# SENG 637
# Dependability, Reliability & Testing of Software Systems

## Software Reliability Models
## (Chapter 2)

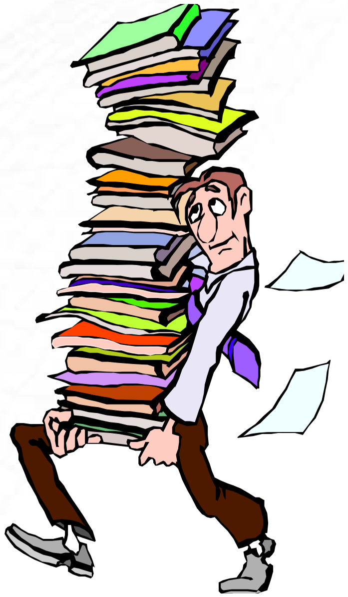
**Department of Electrical & Computer Engineering, University of Calgary**

B.H. Far (far@ucalgary.ca)

http://www.enel.ucalgary.ca/People/far/Lectures/SENG637/

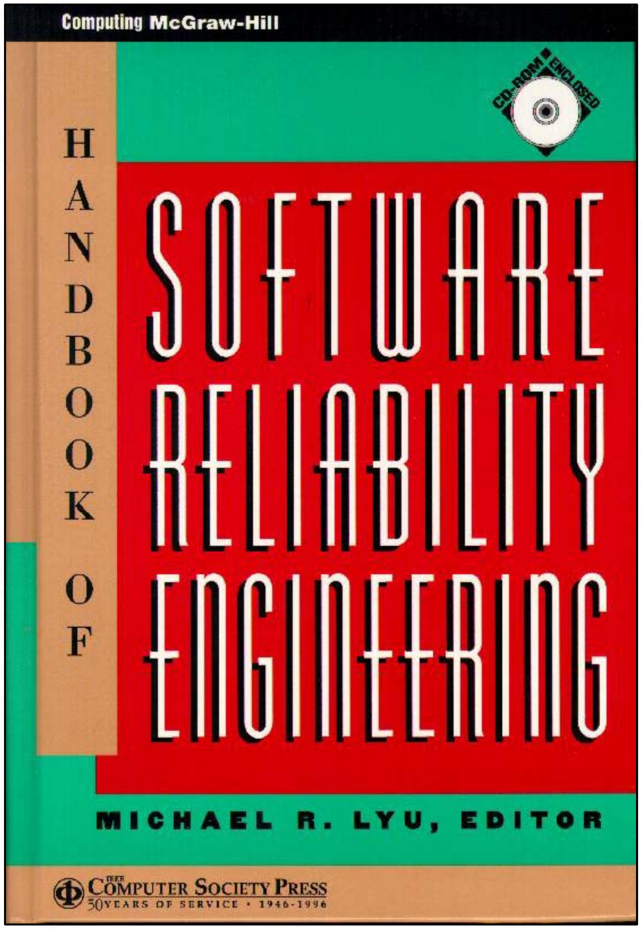
# Contents

- Basic Features of the Software Reliability Models
- Single Failure Model
- Reliability Growth Model
- Weibull and Gamma Failure Class Models
- Bayesian Models
- Early Life-Cycle Prediction Models



# Reference

- Software Reliability Engineering Handbook
- Chapter 3: Software Reliability Modeling Survey



# Chapter 2 Section 1

## Basic Features of SRE Models

# Goal

- It is important to be able to
  - Predict probability of failure of a component or system
  - Estimate the mean time to the next failure
  - Predict number of (remaining) failures

  during the development.
- Such tasks are the target of the **reliability management models**.

# Software Reliability Models

Input data (Failure Specification) → **Reliability Model** → Prediction, Estimation

**Fault introduction:**

Characteristics of the product (e.g., program size)

Development process (e.g., SE tools and techniques, staff experiences, etc.)

**Reliability Model**

**Fault removal:**

Failure discovery (e.g., extent of execution, operational profile)

Quality of repair activity

**Environment**

**(Usage)**

# Failure Specification /1

1) Time of failure
2) Time interval between failures
3) Cumulative failure up to a given time
4) Failures experienced in a time interval

**Time based failure specification**

| Failure no. | Failure times (hours) | Failure interval (hours) |
|---|---|---|
| 1 | 10 | 10 |
| 2 | 19 | 9 |
| 3 | 32 | 13 |
| 4 | 43 | 11 |
| 5 | 58 | 15 |
| 6 | 70 | 12 |
| 7 | 88 | 18 |
| 8 | 103 | 15 |
| 9 | 125 | 22 |
| 10 | 150 | 25 |
| 11 | 169 | 19 |
| 12 | 199 | 30 |
| 13 | 231 | 32 |
| 14 | 256 | 25 |
| 15 | 296 | 40 |

# Failure Specification /2

1) Time of failure
2) Time interval between failures
3) Cumulative failure up to a given time
4) Failures experienced in a time interval

**Failure based failure specification**

| Time(s) | Cumulative Failures | Failures in interval |
|---|---|---|
| 30 | 2 | 2 |
| 60 | 5 | 3 |
| 90 | 7 | 2 |
| 120 | 8 | 1 |
| 150 | 10 | 2 |
| 180 | 11 | 1 |
| 210 | 12 | 1 |
| 240 | 13 | 1 |
| 270 | 14 | 1 |

# Two Reliability Questions

## Single failure specification:

- What is the probability of failure of a system (or a component)?

## Multiple failure specification:

- If a system (or a component) fails at time $t_1$, $t_2$, …, $t_{i-1}$,
  - What is the expected time of the next failure?
  - What is the probability of the next failure?

# Various Reliability Models /1

- **Exponential Failure Class Models**
  - Jelinski-Moranda model (JM)
  - Nonhomogeneous Poisson Process model (NHPP)
  - Schneidewind model
  - Musa's Basic Execution Time model (BET)
  - Hyperexponential model (HE)
  - Others

# Various Reliability Models /2

- **Weibull and Gamma Failure Class Models**
  - Weibull model (WM)
  - S-shaped Reliability Growth model (SRG)
- **Infinite Failure Category Models**
  - Duane's model
  - Geometric model
  - Musa-Okumoto Logarithmic Poisson model

# Various Reliability Models /3

- **Bayesian Models**
  - Littlewood-Verrall Model
- **Early Life-Cycle Prediction Models**
  - Phase-based model

# How to Choose a SRE Model?

- Collect failure data (failure specification)
- Examine data (Density distribution vs. Cumulative distribution)
- Select a model
- Estimate model parameters
- Customize model using the estimated parameters
- Goodness-of-fit test
- Make reliability predictions

# Chapter 2 Section 2

**Background:**

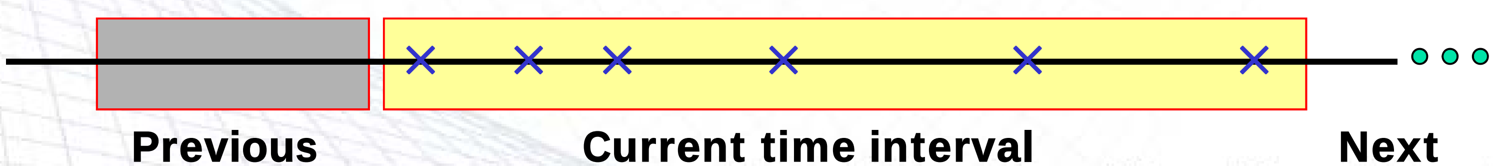
**Randomness & Probability**

# Randomness /1

- Random actions in reliability engineering:
  - Introduction of defects into the code and their removal
  - Execution of the test-cases, etc.
- We should define some ***random processes*** to represent the randomness
- How to handle randomness?
  - Collect failure data through testing
  - Find a ***distribution function*** that is a best-fit for the collected data
  - Make assumptions about the presence of errors and reliability

# Randomness /2

- **What is a random variable?**
  - A random variable $x$ on a sample space $S$ is a rule that assigns a numerical value to each outcome of $S$ (a function of $S$ into a set of real numbers)
- **In reliability modeling what can be represented by random variable?**
  - Number of failures in an interval
  - Time of failure within an interval
  - etc.



# Probability Distribution /1

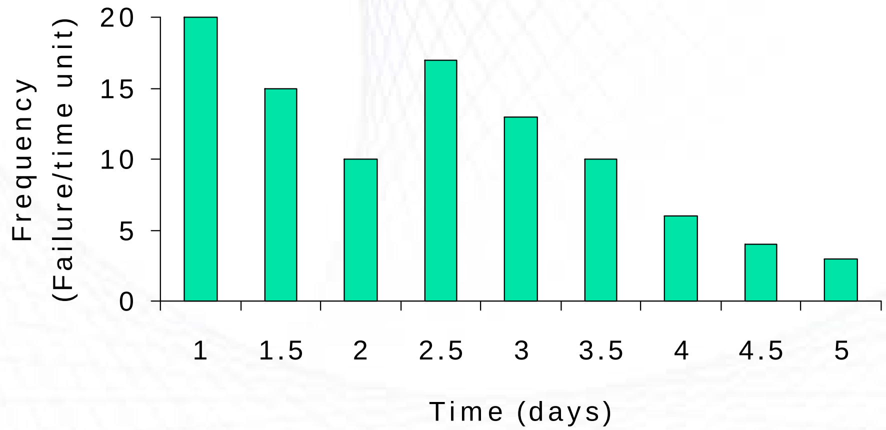
- Suppose that a random variable $X$ assigns a finite number of values to a sample space $S$
- Then $X$ induces a ***distribution function*** $f$ that assigns probabilities to the points in $R_x$

$$R_x = \{x_1, x_2, x_3, \ldots, x_n\}$$

$$f(x_k) = P(X = x_k)$$

# Probability Distribution /2

- The set of ordered pairs [$x_k$, $f(x_k)$] is usually represented by a table or a graph (histogram)




- The expected value of $X$, denoted by $E(X)$ is defined by

$$E(X) = x_1 f(x_1) + x_2 f(x_2) + \ldots + x_n f(x_n)$$

# Probability Distribution /3

- Let $M(t)$ be a random process representing the ***number of failures*** at time $t$
- The mean function $\mu(t)$ represents the expected number of failures at time $t$

$$\mu(t) = \mathrm{E}(M(t))$$

- Failure intensity is the rate of change of the expected number of failures with respect to time

$$\lambda(t) = d\,\mu(t) / dt$$

- $\lambda(t)$ is the number of failures per unit time
- $\lambda(t)$ is an instantaneous value

# Probability Distribution /4

- Discrete distributions:
  - Binomial distribution
  - Poisson distribution
- Continuous distributions:
  - Normal / Gaussian distribution
  - Lognormal distribution
  - Weibull distribution
  - Rayleigh distribution
  - Exponential distribution
  - $\Gamma$ (Gamma) distribution
  - $\chi 2$ (Kai square) distribution

# Binomial Distribution /1

- Gives probability of exact number of successes in $n$ independent trials, when probability of success $p$ on single trial is a constant.
- Situations with only 2 outcomes (*success* or *failure*)
- Probability remains the same for all *independent* trials (Bernoulli trials)

Jakob Bernoulli
(1645-1705)

# Binomial Distribution /1

- Probability of exactly *x* successes:

$$f(x) = \frac{n!}{x!\,(n-x)!}\, p^x\, q^{n-x} = \binom{n}{x} p^x\, q^{n-x}$$

| | |
|---|---|
| $n$ | : number of trials |
| $f(x)$ | : probability of $x$ successes in $n$ trials |
| $p$ | : probability of success |
| $q$ | : probability of failure |
| | $p + q = 1$ |

# Binomial Distribution /2

- Probability of having upto $r$ successes:

$$F(r) = \sum_{x=0}^{r} \binom{n}{x} p^{x} q^{n-x}$$

$n$ : number of trials
$f(x)$ : probability of $x$ successes in $n$ trials
$p$ : probability of success
$q$ : probability of failure
$p + q = 1$
$F(r)$ : probability of obtaining $r$ or fewer successes in $n$ trials

Calculating $F(r)$ becomes increasingly difficult as $n$ (sample set) gets larger

It is possible to find an approximate solution by means of a normal distribution

# Binomial Distribution /3

- Common shapes of binomial distribution

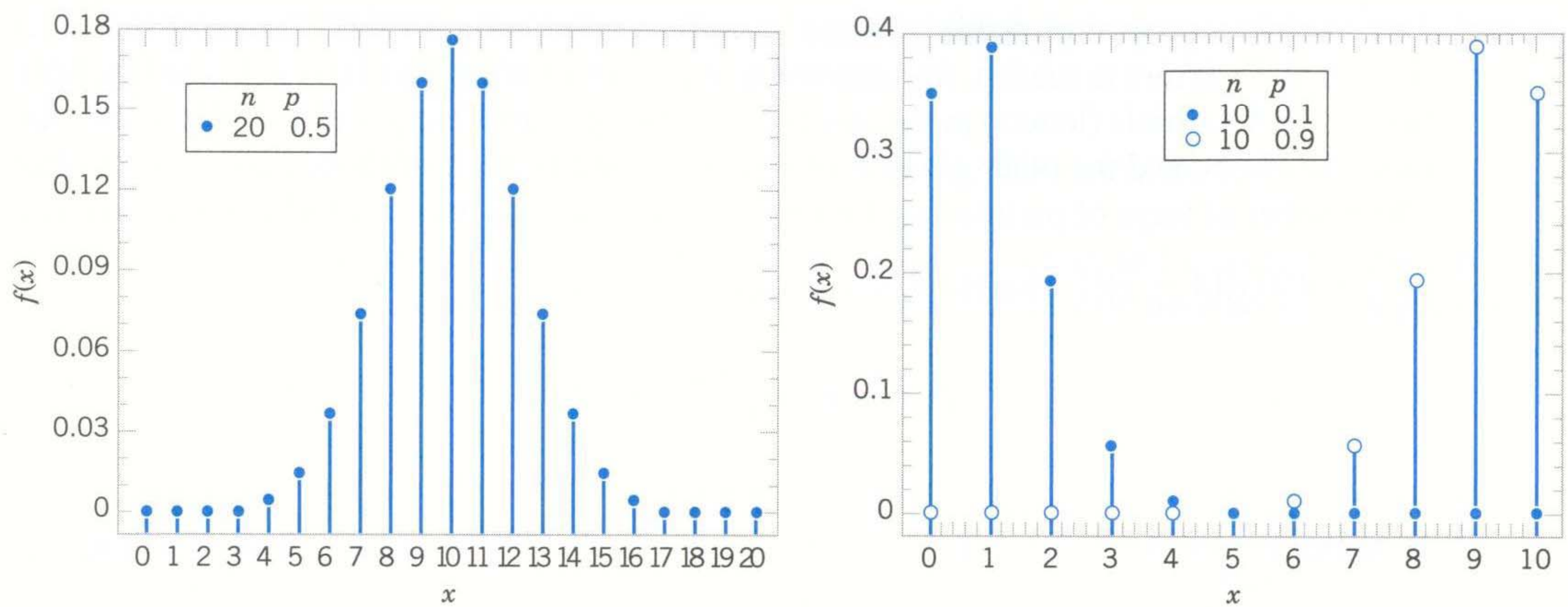



*Figure from: Montgomery et al. "Engineering Statistics"*

# Example: Binomial Distribution

- **Acceptance sampling:**
  - A lot is accepted if not more than 2 defectives are found in a sample of 6. The defect probability is 25%.
  - Probability of having exactly 2 defects in the lot is:

$$f(2) = \binom{6}{2} 0.25^2 \times 0.75^4 = 0.297$$

  - Probability of having more than 4 defects in the lot is:

$$F(r > 4) = F(5) + F(6) = \binom{6}{5} 0.25^5 \times 0.75^1 + 0.25^6 = 0.0046$$

# Example 2

- In a company there are 4 file servers each having the exact replica of a data set. The probability of failure for each server is 10%.
- Probability of having 2 servers fail:

$$f(2) = \binom{4}{2} 0.1^2 \times 0.9^2 = 0.0486$$

- Probability of having more than 3 servers fail:

$$F(r > 2) = F(3) + F(4) = \binom{4}{3} 0.1^3 \times 0.9^1 + 0.1^4 = 0.00036$$

# Poisson Distribution /1

- Some events are rather rare, they don't happen that often. Still, over a period of time, we want to say something about the nature of rare events.
- Poisson distribution is special case of binomial distribution (either $p$ or $q$ is very small and $n$ very large)
- Conditions under which a Poisson distribution holds
  - counts of rare events, i.e. small probability
  - all events are independent
  - average rate (usually denoted by $\mu$) does not change over the period of interest

Simeon Poisson (1781-1840)

# Poisson Distribution /2

- Poisson distribution is a special case of binomial distribution (either *p* or *q* is very small and *n* very large): $\mu = np$

$$f(x) = \frac{\mu^x}{x!} e^{-\mu} \quad x = 0,1,2,\ldots$$

μ: mean rate of occurrence (in statistics literature is usually denoted by λ)

*x* : observed number of failures

# Poisson Distribution /3

- Common shapes of Poisson distribution

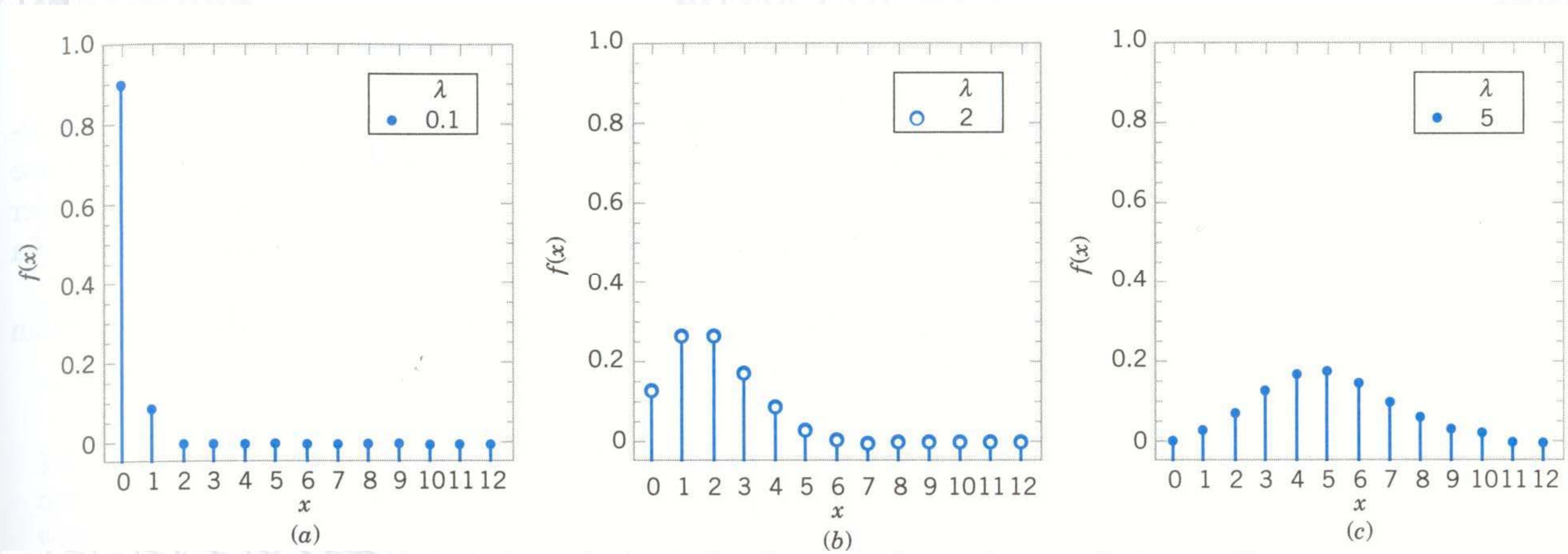



*Figure from: Montgomery et al. "Engineering Statistics"*

# Example: Poisson Distribution

- Suppose that the defect rate is only 2%

  find the probability that there are 3 defective items in a sample of 100 items.

$$\mu = np = 100 \times 0.02 = 2$$

$$f(x=3) = \frac{2^3 \times e^{-2}}{3!} = 0.18$$

# Chapter 2 Section 3

## Single Failure Model

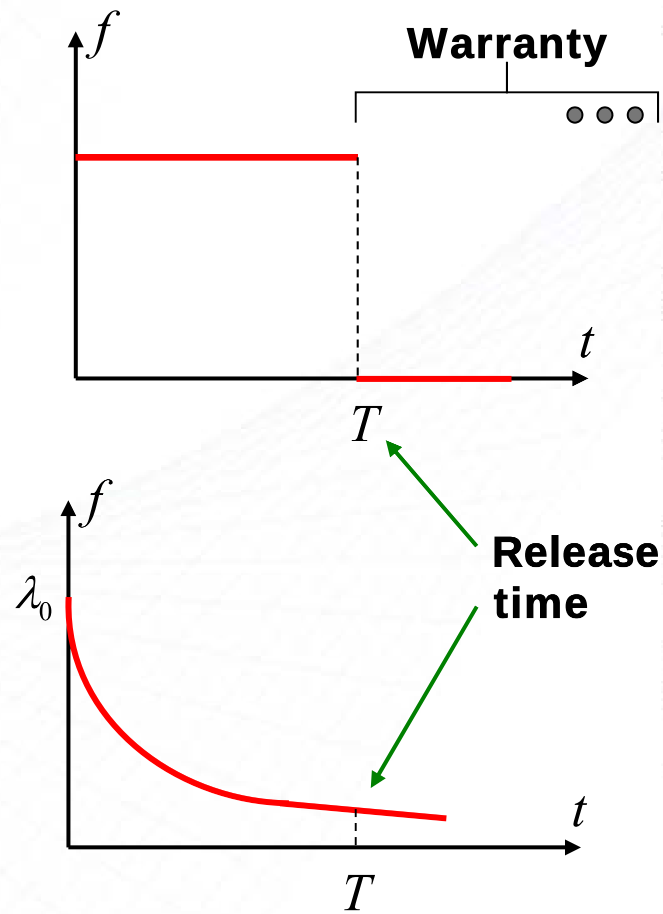
# Hardware Reliability Models

- **Uniform model:**
  - Probability of failure is fixed.

$$f(t) = \begin{cases} 1 & t \le T \\ 0 & t > T \end{cases}$$

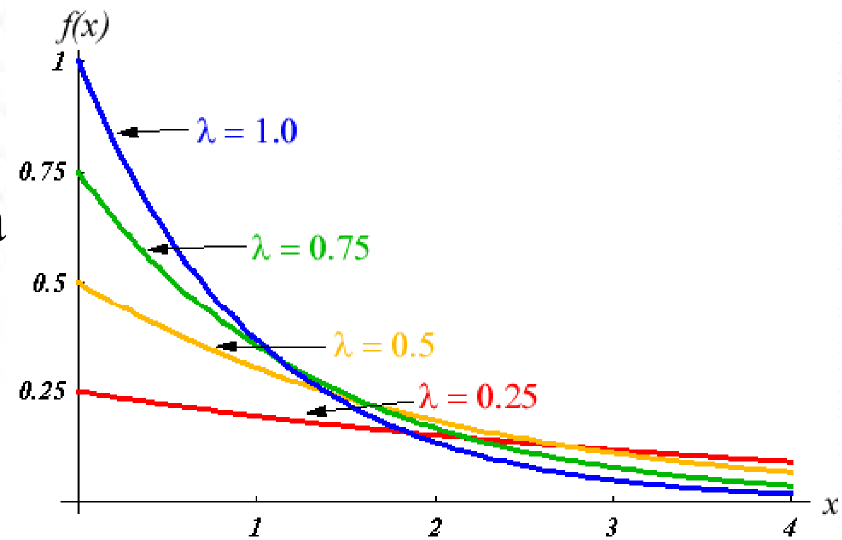
- **Exponential model:**
  - Probability of failure changes exponentially over time

$$f(t) = \lambda_0 e^{-\beta t}$$



# Single Failure Model /1

- **Probability Density Function (PDF):** depicting changes of the probability of failure up to a given time *t*.
- A common form of PDF is exponential distribution
- Usually we want to know how long a component will behave correctly before it fails, *i.e.*, the probability of failure from time *0* up to a given time *t*.



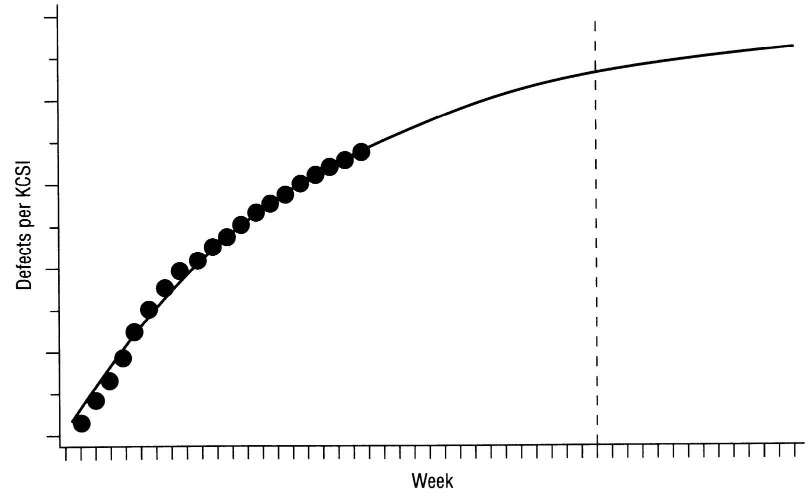

Exponential PDF:

$$f(t) = \begin{cases} \lambda e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

# Single Failure Model /2

- **Cumulative Density Function (CDF):** depicting cumulative failures up to a given time *t*.

$$F(t) = \int_0^t f(t)$$




- For exponential distribution, CDF is:

$$F(t) = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

*Figure from Musa's book*

# Single Failure Model /3

- **Reliability function *R(t)*:** defined as a component functioning without failure until time *t*, that is, the probability that the time to failure is greater than *t*.

$$R(t) = \int_t^{\infty} f(t) \quad = 1 - F(t)$$

- For exponential distribution, with a constant failure rate λ:

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

# Single Failure Model /4

- What is the expected value of failure at time *T*?
- It is the mean of the probability density function (PDF), named **mean time to failure (MTTF)**

$$E(T) = \int_0^{\infty} t\ f(t)\,dt$$

- For exponential distribution, MTTF is:

$$MTTF = \frac{1}{\lambda}$$

# Single Failure Model /5

- **Median time to failure ($t_m$):** a point in time that the probability of failure before and after $t_m$ are equal.

$$\int_0^{t_m} t\, f(t)\, dt = \tfrac{1}{2} \quad or \quad F(t_m) = \tfrac{1}{2}$$

- **Failure (hazard) Rate *z(t)*:** Probability density function divided by reliability function.

$$z(t) = \frac{f(t)}{R(t)}$$

For exponential distribution, *z(t)* is: λ

# Single Failure Model /6

- **System Reliability:** is the multiplication of the reliability of its components. (for serial systems, i.e., with no redundancy)

$$R_{system}(t) = \prod_{i=1}^{n} R_i(t)$$

- For exponential distribution:

$$R_{system}(t) = e^{-\lambda_1 t} e^{-\lambda_2 t} \cdots e^{-\lambda_n t} = e^{-(\lambda_1 + \lambda_2 + \cdots + \lambda_n)t}$$

$$R_{system}(t) = e^{-t\left(\sum_{i=1}^{n} \lambda_i\right)}$$

far@ucalgary.ca

# Single Failure Model /7

- **System Cumulative Failure (hazard) Rate:** is the sum of the failure rate of its components.

$$z_{system}(t) = \sum_{i=1}^{n} z_i(t)$$

- For exponential distribution:

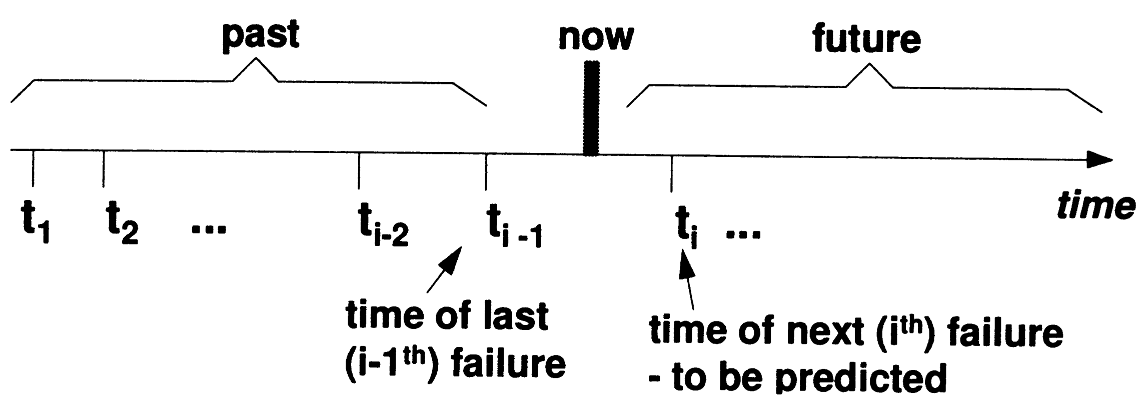
$$z_{system}(t) = \lambda_1 + \lambda_2 + \cdots + \lambda_n = \sum_{i=1}^{n} \lambda_i$$

# Chapter 2 Section 4

## Reliability Growth Model

# Reliability Growth Models /1

- For hardware systems one can assume that the probability of failure (probability density function, PDF) ***for all failures*** are the same (*e.g.*, replacing a faulty hardware component with an identical working one).
- In software, however, we want to "fix" the problem, i.e., ***have a lower probability for the remaining failures*** after a repair (or longer $\Delta t_i = t_i - t_{i-1}$). Therefore, we need a model for **reliability growth** (i.e., reliability change over time).



# Reliability Growth Models /2

- In reliability growth models we are assuming some effort of fault removal. This leads to a variable failure intensity $\lambda(t)$.
- Every reliability growth model is based on specific assumptions concerning the change of failure intensity $\lambda(t)$ through the process of fault removal.

# Reliability Growth Models /3

- Common software reliability growth models are:
  - **Basic Exponential model**
  - **Logarithmic Poisson model**
- The basic exponential model assumes finite failures ($\nu_0$) in infinite time.
- The logarithmic Poisson model assumes infinite failures.

# Validity of the Models

- Software systems are changed (updated) many times during their life cycle.
- The models are good for one revision period rather than the whole life cycle.

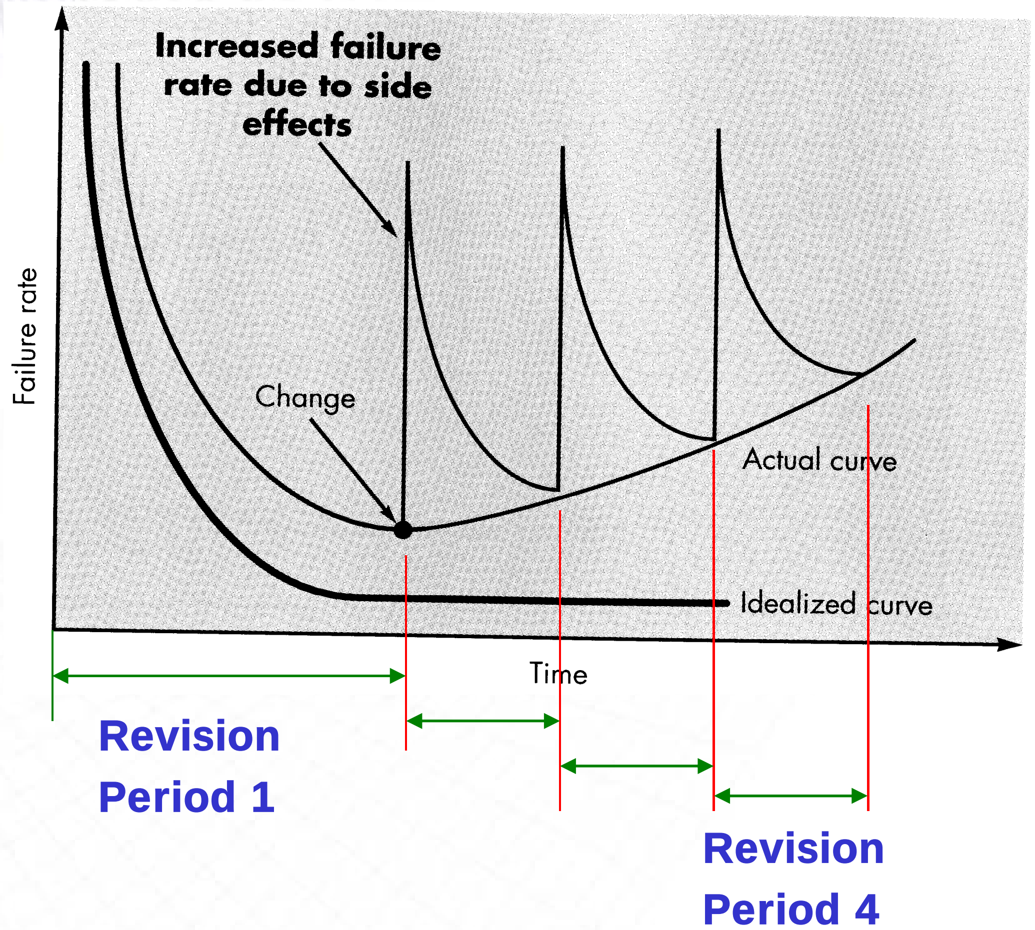



*Figure from Pressman's book*

# Reliability Growth Models /4

- Variables involved in reliability growth models:
  1) Failure intensity (λ): number of failures per natural or time unit.
  2) Execution time (τ): time since the program is running. Execution time may be different from calendar time.
  3) Mean failures experienced (μ): mean failures experienced in a time interval.

# Reliability Growth Models /9

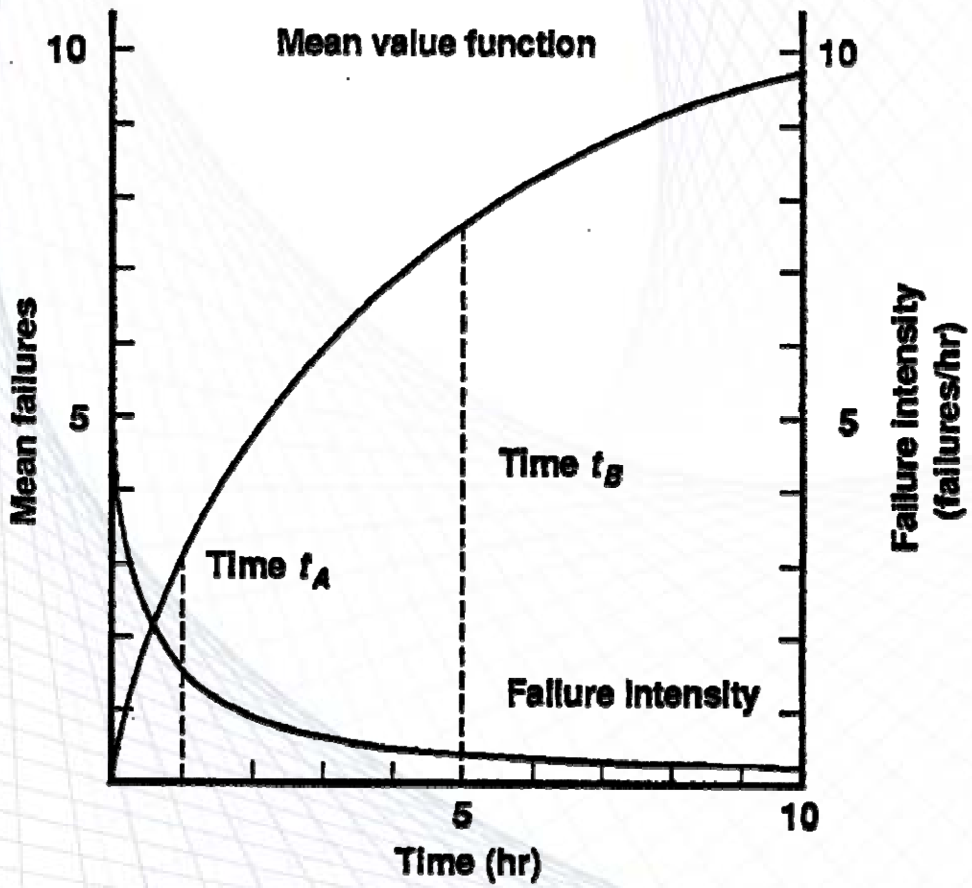
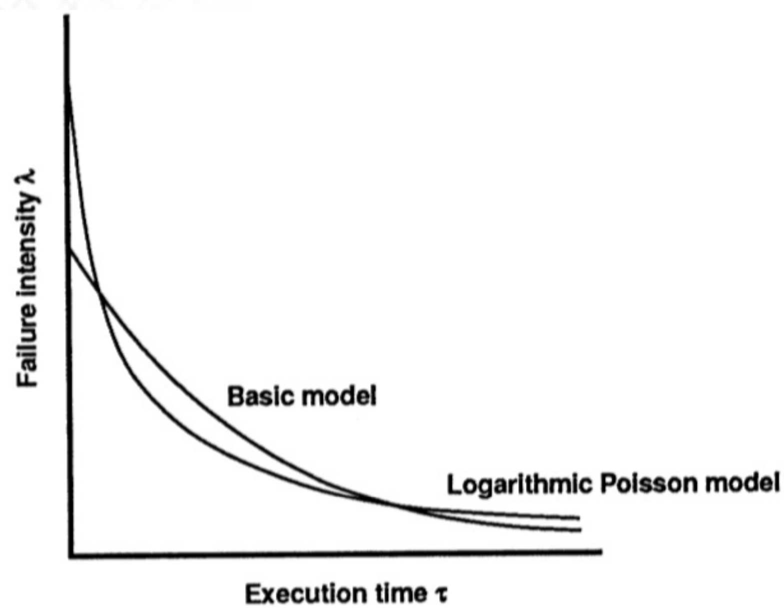



*Figure from Musa's book*

| Failure(s) in time period | Probability | |
|---|---|---|
| | Elapsed time (1 hour) | Elapsed time (5 hours) |
| **0** | **0.10** | **0.01** |
| **1** | **0.18** | **0.02** |
| **2** | **0.22** | **0.03** |
| **3** | **0.16** | **0.04** |
| **4** | **0.11** | **0.05** |
| **5** | **0.08** | **0.07** |
| **6** | **0.05** | **0.09** |
| **7** | **0.04** | **0.12** |
| **8** | **0.03** | **0.16** |
| **9** | **0.02** | **0.13** |
| **10** | **0.01** | **0.10** |
| **11** | **0** | **0.07** |
| **12** | **0** | **0.05** |
| **13** | **0** | **0.03** |
| **14** | **0** | **0.02** |
| **15** | **0** | **0.01** |
| **Mean** | **3.04** | **7.77** |

# Reliability Growth Models /6

- Failure intensity (λ) versus execution time (τ)




(B) $$\lambda(\tau) = \lambda_0 \, e^{\left(-\frac{\lambda_0}{\nu_0}\right)\tau}$$

(P) $$\lambda(\tau) = \frac{\lambda_0}{\lambda_0 \theta \tau + 1}$$

$\lambda_0$ Initial failure intensity

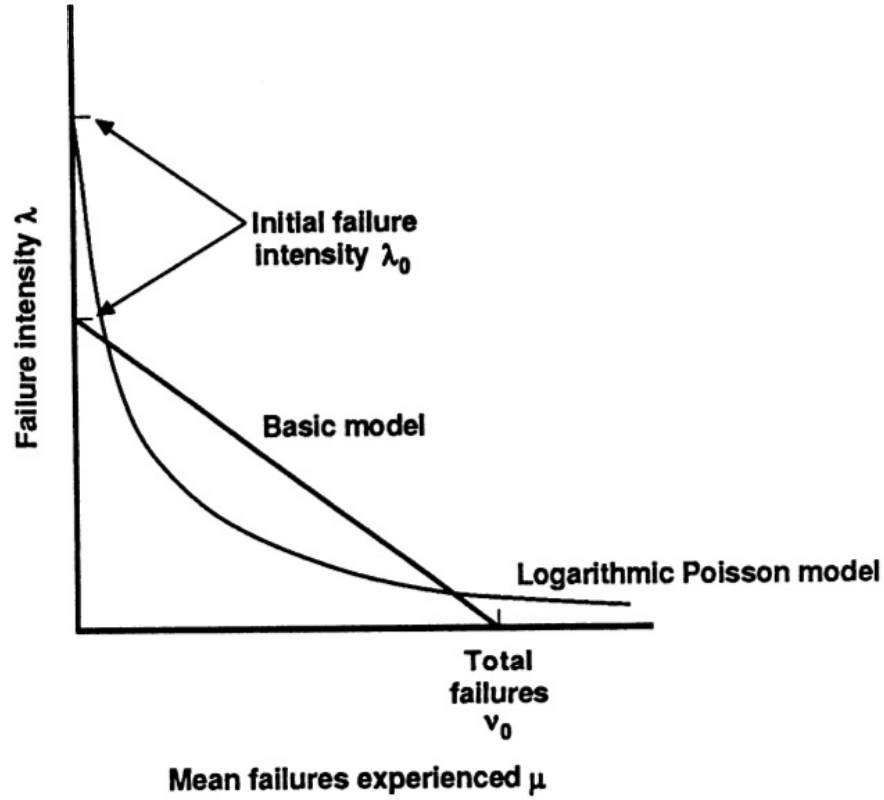
$\nu_0$ Total failures

$\theta$ Decay parameter

*Figure from Musa's book*

# Reliability Growth Models /7

- Failure intensity (λ) versus mean failures experienced (μ)



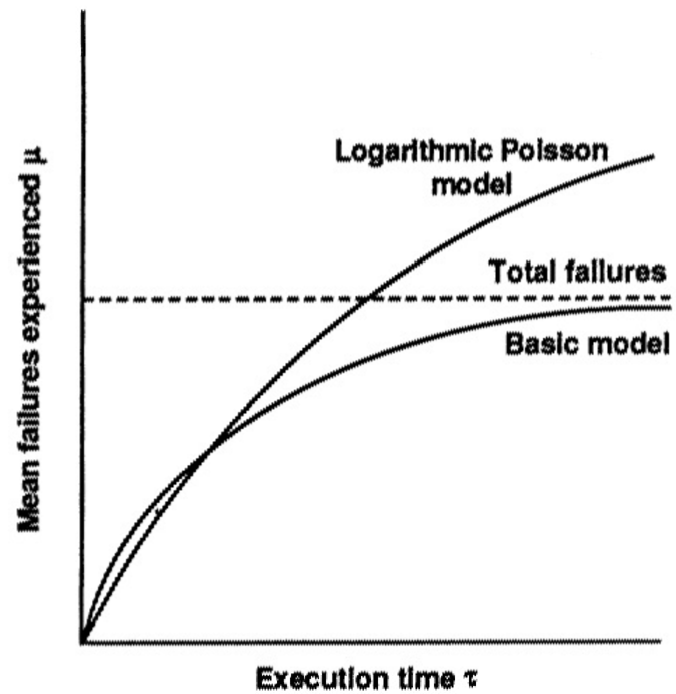

(B) $$\lambda(\mu) = \lambda_0 \left(1 - \frac{\mu}{\nu_0}\right)$$

(P) $$\lambda(\mu) = \lambda_0 \, e^{-\theta\mu}$$

*Figure from Musa's book*

# Reliability Growth Models /8

- Mean failures experienced (μ) versus execution time (τ)




(B) $$\mu(\tau) = \nu_0 \left[ 1 - e^{-\left(\frac{\lambda_0}{\nu_0}\tau\right)} \right]$$

(P) $$\mu(\tau) = \left(\frac{1}{\theta}\right) \ln\left(\lambda_0 \theta \tau - 1\right)$$

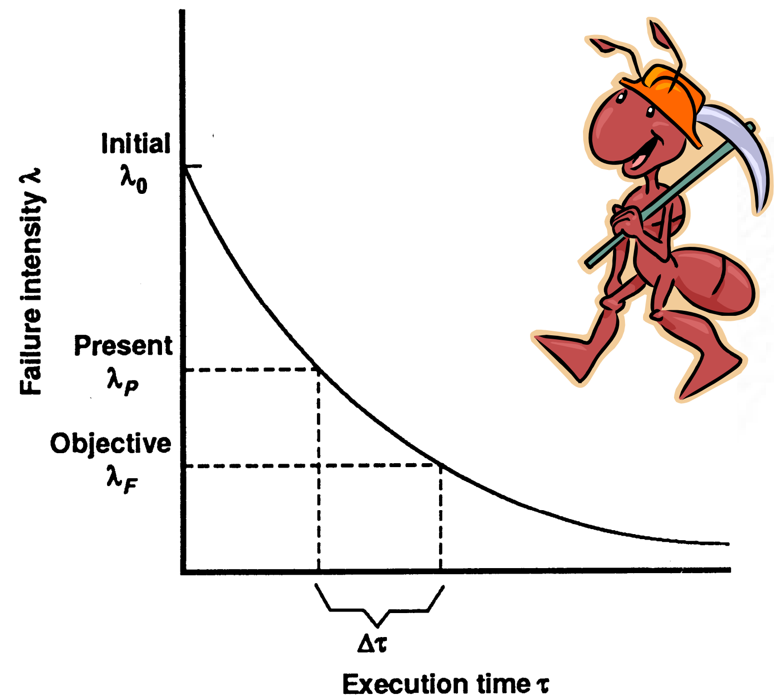
*Figure from Musa's book*

# How to Use the Models?

- **Release criteria (time):** time required to test the system to reach a target failure intensity:

$$(B) \quad \Delta\tau = \frac{\nu_0}{\lambda_0} \ln \frac{\lambda_P}{\lambda_F}$$

$$(P) \quad \Delta\tau = \frac{1}{\theta}\left(\frac{1}{\lambda_F} - \frac{1}{\lambda_P}\right)$$




$\lambda_P$ : Present failure intensity

$\lambda_F$ : Target failure intensity

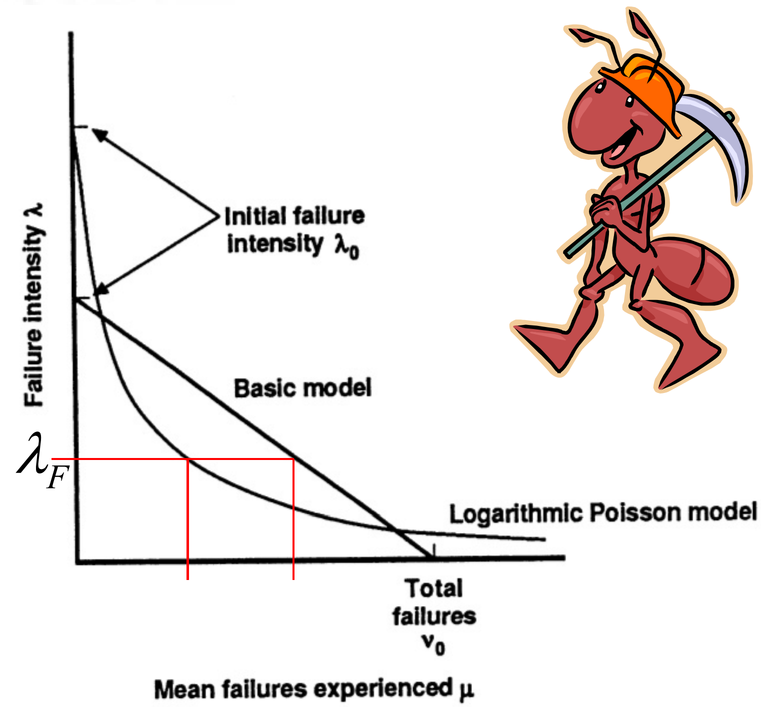
*Figure from Musa's book*

# How to Use the Models?

- **Release criteria (failure):** time required to test the system to reach a target number of failures:

(B) $$\Delta\mu = \frac{\nu_0}{\lambda_0}\left(\lambda_P - \lambda_F\right)$$

(P) $$\Delta\mu = \frac{1}{\theta}\ln\frac{\lambda_P}{\lambda_F}$$




$\lambda_P$ : Present failure intensity

$\lambda_F$ : Target failure intensity

*Figure from Musa's book*

# Reliability Metrics

- **Mean time to failure (MTTF):** Usually calculated by dividing the total operating time of the units tested by the total number of failures encountered (assuming that the failure rate is constant).
- **Example:**
  - MTTF for Windows 2000 Professional is 2893 hours or 72 workweeks (40 hours per week).
  - MTTF for Windows NT Workstation is 919 hours or 23 workweeks.
  - MTTF for Windows 98 is 216 hours or 5 workweeks.
- **Mean time to repair (MTTR):** mean time to repair a (software) component.
- **Mean time between failures (MTBF)**:

  MTBF = MTTF + MTTR

# Reliability Metrics: Availability

- Software System Availability ($A$):

$$A(t) = \frac{1}{1 + t_m \lambda(t)} \quad or \quad \lambda(t) = \frac{1 - A(t)}{t_m A(t)}$$

  $\lambda$ is failure intensity

  $t_m$ is downtime per failure

- Another definition of availability:

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{MTTF}{MTBF}$$

- **Example:** If a product must be available 99% of time and downtime is 6 min, then $\lambda$ is about 0.1 failure per hour (1 failure per 10 hours) and MTTF=594 min.

# Reliability Metrics: Reliability

- Software System Reliability (*R*):

$$\lambda(t) = \frac{-\ln\ R(t)}{t} \quad or \quad \lambda \approx \frac{1-R(t)}{t} \quad for \quad R(t) \geq 0.95$$

  λ is failure intensity

  *R* is reliability

  *t* is natural unit (time, etc.)

- **Example:** for λ=0.001 or 1 failure for 1000 hours, reliability (*R*) is around 0.992 for 8 hours of operation.

# Example

- In reliability growth testing of a software system the failure data fits to both the Exponential model and Logarithmic Poisson model. The model parameters are given below:

| ***Exponential model*** | ***Logarithmic Poisson model*** |
| --- | --- |
| $\lambda_0$ = 20 failures/CPU hour | $\lambda_0$ = 50 failures/CPU hour |
| $\nu_0$ = 120 failures | $\theta$ = 0.025 failures |

- Assume for both cases below that you start from the initial failure intensity.

# Example (cont'd)

- Determine the additional failure and additional execution time required to reach a failure intensity objective of 10 failure/CPU hour for both models.

For Basic model:

$$\Delta\mu = \frac{\nu_0}{\lambda_0}(\lambda_P - \lambda_F) = \frac{120}{20}(20-10) = 60 \quad \textit{failures}$$

$$\Delta\tau = \frac{\nu_0}{\lambda_0}\ln\frac{\lambda_P}{\lambda_F} = \frac{120}{20}\ln\frac{20}{10} = 4.16 \quad \textit{CPU hour}$$

For Poisson model:

$$\Delta\mu = \frac{1}{\theta}\ln\frac{\lambda_P}{\lambda_F} = \frac{1}{0.025}\ln\frac{50}{10} = 64 \quad \textit{failures}$$

$$\Delta\tau = \frac{1}{\theta}\left(\frac{1}{\lambda_F} - \frac{1}{\lambda_P}\right) = \frac{1}{0.025}\left(\frac{1}{10} - \frac{1}{50}\right) = 3.2 \quad \textit{CPU hour}$$

# Example (cont'd)

- Repeat the same calculation for a failure intensity objective of 1 failure/CPU hour for both models.

For Basic model:

$$\Delta\mu = \frac{\nu_0}{\lambda_0}\left(\lambda_P - \lambda_F\right) = \frac{120}{20}\left(20-1\right) = 114 \quad \textit{failures}$$

$$\Delta\tau = \frac{\nu_0}{\lambda_0}\ln\frac{\lambda_P}{\lambda_F} = \frac{120}{20}\ln\frac{20}{1} = 18 \quad \textit{CPU hour}$$

For Poisson model:

$$\Delta\mu = \frac{1}{\theta}\ln\frac{\lambda_P}{\lambda_F} = \frac{1}{0.025}\ln\frac{50}{1} = 156 \quad \textit{failures}$$

$$\Delta\tau = \frac{1}{\theta}\left(\frac{1}{\lambda_F} - \frac{1}{\lambda_P}\right) = \frac{1}{0.025}\left(\frac{1}{1} - \frac{1}{50}\right) = 39.2 \quad \textit{CPU hour}$$

# Example (cont'd)

- Based on the experiment's results compare the two model results and explain what will happen to the additional execution time required when the failure intensity objective gets smaller.

As failure intensity gets smaller the additional failures and execution time required to reach them become substantially larger for Logarithmic Poisson model than the Basic model.

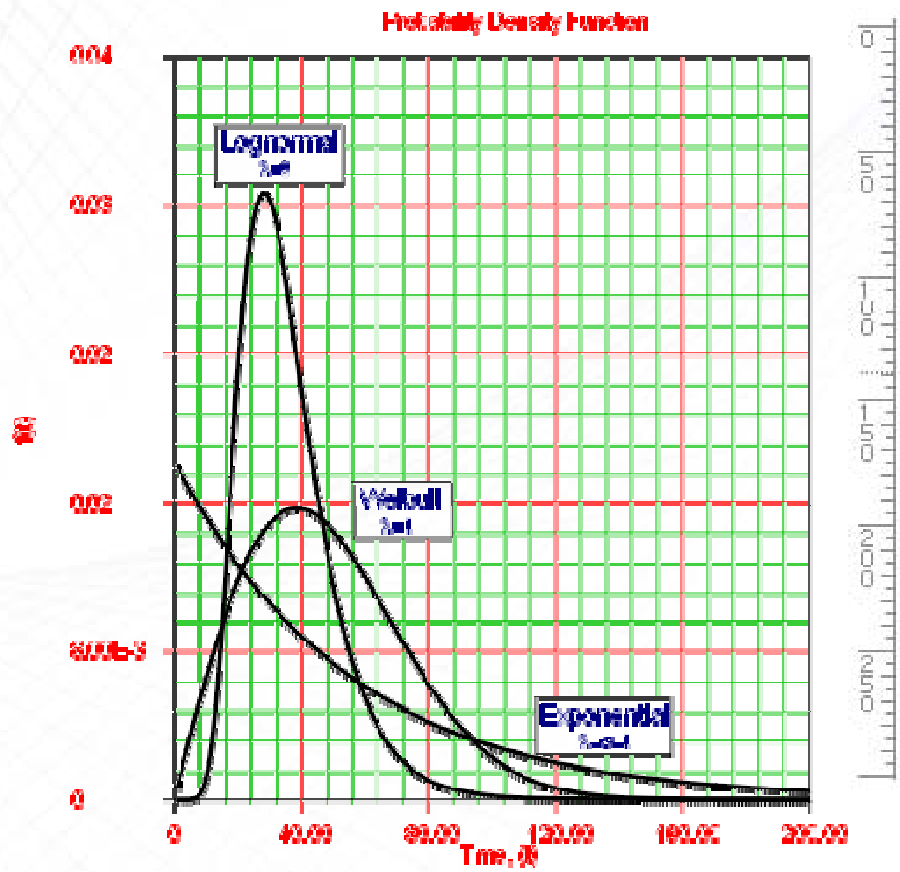
Therefore the Basic model gives an optimistic estimation while Logarithmic Poisson model offers a pessimistic one.

# Chapter 2 Section 5

## Weibull and Gamma Failure Class Models

# Gamma Failure Class Models

- The generalized gamma distribution (total 3 parameters, i.e., scale, shape, location) includes other distributions as special cases based on the values of the parameters.




Reference: http://www.weibull.com/

# Weibull Model /1

- Failure distribution is Weibull distribution rather than exponential
- Depending on the values of the parameters (total 3 parameters, i.e., scale, shape, location), the Weibull distribution can be used to model a variety of behaviours

# Weibull Model /2

- **Assumptions:**
  - Total number of faults is bounded
  - The time to failure is distributed as Weibull distribution
  - The number of faults detected in each interval are independent for any finite collection of times
- **Data requirement:**
  - Fault counts on each testing interval: $f_1, f_2, \ldots, f_n$
  - Completion time of each period: $t_1, t_2, \ldots, t_n$

# Weibull Model /3

- The Weibull distribution has the probability density function

$$f(T) = \frac{\beta}{\eta}\left(\frac{T-\gamma}{\eta}\right)^{\beta-1} e^{-\left(\frac{T-\gamma}{\eta}\right)^{\beta}}$$

where $f(T)\geq 0, T\geq 0 \text{ or } \gamma, \beta>0, \eta>0, -\infty<\gamma<\infty$

  - η = scale parameter
  - β = shape parameter (i.e., slope)
  - γ = location parameter

- Cumulative density function, *cdf* :

$$F(T) = 1 - e^{-\left(\frac{T-\gamma}{\eta}\right)^{\beta}}$$

# Weibull Model /4

- Reliability

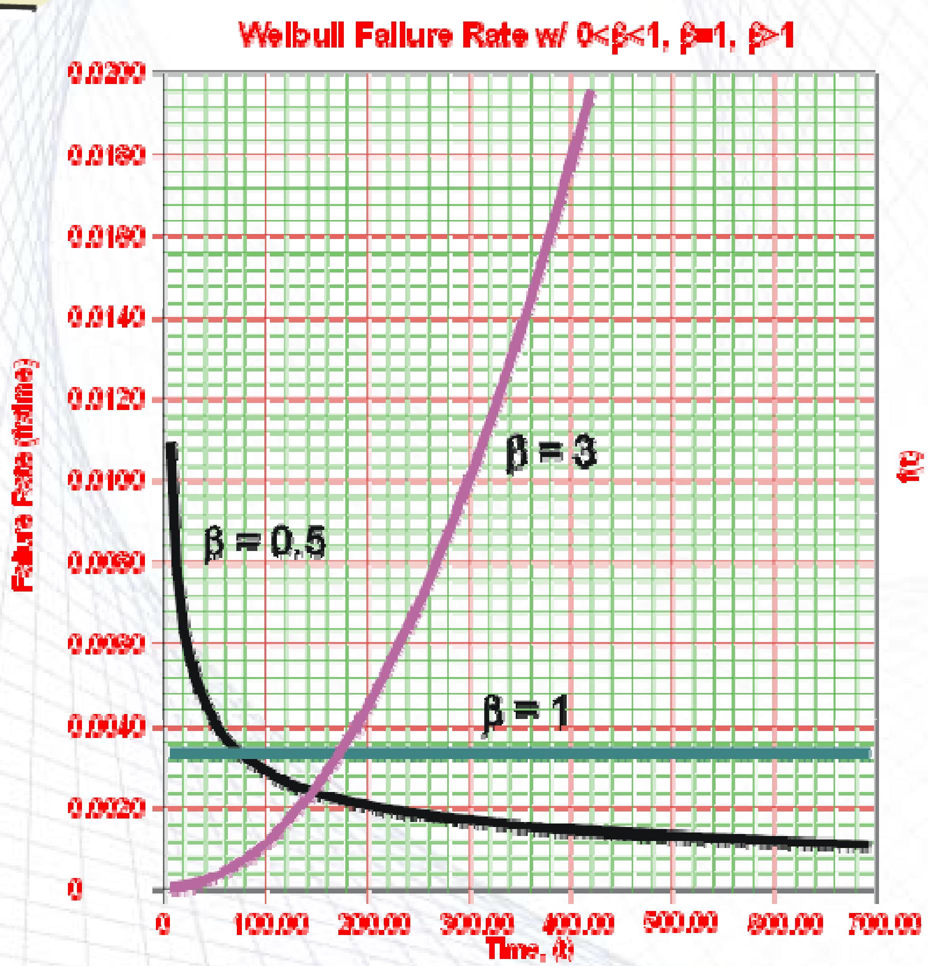
$$R(T) = e^{-\left(\frac{T-\gamma}{\eta}\right)^{\beta}}$$

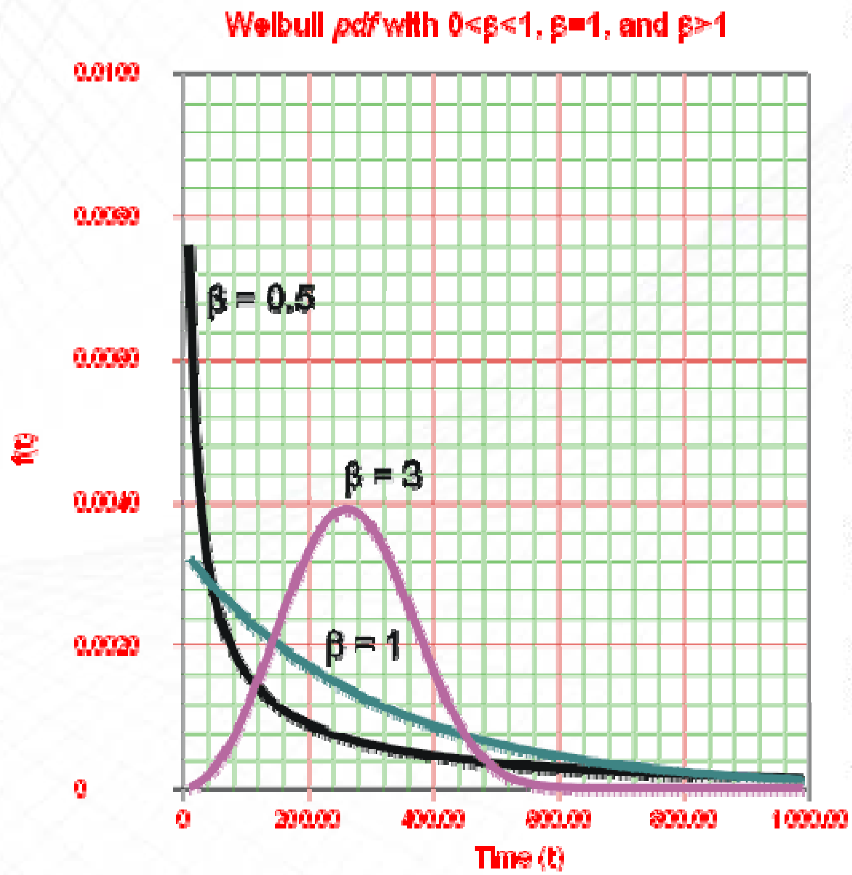
- Failure intensity

$$\lambda(T) = \frac{f(T)}{R(T)} = \frac{\beta}{\eta}\left(\frac{T-\gamma}{\eta}\right)^{\beta-1}$$

- Model parameters: η (scale parameter), β (shape or slope parameter) and γ (location parameter)

# Weibull Model /5




Reference: http://www.weibull.com/

# Chapter 2 Section 6

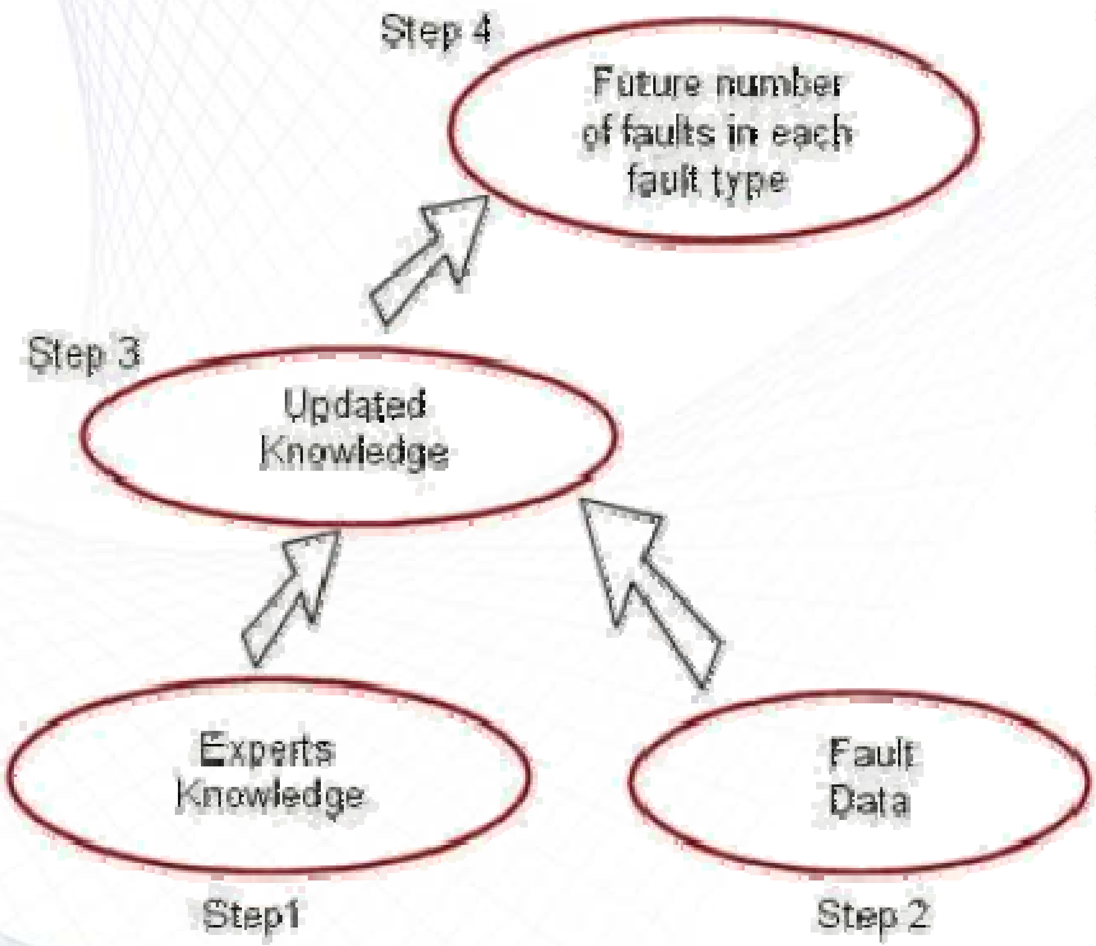
## Bayesian Models

# Bayesian Models /1

- The previous models allow change in the reliability only when an error occurs.
- **Assumptions:**
  - If no failures occur while the software is tested then the reliability should increase, reflecting the growing confidence in the software by the user.
  - Different faults have different impacts on reliability. The number of faults is not as important as their impacts.
- Reliability is a reflection of both the number of faults that have been detected and the amount of failure-free operation.
- Uses prior distribution representing the "past data" and a posterior distribution to incorporate past and "current data".

# Bayesian Models /2

- Takes more "subjective" view of failure
- Steps of Bayesian approach



# Chapter 2 Section 7

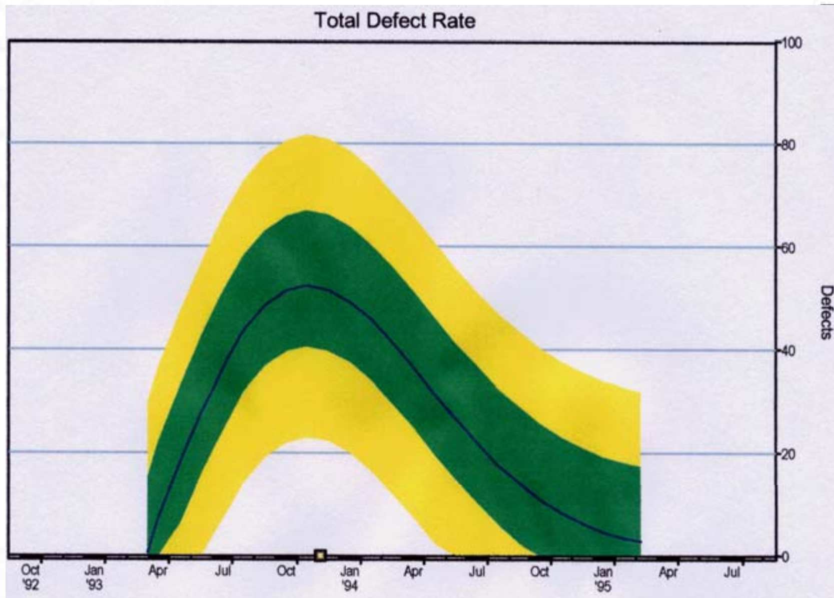
## Early Life-Cycle Prediction Models

# Early Life-Cycle Models /1

- Most of the reliability growth models ***predict*** reliability at the later stages of development life cycle, i.e., during testing stage. However, if a software organization waits until all necessary data is available to collect, what usually happens is that it is already too late to make proper improvements in software reliability to achieve the reliability goal.
- **Question:** Would it be possible to predict reliability at the earlier stages?
- Models that relate early stage data to reliability are needed.
- Typical early stage data: requirements, man-power build-up, development process, error injection rate and trends, code size estimates, etc.

# The Raleigh-Putnam Model

- It is usually assumed that over the life of the project the faults detected (per month) will resemble a Raleigh curve
- The rate of expending effort is proportional to the rate of committing errors. Putnam assumes that **the rate of effort expenditure is a Raleigh curve**.




Source: QSM Web page
http://www.qsm.com/

# ELCM: The Process /1

- Obtain the milestones for the schedule:
  - Start date and total months in project
  - Date of expected full operational capability - $t_d$
  - Estimate the total number of faults over the life of the project - Er.
- A Raleigh curve can be generated by solving for each month t using this equation:

$$E_m = \left( \frac{6E_r}{t_d^2} \right) t \times e^{\left( \frac{-3t^2}{t_d^2} \right)}$$

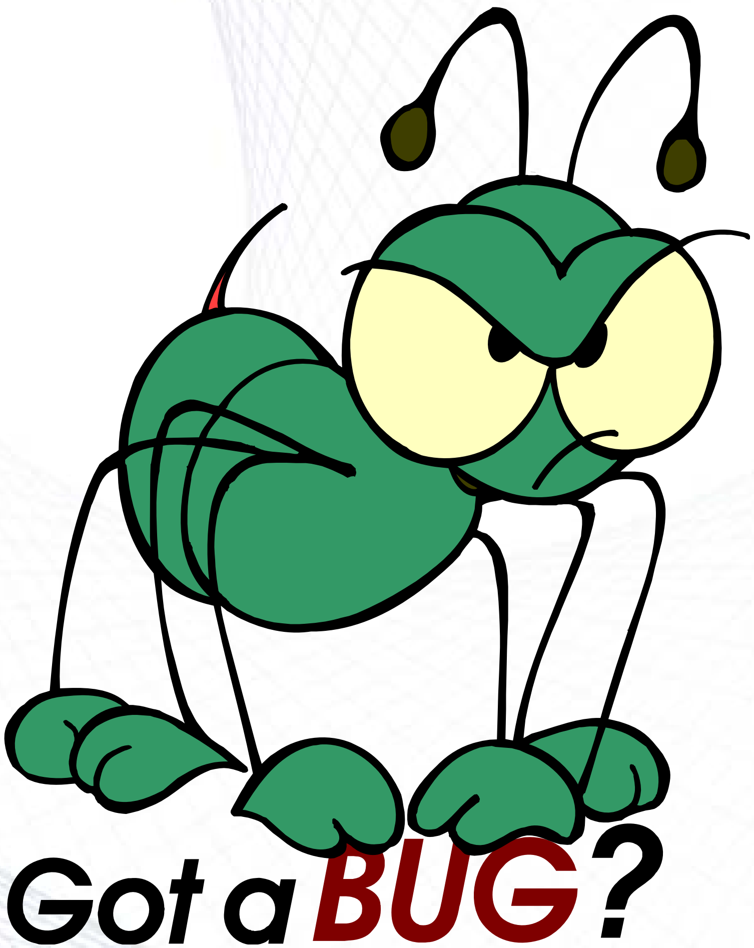
# ELCM: The Process /2

- Use this profile to gauge the fault detection process during each phase of development.
- In particular, this profile can be used to gauge the original schedule estimate and the prediction for the total number of defects.
- For example, the estimated number of defects impacts the height of the curve while the schedule impacts the length of the curve.
- If the actual defect curve is significantly different from the predicted curve then one or both of these parameters may have been estimated incorrectly and should be brought to the attention of management.

# Reliability Growth Model Assumptions vs. Practice

| Assumptions | In practice |
|---|---|
| Software does not change and defects are fixed immediately | Software changes rapidly and certain defects are scheduled to be fixed in a later date |
| Testing Operational Profile (OP) | Focus on functional testing. It is difficult to define OP and perform operational tests. |
| Constant test effort and independent failure intervals | Varying test effort (due to holidays and vacations) and failure intervals may vary |
| All failures are observable | Testing in a controlled environment may be different from running software in live environment |
| Collect failure reports | Collect defect reports. Any kind of defect observed is recorded. Not all reports address a failure |
| Remaining failures are either constant or decreasing | Remaining failures may actually increase due to improper bug fixes |

*[Stringfellow & Andrews 2002]*


Got a BUG?