

Software Reliability

Introduction

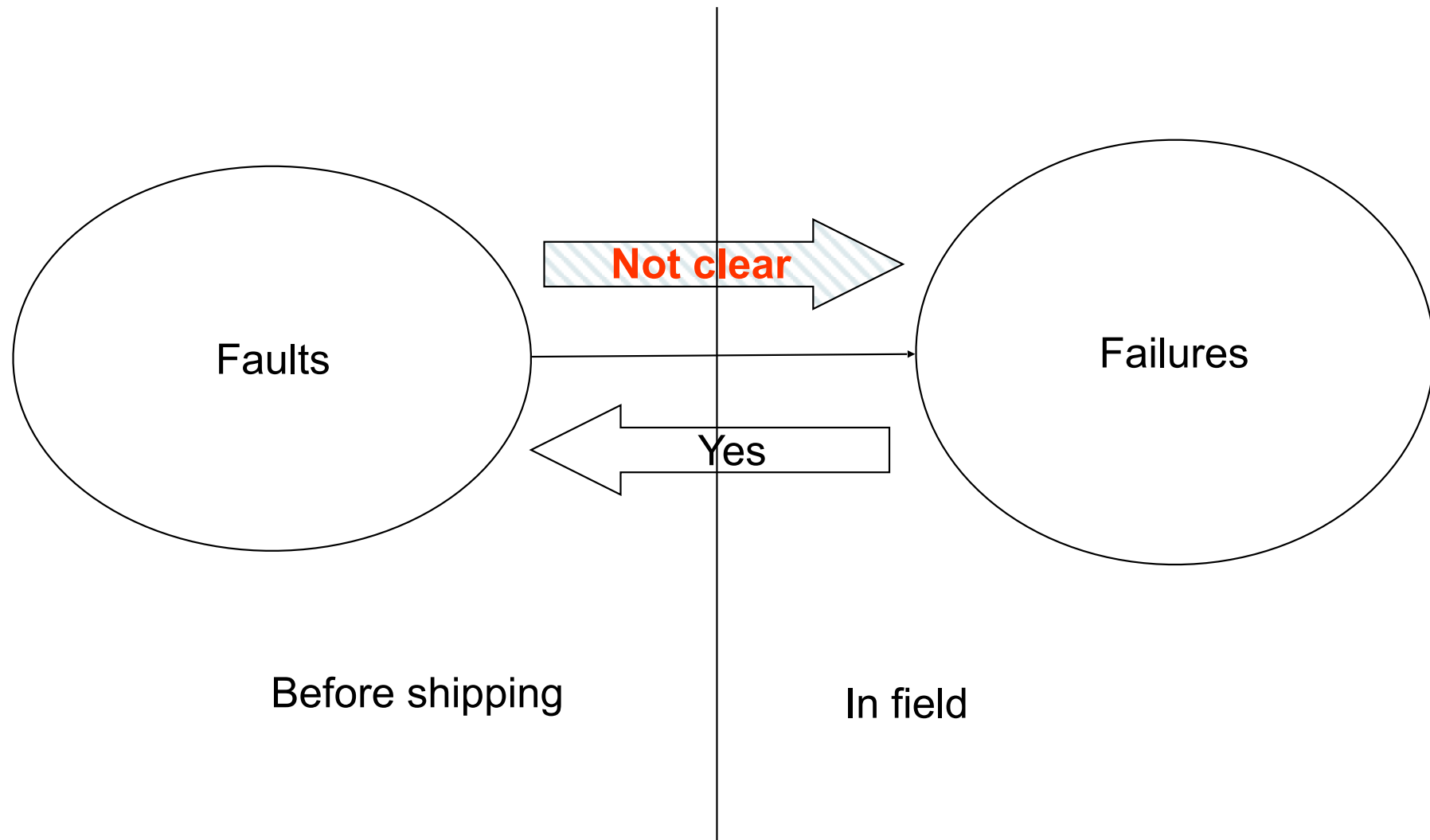
Barbara Russo

SwSE - Software and Systems Engineering

Lessons learned

- Reliability concerns the study of failures as effects of an error
- There are different notions of *effect of an error*
 - Often the difference concerns where and when an error occurs and who is involved in the event
- There is a not clear bidirectional relationship between faults (**cause**) and failures (**effect**)

Relationship btw Faults & Failures



Lessons learned

- Faults describe reliability of the system as a **internal** product attribute, like complexity, functionality etc.
- Failures describe reliability as an **external** product attribute, like usability, robustness, etc

Reliability as a Quality Attribute

- There are many different models for software quality, but in almost all models, reliability is one of the criteria, attribute or characteristic that is incorporated
- **ISO 9126** [1991] defines six quality characteristics, one of which is reliability.
- **ISO/IEC 25010**, which supersedes ISO/IEC 9126-1, March 2011
 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (**SQuaRE**) — System and software quality models (last reviewed in 2017)

- ISO 25010 has eight product quality characteristics:
 - Reliability has a new sub-characteristic :
Availability

Software Reliability Management

- IEEE 982.1-1988 defines Software Reliability Management as
- "The process of optimizing the reliability of software through a program that emphasizes software **error prevention, fault detection and removal**, and the use of **measurements to maximize reliability** in light of project constraints such as **resources, schedule and performance.**"

Did you ever performed any of these?

- Software reliability comprises three activities:
 - **Error/Faults prevention**
 - **Fault detection and removal**
 - **Measurements support the first two activities and predict future failures**
- **Fault tolerance:** how to deliver correct service in the presence faults

Example of error/fault prevention

Barbara Russo

SwSE - Software and Systems Engineering

Error prevention

- To increase the reliability by preventing software errors;
- The focus must be on
 - Comprehensive requirements and comprehensive testing plan, ensuring all requirements are tested
 - Maintainability of the software
 - Ensure the code can easily be engineered without injecting additional errors

Fault prevention in the requirements

- Seven measures
 - **Lines of Text** - Physical lines of text as a **measure of size**
 - **Imperatives** - Words and phrases that command that something must be done or provided. The number of imperatives is used as a base **requirements count**
 - **Continuances** - Phrases that follow an imperative and introduce the specification of requirements at a lower level, for a supplemental **requirement count**
 - **Directives** – References provided to figures, tables, or notes
 - **Weak Phrases** - Clauses that are apt to cause uncertainty and leave room for multiple interpretation **measure of ambiguity**
 - **Incomplete** – Statements within the document that have TBD (To be Determined) or TBS (To Be Supplied)
 - **Options** - Words that seem to give the developer latitude in satisfying the specifications but can be **ambiguous**

Fault prevention in coding

- For example, Compilers and Type Systems are common instruments to perform Fault Prevention
 - Syntax Fault
 - Type Mismatch Fault

Fault prevention in coding

- **Constants Propagation:** finding the program variables assigned once and never more (to be replaced by constants)
- **Dead variables:** a variable x is dead at some point in the program if its value is no longer necessary, otherwise is alive.

Fault prevention in coding

- Information flow security:
 - Example: in a program data is partitioned in two different security levels.
 - These levels cannot be violated: for example, do not copy a high level variable into a lower level variable

Measurements in reliability

- To support fault prevention
- To detect faults
- To remove faults
- To predict future occurrences of failures

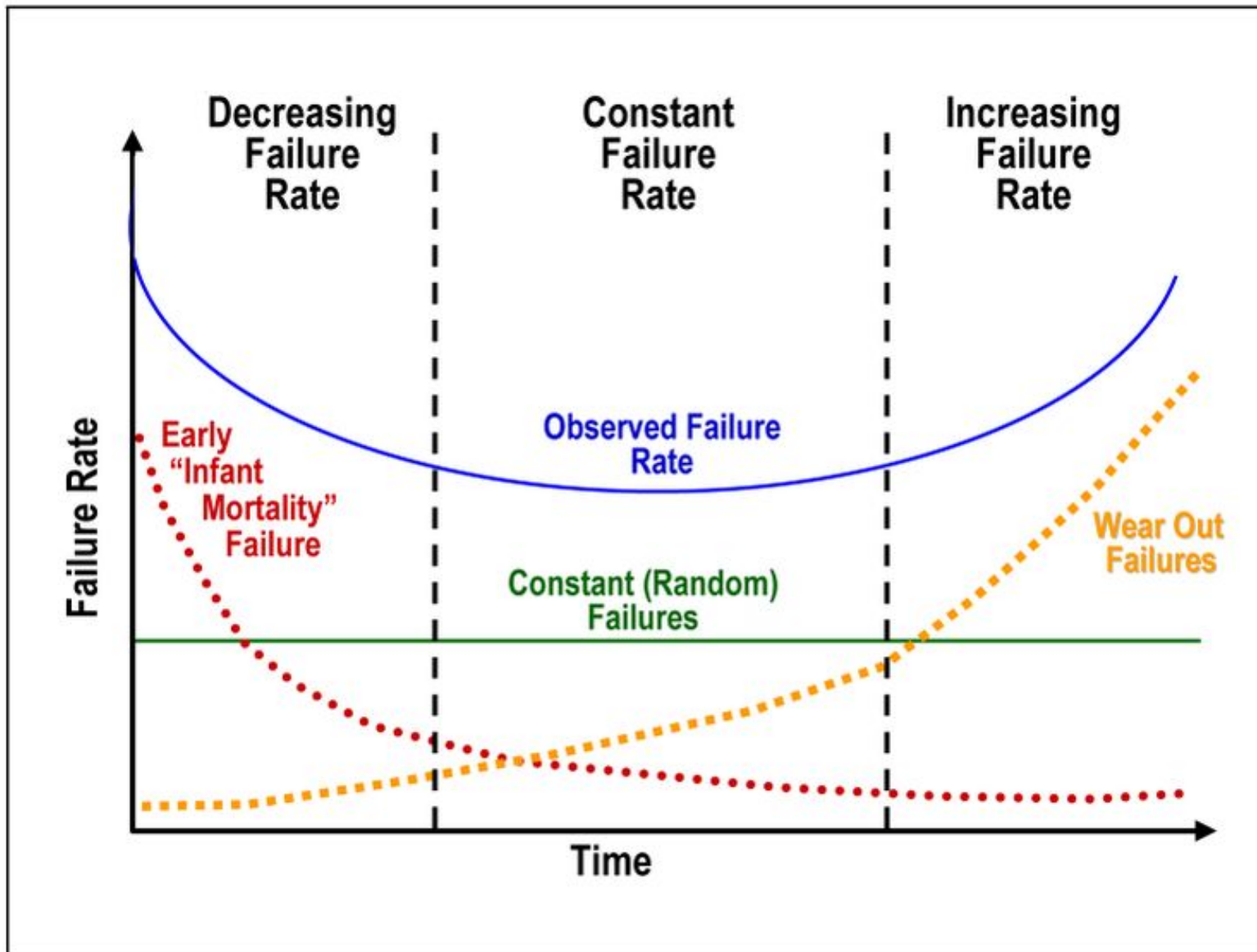
Measurement in Software Reliability

- From Hardware reliability
 - Static analysis of failures, failure occurrences, failure fix...
- Initial research:
 - Models defined on failure rates
 - Models derived from hardware reliability
- Later:
 - Deviation of hardware models
 - Models coming from social and economic sciences

Failure rate

- The probability that in a given interval of time a failure occurs **given that no failure has occurred before**
 - Measure: number of failures per hour

Bath tube curve: three behaviors



Hardware & software reliability

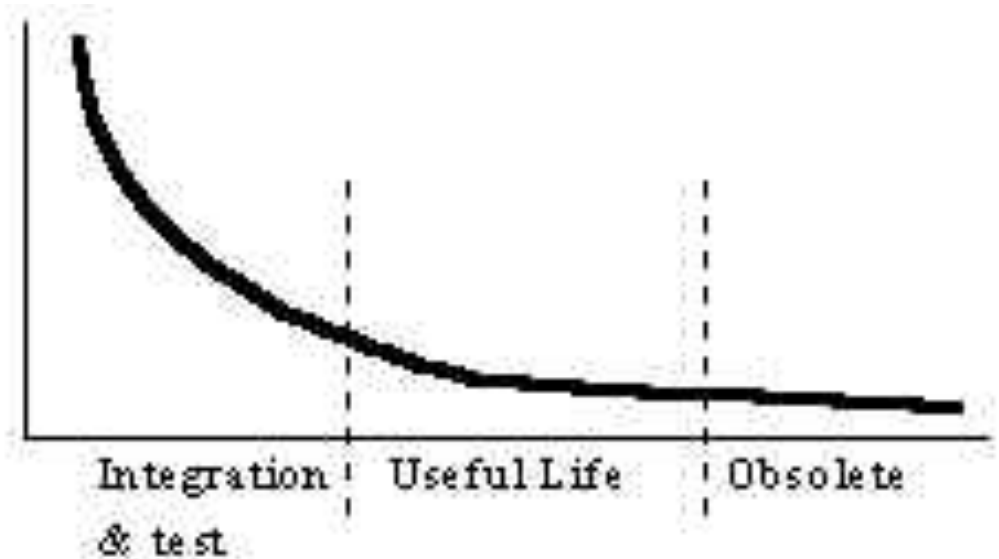
- Software reliability attempted to **extrapolate the mathematics of hardware** reliability to the **prediction of software** - W.H. von Alvin, in his book *Reliability Engineering*
- However, **most hardware reliability models** are predicted on failures due to **wear out rather than to design defects** - Pressman (R. S. Pressman. *Software Engineering: A Practitioner's Approach*, 6th Edition, McGraw Hill, 2005.)
- Software failures can be traced back to **design or implementation failures**

Hardware and software reliability

- Hardware reliability discipline since '70



Hardware Failure Rate



Software Failure Rate

Hardware and software reliability

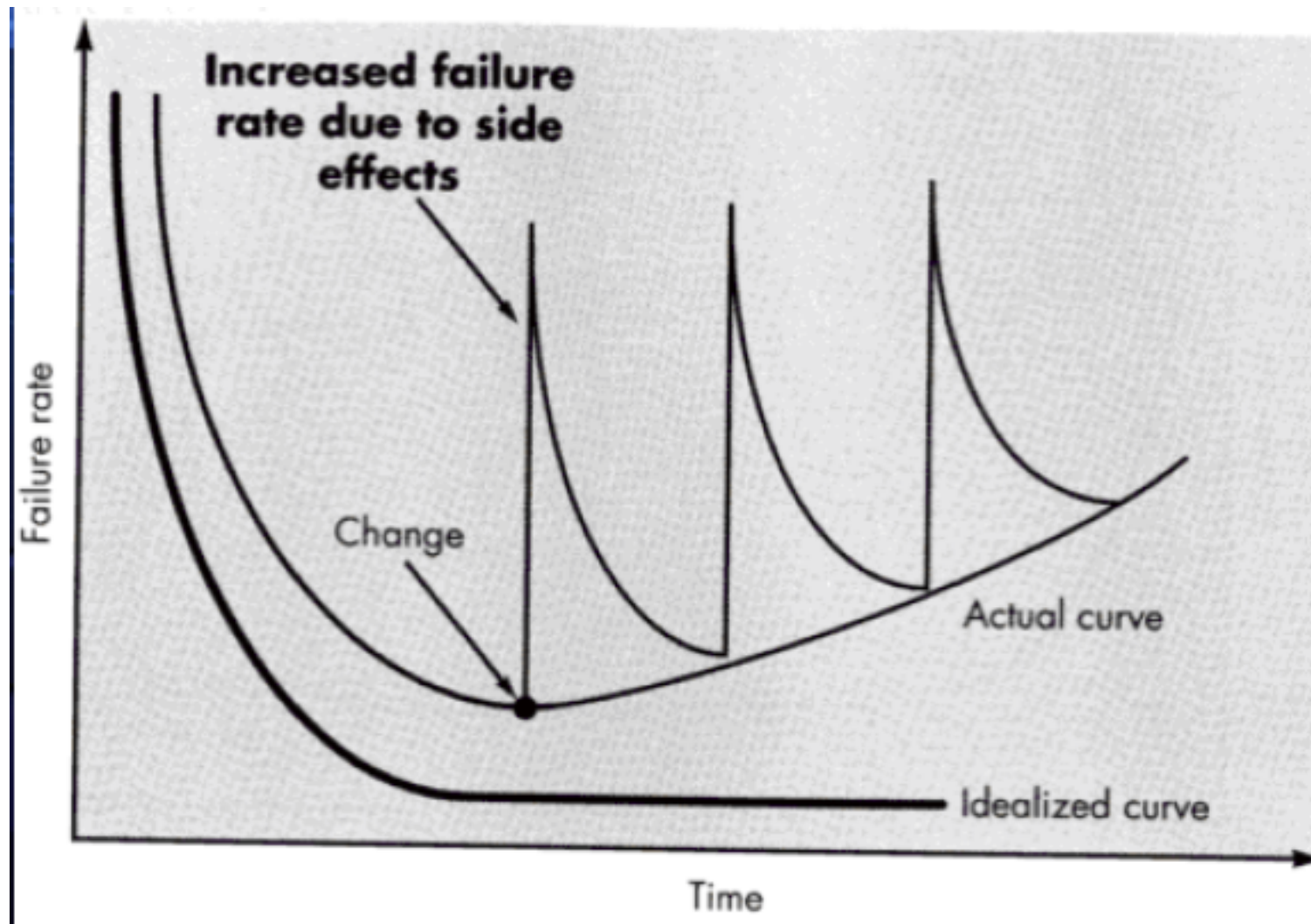
- **Difference** between hardware failure rate and software failure rate
- For **hardware**, the **initial number of faults is high** but then decreases as the **faulty components are identified and removed** or the components **stabilize**
- The component then enters the useful life phase
- As the component physically **wears out**, the fault rate starts to increase

Hardware and software reliability

- Software has a different fault / error rate
 - The error rate is at the highest level at **integration and test**
 - As it is tested, errors are identified and removed
 - This removal continues at a slower rate during its operational use
 - **The number of errors continually decreases, assuming no new errors are introduced**

- Software does not have **moving parts** and does **not physically wear out** as hardware, but is **does outlive its usefulness and becomes obsolete**

Software reliability curve



Software Reliability theory

- Firstly ...few statistical notions
 - Random variables
 - Conditional probability
- Reliability theory
 - Definitions
 - Failure rate
 - Mean Time To Failure
 - Failure Intensity

Few notion of statistics

- **Discrete random variable X**
 - E a **countable** space of independent and complementary **events**
 - O a set of output values of these events
 - **Probability density function (pdf), $x \in O$**

$$p(x) = P \{X=x\}$$

- Example, the probability of X =“tossing coin”
- $p(\text{head})=P(X=\text{“head”}) = 1/2$
- Note: a function $p(x)$ is a density function iff
$$p(x) \geq 0 \text{ and } \sum p(x_i) = 1$$

Few notion of statistics

- When there is an order on x , we can define the
- Cumulative distribution function (**cdf**)

$$P(X \leq x) = \sum p(x_i) \text{ for } x_i \leq x$$

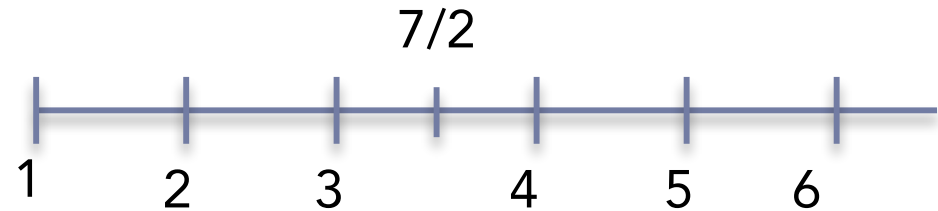
Few notion of statistics

- Expected mean of X
 - $E[X] = \sum x_i p(x_i)$
 - Weighted average of x values weighted by the probability density of the x values

- Variance of X
 - $\text{Var}(X) = E[X^2] - (E[X])^2 = E[(X - E[X])^2]$
 - and $E[X^2] = \sum x_i^2 p(x_i)$
 - $\text{Var}(X) = \sum x_i^2 p(x_i) - (\sum x_i p(x_i))^2$

Example

- S = set of die sides
- X “roll of a die”
- $p(x) = 1/6$ for every x in S
- $E[X] =$



$$1 * 1/6 + 2 * 1/6 + 3 * 1/6 + 4 * 1/6 + 5 * 1/6 + 6 * 1/6 = 7/2$$

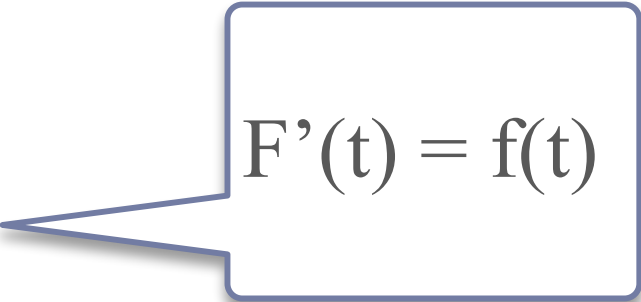
- $\text{Var}(X) = 1/6 + 4/6 + 9/6 + 16/6 + 25/6 + 36/6 - 49/4 = 35/12$

Few notion of statistics

- **Continuous random variable**

- $f(x)$ as $p(x)$: f is continuous and satisfies $f(x) \geq 0$ and $\int f(x) dx = 1$

- $F(x) = P(X \leq x) = \int_{-\infty}^x f(x) dx$


$$F'(t) = f(t)$$

- $E[X] = \int x * f(x) dx$ – “weighted mean”

- $Var[X] = \int x^2 p(x) - (\int x * p(x))^2$

Properties we need

- $P(A \cap B) = 0$ independent
- $P(A \cap B) = P(A|B)P(B)$ dependent
- $P(A \cup B) = P(A) + P(B)$ independent
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ dependent
- $P(C(A)) = 1 - P(A)$ complement

Conditional probability

- The probability that an event happens when a condition holds

$$P(A|B)=P(A \cap B)/P(B)$$

Michael Baron “Probability and statistics for computer scientists”

Example

- Urn with 3 white balls, 5 black balls and 7 red balls
- Pick two white balls (two cases)
- $1/25$ or $1/5 * 1/7 = 1/35$? when and why?
- Rephrase in case of failure occurrences
- 15 failures occur ...

Software Reliability theory

- Firstly ...few statistical notions
 - Random variables
 - Conditional probability
- **Reliability theory**
 - **Definitions**
 - **Failure rate**
 - **Mean Time To Failure**
 - **Failure Intensity**

Reliability theory

Barbara Russo

SwSE - Software and Systems Engineering

Reliability theory

- The goal is to estimate the **expected life** of a system, that is the time during which the system will function successfully without maintenance or repair

Reliability theory

- We proceed in two steps:
 - Firstly we consider the case of **one failure**
 - In this case, we introduce the random variable **time of failure**
 - This is the local view surrounding only one failure
 - Then we consider the case in which systems experience **more than one failure**

Reliability theory – one failure case

- T is the time in which the failure occurs:
 - **Time of Failure**
- We analyze the probability of T (continuous random variable) in some interval $(t, t+\Delta t)$

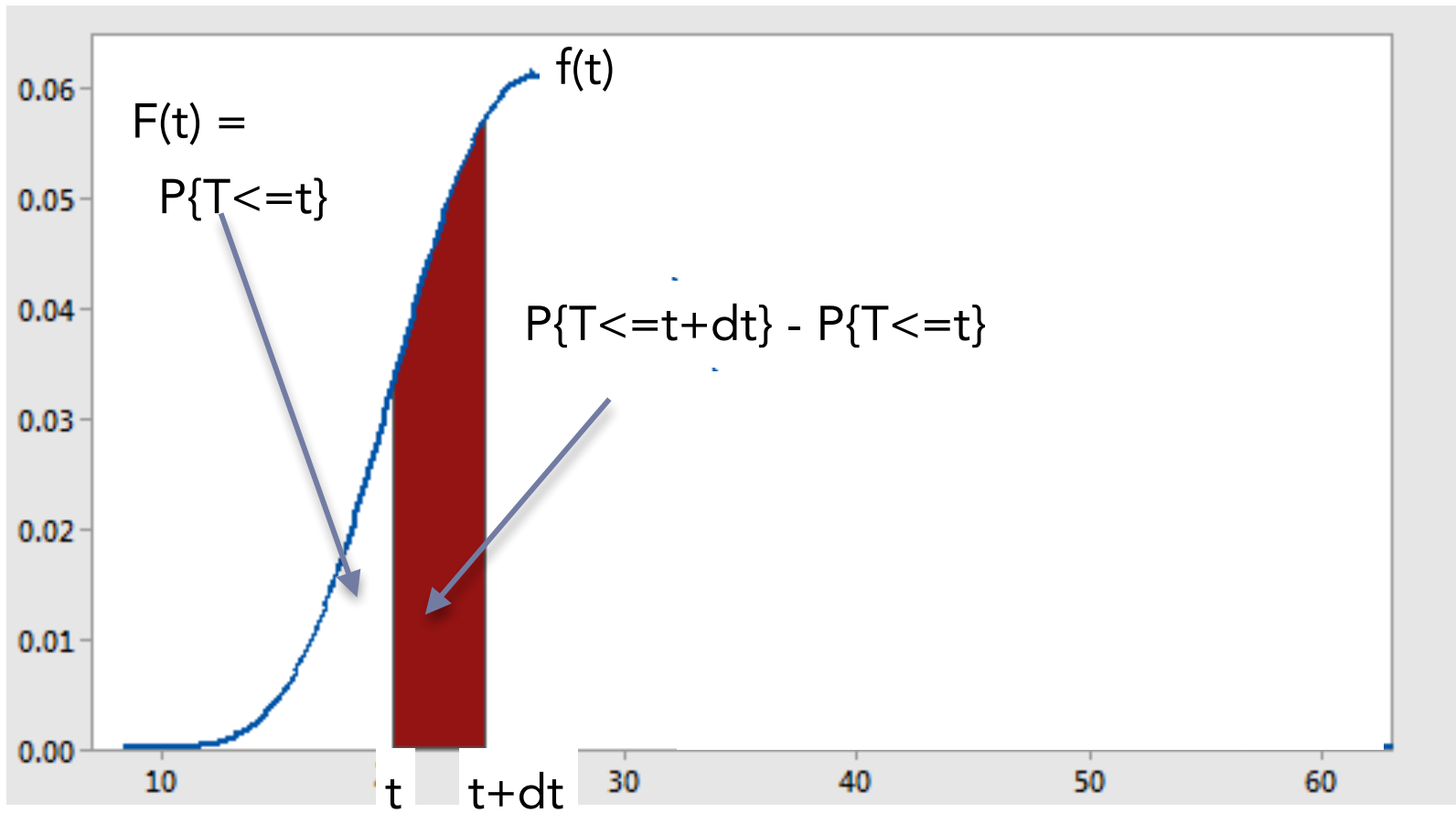
$$P(\Delta t) = P \{ t \leq T \leq t + \Delta t \}$$

One failure case - relations

$$\begin{aligned} P(\Delta t) &= P(T \leq t + \Delta t) - P(T \leq t) = \\ &= F(t + \Delta t) - F(t) = \int_t^{t + \Delta t} f(x) dx \end{aligned}$$

as

$F'(t) = f(t)$ (the derivative of the cdf
is the pdf)



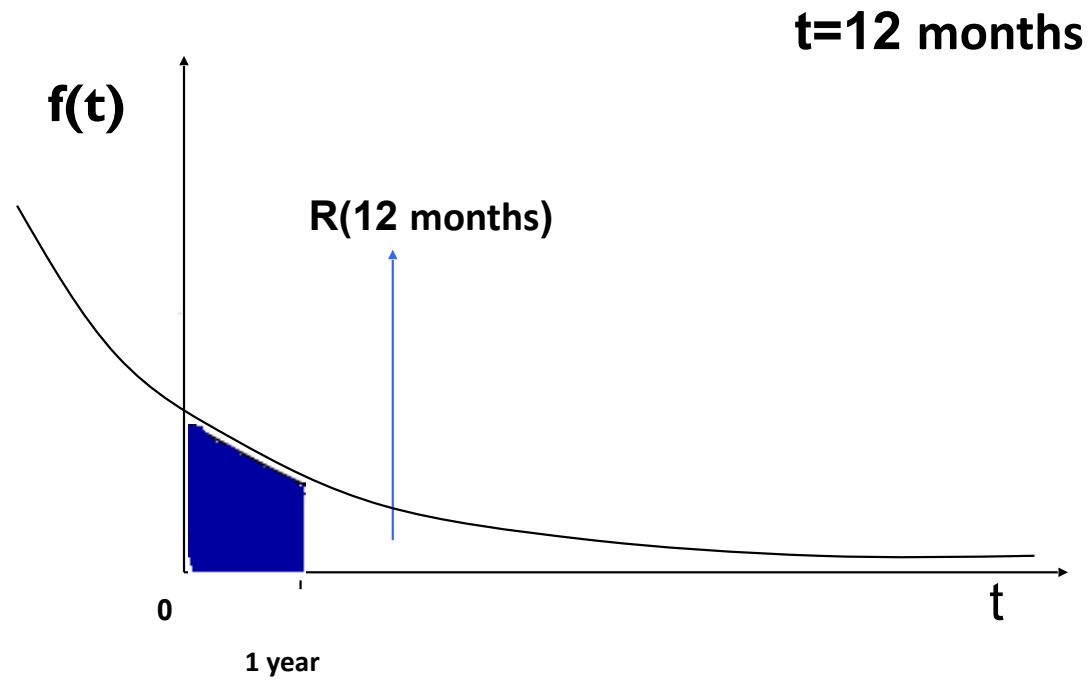
Reliability function

- The probability of success at time t
- $R(t)$, is the probability that Time of Failure is larger than t ($T > t$):

$$R(t) = P(T > t) = 1 - F(t) = \int_t^{+\infty} f(x) dx$$

- The system is reliable until t

Example



Failure rate

- The failure rate is a conditional probability:
 - The probability that a failure occurs in the interval $[t, t+\Delta t]$, given that a failure has not occurred before t , per unit of time

$$P(t \leq T \leq t+\Delta t \mid T \geq t) / \Delta t$$

- Failure rate is also equal to

$$P(\Delta t) / (\Delta t * R(t)) = F(t+ \Delta t) - F(t) / (\Delta t * R(t))$$

- in an interval Δt

Search or type a command

$$\begin{aligned}
 & P\{t \leq T \leq t + \Delta t \mid T \geq t\} = \\
 & \frac{P\{(t \leq T \leq t + \Delta t) \cap (T \geq t)\}}{P\{T \geq t\}} \\
 & = \frac{P\{t \leq T \leq t + \Delta t\}}{P\{T \geq t\}} \\
 & = \frac{P\{T \leq t + \Delta t\} - P\{T \leq t\}}{P\{T \geq t\}}
 \end{aligned}$$

People

Invite someone

Currently in this meeting (1)

RB Russo Barbara Organizer

Suggestions (5)

Camilli Matteo

BD Blaeske Vincent Cedric Daniel (Stud...

RA Ravi Abimanyu (Student Com13)

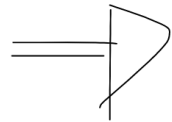
LF Lecini Fehemi (Student Com19)

LC labtest cuc (Student Com14)

Search or type a command

- Activity
- Chat
- Teams
- Assignments
- Calendar
- Calls
- Files
- Apps
- Help

$$= \frac{F(t+\Delta t) - F(t)}{1 - F(t)}$$

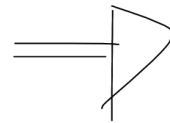


$$h(t) = \lim_{\Delta t \rightarrow 0} \frac{F(t+\Delta t) - F(t)}{\Delta t R(t)} = \frac{f(t)}{R(t)}$$



- Activity
- Chat
- Teams
- Assignments
- Calendar
- Calls
- Files
- ...
- Apps
- Help

$$= \frac{F(t+\Delta t) - F(t)}{1 - F(t)}$$



$$h(t) = \lim_{\Delta t \rightarrow 0} \frac{F(t+\Delta t) - F(t)}{\Delta t R(t)} = \frac{f(t)}{R(t)}$$

$$f(t) = F'(t) = \lim_{\Delta t \rightarrow 0} \frac{F(t+\Delta t) - F(t)}{\Delta t}$$



Hazard rate

- Hazard rate is the limit of $\Delta t \rightarrow 0$ of the Failure rate

$$h(t) = f(t) / R(t)$$

- The Hazard rate is the instantaneous probability that a failure occurs in a (very small) interval dt given that it had not occurred before

Ingredients for reliability analysis

- $F(t)$, cumulative distribution function
- $f(t)$, probability density function
- $R(t)$, reliability function and
- $h(t)$ hazard rate
- Do you remember the expected value for a random variable, $E[T]$?

Ingredients for reliability analysis

- We define the **mean time of failure**

$$E[T] = \int_{-\infty}^{+\infty} t * f(t) dt$$

- which is the expected value for the time of failure T
 - We expect that the time of failure in the whole interval of time considered will be at $E[T]$
 - We can use the expected value to predict the time of failure

How do we derive $E[T]$?

Barbara Russo

SwSE - Software and Systems Engineering

Example - uniform probability density

- $f(t) = 1/3, t \leq 3$ otherwise 0
- $F(t) = 1/3 * t, t \leq 3$ otherwise 1
- $R(t) = 1 - 1/3 * t, t \leq 3$ otherwise 0
 - A failure for sure has occurred after $t=3$

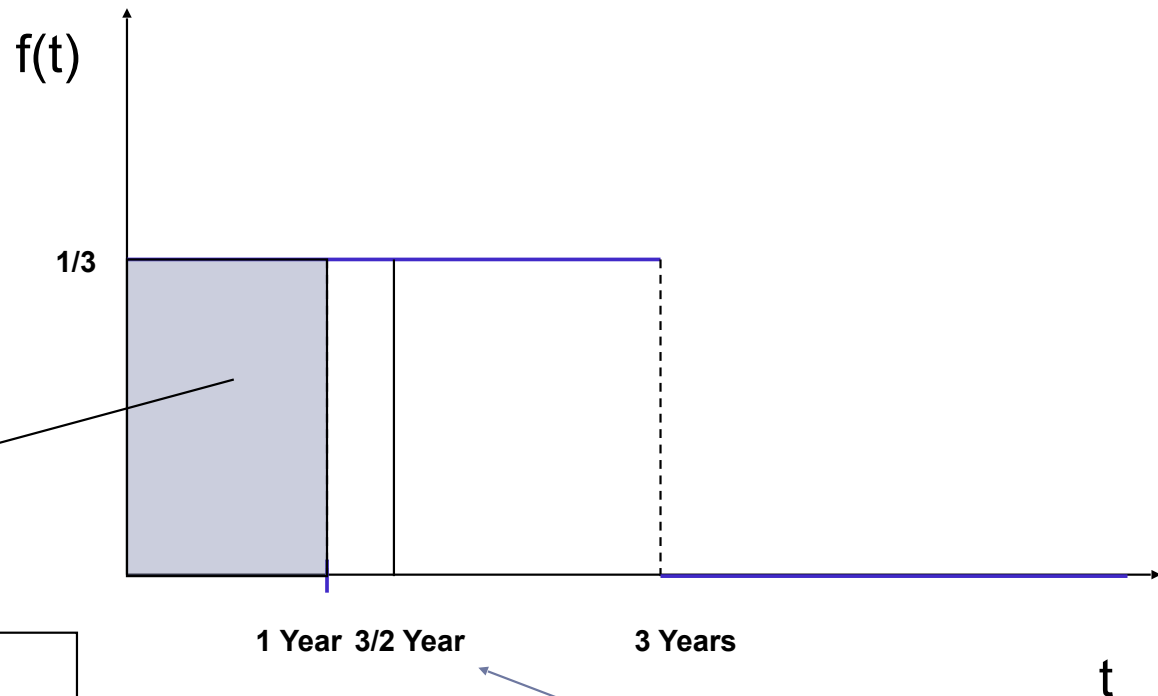
- $h(t) = \frac{1/3}{(1-1/3*t)} \quad t < 3$ otherwise is not defined:

- $E[T] = \int x f(x) dx = \int x * 1/3 dx = 1/6 * (x^2) \Big|_0^3 = 3/2$

- Each instant is equivalent to another : failures have the same probability of occurrence in different instants

Expected value

Probability that a single failure occurs



Probability that a failure occurs in one year

$$F(1 \text{ Year}) = P(T \leq 1 \text{ Year}) = 1/3$$

Expected time to failure

Typical Hazard rate of a sw system

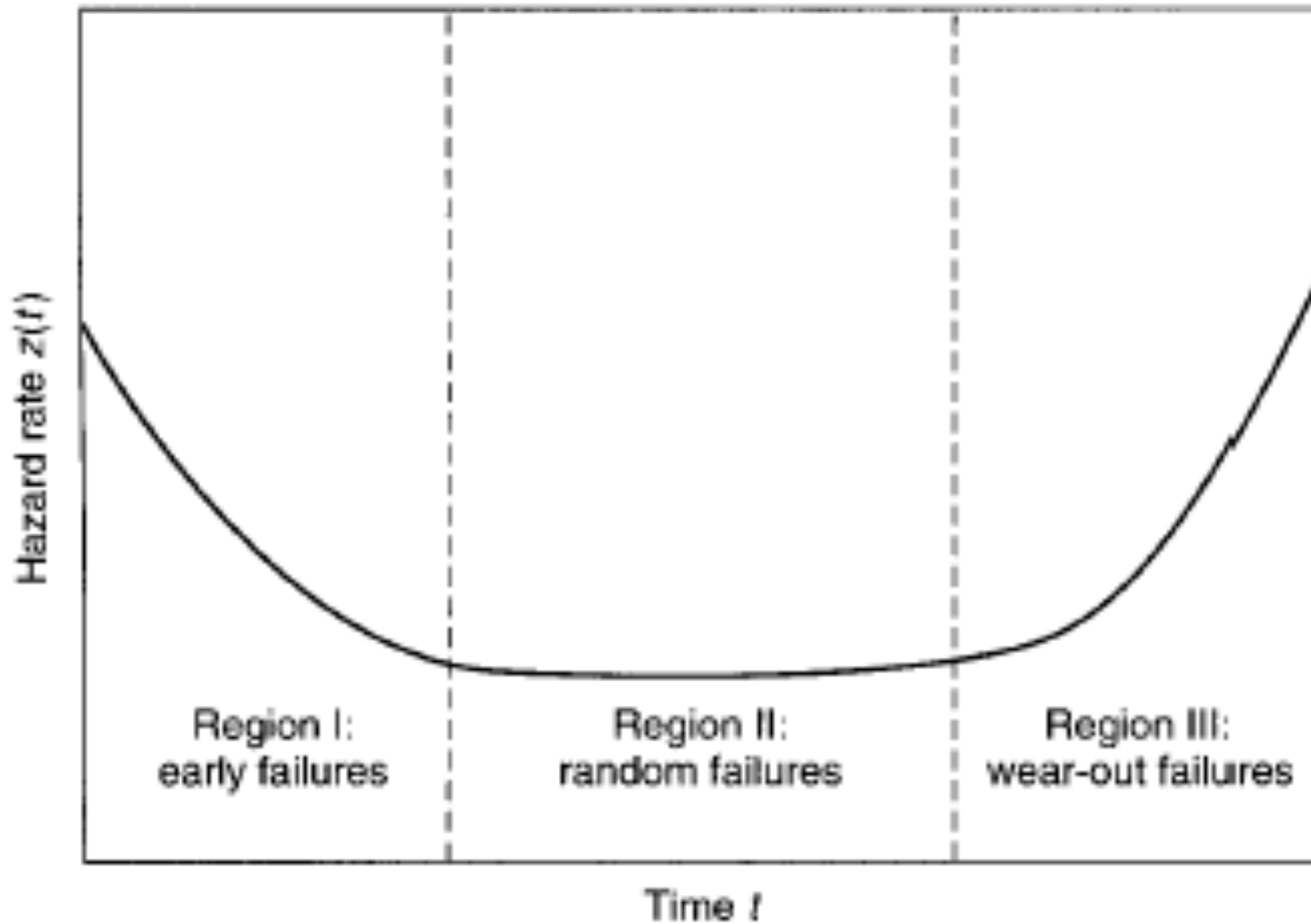
- The previous example **does not draw the case of a software system.**

- The hazard rate is a function that tends to infinite at $t=3$

$$h(t) = \frac{1/3}{(1-1/3*t)}$$

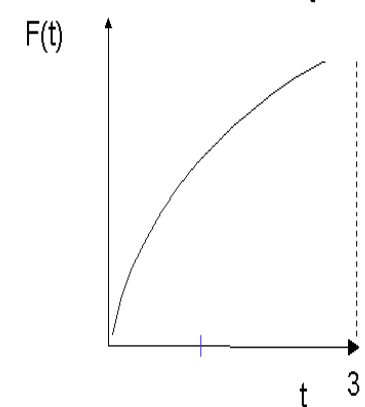
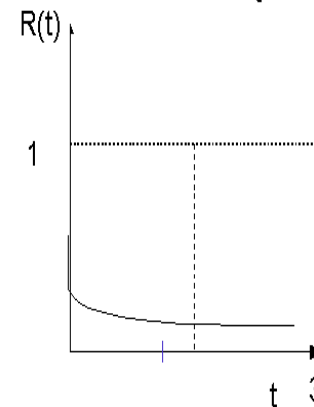
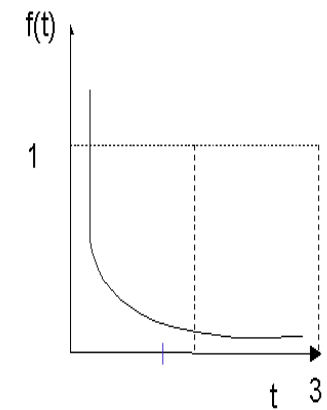
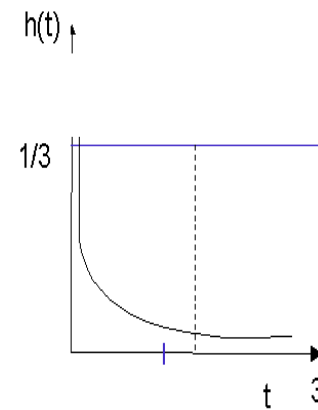
- Which are the typical functions for the hazard rate?

Hazard rate the bath tube curve



Debugging phase – region 1

- $h(t) = \lambda / (2\sqrt{t})$ with $\lambda > 0$
- $F(t) = -e^{-\lambda\sqrt{t}} + 1$
- $f(t) = \lambda / (2\sqrt{t}) * e^{-\lambda\sqrt{t}}$
- $R(t) = e^{-\lambda\sqrt{t}}$
- $E[T] = 1 / \sqrt{2\sqrt{\lambda}}$



Comments

- Region 2: Useful life period or **normal operating phase**
- The probability that a failure occurs is **equal in each instant of time**. Constant hazard rate
- It represents chance failures caused by **sudden stress or extreme conditions**

Example $f(t)=\lambda e^{-\lambda t}$

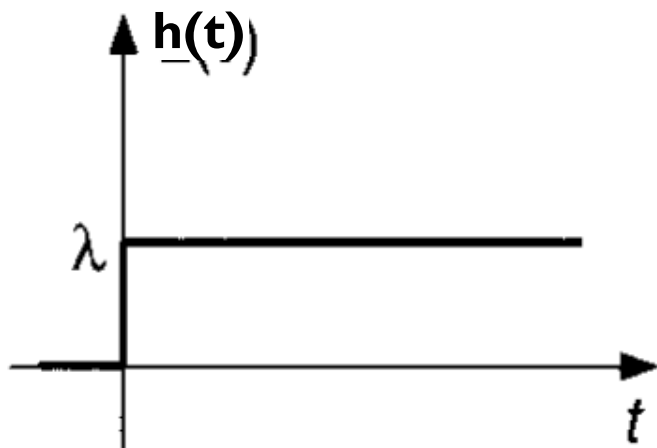
- $f(t)=\lambda e^{-\lambda t}$
- $F(t)=1-e^{-\lambda t}$
- $R(t)=e^{-\lambda t}$
- $h(t)=\lambda e^{-\lambda t}/e^{-\lambda t}=\lambda \rightarrow$
- hazard rate is the instantaneous probability that a failure occurs given that it has never occurred before $t \rightarrow$ as it is constant $h(t)$ does not depend on time \rightarrow the event occurs randomly

Note

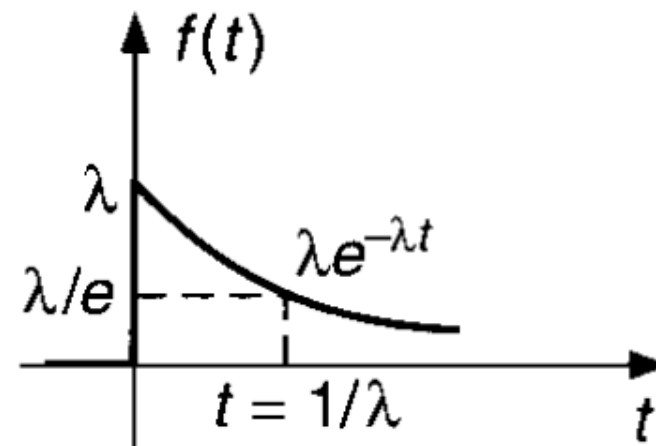
- $h(t) = \lambda = f(t)/R(t) = f(t)/(1-F(t))$
- $dF(t)/dt = \lambda * (1-F(t))$ differential equation
- The solution is $F(t) = 1 - e^{-\lambda t}$
- Exercise:

By integrating by parts between 0 and +infinite, we get:

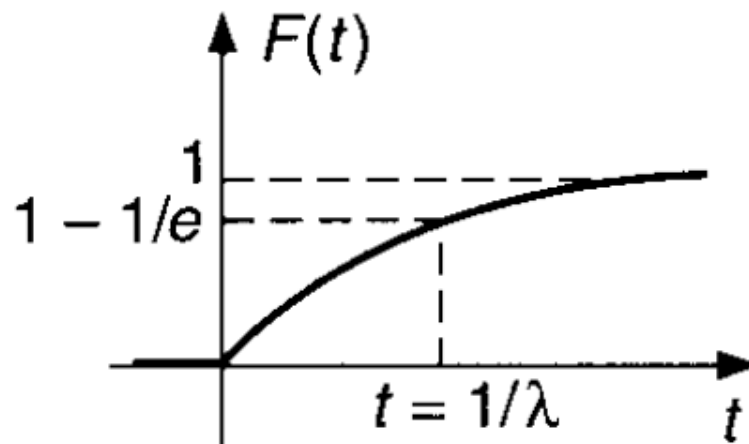
$$E[T] = \int_0^{\infty} t * f(t) dt = \int_0^{\infty} \lambda t * e^{-\lambda t} dt = 1 / \lambda$$



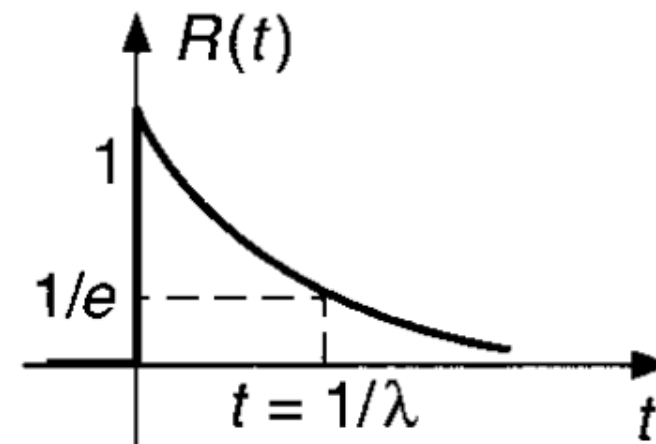
(a)



(b)



(c)



(d)

Comments

- Region 3: **wear-out failures**
- It is characterized by a rapid increase of the hazard rate
- This is **not suitable for software** as software does not wear out

Linearly increasing hazard rate– region 3

- $h(t) = Kt$ with $K > 0$
- $h(t) = Kt = f(t)/R(t) = f(t)/(1-F(t))$
- $dF(t)/dt = Kt * (1-F(t))$ differential equation whose solution is:

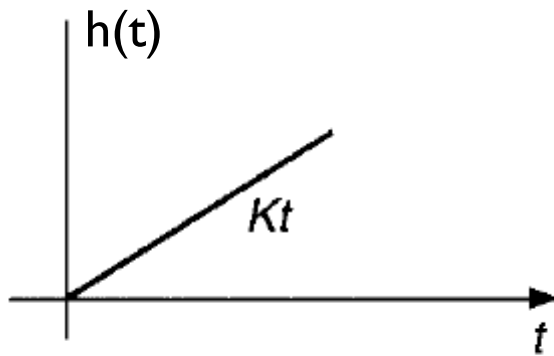
$$F(t) = -e^{-Kt^2/2} + 1$$

$$f(t) = Kt * e^{-Kt^2/2}$$

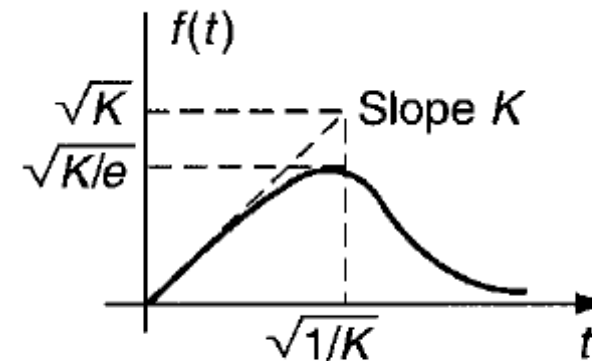
$$R(t) = e^{-Kt^2/2}$$

- $E[T] = \int tf(t)dt = \int Kt^2 * e^{-Kt^2/2} dt$ $E[T] = \sqrt{\pi/2K}$

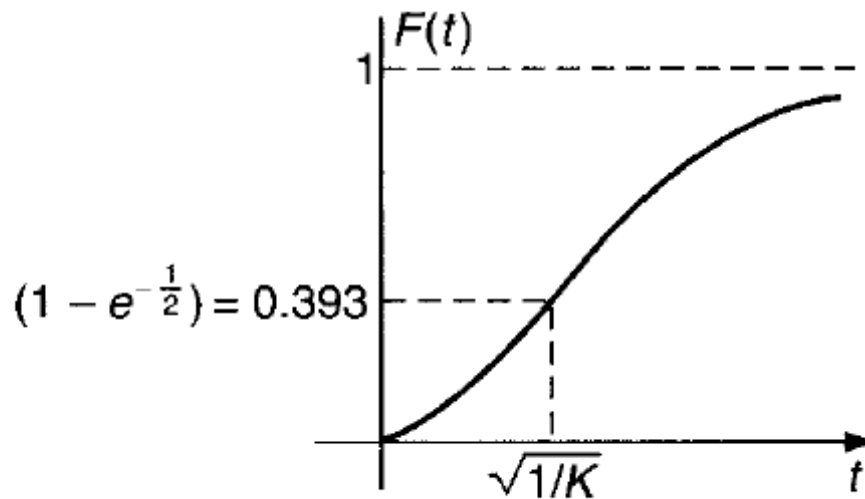
Linearly increasing hazard rate – region 3



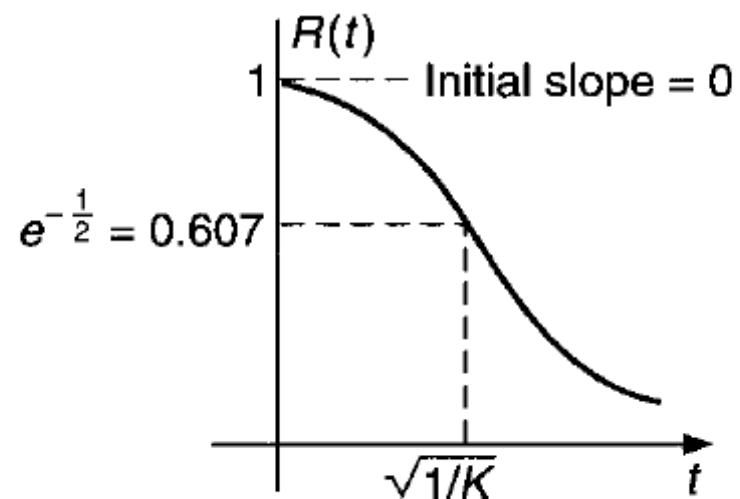
(a)



(b)



(c)



(d)