UNIVERSITÀ DEGLI STUDI DI ROMA TOR VERGATA

FACOLTÀ DI INGEGNERIA

LAUREA SPECIALISTICA IN INGEGNERIA INFORMATICA

Corso di Ingegneria del Software 2

A.A. 2007/2008

PROGETTO DI LABORATORIO

$$N = 48.9047$$

$$\phi = 0.00070413$$

Professore

Giuseppe Iazeolla

Studente

Stefano Perna
Matricola 114505
Gruppo 7

Indice

Introduzione

L'obiettivo del progetto di laboratorio è la stima del numero di difetti N e del tasso di failure per difetto ϕ relativi ad un progetto software.

Allo scopo di stimare i valori è stato richiesto di stimare il tasso iniziale ϕ_0 di failure per difetto simulando un **pre-run** con T_i esponenzioale di media $E(T_i) = 127,77$ ore, e di durata τ sufficiente a produrre $n = 100$ failure. Si è inoltra assunta l'esistenza di $N_0 = 60$ difetti latenti iniziali. Al termine di tale fase è stata simulata la fase di **run** di testing statistico con T_i esponenziale di media $E(T_i)$ e di durata sufficiente a produrre 50 failure, ed è stata rilevata la sequenza $(T_1, \ldots, T_{50})$ dei tempi di lifetime.

I valori di N e ϕ sono poi stati stimati utilizzando il metodo Maximum Likelyhood e modello Musa, dato un valore di $A = 0.95$, adottando il procedimento Newton-Raphson per la ricerca delle radici. Tale procedimento è stato ripetuto fino a convergenza con distanza tra radici successive inferiore a 10^{-6}.

Capitolo 1

La fase di Pre-Run

La fase di Pre-Run è un'esecuzione molto lunga tale da produrre nel nostro caso 100 failure.

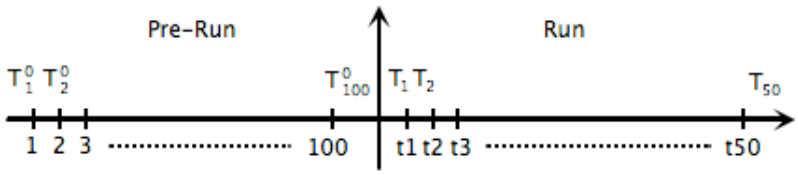

Figura 1.1: Distribuzione dei tempi di Pre-Run e di Run di Testing statistico.

Per soddisfare tale condizione la fase dovrà durare un tempo $\tau = \sum_{i=1}^{100} T_i$.

Si potrà quindi calcolare il valore di ϕ_0 come:

$$\phi_0 = \frac{n}{\tau} * \frac{1}{N_0} = \frac{n}{\tau * N_0}$$

1.1 La sequenza T_i

La sequenza dei valori di T_i esponenziali, di media $E(T_i) = 127.77$ ore, è stata creata mediante un generatore di numeri pseudo-casuali di tipo moltiplicativo realizzato in Java (vedi Appendice A).

L'espressione del generatore moltiplicativo è del tipo $X_i = X_{i-1} * a \mod m$ avendo $a = 1220703125$ e $m = 2147483648$, con seme $X_0 = 2473$.

Per la generazione della sequenza dei valori di T_i dovremo creare la sequenza dei valori di X_i e normalizzarla nell'intervallo $[0, 1]$ mediante la trasformazione nel vettore $R_i = X_i/m$. Possiamo quindi riscrivere la sequenza T_i come

$$T_i = E(T_i) * -\ln(R_i)$$

Calcolando il valore esatto di X_0 in base ai valori assegnati otteniamo: $X_0 = 1220703125 * 2473 \mod 2147483648 = 1584302685$. I successivi X_i saranno invece calcolati mediante il generatore riportato in Appendice A.

1.2 Il calcolo di ϕ_0

Il tasso iniziale di failure per difetto potrà ora essere valutato secondo la formula

$$\phi_0 = \frac{n}{\tau} * \frac{1}{N_0} = \frac{n}{\tau * N_0}$$

con

$$\tau = \sum_{i=1}^{100} T_i$$

Usando la sequenza di T_i ottenuta in precedenza otteniamo

$$\tau = 11142.101687883207$$

e

$$\phi_0 = 0.000149583$$

.

Capitolo 2

La fase di Run

La simulazione della fase di Pre-Run è seguita da quella di un **Run** di testing statistico con T_i esponenziali di media $E(T_i)$, di durata tale da produrre 50 failure. I tempi di lifetime usati sono stati rilevati come seguito della sequenza generata nella precedente fase di Pre-Run secondo il modello Musa.

2.1 Testing statistico

Il testing statistico ha lo scopo di valutare l'impatto che i difetti rimanenti hanno sull'affidabilità del software. Per affrontare questo tipo di test si dovranno preparare i dati di test in modo da stressare le aree critiche del sistema (*core*) e si dovrà costruire il profilo operativo del sistema (o sottosistema) che si vuole valutare.

Il profilo operativo di un software è un grafico che mette in relazione le classi di dati di input (rappresentate in ascissa) con le rispettive frequenze di utilizzo (rappresentate in ordinata).

Il testing statistico può essere suddiviso in quattro fasi:

1. Individuazione del profilo operativo.

2. Creazione dei dati di test.

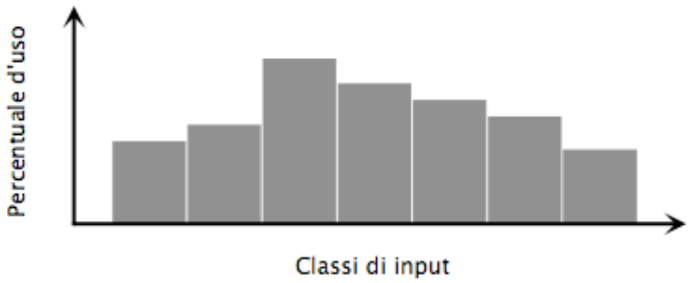

Figura 2.1: Frequenza con cui le classi sono usate.

3. Applicazione dei dati di test al sistema.

4. Stima dell'affidabilità del sistema.

2.1.1 Individuazione del profilo operativo

In questa fase si devono costruire partizioni e path, usando il profilo operativo (una rappresentazione probabilistica degli input possibili al prodotto software). Si costruisce un istogramma di probabilità, con input divisi in categorie, che potrebbero essere quelle dell'Equivalent Partitioning. Si associa ad ogni classe una probabilità di uso e si ottine il numero delle frequenze, che è dato da $\frac{\mu_1}{N}, \frac{\mu_2}{N}, \ldots, \frac{\mu_n}{N}$. Quindi, si costruisce un'urna in cui è rappresentato un insieme degli input con la distribuzione del profilo operativo; l'input può essere corretto oppure erroneo; se è erroneo si lancia di nuovo il prodotto con un altro input. Si assume che il processo sia istantaneo. Si costruisce la storia delle failure: si rimette un input nell'urna, si vede il tempo in cui si è verificata la seconda failure. Si cerca il difetto, si corregge e si rilancia il prodotto.

2.2 L'Affidabilità

L'affidabilità di un software rappresenta la probabilità che il software lavori correttamente nell'arco di un certo periodo di tempo, ovvero la probabilità che una failure non

si manifesti prima di un determinato intervallo di tempo dalla release del software. Per $t \to \infty$ l'affidabilità del software sarà sempre pari a zero in quanto prima o poi accadrà che un determinata sequenza di input generi una failure.

Un software viene rilasciato nel momento in cui la sua affidabilità è considerata accettabile rispetto a quanto richiesto (*stopping by optimal reliability*) o quando il costo della fase di testing è ottimizzato rispetto alle ripercussioni che si potrebbero manifestare durante la vita del software (*stopping by optimal cost*).

La stima dell'affidabilità e effettuata mediante modelli statici o dinamici.

In particolare la stima dei difetti iniziali è effettuata mediante modelli statici, attraverso metodi di regressione.

2.2.1 Modelli statici

I modelli statici consentono di stimare il numero di difetti latenti alla fine della fase di Defect Testing. Tali stime vengono effettuate basandosi sulle caratteristiche intrinseche del prodotto (*size*, numero di elementi decisionali, complessità della nidificazione, strutturazione, complessità ciclomatica) e del tipo di processo adottato (waterfall, iteartivo, ide, etc.).

2.2.2 Modelli dinamici

I modelli dinamici, nota la stima dei difetti iniziali, si propongono di prevedere la frequenza con cui le failure si manifesteranno (tasso futuro di failure).

Sono rappresentati da una funzione nella forma:

$$y'_{\tau+} = f(\hat{d}, y'_{\tau})$$

in cui la previsione futura $y'_{\tau+}$ è funzione della stima statica dei difetti $\hat{d}$ e del numero di failure y'_{τ} verificatosi nell'intervallo di tempo τ.

Modelli *single failure*

Indicando con T, variabile aleatoria continua, il tempo di vita (*lifetime*) del prodotto software, inteso come il tempo intercorso tra il rilascio del prodotto e il manifestarsi della prima failure, si introduce la sua densità:

$$f(T) = \int_{t-\frac{dt}{2}}^{t+\frac{dt}{2}} f(T)dT = prob(t - \frac{dt}{2} \leq T \leq t + \frac{dt}{2}$$

Si definisce cumulativa di probabilità $f(t)$ la probabilità che una failure si presenti nell'intervallo t:

$$F(t) = prob(T \leq t) = \int_0^t f(T)dT$$

con $F(\infty) = 1$ e $F(0) = 0$.

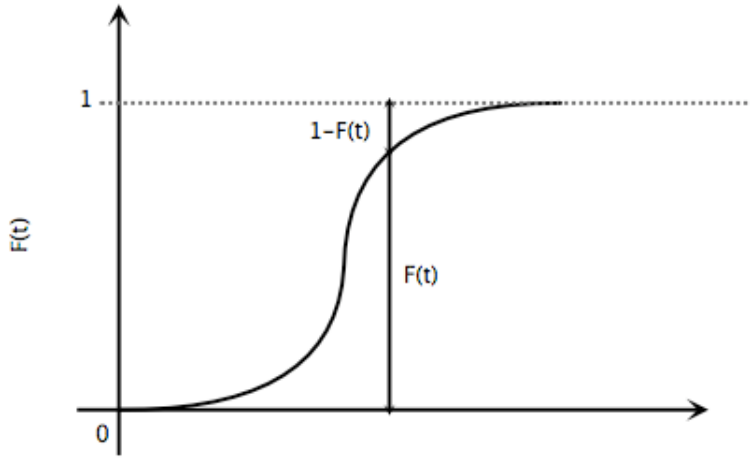

Figura 2.2: Cumulativa di probabilità F(t).

L'affidabilità $R(t)$ si definisce come la probabilità che la failure si manifesti dopo un tempo $T \geq t$:

$$R(t) = prob(T \geq t) = 1 - F(t)$$

con $R(\infty) = 1 - F(\infty) = 0$ e $R(0) = 1 - F(0) = 1$.

Possiamo ora calcolcolare la probabilità che si verifichi un guasto nell'intervallo $t \pm \frac{\triangle t}{2}$, assunto che non ci siano stati guasti in precedenza, tramite il rapporto

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

il cui limite per $\triangle t \to 0$ diviso per $\triangle t$

$$\lim_{\triangle t \to \infty} \frac{F(t+\triangle t) - F(t)}{1 - F(t)} \frac{1}{\triangle t} = \frac{f(t)}{R(t)} = h(t)$$

La funzione $h(t)$ esprime la funzione di rischio (*hazard rate*) che rappresenta il tasso medio con cui si manifestano le failure.

Possiamo quindi scrivere la relazione

$$f(t) = \frac{d}{dt}F(t) = \frac{d}{dt}(1 - R(t)) = -\frac{d}{dt}R(t)$$

Si definisce MTTF (*mean time to failure*) o media del tempo di vita la grandezza

$$E(T) = MTTF = \int_0^\infty t f(t)dt = \int_0^\infty t dF(t) = \int_0^\infty t d(1-R(t)) = -\int_0^\infty t dR(t) = -\int_0^\infty t R'(t)dt$$

da cui

$$MTTF = \int_0^\infty R(t)dt$$

Modelli *multiple failure*

Tali modelli estendono quelli *single failure* al caso in cui n failure si manifestano ad intervalli $T_1, T_2, \ldots, T_n, t_{i-1} \leq T_i \leq t_i$.

2.3 Il modello *MUSA*

Il modello MUSA è un modello d'affidabilità dinamico a failure multipla di tipo *time between failure*. Tale modello tiene conto di:

- **difetti indotti**, ovvero difetti causati dagli interventi mirati a correggerne altri

- **difetti scoperti indirettamente**. ovvero difetti scoperti per caso mentre se ne indagavano altri
- **difetti introvabili**, ovvero difetti che si assume non saranno mai trovati.

A differenza del modello J& M non siamo in presenza di *perfect debug*.

Il modello si basa su campioni di tempi di interfailure ricavati dal profilo operativo

La funzione *hazard rate* diventa:

$$h(t) = \phi(N - M(t))$$

dove N è il numero iniziale stimato di difetti e $M(t)$ è il numero di failure sperimentate fino all'istante t. Il valore ϕ rappresenta il tasso medio di failure prodotto da ciascuno dei rimanenti difetti. Chiamato $m(t)$ il numero di difetti individuati e corretti si all'istante t possiamo definire il *fault reduction factor* $B(t) = \frac{M(t)}{m(t)}$.

Supponendo che il rapporto tra numero di failure sperimentate e il numero di difetti individuati sia più o meno costante, allora è possibile approssimare $B(t)$ con la costante $B = \frac{N}{m}$.

Definiamo A come $A = BC$ dove C è un fattore di compressione che tiene conto delle caratteristiche del software. Il fattore A esprime la proporzionalità tra la derivata di $M(t)$ e la funzione *hazard rate* $h(t)$:

$$\frac{dM(t)}{dt} = A \cdot h(t)$$

Risolvendo l'equazione si ricava:

$$M(t) = N \cdot (1 - e^{-\phi AT})$$

e

$$h_i(t) = \phi \cdot N \cdot e^{-\phi A t_i}$$

da cui la funzione dell'affidabilità

$$R_i(t) = e^{-\int_0^t h_i(u)\cdot du} = e^{-h_i(t)\cdot t} = e^{-\phi\cdot N\cdot e-\phi\cdot A\cdot t_i\cdot t}$$

con $R_0(t) = e^{-\phi N t}$

e la distribuzione di probabilità

$$f_i(T) = h_i(T)\cdot R_i(T) = \phi\cdot N\cdot e^{-\phi\cdot A\cdot t_i}\cdot e^{-\phi\cdot N\cdot e^{-\phi\cdot A\cdot t_i}\cdot T_i}$$

dove t_i rappresenta il tempo in cui si verifica l'i-esima failure. Il valore di MTTF potra essere calcolato come

$$MTTF_i = \int_0^\infty R_i(t)\cdot dt = \frac{1}{\phi\cdot N}\cdot e^{\phi\cdot A\cdot t_i} = MTTF_0\cdot e^{\phi\cdot A\cdot t_i}$$

.

Il calcolo dei T_i secondo *MUSA*

Come appena visto possiamo calcolare i valori T_i utilizzando la forma:

$$T_i = -\frac{1}{\phi N e^{-\phi A t_i}}\ln(R_i)$$

dove R_i rappresenta il vettore generato in modo pseudo-casuale.

2.4 Il metodo *Maximum Likelihood*

Il metodo maximum likelihood è un metodo statistico utilizzato per inferire i parametri di una distribuzione che genera un data set noto. Nel nostro esercizio verrà utilizzato per stimare i valori iniziali dei parametri N e ϕ. Noto un campione dei tempi di inter-failure $E : T_1, T_2, \ldots, T_n$ possiamo associargli una distribuzione probabilistica

$$p(E|\theta) = f(T_1)\cdot f(T_2)\ldots f(T_n) = \sum_{i=1}^{n} f(T_i)$$

Considerando lo stimatore $\hat{\theta}$ di θ la corrispondente densità probabilità stimata sarà

$$\hat{p}(E|\hat{\theta}) = \hat{f}(T_1) \cdot \hat{f}(T_2) \dots \hat{f}(T_n) = \sum_{i=1}^{n} \hat{f}(T_i)$$

Il metodo maximum likelihood indica come migliore stimatore quello che gode delle proprietà

$$\frac{\partial}{\partial \hat{\theta}} \hat{p}(E|\hat{\theta}) = 0$$
$$\frac{\partial^2}{\partial \hat{\theta}^2} \hat{p}(E|\hat{\theta}) < 0$$

Per il modello Musa vale

$$\hat{f}_i(T) = h_i(T) \cdot R_i(T) = \phi \cdot N \cdot e^{-\phi \cdot A \cdot t_i} \cdot e^{-\phi \cdot N \cdot e^{-\phi \cdot A \cdot t_i} \cdot T_i}$$

che sostituita nell'equazioni precednti per il metodo maximum likelihood porta a

$$\frac{\partial}{\partial \hat{N}} \sum_{i=0}^{n-1} \hat{\phi} \cdot \hat{N} \cdot e^{-\hat{\phi} A t_i} \cdot e^{-\hat{\phi} \hat{N} e^{-\hat{\phi} A t_i} T_{i+1}} = 0$$

$$\frac{\partial}{\partial \hat{\phi}} \sum_{i=0}^{n-1} \hat{\phi} \cdot \hat{N} \cdot e^{-\hat{\phi} A t_i} \cdot e^{-\hat{\phi} \hat{N} e^{-\hat{\phi} A t_i} T_{i+1}} = 0$$

Dovendo trovare il massimo $\hat{p}(E|\hat{N}, \hat{\phi})$, e in virtù del fatto che il logaritmo conserva la posizione del massimo possiamo risolvere il problema logaritmico corrispondente di più facile risoluzione

$$\frac{\partial}{\partial \hat{N}} \sum_{i=0}^{n-1} \ln(\hat{\phi} \cdot \hat{N} \cdot e^{-\hat{\phi} A t_i} \cdot e^{-\hat{\phi} \hat{N} e^{-\hat{\phi} A t_i} T_{i+1}}) = 0$$

$$\frac{\partial}{\partial \hat{\phi}} \sum_{i=0}^{n-1} \ln(\hat{\phi} \cdot \hat{N} \cdot e^{-\hat{\phi} A t_i} \cdot e^{-\hat{\phi} \hat{N} e^{-\hat{\phi} A t_i} T_{i+1}}) = 0$$

ovvero risolvere il sistema

$$\frac{n}{\hat{N}} - \sum_{i=0}^{n-1} \hat{\phi} \cdot e^{-\hat{\phi} \cdot A \cdot t_i} \cdot T_{i+1} = 0$$

$$\frac{n}{\hat{N}} - \sum_{i=0}^{n-1} A \cdot t_i - \sum_{i=0}^{n-1} \hat{N} \cdot e^{-\hat{\phi} \cdot A \cdot t_i} \cdot T_{i+1} - \sum_{i=0}^{n-1} \hat{\phi} \cdot e^{-\hat{\phi} \cdot A \cdot t_i} \cdot T_{i+1} = 0$$

2.5 Il procedimento di Newton-Raphson

La risoluzione del sistema può essere effettuata mediante il procedimento Newton-Raphson, che tra i metodi iterativi è il più efficiente.

Il metodo è basato sullo sviluppo in serie di Taylor arrestato al primo ordine di un vettore colonna in cui le due funzioni sono proprio $f1$ e $f2$. Partendo da opportuni valori iniziali il procedimento consente di ricavare soluzioni via via più esatte del sistema.

La funzione d'interesse è

$$U(\beta) = \frac{\partial}{\partial \beta_k} \ln L(\beta) \quad k = \hat{N}, \hat{\phi}$$

e nel caso di MUSA possiamo scrivere

$$U_{\hat{N}}(\beta) = \frac{n}{\hat{N}} - sum_{i=0}^{n-1} \hat{\phi} \cdot e^{-\hat{\phi} A t_i} T_{i+1}$$

$$U_{\hat{\phi}}(\beta) = \frac{n}{\hat{\phi}} - sum_{i=0}^{n-1} A \cdot t_i - sum_{i=0}^{n-1} \hat{N} \cdot e^{-\hat{\phi} A t_i} T_{i+1} + sum_{i=0}^{n-1} \hat{\phi} \cdot \hat{N} \cdot A \cdot t_i \cdot e^{-\hat{\phi} A t_i} T_{i+1}$$

e lo sviluppo in serie di Taylor sarà il seguente

$$U(\beta =\backsimeq U(\beta_t) \cdot (\beta - beta_t)$$

dove $\beta_t = \binom{\hat{N}_0}{\hat{\phi}_0}$ è il vettore che ha per componenti la stima iniziale di N e di ϕ ed $H(\beta_t)$ è la matrice

$$H(\beta_t) = \frac{\partial^2 \ln L(\beta_t)}{\partial \beta_{\hat{N}} \cdot \partial \beta_{\hat{\phi}}} = J(U(\beta_t)) = \begin{vmatrix} \frac{\partial U_{\hat{N}} \beta_t}{\partial \hat{N}} & \frac{\partial U_{\hat{N}} \beta_t}{\partial \hat{\phi}} \\ \frac{\partial U_{\hat{\phi}} \beta_t}{\partial \hat{N}} & \frac{\partial U_{\hat{\phi}} \beta_t}{\partial \hat{\phi}} \end{vmatrix}$$

Il nostro obiettivo è quello di trovare la β tale che $U(\beta) = 0$ ed iterare il procedimento finchè $|U(\hat{\beta}_k)|$ 10^{-6}.