $\mathrm{R(t)} = e^{-\lambda t}$ if the failure rate λ is constant.
- λ is often expressed as failures per 10^6 hours or FIT (failures per 10^9 hours).
- If "λt" small then R(t) = 1 - λt

Mean Time Between Failure (MTBF)

MTBF

Mean Time Between Failures (MTBF) is the average time a system will run between failures. The MTBF is usually expressed in hours.

$\Theta = \int_0^\infty R(t)dt = \int_0^\infty e^{-\lambda t}\,\mathrm{dt} = \lambda^{-1}$, λ = const.

The observed MTBF (not all items have failed but k):

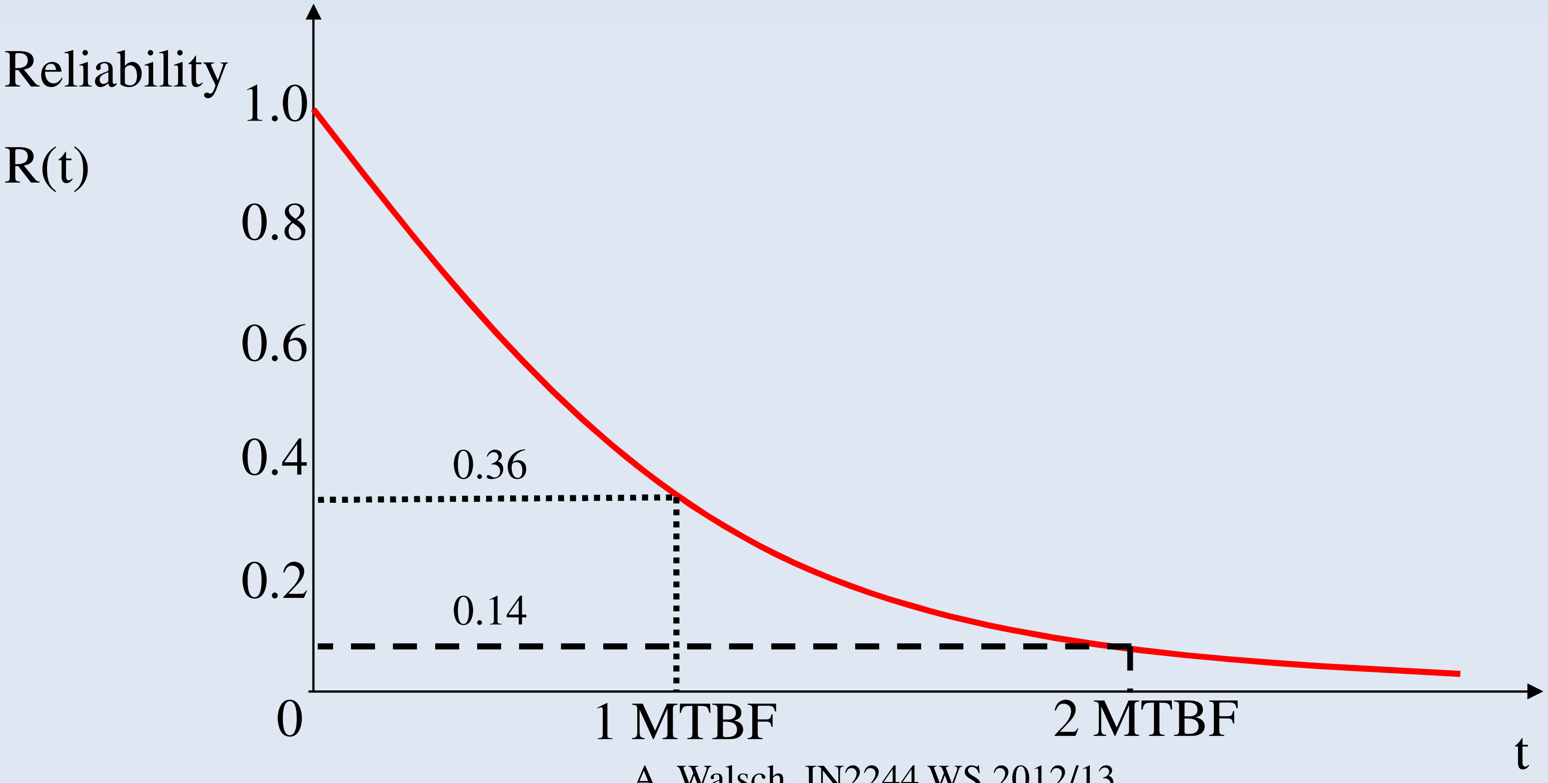
$\widehat{\Theta}$ = T/k; T = total time, k = failed items (total N)

Relation between Reliability and MTBF

$$\mathrm{R(t)} = e^{-\lambda t} = e^{-t/\Theta}$$

$\mathrm{t} = \Theta$ -> $\mathrm{R} = e^{-1} \approx 0.37$

$\mathrm{t} = 2\Theta$ -> $\mathrm{R} = e^{-2} \approx 0.14$



Failure Rate Example

A system (S) has 10 components (C) with a failure rate of 5 per 10^6 hours each. Calculate λ_S and MTBF $_S$. Consider two cases:

- All components are required to perform the function (single point of failure)

- Each component performs a different function

Failure Rate Example II

A system (S) has 10 components (C) with a failure rate of 5 per 10^6 hours each. Calculate λ_S and MTBF $_S$. Consider two cases:

A) All components are required to perform the function (single point of failure)

B) Each component performs a different function

Solution A)

$\lambda_C = 5 * 10^{-6}$ failures/hour

$\lambda_S = 10 * 5 * 10^{-6}$ failures/hour $= 5 * 10^{-5}$ failures/hour

$\Theta_S = \frac{1}{\lambda_s} = 20000\text{h}$

Solution B)

$\lambda_C = \lambda_S = 5 * 10^{-6}$ failures/hour; $\Theta_S = \frac{1}{\lambda_s} = 200000\text{h}$

Mean Down Time (MDT)

MDT

Mean Down Time (MDT) is the average time a system is in a failed state and can not execute its function.
MTBF can be understood as the mean up time.

MTTR

Mean Time to Repair (MTTR) is overlapping with MDT. Used for maintenance calculations. It can be visualized as the average time it takes (a technician) to repair the system such that it is up again.
We will not use MTTR in this lecture anymore.

Availability

<u>Availability</u>

Availability is the probability that a system is functioning at any time during its scheduled working period.

$$A = \frac{up\ time}{total\ time} = \frac{up\ time}{up\ time+down\ time} = \frac{MTBF}{MTBF+MDT}$$

similar:

calculation of unavailability (PFD)

Unavailability Example

$\lambda = 10^{-6}$ failures/hour ; MDT = 10h

Unavailability = ?

Unavailability Example

λ = 10^{-6} failures/hour ; MDT = 10h

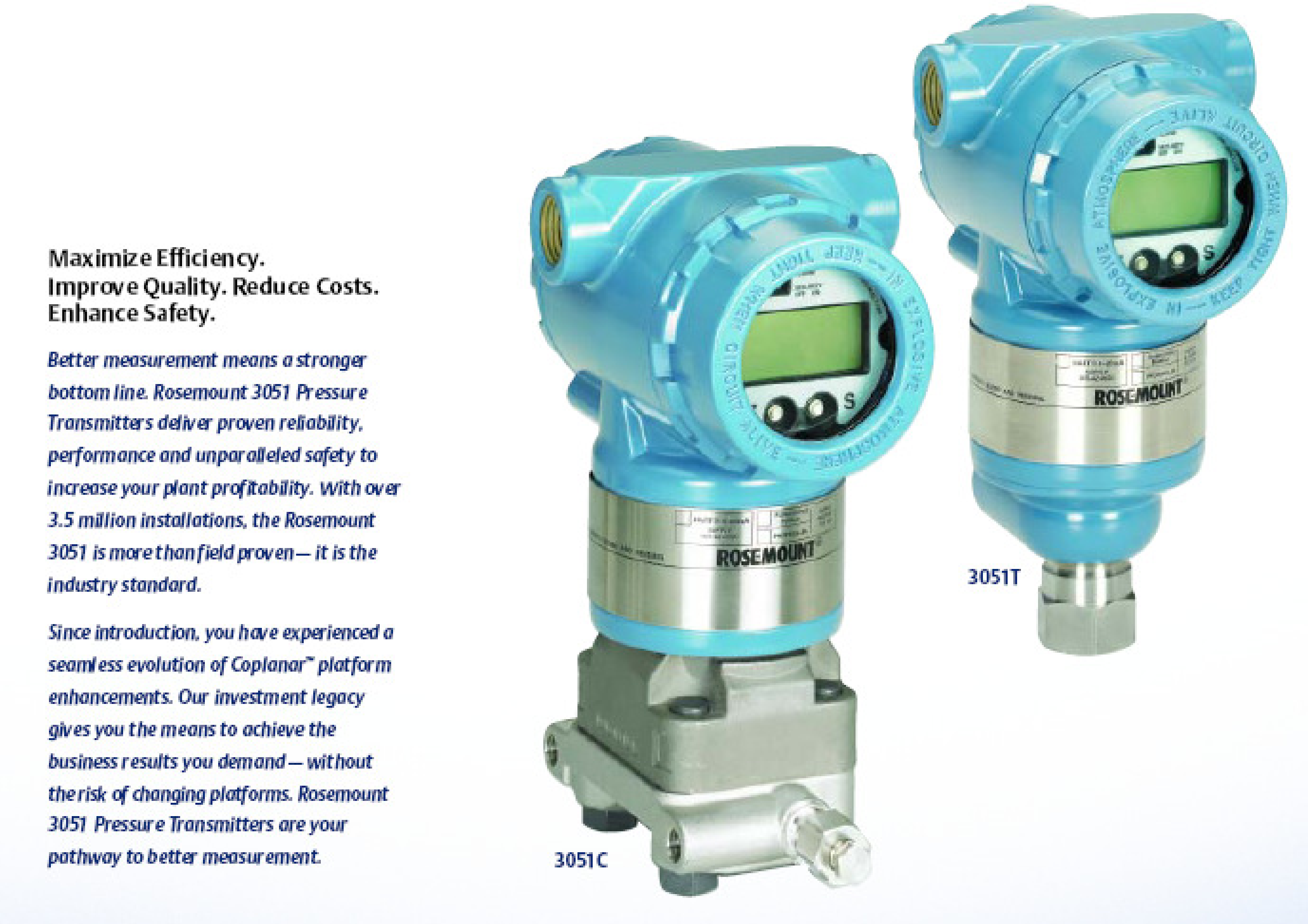
Unavailability = ?

$$\bar{A} = \frac{down\ time}{total\ time} = \frac{MDT}{MTBF+MDT} \approx \lambda * MDT$$

-> $\bar{A} = 10^{-5}$

We will need this metric when we look into safety later. This is an important metric if a failure is dormant meaning the function is not performed immediately.

Reliability in Product Descriptions



Maximize Efficiency.
Improve Quality. Reduce Costs.
Enhance Safety.

Better measurement means a stronger bottom line. Rosemount 3051 Pressure Transmitters deliver proven reliability, performance and unparalleled safety to increase your plant profitability. With over 3.5 million installations, the Rosemount 3051 is more than field proven — it is the industry standard.

Since introduction, you have experienced a seamless evolution of Coplanar™ platform enhancements. Our investment legacy gives you the means to achieve the business results you demand — without the risk of changing platforms. Rosemount 3051 Pressure Transmitters are your pathway to better measurement.

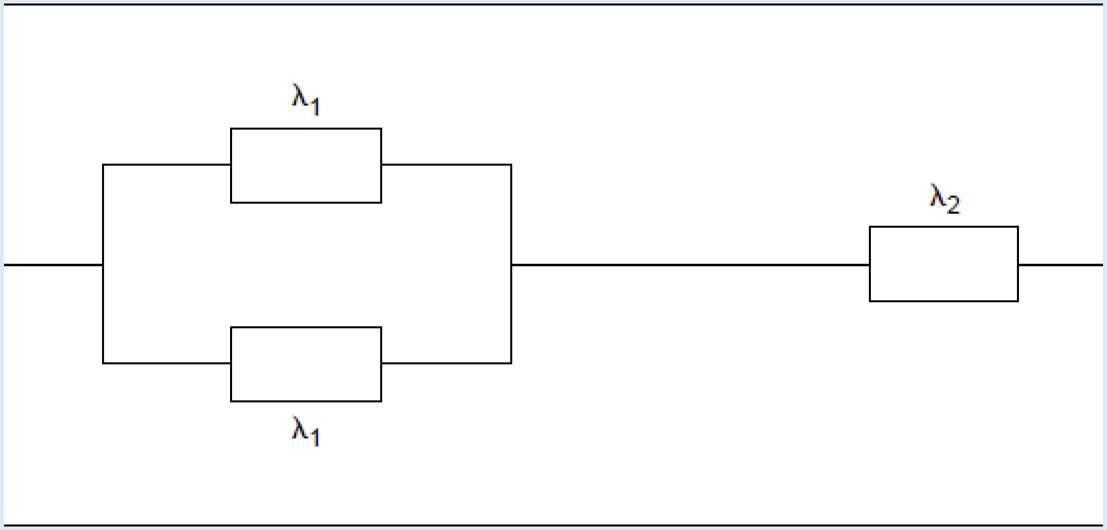
Source:
Rosemount

What is Rosemount marketing advertising with in this example?

Reliability Improvement

- We need two things to compare different architectures:
  - A probabilistic model – probability law
  - A notation – Reliability Block Diagram (RBD) which assume probabilistic independent blocks
  - Each block has a defined function, a failure mode with a failure rate

Source:
Smith: Reliability, Maintainability and Risk



The Bernoulli Experiment applied to Reliability

We have a total number of n identical components. For each component only two states are defined: “functioning” or “has failed”. Both states have a certain probability assigned.
The Bernoulli experiment gives us the probability of finding k (out of n) components in a functioning state.

We state:

P(functioning) = 1 – P (failed);
P(functioning) = p; P (failed) = q

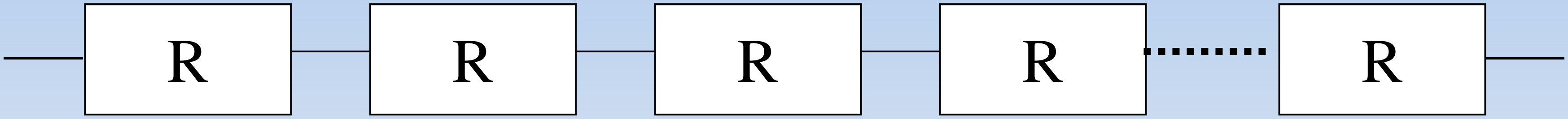
The Bernoulli Experiment II

The probability of k functioning components out of n total is P(n,p,k) = $\binom{n}{k}\ p^k\ q^{n-k}$

Now we need the probability that a system function (made out of components) is working -> reliability ("probability of survival")

P(n,p,k) = $\binom{n}{k}\ R^k\ (1-R)^{n-k}$ is the probability of having k functioning components in an assembly of n total.

Series Reliability Calculation



All n components above need to work such that the series assembly (system) is functioning.

The probability of having n functioning blocks out of n total is

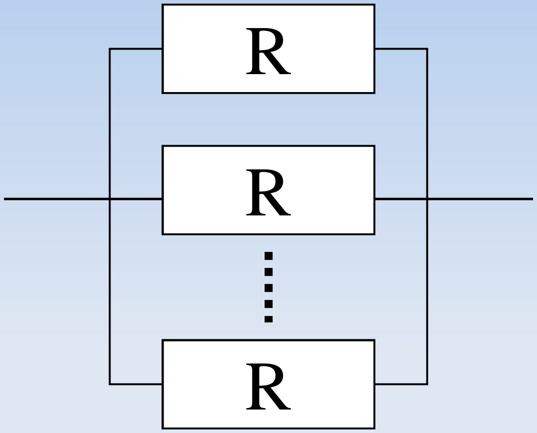
R_S = P(n,n,k) = $\binom{n}{n} R^n (1-R)^{n-n} = R^n$ when using a Bernoulli experiment

R_S= R × R × … × R = R^n when using the probability law for independent events

Parallel Reliability Calculation
- full active redundancy -

At least 1 component needs to be functioning in full active redundancy configuration.

Therefore, the assembly is working if n or (n-1) or ... or 1 component work.



n=2: 2 or 1 component must be functioning.

$$R_S = \binom{2}{2} R^2 (1-R)^0 + \binom{2}{1} R^1 (1-R)^1 = 2R - R^2$$

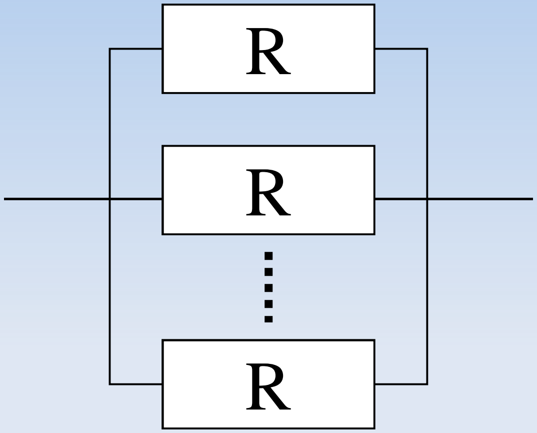
n=n:

$$R_S = \binom{n}{n} R^n (1-R)^0 + \ldots + \binom{n}{1} R(1-R)^n = 1-(1-R)^n$$

Parallel Reliability Calculation
- partial active redundancy -

At least m components need to be functioning in partial active redundancy configuration.

Therefore, the assembly is working if n or (n-1) or ... or m components work.



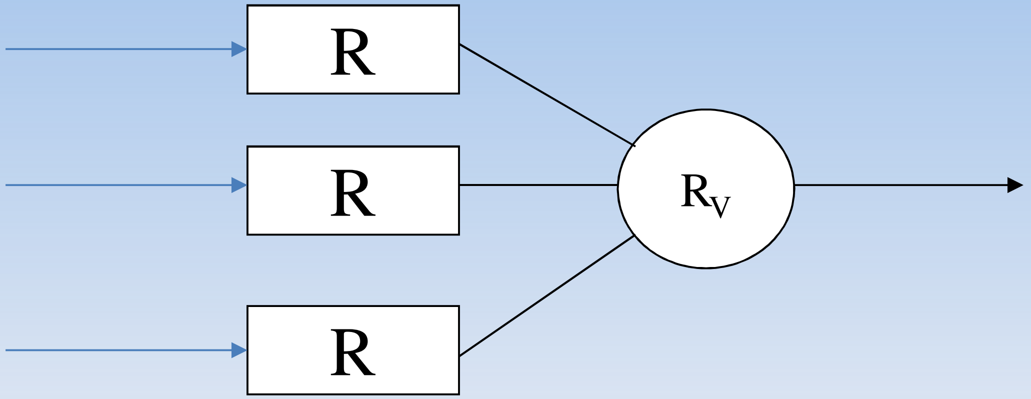
n=3: m = 2 (2oo3 = “two out of three”)

$$R_S = \binom{3}{3} R^3 (1-R)^0 + \binom{3}{2} R^2 (1-R)^1 = 3R^2 - 2R^3$$

n=N, m = M: (MooN = “M out of N”)

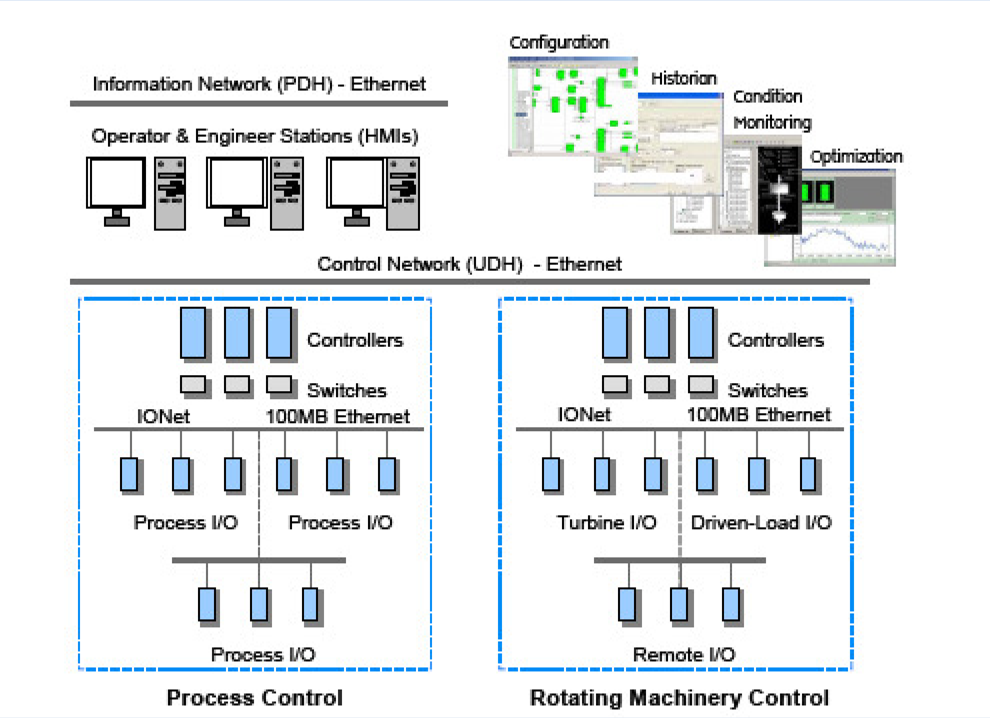
$$R_S = \binom{n}{n} R^n (1-R)^0 + \ldots + \binom{n}{m} R^m (1-R)^{n-m}$$

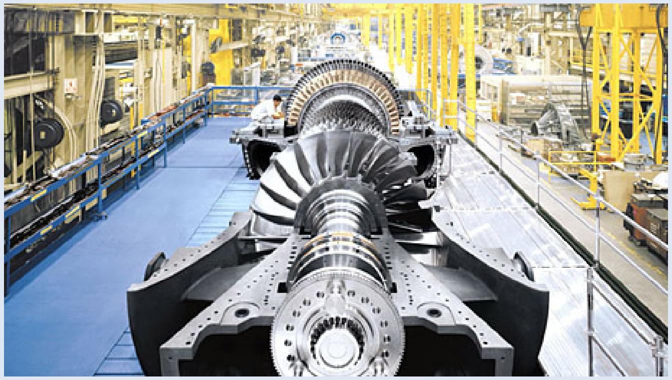
Partial Active Redundancy Example
- 2oo3 majority voter -



- Three identical inputs. One single real number output. Triple Modular Redundancy (TMR)
- Input stages have reliability R, Voter and output stage have reliability R_V
- One unit may fail but no more (partial redundancy)
- Reliability: $R_S = 3R^2 - 2R^3$

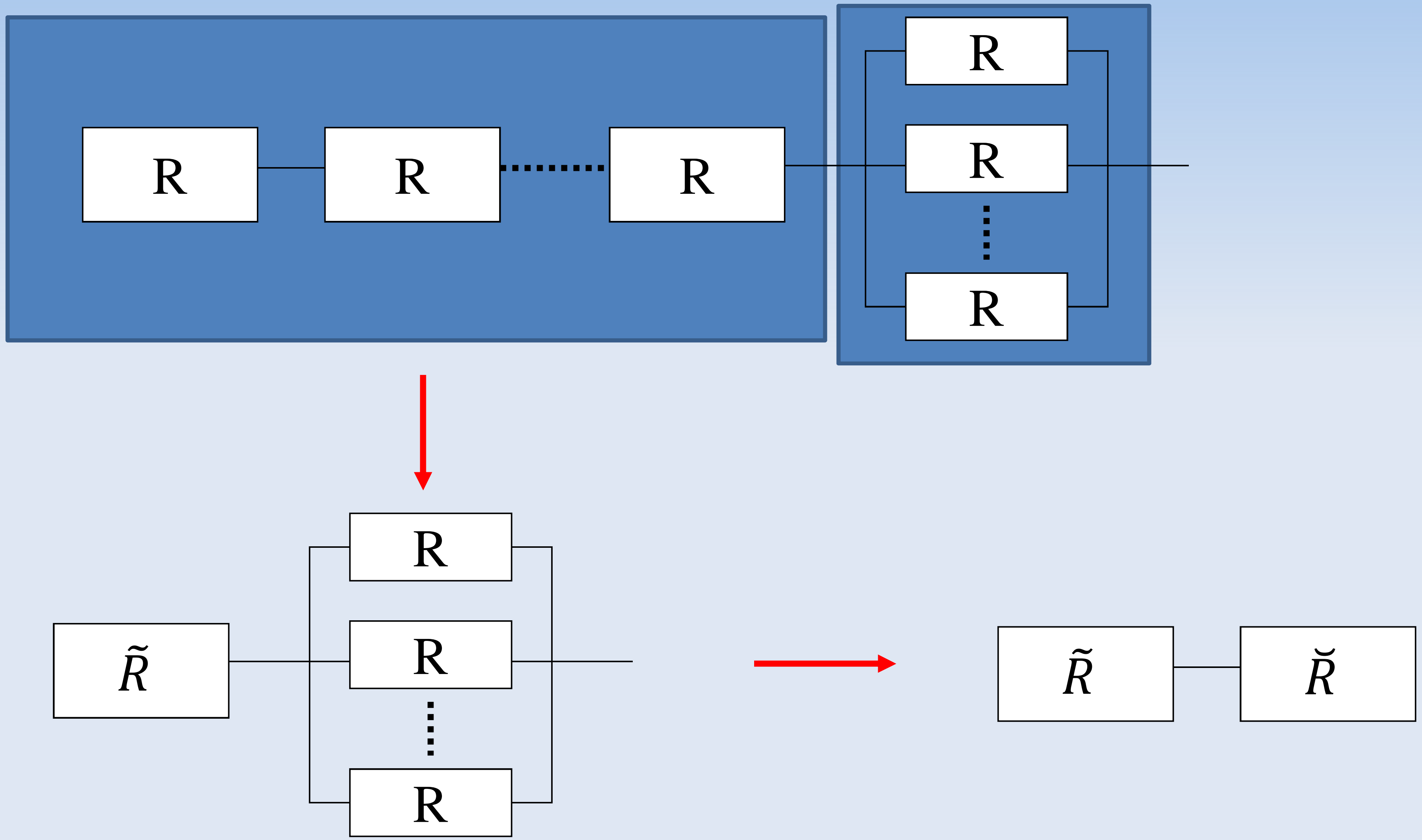
Partial Active Redundancy Example
- 2oo3 majority voter -



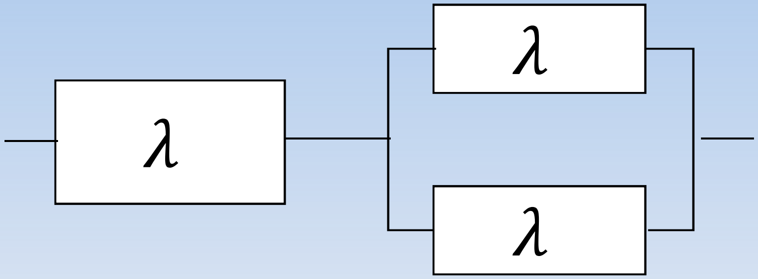


Source:
GE Energy

Complex Configurations



Complex Configuration Example



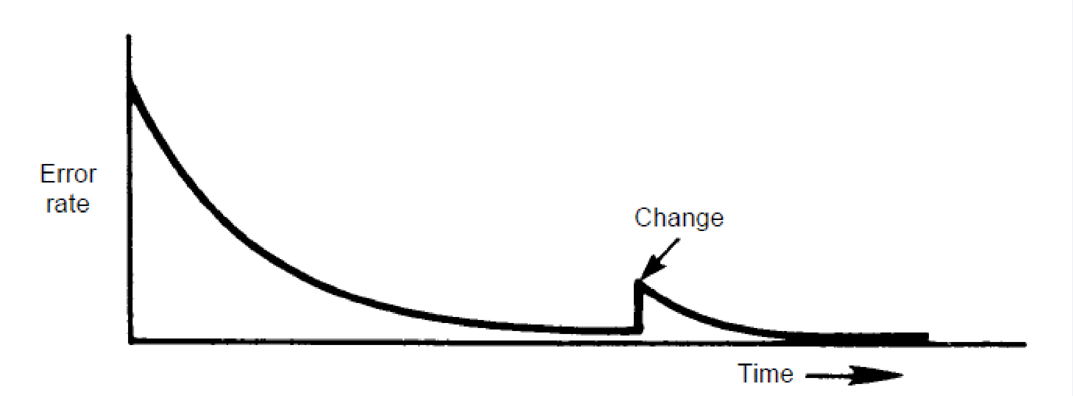
Calculate the MTBF of this system (S) made of identical components (C). $\lambda = const$.

$$R_S = R_C * (2R_C - {R_C}^2) = 2e^{-\lambda t} - e^{-3\lambda t}$$

$$\Theta = \int_0^\infty (2e^{-\lambda t} - e^{-3\lambda t})dt = \ldots = \frac{2}{3\lambda}$$

Systematic Failures
- software -

- A failure has been defined as deviation from the specification. This deviation can happen in two ways
  - Random (Hardware) – due to degradation (fault not present at time of commissioning).
    Random failures happen randomly in time. The rate is predictable (statistical quantification).
  - Systematic (Hardware and Software) – linked to a certain cause (fault (bug) present but maybe dormant at time of commissioning)
    Systematic failures happen systematically in time. They are not predictable. A rigorous design and qualification process must be applied.



Source:

Smith: Reliability, Maintainability and Risk

Questions?