

CAPITOLUL 5

FIABILITATEA OPERATIONALA SOFTWARE

5.1 Locul si rolul fiabilitatii operationale in ciclul de viata al produsului

Modelele de crestere a fiabilitatii software, dupa cum s-a aratat si in capitolul 3, permit predictia comportarii sistemelor software pe baza unor date experimentale acumulate in perioadele de testare si de mentenanta, fiind in concordanta cu metodologia bine pusa la punct a sistemelor hardware. Defectariile detectate nu trebuie utilizate inasa numai in vederea evaluarii indicatorilor de fiabilitate. Cel putin la fel de importanta este si reactia pe care analiza acestor defectari o determina in procesul de dezvoltare a produselor software, avand ca scop asigurarea unui nivel de fiabilitate minim cerut.

Perceptia defectarii depinde de punctul de vedere al celui care o detecteaza, raporteaza sau remediază. Beneficiarul identifica o problema legata de program, care impieteaza asupra atingerii obiectivelor sale concrete. Ceea ce sesizeaza beneficiarul nu este natura defectului ci implicatiile lui mai mult sau mai putin grave asupra misiunii de indeplinit. Persoana care asigura legatura informationala dintre beneficiar si producator are sarcina sa raporteze cat mai exact simptomele defectarii, sa localizeze defectul la nivel de componenta si sa evalueze posibilitatile de remediere ca volum de munca si, implicit, ca durata de indisponibilizare a produsului. In sfarsit responsabilul cu mentenanta sistemului software percepe defectul localizat la nivelul codului si este interesat de cauza lui, de prioritatea remedierii si de dificultatea asociata acestei operatii.

Aplicatiile cu mare raspundere functionala necesita un software fiabil, dar realizarea acestuia este una din cele mai dificile probleme careia trebuie sa-i faca fata industria software-lui. Dupa ce produsele software au fost livrate la beneficiari, producatorii primesc informatii de la clienti despre fiabilitatea software-lui. Totusi atunci este prea tarziu, producatorul dorind sa stie cat de fiabile sunt produsele lor inainte ca acestea sa fie livrate beneficiarilor. Modelele de crestere a fiabilitatii ajuta la furnizarea acestui tip de

informatie. Din pacate, foarte putine date reale provenite din aplicatii comerciale au fost publicate, posibil si din cauza proprietarilor acestor date. Datele defectiunilor utilizate in crearea modelelor sunt considerate proprietate, dar se poate dezvolta o metoda de transformare acestor date astfel incat ele sa poata fi publicate s159t.

Astfel in cadrul firmei Tandem s-a aplicat modelarea fiabilitatii unui numar de 4 proiecte. S-au colectat momentele de aparitie ale aparitiei defectelor pe timpul testelor de sistem si statistice corelate cu datele modelelor de crestere a fiabilitatii software. S-a constatat astfel ca exista o buna corelatie cu un *model exponential simplu* si acest model poate estima, in mod rezonabil, numarul de rezidual de defecte in produsele livrate.

In tabelul 5.1 sunt prezentate numarul defectelor reziduale si numarul de defecte gasite in primul an de functionare normala. Estimarea este foarte buna mai putin cazul produsului 2. De altfel dupa livrarea produsului 2 au fost putine date primite de la utilizator deoarece a urmat livrarea produsului 3 astfel incat nu se poate face o comparatie corecta intre estimatia calculata si valorile obtinute din datele de pe teren. Pentru a compensa aceasta lipsa a datelor din exploatare s-a dezvoltat un model care combina datele rezultate in urma testarii sistemelor 2 si 3. Asa cum se observa din tabelul 5.1 modelul de predictie combinat 2&3 se coreleaza bine cu datele din exploatare.

Tabel 5.1 Comparatie intre estimatia teoretica si datele din exploatare

Livrare sistem	Defecte reziduale estimate	Defecte din exploatare in primul an
1	33	34
2	65	8
3	24	20
4	10	9
2&3	33	28

Un sistem trebuie sa functioneze suficient de fiabil pentru functiile (aplicatiile) pe care le indeplineste, dar trebuie de asemenea sa fie livrat pe piata in acelasi moment,

de preferat chiar înainte, de produse similare ale concurenței și de asemenea să aibă un cost competitiv. Anumite sisteme pot să fie mai puțin dependente de piață dar echilibrul dintre fiabilitate, timp de livrare și cost este totdeauna important. Una din modalitățile cele mai adecvate pentru a face acest lucru este conducerea științifică a procesului de testare, prin planificare și urmărire în timp [19]. Din păcate, exploatarea multor produse software nu este făcută științific, și produsul obținut nu este suficient de fiabil cum ar trebui să fie sau este prea costisitor.

[28] se prezintă anumite rezultate în urma unui studiu experimental și analitic pentru analiza defectiunilor aparute în perioada de funcționare normală (operatională) a unui sistem de programe pentru diferite tipuri de solicitări. În mod principal se analizează ceea ce se numește "software-ul de mare fiabilitate" (highly reliable software) dar despre care se știe că mai conține greseli. Pentru scopul acestui studiu, o versiune considerată de referință (de aur - gold version) a fost luată în considerare. Erorile judicios selectate au fost introduse, una la un anumit moment de timp, și fiecare program mutant (program ce a fost obținut în urma inserării unei erori în versiunea de referință a programului analizat) a fost supus la trei tipuri de solicitări (incercări): *constante*, *aleatoare* și *variate sistematic*.

Pentru a veni în sprijinul utilizatorului, în sensul de a ști când să aplice măsurile respective software-ului, ciclul de viață s-a divizat în (fig.5.1): **timpuriu; mijlociu; târziu**.

În segmentul timpuriu măsurile vizează cauzele potențiale ale fiabilității sistemului. Diviziunea mijlocie se referă la reducerea erorilor proces, care pot îmbunătăți eficiența dezvoltării software. Ultimul segment se referă la măsurile de performanță ale fiabilității sistemului.

1. Ciclul de viață timpuriu

Segmentul timpuriu, incluzând **stabilirea conceptelor, cerințelor și etapa de proiectare**, este perioada determinării multora dintre caracteristicile sistemului, incluzând fiabilitatea potențială a sistemului; problemele care pot apărea în această fază au ca origine constrângerile proiectării sistem care limitează alternativele în darea soluției și afectează obiectivele sistem, inclusiv fiabilitatea.

Masurile dezvoltate în faza aceasta ajuta la atingerea **completitudinii si consistentei** în definirea cerintelor utilizatori cât si în etapa de proiectare.

2. Ciclul de viata mijlociu

Include **implementarea, testarea, instalarea si verificarea functionarii software**.

Obiectivele masurilor fiabilitatii în aceasta diviziune includ nu numai proiectia fiabilitatii, ci si ghidarea procesului de dezvoltare eficienta (fara defecte).

Defectele depistate în acest segment pot sa determine disponibilitatea software-lui de-a fi livrat. Aceste defecte pot fi considerate erori proces si pot fi legate direct de efectivitatea si eficienta proceselor anterioare, putându-se stabili, prin tehnici de control proces adecvate aplicate masurilor acestui segment, ce proces trebuie reluat si ce trebuie schimbat.

3. Ciclul de viata târziu

Cuprinde **operarea, mentenanta si retragerea**, oferind posibilitatea aprecierii performantei sistemului software, ca o expresie a fiabilitatii si disponibilitatii, prin masuri traditionale ale fiabilitatii software.

În acest segment sunt doi auditori pentru masurile fiabilitatii software - utilizatorul si inginerul software. Utilizatorii sunt proprietarii, operatorii si cei care receptioneaza produsul final, apreciind capacitatea acestuia de a raspunde cerintelor formulate.

Inginerii software includ pe cei care au dezvoltat software-ul si cei care ulterior asigura mentenanta. În majoritatea cazurilor, utilizatorul este mai interesat de disponibilitatea si modul de operare al sistemului; inginerul software este mai interesat de originea defectelor software care cauzeaza defectarile.

Deci, utilizatorul isi focalizeaza atentia asupra capabilitatii sistemului, iar inginerul software asupra defectelor (numai în software). Masurile fiabilitatii ar trebui, in acest caz, sa se refere atât la caderile sistem, cât si la defectele software, pentru a furniza informatii utile ambelor categorii interesate.

Exista masuri traditionale ale performantei software care exprima cele doua puncte de vedere asupra fiabilitatii sistemului, cum ar fi: MTBF, din punct de vedere utilizator, si rata de descoperire a defectelor software, din punct de vedere al inginerului software.

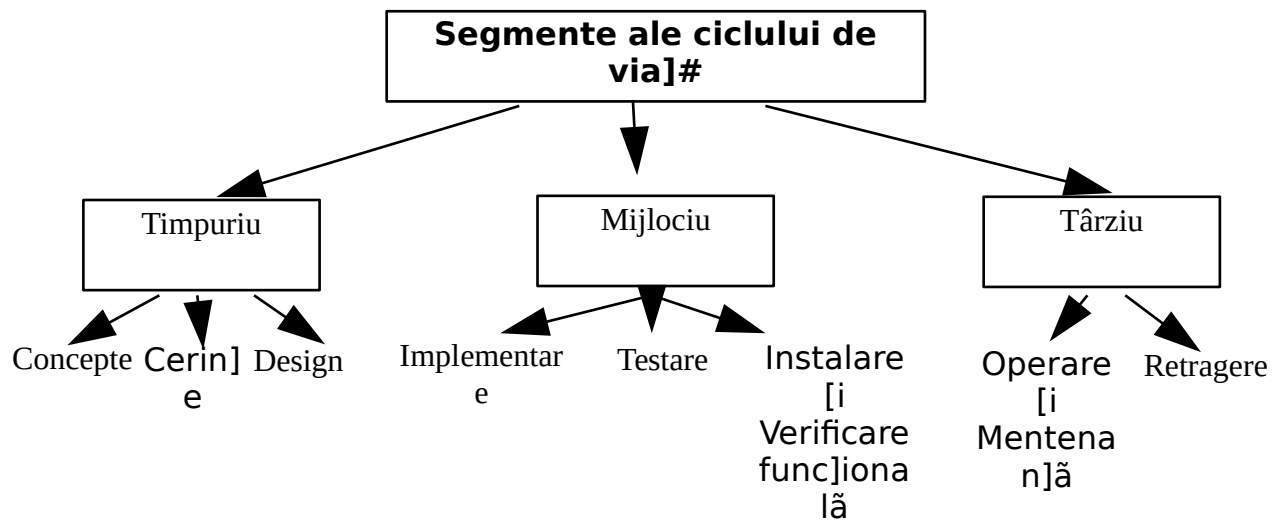


Fig.5.1 Clasificarea ciclului de viata