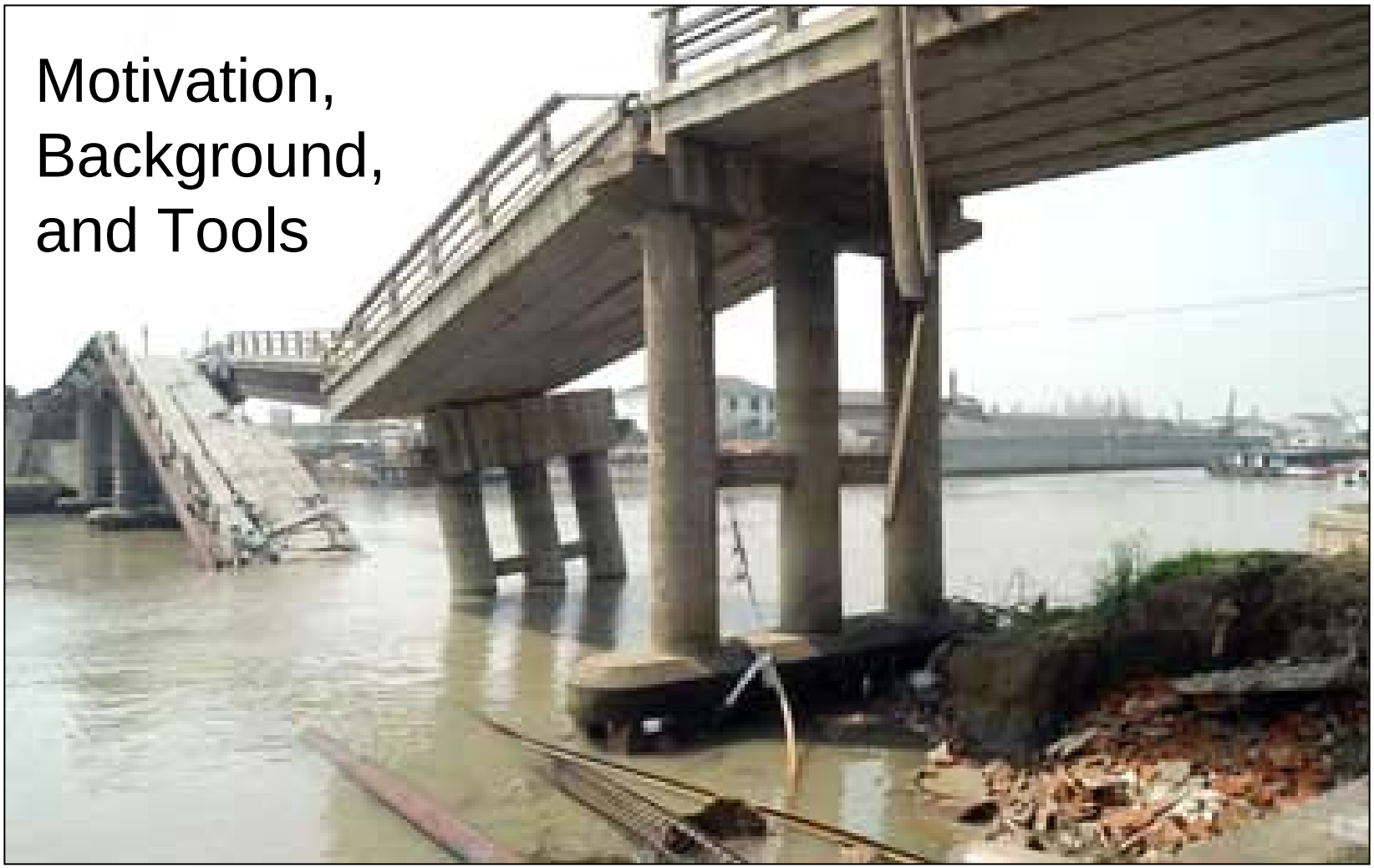
# Fault-Tolerant Computing

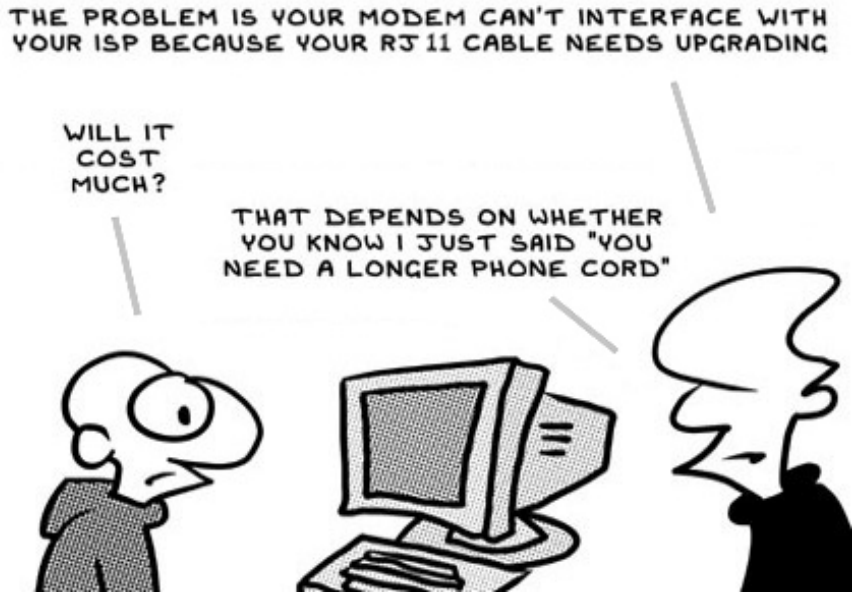


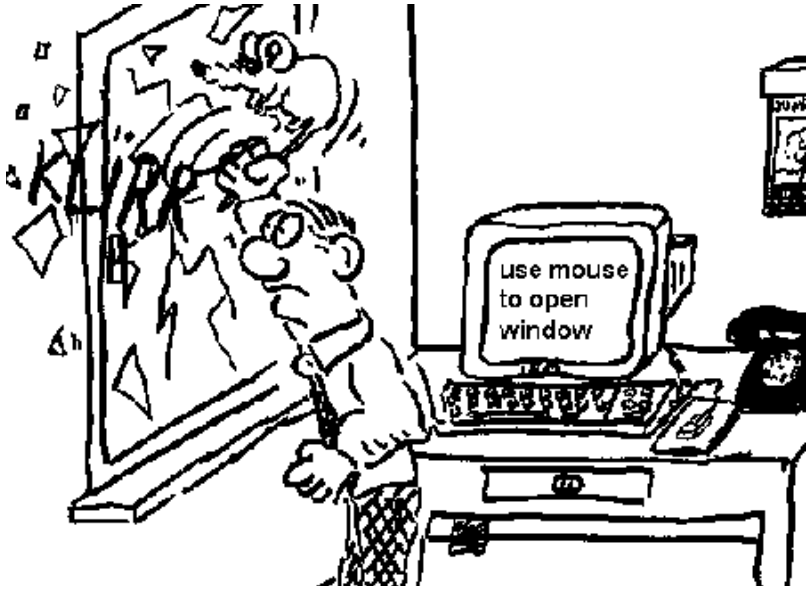
# About This Presentation

This presentation has been prepared for the graduate course ECE 257A (Fault-Tolerant Computing) by Behrooz Parhami, Professor of Electrical and Computer Engineering at University of California, Santa Barbara. The material contained herein can be used freely in classroom teaching or any other educational setting. Unauthorized uses are prohibited. © Behrooz Parhami

| Edition | Released | Revised | Revised |
|---|---|---|---|
| First | Oct. 2006 | | |

# Terminology, Models, and Measures for Dependability



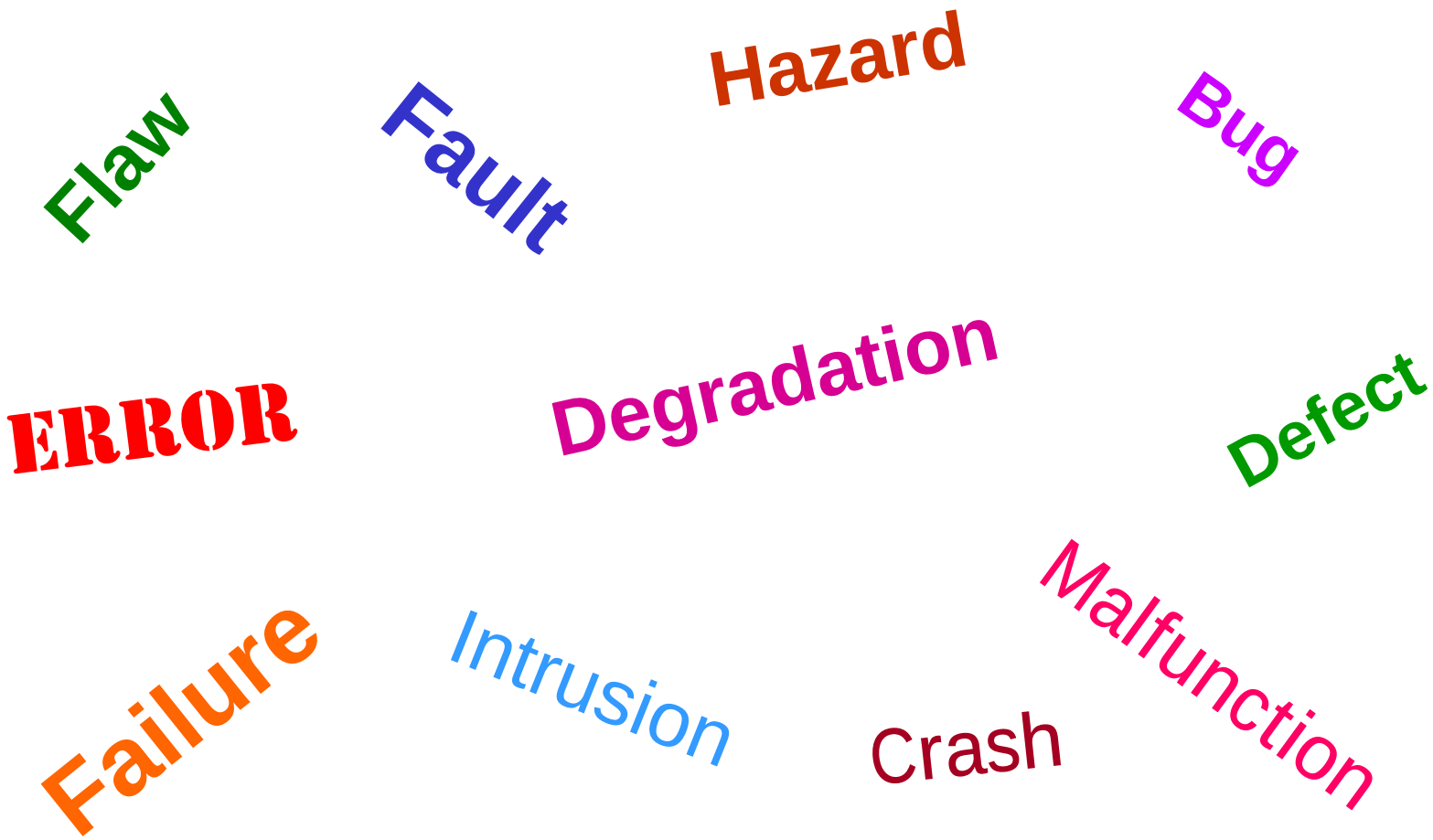





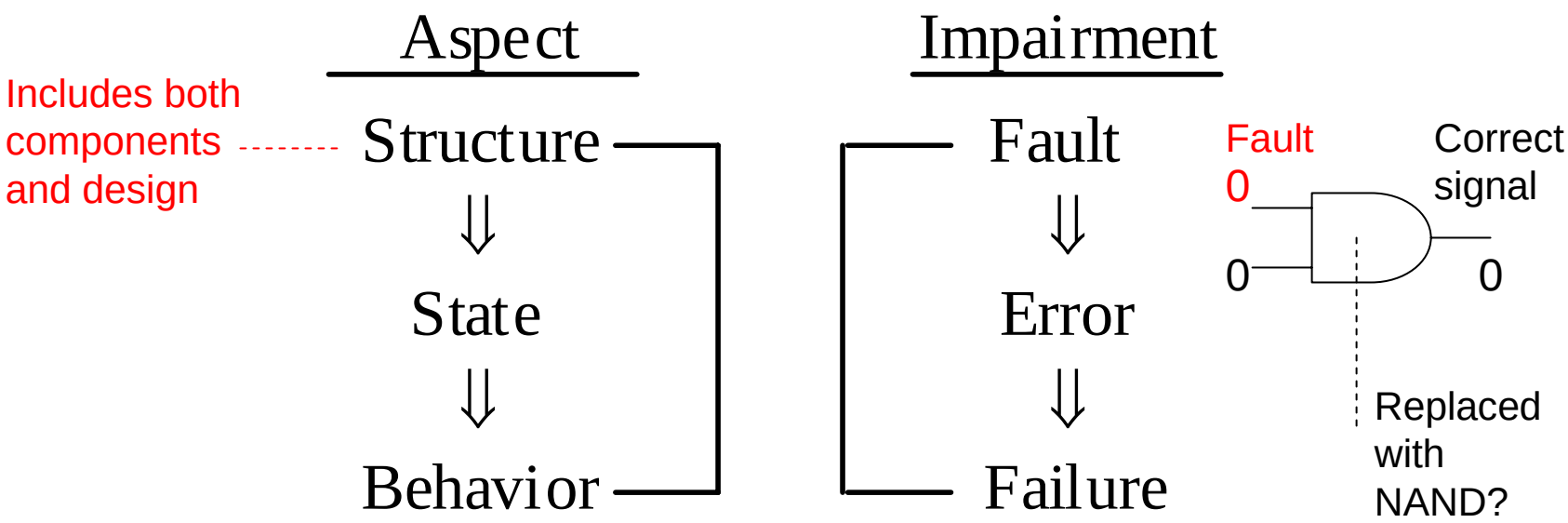
"Every time we successfully recover from a technical problem, the computer likes a high five."


"That foul smell is coming from your computer. You've got some old data in there that's gone bad."

# Impairments to Dependability



# The Fault-Error-Failure Cycle



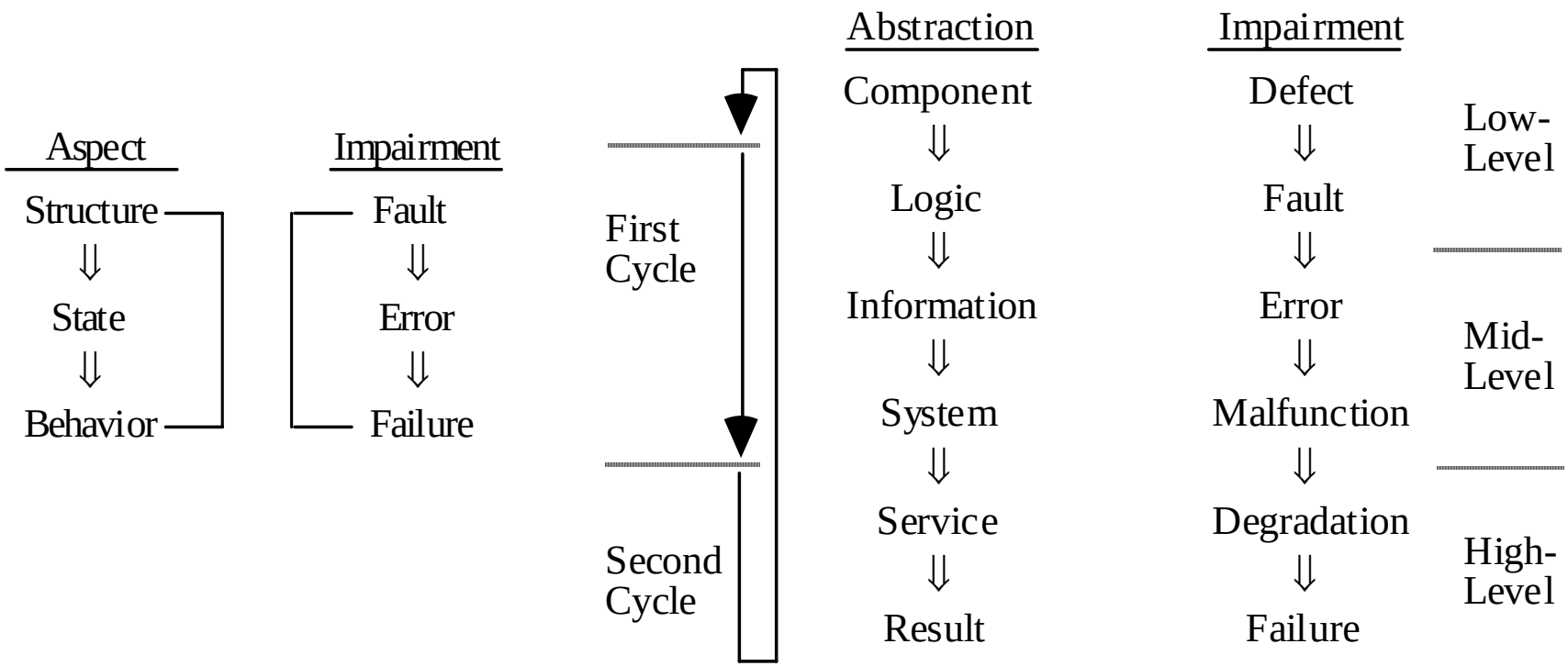

Schematic diagram of the Newcastle hierarchical model and the impairments within one level.

# The Four-Universe Model

| Universe | Impairment |
|---|---|
| Physical | Failure |
| ⇓ | ⇓ |
| Logical | Fault |
| ⇓ | ⇓ |
| Informational | Error |
| ⇓ | ⇓ |
| External | Crash |

Cause-effect diagram for Avižienis' four-universe model of impairments to dependability.

# Unrolling the Fault-Error-Failure Cycle




Cause-effect diagram for an extended six-level view of impairments to dependability.

# Multilevel Model

Component

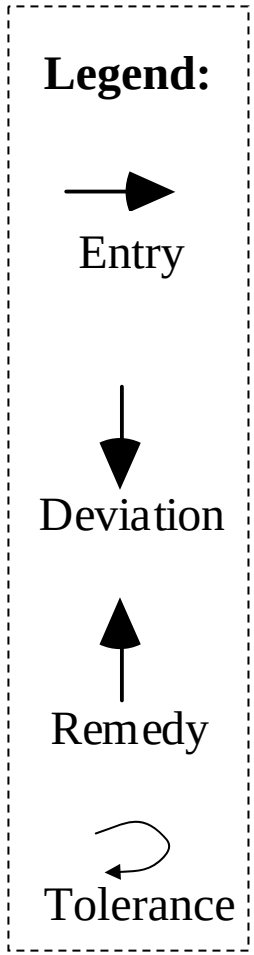
Logic

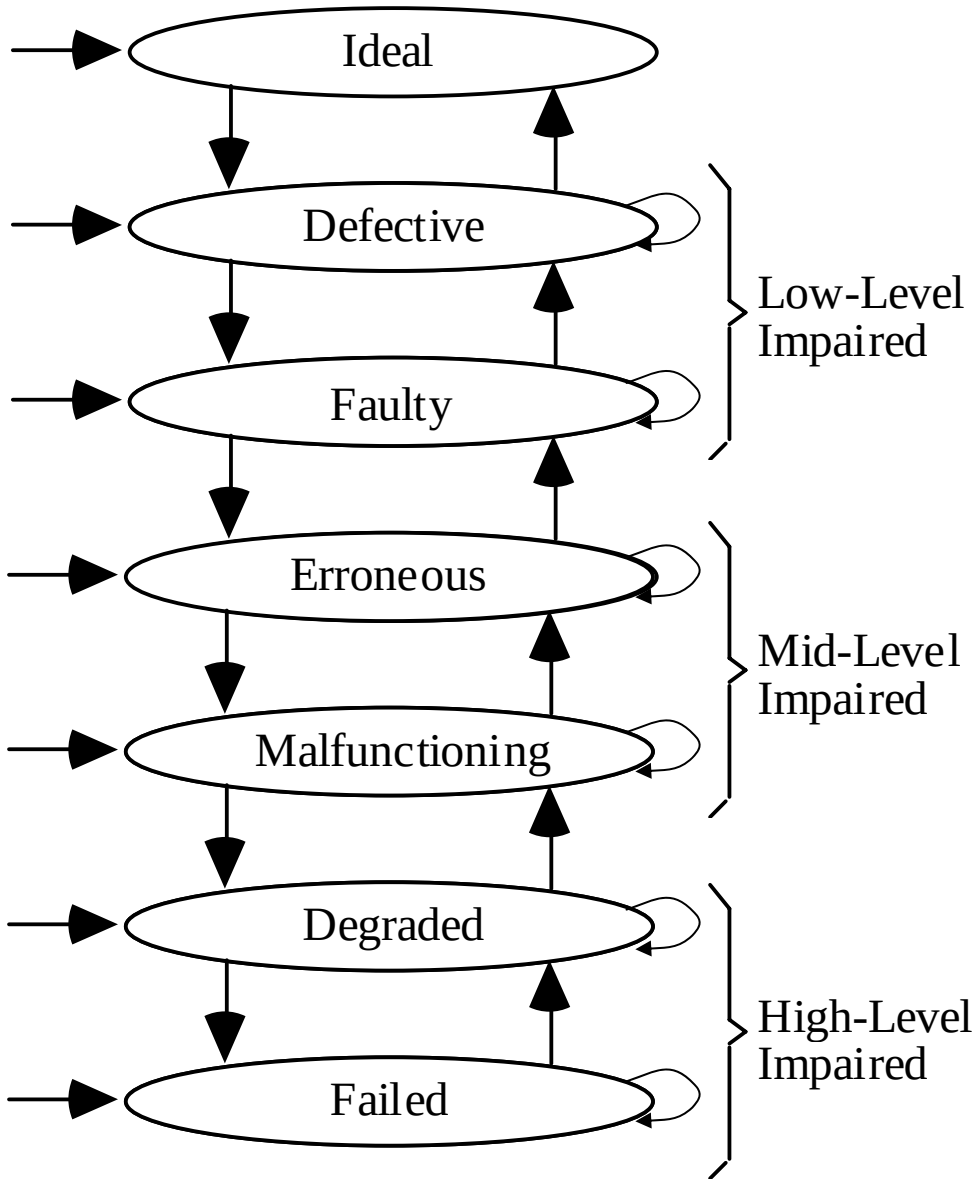
Information

System

Service

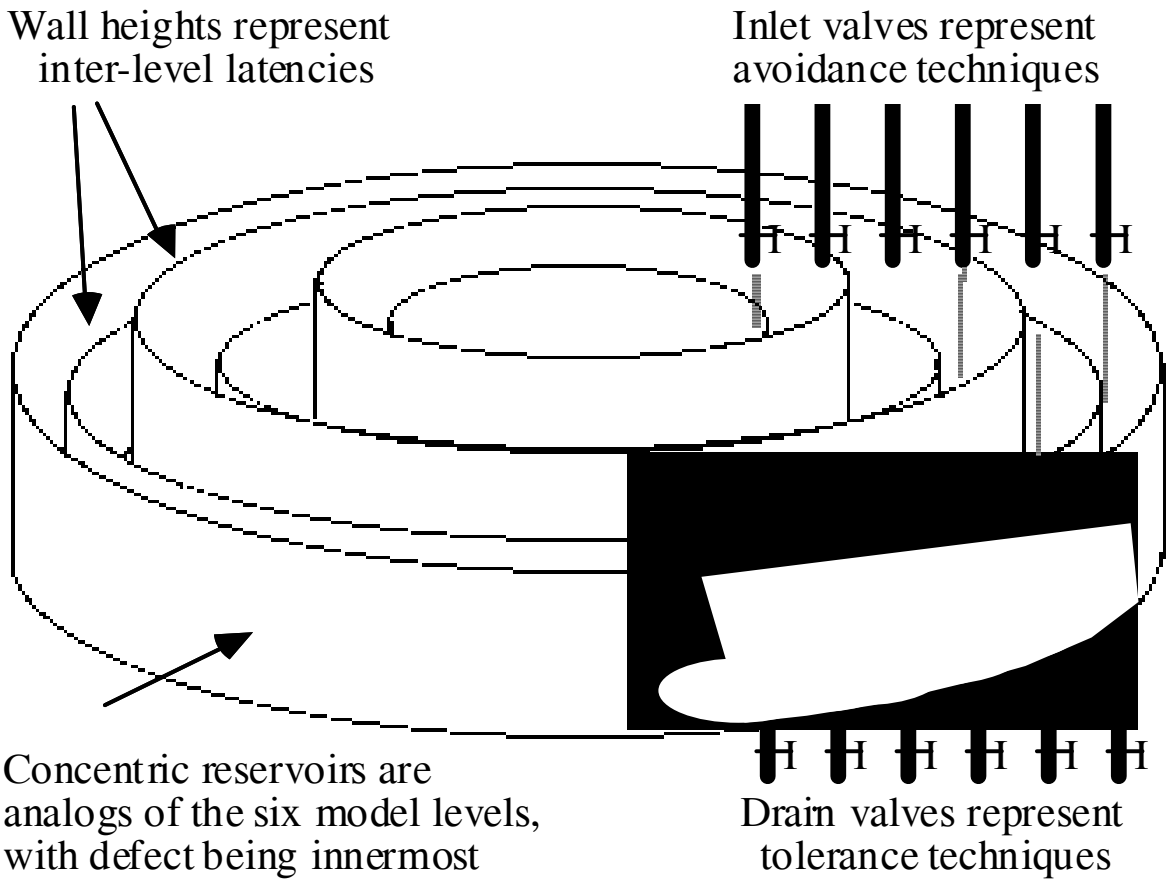
Result



# Analogy for the Multilevel Model

An analogy for our multi-level model of dependable computing.
Defects, faults, errors, malfunctions, degradations, and failures are represented by pouring water from above. Valves represent avoidance and tolerance techniques. The goal is to avoid overflow.



# Why Our Concern with Dependability?

**Reliability of *n*-transistor system, each having failure rate λ**

$$R(t) = e^{-n\lambda t}$$

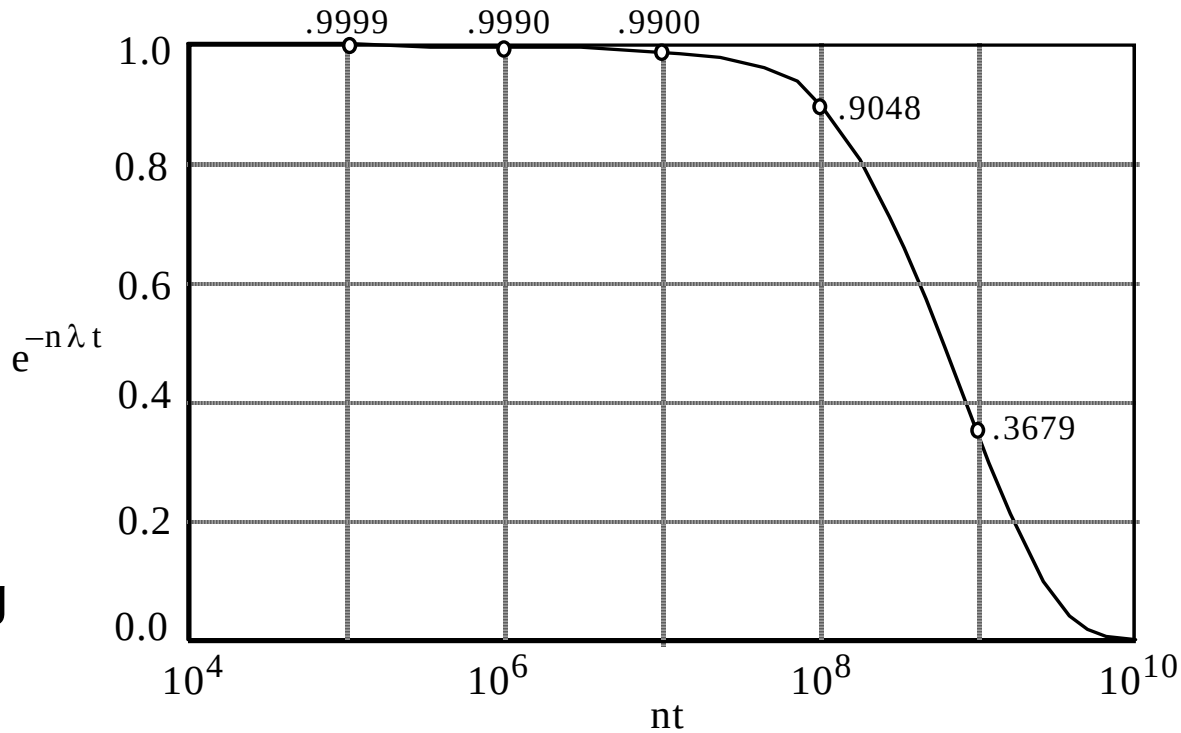
There are only 3 ways of making systems more reliable

**Reduce λ**

**Reduce *n***

**Reduce *t***

**Alternative:**
Change the reliability formula by introducing redundancy in system



# Highly Dependable Computer Systems

**Long-life systems: Fail-slow, Rugged, High-reliability**
Spacecraft with multiyear missions, systems in inaccessible locations
Methods: Replication (spares), error coding, monitoring, shielding

**Safety-critical systems: Fail-safe, Sound, High-integrity**
Flight control computers, nuclear-plant shutdown, medical monitoring
Methods: Replication with voting, time redundancy, design diversity

**High-availability: Fail-soft, Robust, High-availability**
Telephone switching centers, transaction processing, e-commerce
Methods: HW/info redundancy, backup schemes, hot-swap, recovery

Just as performance enhancement techniques gradually migrate from supercomputers to desktops, so too dependability enhancement methods find their way from exotic systems into personal computers

# Aspects of Dependability

# Concepts from Probability Theory

**Probability density function: pdf**

$f(t) = \text{prob}[t \le x \le t + dt] / dt = dF(t) / dt$

**Cumulative distribution function: CDF**

$F(t) = \text{prob}[x \le t] = \int_0^t f(x)\,dx$

**Expected value of *x***

$E_x = \int_{-\infty}^{+\infty} x\,f(x)\,dx = \sum_k x_k\,f(x_k)$

**Variance of *x***

$\sigma_x^2 = \int_{-\infty}^{+\infty} (x - E_x)^2\,f(x)\,dx$

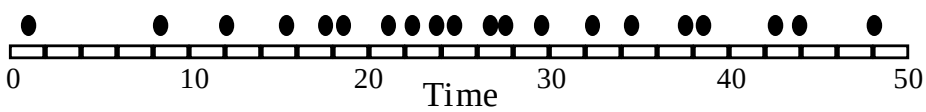
$= \sum_k (x_k - E_x)^2\,f(x_k)$

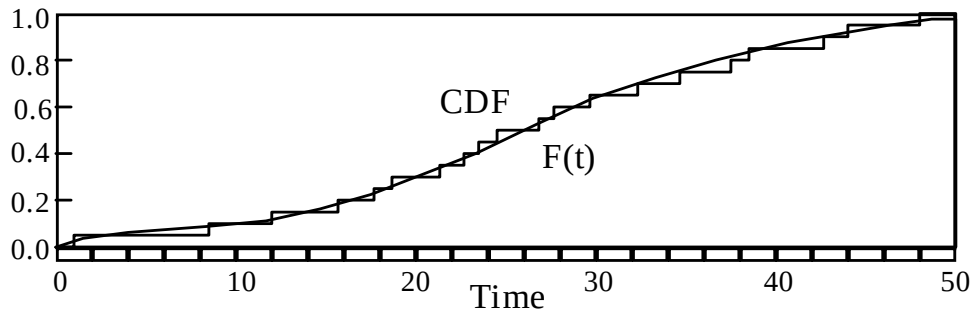
**Covariance of *x* and *y***

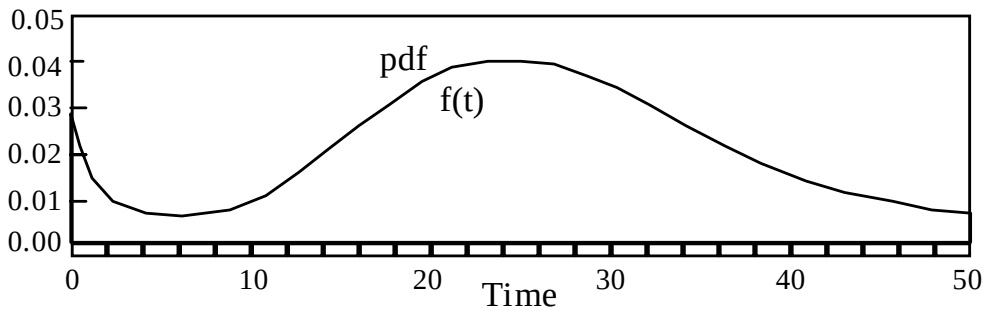
$\psi_{x,y} = E[(x - E_x)(y - E_y)]$

$= E[xy] - E_x E_y$

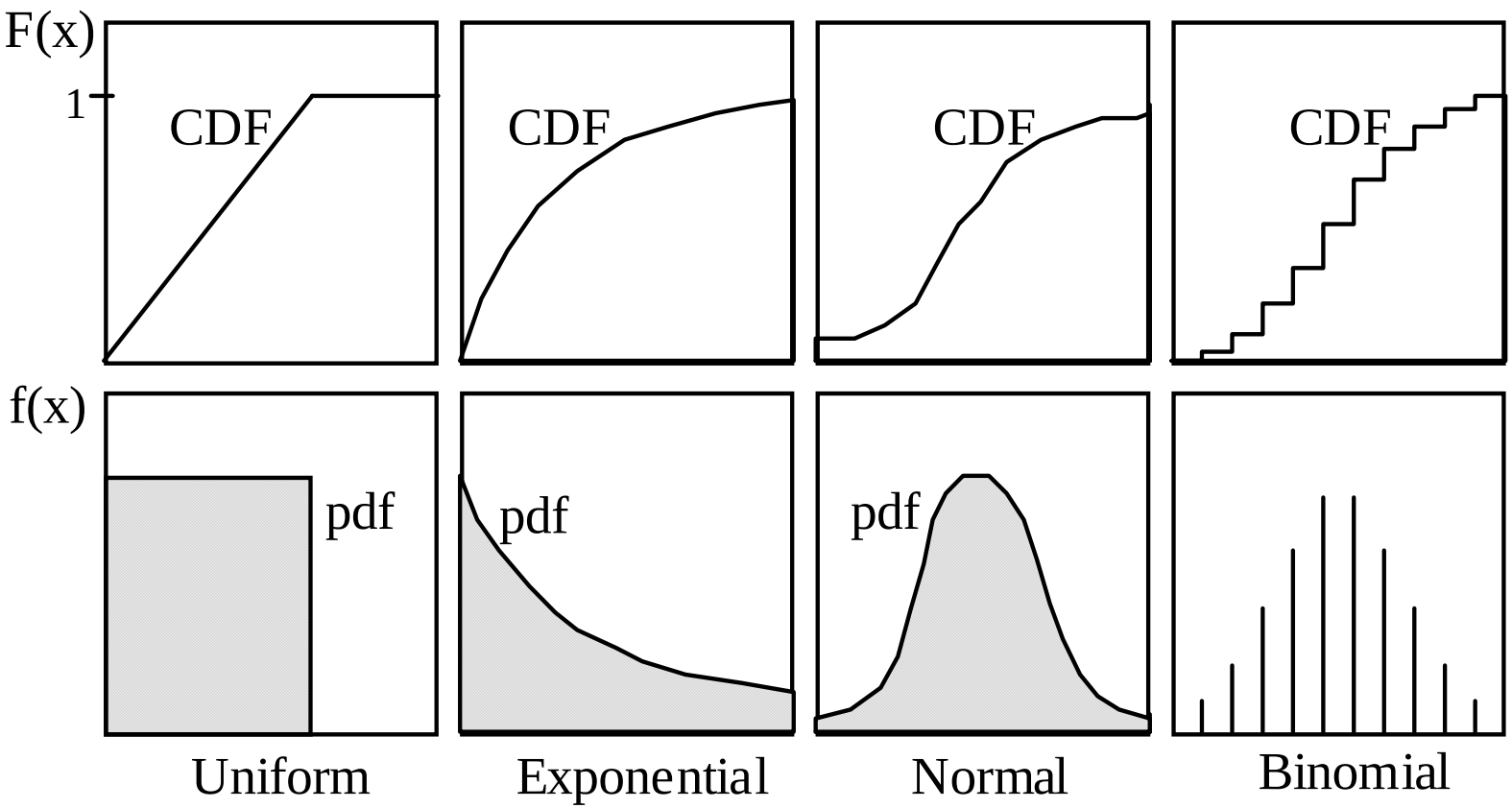
Liftimes of 20 identical systems



# Some Simple Probability Distributions



# Reliability and MTTF

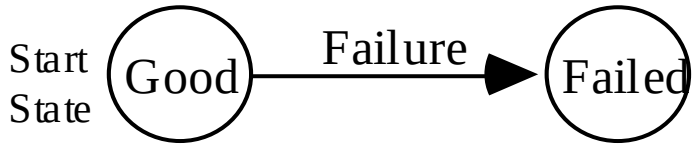
**Reliability: *R*(*t*)**

Probability that system remains in the "Good" state through the interval [0, *t*]

$R(t + dt) = R(t)\,[1 - z(t)\,dt]$

Hazard function

Two-state nonrepairable system




$R(t) = 1 - F(t)$ ------- CDF of the system lifetime, or its unreliability

Constant hazard function $z(t) = \lambda \Rightarrow R(t) = e^{-\lambda t}$
(system failure rate is independent of its age)

Exponential reliability law

**Mean time to failure: MTTF**

$\text{MTTF} = \int_0^{+\infty} t\,f(t)\,dt = \int_0^{+\infty} R(t)\,dt$

Expected value of lifetime

Area under the reliability curve (easily provable)

# Failure Distributions of Interest

Discrete versions

**Exponential: $z(t) = \lambda$**

$R(t) = e^{-\lambda t}$ MTTF = $1/\lambda$

**Geometric**

$R(k) = q^k$

**Rayleigh: $z(t) = 2\lambda(\lambda t)$**

$R(t) = e^{(-\lambda t)^2}$ MTTF = $(1/\lambda)\ \sqrt{\pi}\ /\ 2$

**Weibull: $z(t) = \alpha\lambda(\lambda t)^{\alpha-1}$**

$R(t) = e^{(-\lambda t)^\alpha}$ MTTF = $(1/\lambda)\ \Gamma(1 + 1/\alpha)$

**Discrete Weibull**

**Erlang:**

MTTF = $k/\lambda$

**Gamma:**

Erlang and exponential are special cases

**Normal:**

Reliability and MTTF formulas are complicated

**Binomial**

# Comparing Reliabilities

**Reliability difference:** $R_2 - R_1$

**Reliability gain:** $R_2 / R_1$

**Reliability improvement factor**

$\text{RIF}_{2/1} = [1 - R_1(t_M)] / [1 - R_2(t_M)]$

Example:

$[1 - 0.9] / [1 - 0.99] = 10$

**Reliability improv. index**

$\text{RII} = \log R_1(t_M) / \log R_2(t_M)$

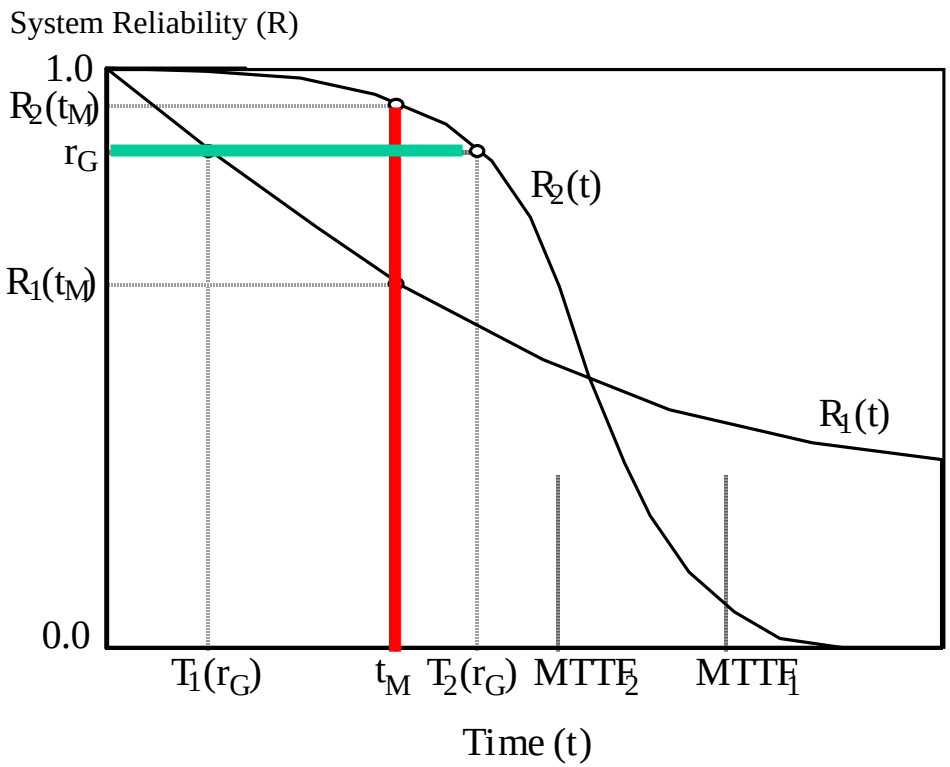
**Mission time extension**

$\text{MTE}_{2/1}(r_G) = T_2(r_G) - T_1(r_G)$

**Mission time improv. factor:**

$\text{MTIF}_{2/1}(r_G) = T_2(r_G) / T_1(r_G)$

Reliability functions for Systems 1/2



# Availability, MTTR, and MTBF

**(Interval) Availability: $A(t)$**

Fraction of time that system is in the "Up" state during the interval [0, $t$]

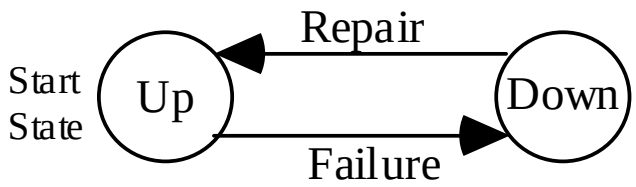
**Steady-state availability: $A = \lim_{t\to\infty} A(t)$**

**Pointwise availability: $a(t)$**

Probability that system available at time $t$

$A(t) = (1/t) \int_0^t a(x)\,dx$

Two-state repairable system




Availability = Reliability, when there is no repair

Availability is a function not only of how rarely a system fails (reliability) but also of how quickly it can be repaired (time to repair)

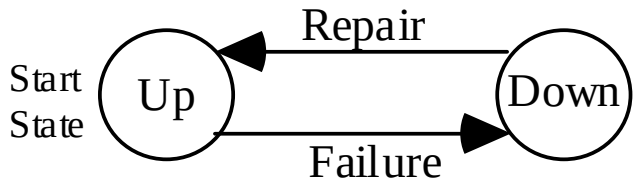
$$A = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} = \frac{\text{MTTF}}{\text{MBTF}} = \frac{\mu}{\lambda + \mu}$$

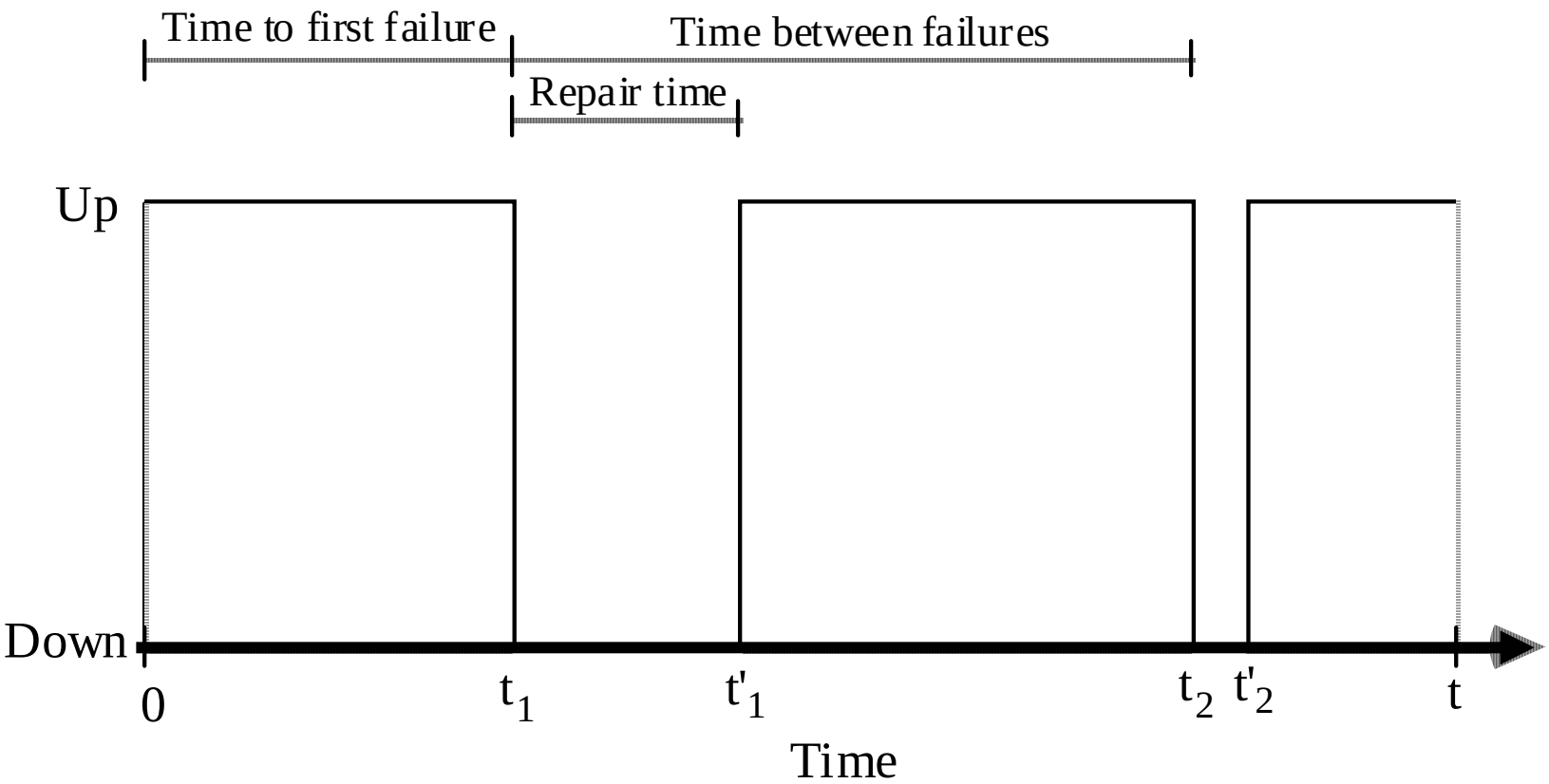
Repair rate $1/\mu$ = MTTR (Will justify this equation later)

In general, $\mu >> \lambda$, leading to $A \cong 1$

# System Up and Down Times

Short repair time implies
good **Maintainability** (**serviceability**)



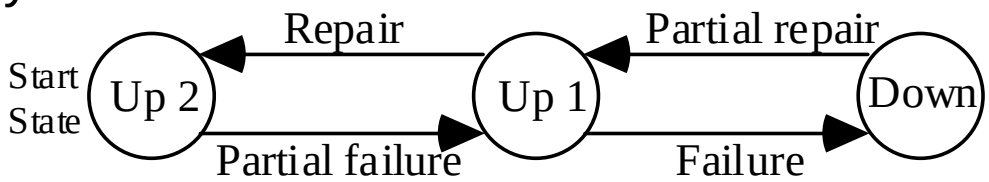



# Performability and MCBF

**Performability: *P***

Composite measure, incorporating both performance and reliability

Three-state degradable system




**Simple example**

Worth of "Up2" twice that of "Up1"

$p_{Upi}$ = probability system is in state Up*i*

$P = 2p_{Up2} + p_{Up1}$

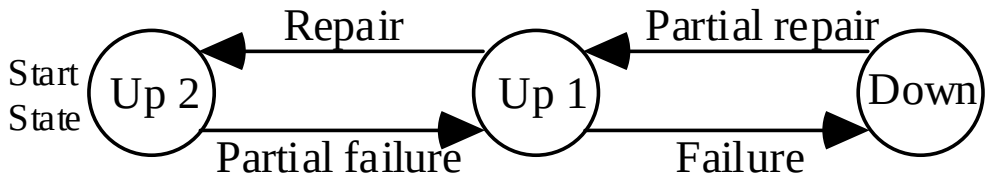
**Question:**
What is system availability here?

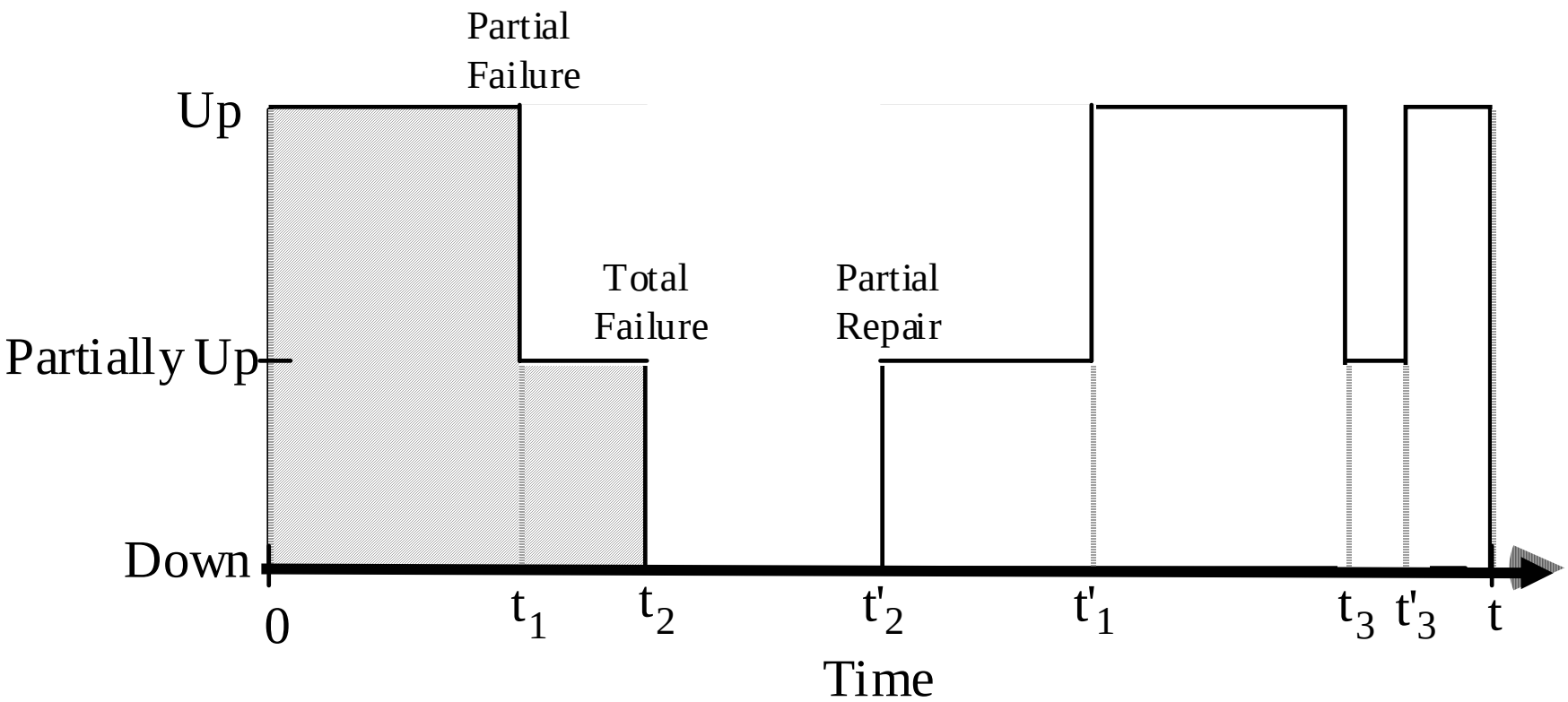
$p_{Up2} = 0.92$, $p_{Up1} = 0.06$, $p_{Down} = 0.02$, $P = 1.90$
(system performance equiv. To that of 1.9 processors on average)

Performability improvement factor of this system (akin to RIF) relative to a fail-hard system that goes down when either processor fails:
PIF = $(2 - 2 \times 0.92) / (2 - 1.90) = 1.6$

# System Up, Partially Up, and Down Times

Important to prevent direct transitions to the "Down" state (**coverage**)



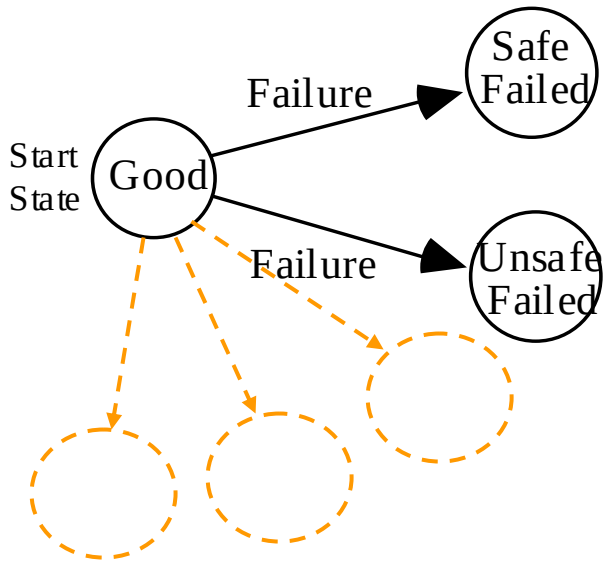



# Integrity and Safety

**Risk: Prob. of being in "Unsafe Failed" state**

There may be multiple unsafe states, each with a different consequence (cost)

Three-state fail-safe system




**Simple analysis**

Lump "Safe Failed" state with "Good" state; proceed as in reliability analysis

**More detailed analysis**

Even though "Safe Failed" state is more desirable than "Unsafe Failed", it is still not as desirable as the "Good" state; so keeping it separate makes sense

For example, if a repair transition is introduced between "Safe Failed" and "Good" states, we can tackle questions such as the expected outage of the system in safe mode, and thus its availability