

## Affidabilità (Reliability)

1.1

## Outline

- Definizioni
- Indicatori
- Modelli

1.2

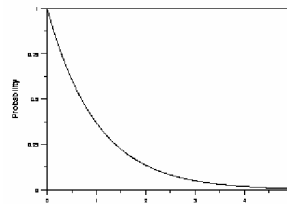
## Affidabilità

- Affidabilità**
  - R(t) probabilità che un componente o sistema svolga correttamente ed ininterrottamente la sua funzione per un periodo di tempo assegnato, in condizioni operative ed ambientali ben definite.
- Rateo di guasto  $\lambda$** 
  - N guasti per unità di tempo
- MTTF**
  - Tempo medio al guasto
  - $1/\lambda$

1.3

## Esempio

- Esponenziale, con rateo guasto costante nel tempo



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

1.4

- Inaffidabilità**
  - $F(t) = 1 - R(t)$
- Disponibilità (availability)**
  - probabilità che un componente o sistema svolga correttamente la sua funzione in un istante prefissato, in condizioni operative ed ambientali ben definite.

1.5

- Disponibilità**

$$A(t) = \frac{MTTF}{MTTF + MTTR} = \frac{MTTF}{MTBF}$$

- MTTF: *Mean Time To Failure*, tempo medio di corretto funzionamento del componente o sistema prima del guasto.
- MTTR: *Mean Time To Repair*, tempo medio di durata della riparazione, considerando sia il tempo necessario a rilevare il guasto che quello utilizzato per ripristinare il corretto funzionamento.

1.6

•MTBF: *Mean Time Between Failures*, tempo medio che intercorre tra un guasto ed il successivo

$$MTBF = MTTF + MTTR$$

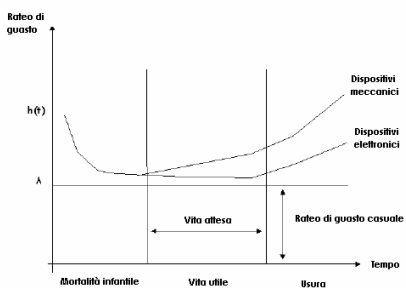
1-7

### Affidabilita' hw

- Identificazione modi di guasto
- Identificazione cause
  - Progettazione, produzione, usura
- Test accelerati (stress)
- Stima affidabilita in operation
  - A partire da risultati test accelerati
- Modello affidabilita del sistema
  - Come f(affidabilita componenti)

1-8

### Affidabilita hw



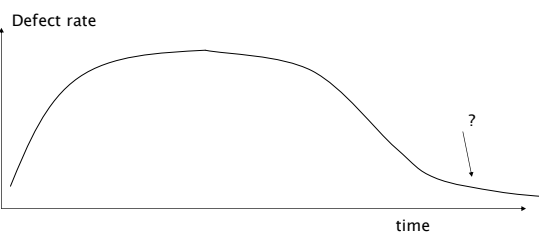
1-9

### Affidabilita' sw

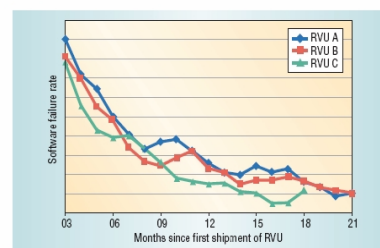
- Modi di guasto
  - Identificazione difficile
- Cause
  - Progettazione
- Test accelerati
  - Non possibile
- Stima affidabilita in operation
  - Da risultati test (caratteristiche processo)
  - Da caratteristiche strutturali
- Modello affidabilita' sistema

1-10

### Affidabilita' sw



1-11



1-12

## Affidabilità - livelli

Product type	Failure rate (per hour)	Reliability level
Safety-critical software	$< 10^{-7}$	Ultra-high
Commercial software	$10^{-3}$ to $10^{-7}$	Moderate to high
Auxiliary software	$> 10^{-3}$	Low

1-13

## Modelli di affidabilità

Y = defect rate in operation

- **Y = f (caratteristiche di prodotto)**
  - Ex. moduli e loro complessità
- **Y = f (caratteristiche di processo)**
  - Ex. processo di test e difetti rimossi

1-14

## Affidabilità sw - processo

- **Requirements analysis**
  - Reliability – user needs
  - Failure rates for failure severity classes
- **Allocation**
  - Define failure rate for subsystems/parts
- **Prediction**
  - Predict reliability early from product / process metrics
    - Define how much system test is needed to achieve desired reliability
    - Change product or process
- **Aggregation**
  - Aggregate subsystem reliability models into system reliability model
- **Growth modeling**
  - On integrated system, during system testing
- **Demonstration testing**
  - At end of system testing

1-15

## Modelli di affidabilità

- 1 costruire modello su progetti passati
- 2 applicare modello su progetto in corso
- 3 aggiornare il modello

1-16

## Modelli – processo

- **Rayleigh**
  - Fondato su difetti trovati in tutto processo di sviluppo
- **RGM (reliability growth models)**
  - Fondati su difetti trovati in fase test

1-17

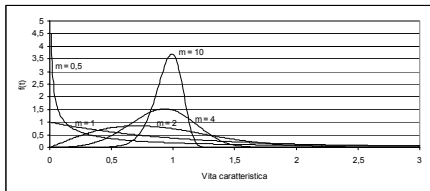
## Modello Rayleigh

1-18

## Rayleigh

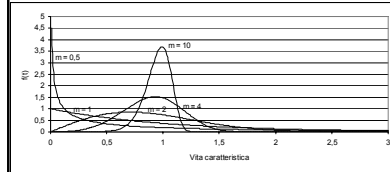
### Basato su distribuzione Weibull con m=2

– Weibull: usata in affidabilità hw



1-18

## Weibull



$$CDF: F(t) = 1 - e^{-(t/c)^m}$$

$$PDF: f(t) = \frac{m}{t} \left(\frac{t}{c}\right)^{m-1} e^{-(t/c)^m}$$

- » CDF: Cumulative distribution function
- » PDF: Probability density function
- » m parametro di forma
- » c parametro di scala
- » Tende a zero in modo asintotico

1-20

## Rayleigh

$$CDF: F(t) = 1 - e^{-(t/c)^2}$$

$$PDF: f(t) = \frac{2}{t} \left(\frac{t}{c}\right) e^{-(t/c)^2}$$

- Sale fino a un massimo (al tempo  $t_m$ ) poi decresce
- c si ottiene 
$$t_m = \frac{c}{\sqrt{2}}$$
- Area sotto curva fino a  $t_m$  e' pari al 39,35% del totale

1-21

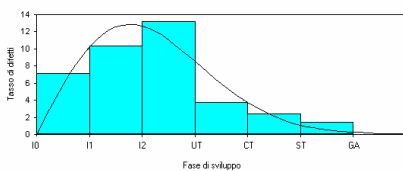
- Aggiungendo K = numero totale difetti
- Interpretando
  - PDF come defect rate
  - CDF come totale difetti

$$CDF: F(t) = K(1 - e^{-(1/2t_m^2)t^2})$$

$$PDF: f(t) = K \left[ \left(\frac{1}{t_m}\right)^2 t e^{-(1/2t_m^2)t^2} \right]$$

1-22

## Costruzione modello

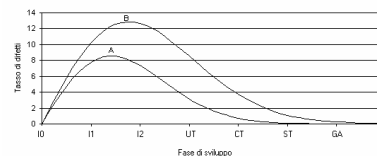


- Avere difetti per fase, fino al massimo
- Fit curva e estrapolazione

1-23

## Ipotesi

1. Maggiore il numero difetti trovato in sviluppo, maggiore quello trovato in operation
2. Prima il picco  $t_m$ , prima il calo



1-24

## Interpretazione

- Confronto tra progetti significativo solo se effort defect removal simile
- Altrimenti

		Defect rate	
		High	Low
Defect removal effort	High	Non negative	Best case
	Low	Worst case	uncertain

1-25

## Reliability Growth Models

1-26

## RGM

- Difetti raccolti in fase di test sistema
- Versione i, difetto, correzione in versione i+1
- Assunzione: difetto viene sempre corretto in modo ideale (reliability growth)

1-27

## Modello esponenziale

- Weibull con  $m=1$ , largamente usato in affidabilità hw
- $c$  = parametro di scala
- $\lambda = 1/c$  = tasso di guasto

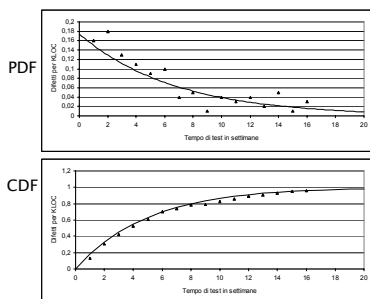
$$CDF : F(t) = 1 - e^{-(t/c)} = 1 - e^{-\lambda t}$$

$$PDF : f(t) = \frac{1}{c} e^{-(t/c)} = \lambda e^{-\lambda t}$$

- K fattore di moltiplicazione, n totale difetti o densità difetti

1-28

## Esponenziale



1-29

## Tempo t

- **t = tempo CPU**
  - Adeguato per progetti piccoli, modelli più precisi
- **t = tempo calendario**
  - Relativo al processo di test
  - Occorre che effort test sia omogeneo nel tempo
    - Controllare e raccogliere effort (person hour) per unità di tempo
    - Se non omogeneo, normalizzare

1-30

## Raccolta dati

### •Cruciale la precisione nella raccolta

- Numero e tempo di identificazione difetti

1-31

## RGM

### •Modelli di stima MTBF

- MTBF cresce secondo ddp ogni volta che difetto trovato e risolto
  - Jelinski Moranda
  - Littlewood
  - Goel Okumoto

### •Modelli di stima K (difetti per unita' di tempo)

- Intervallo di tempo futuro fissato, stimare n difetti che si troveranno
  - Goel Okumoto - Poisson

1-32

## Modelli stima MTBF

1-33

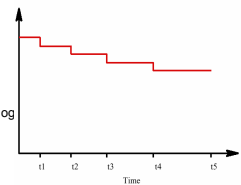
## Jelinski Moranda

### • Ipotesi

- N difetti in totale
- Difetti si manifestano in modo casuale
- Correzione perfetta
- Tempo correzione trascurabile

### • Effetto

- Costante a tratti
- Tasso di guasto decresce di  $\phi$  ad og difetto trovato
- Intervalli tra difetti si allungano



$$Z(t_i) = \phi[N - (i-1)]$$

1-34

## Littlewood

### •Come Jelinski Moranda ma

- Ogni difetto ha *dimensione*
  - Difetti piu' *grandi* vengono trovati prima
  - Difetti piu' *piccoli* possono non essere trovati mai

1-35

## Goel Okumoto – debug imperfetto

### •Correzione di difetto puo' essere imperfetta

$$Z(t_i) = [N - p(i-1)]\lambda$$

### •N numero difetti totale iniziale

### •p probabilita' non inserire altro difetto

### •λ tasso di guasto

1-36

## Modelli stima K

1-37

## Goel Okumoto - NHPP

•Processo Poisson non omogeneo per guasti attesi in intervallo t

$$P\{N(t) = y\} = \frac{[m(t)]^y}{y!} e^{-m(t)}, y = 0, 1, 2, \dots$$

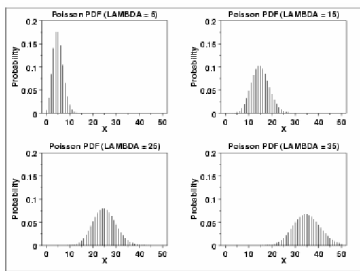
$$m(t) = a(1 - e^{-bt})$$

$$\lambda(t) \equiv m'(t) = abe^{-bt}$$

- $m(t)$  = n guasti attesi in tempo t
- $\lambda(t)$  = tasso di guasto
- a = n guasti osservandi in tempo infinito
- b = tasso di rilevazione errori per difetto

1-38

## Poisson



$$p(x, \lambda) = \frac{e^{-\lambda} \lambda^x}{x!} \text{ for } x = 0, 1, 2, \dots$$

1-39

## NHPP

$$m(t) = a(1 - e^{-bt})$$

$$\lambda(t) \equiv m'(t) = abe^{-bt}$$

•NHPP e' applicazione di esponenziale con

- $m(t)$  = CDF
- $\lambda(t)$  = PDF
- a = K
- b =  $\lambda$

1-40

## NHPP

•Caratteristiche essenziali

- N difetti in tempo infinito e' var casuale (e non costante)
- Tasso di guasto (t) puo essere non decrescente

1-41

## NHPP - variante

$$m(t) = a(1 - e^{-bt^c})$$

$$\lambda(t) \equiv m'(t) = abc \cdot e^{-bt^c} t^{c-1}$$

•Weibull con

- b, c = costanti che caratterizzano la qualita' del test
- a = n difetti rilevati a fine processo test

1-42

### Yamada – S ritardata

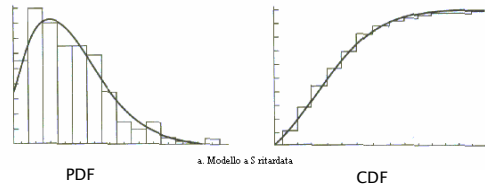
• Simile a NHPP, con

$$m(t) = k[1 - (1 + \lambda t)e^{-\lambda t}]$$

- k numero totale difetti
- $\lambda$  tasso guasti
- Modella apprendimento di testers, picco di detection difetti e' ritardato rispetto a esponenziale

1-43

### Yamada



1-44

### Ohba – S inflessa

• Modella apprendimento dei testers e

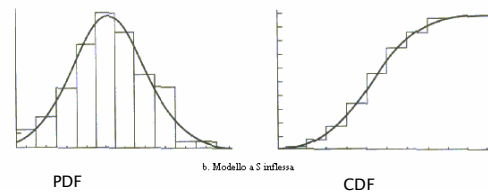
• Non indipendenza difetti

- piu' difetti si trovano in modulo, piu se ne troveranno ..

$$CDF = I(t) = K \frac{1 - e^{-\lambda t}}{1 + i e^{-\lambda t}}$$

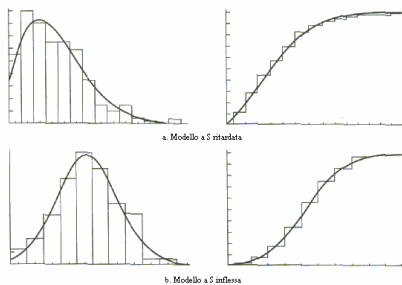
1-45

### Ohba



1-46

### Yamada - Ohba



1-47

### Problemi

• Casualita' del difetto

- Difetto e' deterministico, avviene dato certo input
- Hp dei modelli, difetto casuale (considerando casualita di input rispetto spazio di ingressi)

• Confidenza nel modello aumenta con il numero di difetti

- Possibile alta confidenza nel dire che affidabilita' bassa
- Non possibile dire con alta confidenza (pochi difetti) che affidabilita' alta

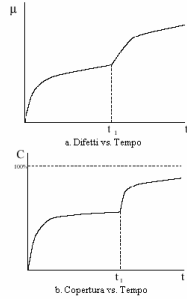
• P trovare difetto non uniforme

- Si abbassa quanto piu' difetti trovati

1-48

## Problemi(2)

- Detection difetti non costante, probabilmente collegato a copertura o altre caratteristiche del processo di test



1-48

## Problemi (3)

- In operation fixing difetti e' piu' lento che in development (test)
  - Non del tutto corretto estrapolare curva di affidabilita da test a operation
- Testing non considera requisiti mancanti (e difetti collegati)

1-50

## Problemi (4)

### •Tasso di guasto

- $\lambda$  in processo test
- $\lambda$  in operation
  - Ragionevole supporre che il primo sia piu' alto (in quanto obiettivo di fase di test e' trovare errori)

### •Proposta 1:

- calcolare tasso conversione

### •Proposta 2:

- Operational profiles

1-51

## Criteri scelta modello

### •Capacita' predittiva

### •Semplicita

- Raccolta e validazione dati
- Concettuale (background matematico richiesto)

### •Vincoli, ipotesi

- Es. Debug perfetto

1-52

## Un' applicazione (HP)

1-53

## Context

•RVU (Release Version Update) is an aggregation of all product changes plus the previous version of unchanged parts.

•HP NonStop Enterprise Division provides a system for failure's data acquisition and RVU production\installation for business critical application.

1-54

### Failure Data (1)

•A database contains all the information for problem analysis and resolution (event cronology, problem symptoms, etc.).

•Problems can be divided into two main categories:

- customer supports (which are excluded from the software failure rate)
- software problems

1.56

### Failure Data (2)

Table 1. Failure categories and associated failure reduction activities.

Failure category	Definition	Reliability-improvement activity
New defect	Previously unknown defect	Feedback characteristics of new defect to improve future inspections or test
Rediscovery, fix available	Known defect; customer could have prevented the failure by installing an existing fix	Publicize fix availability; make fix easy to install
Rediscovery, fix not available	Known defect for which a fix does not yet exist	Create software fixes more quickly
Unknown	Failure that we cannot reproduce or for which we cannot determine the root cause	Develop better diagnostics or improve communication about the use of existing diagnostics
In analysis	Failure still being analyzed at report time	Develop better diagnostics or analysis tools

A further classification of software problems is done as shown in table above.

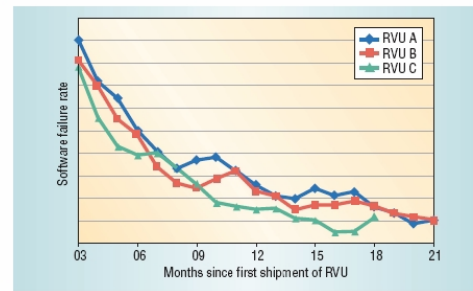
1.56

### Failure Rate Behavior

•In theory software failure rate is constant with time, since it is not subject to early-life failure and wearout.

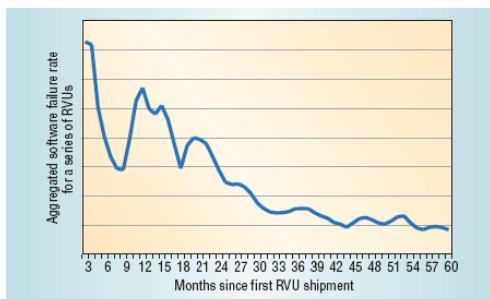
•In practice defect repair coupled with installing bug fixes decreases the failure rate, while new features generally introduce new defects, increasing the failure rate.

1.57



Behavior of the software failure rate for three RVUs

1.58



Aggregated software failure rate for a series of RVUs

1.59

### Un applicazione (IBM)

1.60

